ROBUST KERNEL METHODS IN CONTEXT-DEPENDENT FUSION

By

GYEONGYONG HEO

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2009

© 2009 Gyeongyong Heo

To my family

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Dr. Paul Gader, for all of his guidance and encouragement throughout my studies. My thanks also go to Dr. Howard Beck, Dr. Anand Rangarajan, Dr. Gerhard Ritter, Dr. Clint Slatton, and Dr. Joseph Wilson, for all of their support and valuable suggestions. Additionally, thank to my many former and current labmates for discussion during my studies.

I also want to thank my parents for their love, understanding, and many sacrifices they had to make throughout my studies. Finally, many thanks to my wife and son, who who have been there for me.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS 4

LIST OF TABLES 7

LIST OF FIGURES 8

LIST OF ALGORITHMS 10

ABSTRACT 11

CHAPTER

1 INTRODUCTION 13

2 CONTEXT-DEPENDENT FUSION 19

2.1 Multiple classifier system 20
2.2 Decision fusion 23
2.3 Context-dependent fusion 24
2.3.1 Simultaneous clustering and attribute discrimination 26
2.3.2 Context-dependent fusion 28
2.4 Context-dependent fusion with regularization 32
2.5 Experimental results 36
2.6 Discussion 38

3 RKF-PCA: ROBUST KERNEL FUZZY PCA 41

3.1 Robust PCA (R-PCA) 42
3.2 Robust Fuzzy PCA (RF-PCA) 45
3.3 Kernel PCA (K-PCA) 49
3.4 Robust Kernel Fuzzy PCA (RKF-PCA) 51
3.5 Experimental results 54
3.6 Discussion 58

4 KG-FCM: KERNELIZED GLOBAL FUZZY *C*-MEANS 61

4.1 Global *k*-means (GKM) 63
4.2 Global fuzzy *c*-means (G-FCM) 64
4.3 Kernel-based global fuzzy *c*-means (KG-FCM) 65
4.4 Experimental results 70
4.4.1 Experiments on artificial data sets 70
4.4.2 Experiments on real world data sets 76
4.5 Discussion 79

5 FSVM-N: FUZZY SUPPORT VECTOR MACHINE FOR NOISY DATA 81

5.1 Support vector machine 82
5.2 Previous approaches to membership calculation 84
5.3 Fuzzy SVM for noisy data 91
5.4 Experimental results 95
5.5 Discussion 98

6 KERNEL-BASED CONTEXT-DEPENDENT FUSION 100

6.1 Kernel-based context-dependent fusion 102
6.2 Experimental results 107
6.3 Discussion 112

7 CONCLUSIONS 114

APPENDIX

A UPDATE EQUATIONS FOR CONTEXT-DEPENDENT FUSION WITH REGULARIZATION 116

B RECONSTRUCTION ERROR WITH THE MEMBERSHIPS IN THE FEATURE SPACE 124

C UPDATE EQUATIONS FOR KERNEL-BASED CONTEXT-DEPENDENT FUSION 129

REFERENCES 134

BIOGRAPHICAL SKETCH 144

LIST OF TABLES

Table page

2-1 Update equations of CDF and CDF-R 35

2-2 Landmine data set (N = number of data points, K = number of clusters, d = feature dimension) 36

2-3 Experimental results from data set I 39

2-4 Experimental results from data set II 40

3-1 Classification error on each set of character-pair data 57

4-1 Experimental results on D_7 (N = number of clusters correctly identified, $var(N)$ = variance of N) 72

4-2 Experimental results using UCI data sets (N = number of samples, d = data dimension, K' = number of classes) 79

5-1 SVM and its variants 98

5-2 Error rates on UCI data sets (N = number of data points, d = data dimension) . 98

6-1 Update equations of CDF-R and K-CDF 105

6-2 Experimental results from the subset of data set II 110

6-3 Confidence interval of the number of false alarms with a 90% confidence level 111

6-4 Experimental results from hyperspectral data set 112

6-5 Confidence interval of the number of false alarms with a 90% confidence level 112

LIST OF FIGURES

Figure page

1-1 Block diagram of context-dependent fusion. 14

1-2 Block diagram of the kernel-based context-dependent fusion. 15

2-1 Ensemble methods. 22

2-2 Combination methods of classifier outputs. 25

2-3 Test phase of context-dependent fusion. 30

2-4 ROC curves for a set of classifiers from data set I 38

2-5 ROC curves for a set of classifiers from data set II 39

3-1 Estimated principal components using reconstruction error with (a) one and (b) two principal components. 46

3-2 The first principal component using (a) PCA and (b) RF-PCA. 48

3-3 The first principal component from (a) K-PCA on clean data, (b) K-PCA on noisy data, and (c) RKF-PCA on noisy data. 54

3-4 (a) K-PCA on clean data, (b) K-PCA on noisy data, and (c) RKF-PCA on noisy data. 55

3-5 Error histogram between eigen-systems of clean and noisy data using K-PCA and RKF-PCA. 56

3-6 Confidence intervals of the error rate 58

3-7 Error rate with respect to noise ratio. 59

4-1 Sensitivity to initialization in FCM. 71

4-2 Incremental seed selection in G-FCM 72

4-3 Number of clusters correctly identified with respect to noise ratio. 73

4-4 Clustering results using (a) FCM and (b) K-FCM-C with random initialization. . 74

4-5 Error rate with respect to $d_{between}$ in $D_{parallel}$ 75

4-6 Variance of error rate with respect to $d_{between}$ in $D_{parallel}$ 75

4-7 Non-linearly separable clusters. 76

4-8 Error rate with respect to $d_{between}$ in $D_{non-linear}$. 76

5-1 Mapping function for H-FSVM. 90

5-2 Reconstruction error $e(x)$. 91
5-3 Rescaling function for FSVM-N ($\sigma_{N,1} < \sigma_{N,2} < \sigma_{N,3}$). 94
5-4 Decision boundaries of SVM and FSVM-I. 95
5-5 Decision boundaries of SVM and I-FSVM. 97
6-1 Block diagram of kernel-based context-dependent fusion. 101
6-2 Test phase of kernel-based context-dependent fusion. 107
6-3 Artificial data set. 108
6-4 ROC curves from an artificial data set. 109
6-5 ROC curves for a set of classifiers from the subset of data set II. 110
6-6 ROC curves for a set of classifiers from hyperspectral data. 111

LIST OF ALGORITHMS

Algorithm page

2-1 Context-dependent fusion 32

2-2 Context-dependent fusion with regularization 36

3-1 RF-PCA 48

3-2 RKF-PCA 53

4-1 Fast global k-means 64

4-2 K-FCM with Cauchy kernel 68

4-3 KG-FCM with Cauchy kernel 68

4-4 Average commute time 69

4-5 K-FCM with random walk kernel 70

4-6 KG-FCM with random walk kernel 71

5-1 Fuzzy SVM for noisy data 95

6-1 Kernel-based context-dependent fusion 106

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

ROBUST KERNEL METHODS IN CONTEXT-DEPENDENT FUSION

By

Gyeongyong Heo

December 2009

Chair: Paul Gader
Major: Computer Engineering

Combining classifiers, a common way to improve the performance of classification, has gained popularity in recent years. By combining classifiers, one can take advantage of the diversity of the classifiers, while some defects of a classifier can be compensated for by other classifiers. In this dissertation, a classifier combination method, one specifically applicable to the combination of classifier outputs – commonly called *fusion* – is investigated. The proposed fusion method focuses on the combination of classifier outputs with the help of feature space information that is referred to as *context*. The basic assumption in the proposed method is that a context corresponds to a homogeneous region in the feature space. By dividing the feature space into a given number of homogeneous regions, one can identify the same number of contexts, then a different fusion process can be developed for each context, hence the name *context-dependent fusion* (CDF).

The context-dependent fusion algorithm is an iterative method simultaneously clustering the feature space and learning optimal parameters for fusion. Although CDF has several advantages over previous methods, it is limited in that it is only valid for convex clusters with linearly separable classes. To mitigate the convex cluster assumption, a modified CDF using regularization, called *context-dependent fusion with regularization* (CDF-R), is formulated. By adding the regularization terms, not only does CDF-R achieve noise robustness, the main purpose of regularization, but

the consequent clusters, which need not be convex, result in better performance than CDF. Although CDF-R is better at classification than CDF, the linear separability does not change. To completely remove the limitation, CDF is transformed to be non-linear, termed *kernel-based context-dependent fusion* (K-CDF). K-CDF adopts modified kernel methods to remove the restrictions of CDF and remedies some problems in the original kernel methods. K-CDF consists of three main components: dimension reduction, feature space clustering, and fusion. For each component, *robust kernel fuzzy principal component analysis* (RKF-PCA), *kernel-based global fuzzy c-means* (KG-FCM), and *fuzzy support vector machine for noisy data* (FSVM-N) are formulated, and which correspond to the robust variant of kernel PCA, kernel FCM, and fuzzy SVM, respectively. Although the three modifications were originated to address different shortcomings, one common purpose is to reduce the effect of nose, i.e., making the kernel methods *noise-robust*. By combining the three robust kernel methods, not only does K-CDF overcome the convex cluster assumption and linearly separable restriction, but it achieves noise robustness and better performance than previous methods.

CHAPTER 1
INTRODUCTION

Searching for patterns in data is a fundamental problem that has a long history. One of the main components of the problem, classification, assigns each data point to one of a finite number of categories. Since Fisher introduced the method of discriminant analysis in the 1930s [1], numerous algorithms for classification have been developed ranging from simple linear discriminant analysis to the emerging kernel-based methods [2]. Although the latest classification methods perform better than previous ones and have been applied in many areas successfully, some of the methods are applicable only to specific applications due to the requirement of a large data set and the increase in time and space complexity. The fact that classifiers themselves are approaching their limits has driven researchers into other areas of classification systems. Combining classifiers appears to be a natural step forward when a critical mass of knowledge of single classifier models has been accumulated. Although there are many unanswered questions about matching classifiers to real world problems, combining classifiers is a rapidly growing field in the pattern recognition and machine learning communities.

In this dissertation, a new fusion method, called *context-dependent fusion* (CDF) [3], [4], that uses feature space information to define contexts is described first. The concept of context-based fusion was proposed by Paul Gader of the University of Florida in collaboration with Hichem Frigui of the University of Louisville. There have been several attempts to modify and enhance the performance of CDF of which this study is one. CDF is a generalization of the previous fusion methods and uses feature and decision-level information together. Then two variants of CDF are proposed to improve the performance of CDF. The first one is a noise-robust method using regularization, called *context-dependent fusion with regularization* (CDF-R). CDF-R achieves the noise-robustness by adopting regularization, which is widely used to reduce the effect of

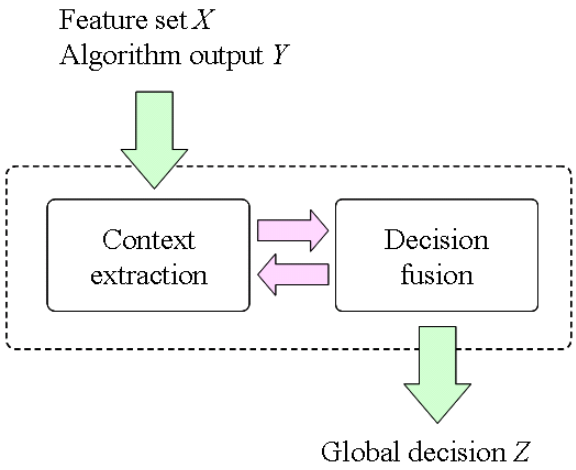


Figure 1-1. Block diagram of context-dependent fusion.

noise. The second one is a non-linear variant adopting kernel methods, called *kernel-based context-dependent fusion* (K-CDF). To implement K-CDF, each component of CDF is kernelized using a robust variant of existing kernel-based method. Specifically, three robust kernel-based methods, *robust kernel fuzzy principal component analysis*, *kernel-based global fuzzy c-means*, and *fuzzy support vector machine for noisy data*, are proposed.

Context-dependent fusion, as shown in Figure 1-1, is composed of two main components, context extraction and decision fusion. To extract context, the input data are divided into homogeneous regions in the feature space and in each of the regions a different fusion process arises. The optimization of each component can be represented as a minimization problem of some objective function. In CDF, the two components are integrated into one objective function that makes it possible to optimize them simultaneously. In Chapter 2, the motivation and structure of CDF is introduced. In CDF, a modified fuzzy c-means (FCM) with feature discrimination is used to extract contexts and the weighted average is used for fusion. During clustering, the feature discrimination function can automatically weight each feature according to its relevance. Its robust variant using regularization, termed *context-dependent fusion with regularization*, is also described in Chapter 2.

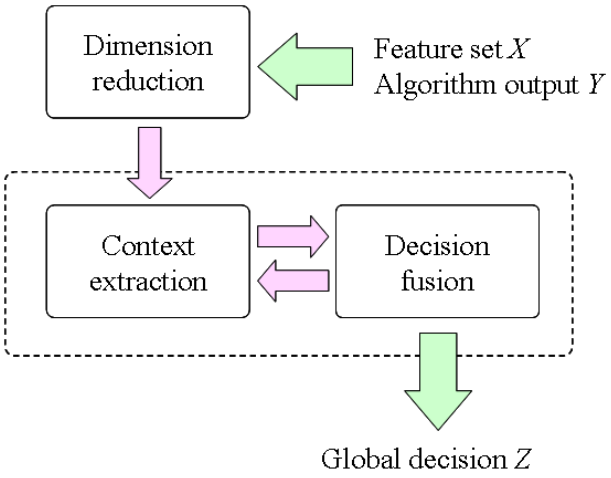


Figure 1-2. Block diagram of the kernel-based context-dependent fusion.

Although CDF is a simple and efficient method for fusion, its basic limitation is that neither modified fuzzy clustering for context extraction nor weighted average for fusion can accommodate non-linear cases. Although CDF-R is better than CDF in classification accuracy, CDF-R has the same problems as CDF. While increasing the number of clusters in context extraction can mitigate the problem, the increase in the number of clusters also increases the time and space complexity. Moreover, to get a stable solution, more training points are needed, which is often impossible in real world applications. Over-fitting also can be a non-trivial problem.

To overcome the limitations of CDF, a generalized CDF, called *kernel-based context-dependent fusion*, is formulated. There have been several attempts to extend or generalize linear methods into corresponding non-linear ones and the kernel-based method is the most promising [5]. Kernel methods approach pattern recognition problems by mapping the data into a high dimensional feature space, where each co-ordinate corresponds to one data item. In that space, a variety of methods can be used to find relations in the data. Since the mapping can be quite general, the relations found in this way are not necessarily linear. Algorithms capable of operating with kernels include support vector machine (SVM), Fisher's linear discriminant analysis, principal components analysis (PCA), ridge regression, spectral clustering, and many others [5].

K-CDF is composed of the two components in CDF – context extraction and decision fusion – and one additional component for dimension reduction, as depicted in Figure 1-2. In CDF, a modified fuzzy c-means simultaneously performs the context extraction and feature weighting. Compared to the original data dimensionality, the dimensionality of the data in kernel methods is very high and sometimes assumed to be infinite. Consequently, feature weight cannot be decided easily in kernel-based methods. In K-CDF, therefore, feature space clustering and feature discrimination are divided into two components and modified kernel PCA and global fuzzy c-means (G-FCM), respectively, are used on each.

First, the input data is processed using modified kernel PCA (K-PCA), which corresponds to feature discrimination in CDF. PCA is widely used for dimensionality reduction and feature extraction in pattern recognition. Although PCA has been applied in many areas successfully, it suffers from sensitivity to noise and is limited to linear principal components. The noise sensitivity problem comes from the sum of squares measure used in PCA and can be alleviated using robust estimation. The limitation of linear components originates from the fact that PCA uses an affine transform defined by eigenvectors of the covariance matrix. This problem can be attacked by using kernels to non-linearly transform the data. In Chapter 3, a robust variant of kernel PCA, which extends kernel PCA of Schölkopf et al. [6] and uses fuzzy memberships is introduced to tackle the two problems simultaneously. To derive the method, first an iterative method to find a robust covariance matrix, robust fuzzy PCA (RF-PCA), is introduced. The RF-PCA method is then extended to a non-linear one, *robust kernel fuzzy PCA* (RKF-PCA) [7], using kernels. By introducing fuzzy memberships into K-PCA, RKF-PCA achieves better noise-resistance than K-PCA.

For context extraction, K-FCM is considered. This is a direct extension of FCM into a non-linear method. FCM is a simple but powerful clustering method using the concept of a fuzzy set. After its introduction, FCM has proven to be successful in many

areas. There are, however, several well known problems with FCM, such as sensitivity to initialization, sensitivity to outliers, and limitation to linearly separable clusters. In Chapter 4, global fuzzy c-means (G-FCM) and kernel fuzzy c-means (K-FCM) are combined and extended to resolve the shortcomings mentioned above. FCM generally requires multiple runs with random initialization to obtain the best model, which is time-consuming, and, even more challenging, finding an optimal initialization is known as a NP hard problem. There are several groups of methods to initialize FCM to find a sub-optimal solution and global FCM (G-FCM) is one of them. G-FCM is a variant of FCM using an incremental and deterministic seed selection method, and is efficient in alleviating the sensitivity to initialization. There are also several approaches to relax the burden of noise and non-convex clusters and K-FCM is one of them. K-FCM is considered in Chapter 4 because it can be easily extended using different kernels, as well as being efficient at dealing with FCM problems. The proposed method, *kernel-based global fuzzy c-means* (KG-FCM), is based on G-FCM to avoid the initialization problem, then extended with the help of the kernel method. Specifically, *KG-FCM with Cauchy kernel* is proposed to mitigate the effect of noise and *KG-FCM with random walk kernel* is proposed to effectively manage non-convex clusters.

Chapter 5 develops a robust variant of fuzzy support vector machine (FSVM), called FSVM for noisy data (FSVM-N). SVM is a theoretically well motivated algorithm developed from statistical learning theory, that has shown good performance in many fields. In spite of its success, it still suffers from a noise sensitivity problem. To relax the problem, SVM was extended by the introduction of fuzzy memberships and resulted in fuzzy SVM, which also has been extended further in two ways: by adopting a different objective function with the help of domain-specific knowledge and by employing a different membership calculation method. In Chapter 5, a new membership calculation method that belongs to the second group is proposed. It is different from previous ones in that it does not rely on circular assumptions about the data distribution and

does not need any prior knowledge. The proposed method is based on reconstruction error, which measures the agreement between the overall data structure and a data point. The reconstruction, previously used to smooth the data by suppressing noise, demonstrated successful results. Thus the reconstruction error can represent the degree of outlier-ness and help to achieve accurate classification. The SVM variant with fuzzy memberships (FSVM) is used as a fusion method in K-CDF to accommodate non-linearly separable cases in each context.

The common aim of Chapters 3, 4, and 5 is to develop a robust non-linear method to address the shortcomings in CDF. By adopting kernel methods, the components in CDF are transformed into non-linear ones and the fuzzy memberships are used to achieve noise-robustness. Using all the robust methods, *kernel-based context-dependent fusion* is formulated in Chapter 6.

Although Chapters 2, 3, 4, and 5 lead to the development of K-CDF, each chapter addresses a separate problem or limitation. The three robust kernel methods (RKF-PCA, KG-FCM, and FSVM-N) are independent of each other and can be used in other pattern recognition problems, as demonstrated in each chapter. Therefore, in this dissertation, each chapter has its own survey of related researches, experiments, and discussions and no separate chapter is devoted to a literature survey.

CHAPTER 2
CONTEXT-DEPENDENT FUSION

Classification problems [2], [8] are a major category of data analysis that are applied to pattern recognition, machine learning, statistical inference, and, recently, data mining. Classification methods represent a set of supervised learning techniques where a set of dependent variables needs to be predicted based on a set of input variables. Classification techniques have attracted the attention of researchers from various fields and a variety of methods, such as decision trees, rule based methods, neural networks, Bayesian methods, and support vector machines, are used to address classification problems. Classification techniques have been successfully applied to many real world problems, although it is also generally accepted that there is no one best way to solve the problems and it may be futile to debate which type of classification technique is best [9].

In spite of the successes, all of the classification methods have their own limitations, which makes researchers investigate other areas of classification system. Combining multiple classifiers to obtain improved performance, generally called *multiple classifier system* (MCS) [10], was developed as a practical and effective approach to solve the limitation of single classifier systems. From the beginning, this approach has produced promising results and research in this domain has increased significantly, partly as a result of advances in the classification technology itself. Combining multiple classifiers can be considered as a generic pattern recognition problem in which the input consists of the results of the individual classifiers, and the output is the combined decision.
For this purpose, many current classification techniques can be applied. MCS has a surprisingly long history. For example, the Borda count for combining multiple rankings is named for the eighteenth century French mathematician Jean-Charles de Borda, who devised the system in 1770. MCS went through parallel routes within several disciplines, for example, pattern recognition, machine learning, and information theory. Although

there is still no agreement on the vocabulary of MCS, the two most common words in this area are *ensemble* and *fusion*. Although they are sometimes used interchangeably, in this dissertation they are used to indicate different parts of MCS. The term *ensemble* or *classifier ensemble* is usually used to indicate the selection and structure of the classifiers used in MCS. On the other hand, the terms *fusion* or *decision fusion* are used to emphasize the process of combining classifier outputs, which is the main concern of this chapter.

In this chapter, we describe an extended fusion method, called *context-dependent fusion* (CDF) [3], [4], which is a starting point of this dissertation. CDF is a combination of traditional fusion and feature space clustering. It is also can be seen as a local approach that takes different fusion processes to different regions of the feature space. It can take advantages of the strengths of a few classifiers in different regions of the feature space without being affected by the weaknesses of the other classifiers. The feature space information is referred to as the *context* of the data point and helps the fuser to make robust decisions.

In the next section, multiple classifier system is briefly overviewed. In section 2.2, we put a focus on *fusion*, and describe the motivation of the proposed method. In section 2.3, *context-dependent fusion*, is described. An extension of CDF, *context-dependent fusion with regularization* (CDF-R), is introduced in section 2.4. Experimental results are given in section 2.5 and a discussion section ends this chapter.

2.1 Multiple classifier system

Multiple classifier system has been applied to various fields of pattern recognition, including character recognition [11], speech recognition [12], and text categorization [13]. The concept of classifier combination is motivated by observation of classifiers' complementary characteristics which lead to better accuracy for the whole than for any of its individual classifiers. It is desirable to take advantage of the strengths of individual classifiers and to avoid their weaknesses. A necessary and sufficient condition for an

ensemble of classifiers to be more accurate than any individual classifiers is that the classifiers are both *accurate* and *diverse* [14]. An accurate classifier is one that has an error rate better than random guessing on a new sample; two classifiers are diverse if they make different errors on new data points. In most applications, the condition is assumed to be satisfied and the superiority of MCS over single classifier system has been demonstrated experimentally.

Dietterich [15] also suggests three reasons why a classifier ensemble might be better than a single classifier. The first reason is *statistical*. A learning algorithm can be viewed as searching a hypothesis space to find the best hypothesis. The statistical problem arises when the amount of training data is too small compared to the size of the hypothesis space. Without sufficient data, the learning algorithm can find many different hypotheses that all give the same accuracy on the training data. By constructing an ensemble out of all accurate classifiers, the algorithms can average their votes and reduce the risk of making a wrong decision. The second reason is *computational*. Many learning algorithms work by performing a local search that may get stuck in local optima. In cases where there is enough training data so that the statistical problem is absent, it may still be very difficult computationally for the learning algorithm to find the best hypothesis. For example, optimal training of neural networks is a NP hard problem [16], [17]. An ensemble constructed by running the local search from many different starting points may provide a better approximation of the true hypothesis than any of the individual hypothesis. The third reason is *representational*. In some applications, the true hypothesis cannot be represented by any of the hypotheses found by a classifier. By forming weighted sums of hypotheses, it may be possible to expand the hypothesis space. These three issues are the three most common ways in which existing learning algorithms fail. Hence, ensemble methods have the promise of relaxing the shortcomings of single classifier system.

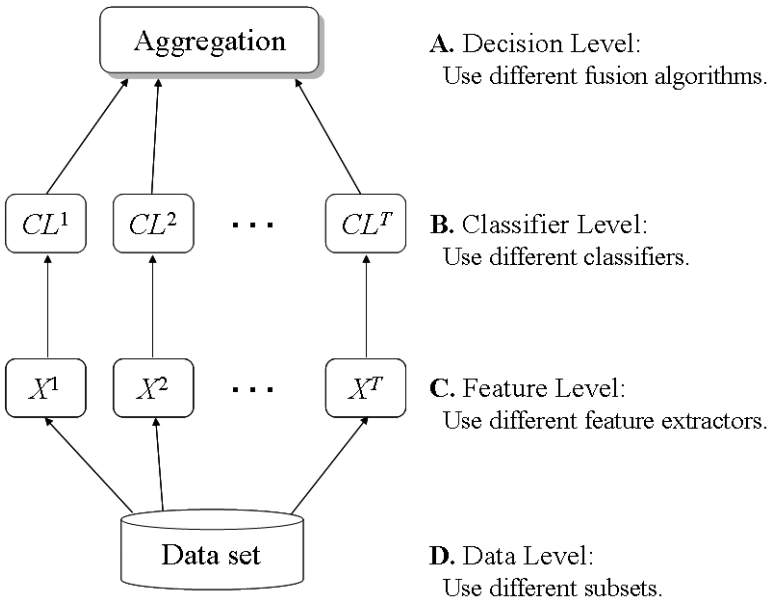


Figure 2-1. Ensemble methods.

MCS has been proved to be superior to single classifier system both theoretically and experimentally, there have been numerous methods proposed to construct MCS [10]. The methods can be represented in several ways and Figure 2-1 shows one of them. The diagram in Figure 2-1 illustrates four levels used in building ensembles of diverse classifiers. At the data level, the data set can be modified with different pre-processing methods or different data subsets can be selected so that each classifier in the ensemble is trained on its own data set. Feature extraction may also be related to a specific classifier and each extractor generates its own feature vector for the corresponding classifier. The methods at the classifier level can be divided roughly into two types: using one base classifier and using several base classifiers. Many ensemble paradigms employ the former approach, but there is no evidence that this strategy is better than using different classifiers and the latter approach is used in this study. The decision level focuses on the methods of combining classifier outputs, then generating one global decision. This is commonly called *decision fusion* or simply *fusion*. *Context-dependent fusion* focuses on the combination level with the help of feature level information.

2.2 Decision fusion

Over the past years, a variety of schemes have been proposed for combining multiple classifier outputs. The most representative approaches include voting [18], [19], Borda count [20], [21], Bayesian [22], [23], neural network [24], [25], Dempster Shafer [26], [27], bagging and boosting [28], [29], and fuzzy integrals [30], [31], just to name a few. These methods for fusion may take one of the two approaches: *classifier fusion* and *classifier selection*.

In classifier fusion, all classifiers are supposed to be equally experienced in the whole feature space and their outputs are combined in some manner to achieve a consensus. Although classifier fusion methods combine the results of several individual classifiers, they do not take into account the local expertise of each classifier. This can mislead the consensus of multiple classifiers by overlooking the opinion of some better skilled classifiers in a specific region to which the given input belongs.

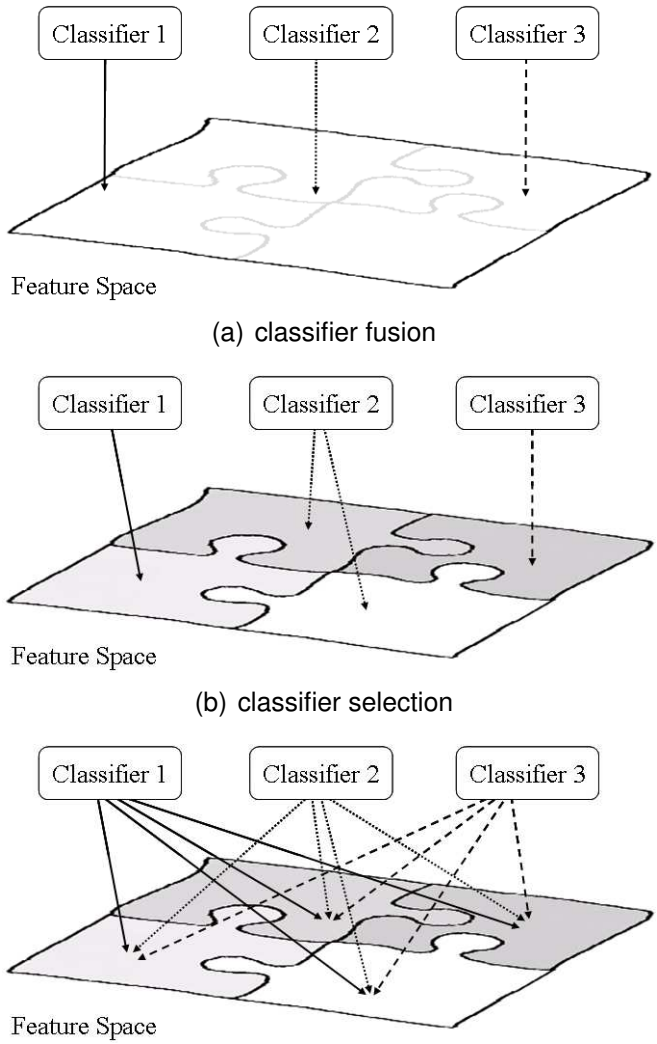
Sometimes it is useful to decompose a complex problem into simpler sub-problems and solve each sub-problem one by one instead of learning the global relation between input variables and target variables. Classifier selection methods take the divide-and-conquer approach by dividing the feature space into some homogeneous regions and assigning one or more classifiers to each region. In classifier selection methods, a method of partitioning the feature space and estimating the performance of each classifier in each partition is crucial. Woods et al. proposed a method called *dynamic classifier selection by local accuracy* [32]. The basic concept of it is to estimate each classifier's accuracy in local regions of feature space surrounding an unknown test sample, and use the decision of the most locally accurate classifier. This method, however, was too time-consuming due to the need for an accuracy estimation for each test sample. In the clustering-and-selection method [33], Kuncheva presented an algorithm to statistically select the best classifier. In this method, the training data are clustered to form the decision regions, and one locally best classifier is selected

based on local accuracy. However, the method was not fully generalized to multiple classifiers for one region. Liu and Yuan [34] proposed a modified version of the clustering-and-selection method, that tried to take advantage of class labels. With each classifier, the training samples are divided into correctly and incorrectly classified samples, which are then clustered to form a partition of the feature space. Due to the difference between the classifiers' error characteristics, the partitions resulting from different classifiers generally are not the same. In the test phase, the most accurate classifier in the vicinity of the input sample is appointed to make the final decision. The main defect of this method is that each classifier should maintain its own partition, which makes the decision process memory- and time-intensive.

Context-dependent fusion (CDF) is a combination of classifier selection and classifier fusion. In CDF, the feature space is divided into K clusters. A specific homogeneous cluster or region in the feature space is referred to as *context* and a different fusion process arises in different contexts. This is where the term *context-dependent* comes from. Figure 2-2 compares the structures of the three methods for fusion: classifier fusion, classifier selection, and context-dependent fusion.

2.3 Context-dependent fusion

Context-dependent fusion (CDF) is a method of combining multiple classifier outputs with the help of feature space clustering. The training part of CDF has two main components: context extraction and decision fusion. In context extraction, a clustering algorithm is used to partition the training vectors into groups of similar vectors. Here it is assumed that the feature vectors that have similar values share some common characteristics and should be assigned to the same context. After partitioning the feature space, training data from each identified context will be used to learn the context-specific optimal fusion parameters and to identify local experts for that region. The two steps, dividing feature space and learning fusion parameters, alternate to find an optimal model. The objective function of CDF can be written as the sum of two



(a) classifier fusion

(b) classifier selection

(c) context-dependent fusion

Figure 2-2. Combination methods of classifier outputs.

sub-functions that corresponds to the two components, respectively. By integrating the two components into one objective function, CDF simultaneously divides the feature space into K contexts and learns the parameters needed for fusion in each context. In CDF, the *simultaneous clustering and attribute discrimination* (SCAD) algorithm [35], [36] is considered for context extraction and a weighted average is used for fusion.

Context-dependent fusion has several advantages over the previous methods. First of all, as the feature space is divided using modified fuzzy clustering, a test point can be easily assigned to a cluster or several clusters with different degrees. In other

words, it is not difficult to find the context of a test point, which is given as a membership vector to K clusters. Moreover, the membership vector makes it possible for CDF to be noise-robust. There are several methods to make an algorithm noise-robust and introducing memberships is one of the most common approaches. Last but not least, CDF is fully generalized to multiple algorithms for one cluster and it is simple to add or remove a classifier in the framework.

2.3.1 Simultaneous clustering and attribute discrimination

Cluster analysis, or clustering, is a method of dividing a data set into groups of similar objects. Since Zadeh [37] proposed fuzzy sets that produced the idea of partial membership described by a membership function, fuzzy clustering has been widely studied and applied in various areas. In fuzzy clustering, fuzzy c-means (FCM), generalized by Bezdek [38], is the most well-known method. FCM is an algorithm derived from a constrained optimization problem in which the following objective function is optimized iteratively:

$$\begin{aligned} J_{FCM} &= \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} D_{ki}^{2} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} ||v_k - x_i||^2, \end{aligned} \tag{2–1}$$

where N is the number of data points, K is the number of clusters, v_k is the center of the kth cluster, u_{ki} is the membership of the ith point in the kth cluster, D_{ki} is the distance between the ith point and the kth cluster center, and m is a fuzzifier constant. The constraint on the memberships can be written as

$$U_{K\times N} = \left\{ u_{ki} \middle| u_{ki} \in [0,1] \ \forall i \text{ and } k; \sum_{k=1}^{K} u_{ki} = 1 \ \forall i \right\}. \tag{2–2}$$

Although FCM is a simple and efficient method, it is well known that the Euclidean distance used in FCM is noise-sensitive and valid only for circular clusters. Several groups of methods have been developed to mitigate these problems and feature

weighting is one of them. Furthermore, feature weighting is useful in clustering high-dimensional data by reducing the effect of irrelevant features. In this chapter, simplified SCAD [35] is considered as a feature space clustering method. The objective function of SCAD can be written as

$$J_{SCAD} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \sum_{k=1}^{K} \delta_k \sum_{l=1}^{L} r_{kl}^{2}, \tag{2–3}$$

where L is the number of features and d_{kil} represents the feature-wise distance. The distance between a data point $x_i = [x_{i1}, \cdots, x_{iL}]^T$ and a cluster center $v_k = [v_{k1}, \cdots, v_{kL}]^T$ is calculated as a weighted sum of feature-wise distances as

$$D_{ki} = D(x_i, v_k) = \sum_{l=1}^{L} r_{kl}^{q} |v_{kl} - x_{il}|^2. \tag{2–4}$$

The value r_{kl} is the weight of the lth feature in the kth cluster and should satisfy the FCM-like constraint,

$$R_{K\times L} = \left\{ r_{kl} \middle| r_{kl} \in (0, 1] \ \forall k \text{ and } l; \sum_{l=1}^{L} r_{kl} = 1 \ \forall k \right\}, \tag{2–5}$$

and q is a fuzzifier constant of feature weights. One important difference between R and U is that $r_{kl} > 0$, i.e., all the features are assumed to participate in the classification with different degrees. The second term in Equation 2–3 is a regularization term and δ_k is a regularization parameter. SCAD performs clustering and feature weighting simultaneously and has several advantages over traditional clustering methods. First, its continuous feature weighting provides a much richer feature relevance representation than binary feature selection. Second, SCAD learns a different feature relevance representation for each cluster in an unsupervised manner.

The objective function of simplified SCAD (S-SCAD) used in the CDF algorithm can be written as

$$J_{S-SCAD} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2}. \tag{2–6}$$

Using the method of Lagrange multipliers, update equations for S-SCAD can be obtained:

$$u_{ki} = \frac{\left(\frac{1}{\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K}\left(\frac{1}{\sum_{l=1}^{L} r_{al}^{q} d_{ail}^{2}}\right)^{\frac{1}{m-1}}} = \frac{\left(\frac{1}{D_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K}\left(\frac{1}{D_{ki}}\right)^{\frac{1}{m-1}}}, \tag{2–7}$$

$$v_{kl} = \frac{\sum_{i=1}^{N} u_{ki}^{m} x_{il}}{\sum_{i=1}^{N} u_{ki}^{m}}, \tag{2–8}$$

$$r_{kl} = \frac{\left(\frac{1}{\sum_{i=1}^{N} u_{ki}^{m} d_{kil}^{2}}\right)^{\frac{1}{q-1}}}{\sum_{a=1}^{L}\left(\frac{1}{\sum_{i=1}^{N} u_{ki}^{m} d_{kia}^{2}}\right)^{\frac{1}{q-1}}}. \tag{2–9}$$

2.3.2 Context-dependent fusion

Assume there are N training points with desired output $O = \{o_i | i = 1, \ldots, N\}$. The data points are processed using T algorithms, each of which generate confidence values for data points $y_t = \{y_{ti} | i = 1, \ldots, N\}$. Each algorithm has its own feature set, $X_t = \{x_{i,t} | i = 1, \ldots, N\}$, and the T feature sets are concatenated to generate one global descriptor:

$$X = \bigcup_{t=1}^{T} X_t = \left\{ x_i = [x_{i1}, \ldots, x_{iL}]^T \mid i = 1, \ldots, N \right\}, \tag{2–10}$$

where L is the overall feature dimension and can be represented as the sum of the number of features used by the tth algorithm, L_t:

$$L = \sum_{t=1}^{T} L_t. \tag{2–11}$$

The contexts are defined as homogeneous regions of the feature space and it is also assumed that there are K contexts or clusters. The objective function of CDF can be

written as

$$
\begin{aligned}
J_{CDF} &= \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i \right)^2, \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} |x_{il} - v_{kl}|^2 + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i \right)^2.
\end{aligned}
\tag{2–12}
$$

The first term of Equation (2–12) partitions the N samples into K clusters using S-SCAD and the second term attempts to learn a cluster-dependent aggregation of the T algorithm outputs. The constant α serves to balance the two terms. There are two sets of weights in Equation (2–12), feature weights r_{kl} and classifier weights w_{kt}. The former puts a weight on the lth feature in the kth cluster. The latter weight, w_{kt}, puts a weight on the output value of the tth algorithm in the kth cluster and affects the cluster output of the kth cluster. In the test phase, when a data point x and its algorithm outputs $y = [y_1, \cdots, y_T]^T$ are given, the memberships are decided using Equation (2–7) and each cluster generates a cluster output as a weighted sum of the T algorithm outputs as

$$
y'_k(x) = \sum_{t=1}^{T} w_{kt} y_t. \tag{2–13}
$$

The final output is given as a membership-weighted sum of the K cluster outputs:

$$
z(x) = \sum_{k=1}^{K} u_k^m y'_k(x) = \sum_{k=1}^{K} u_k^m \left(\sum_{t=1}^{T} w_{kt} y_t \right). \tag{2–14}
$$

Both the weights should satisfy the FCM-like sum-to-one constraint. However, the feature weight r_{kl} should be greater than zero and the classifier weight w_{kt} is allowed to have negative values. In a specific cluster, a classifier might generate negatively correlated output values with the desired outputs, yet negative weight for that classifier can make the classifier still effective in classification. The constraint for the feature weights can be written as Equation (2–5) and the constraint for the classifier weights can be written as

$$
W_{K\times T} = \left\{ w_{kt} \,\middle|\, w_{kt} \in \mathcal{R}\ \forall k \text{ and } t; \sum_{t=1}^{T} w_{kt} = 1\ \forall k \right\}. \tag{2–15}
$$

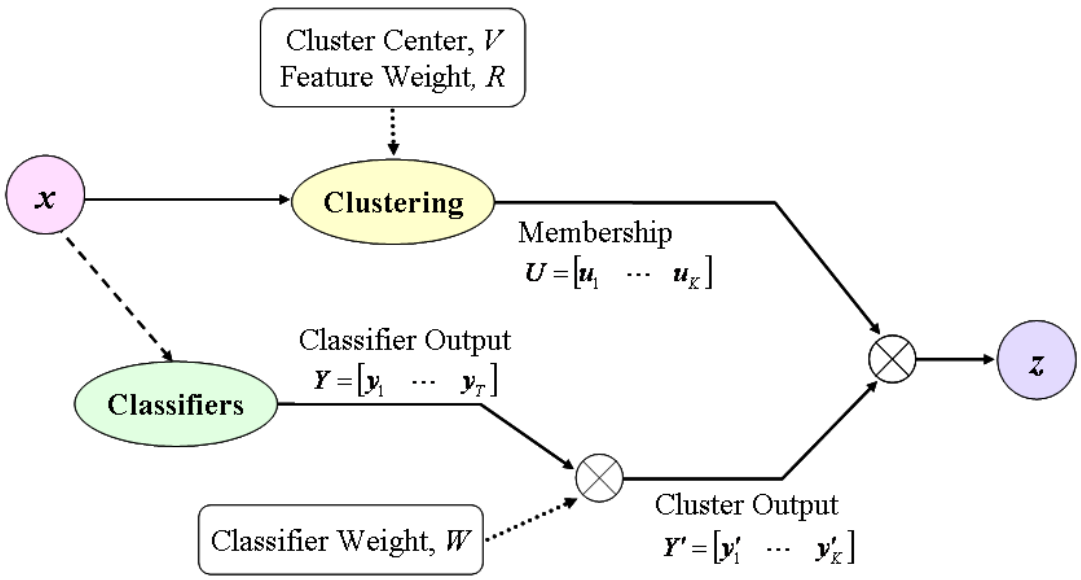


Figure 2-3. Test phase of context-dependent fusion.

Figure 2-3 shows the procedure generating final output value from an input point. In Figure 2-3, three matrices, V, R, and W, are the optimized matrices during the training phase.

In the training phase, four matrixes, V, R, W, and U, that minimize the objective function should be found, and the update equations for them can be obtained using the method of Lagrange multipliers. The update equations can be written as

$$u_{ki} = \frac{\left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K}\left(\frac{1}{\widetilde{D}_{ai}}\right)^{\frac{1}{m-1}}}, \tag{2–16}$$

$$v_{kl} = \frac{\sum_{i=1}^{N} u_{ki}^{m} x_{il}}{\sum_{i=1}^{N} u_{ki}^{m}}, \tag{2–17}$$

$$r_{kl} = \frac{\left(\frac{1}{\overline{D}_{kl}}\right)^{\frac{1}{q-1}}}{\sum_{a=1}^{L}\left(\frac{1}{\overline{D}_{ka}}\right)^{\frac{1}{q-1}}}, \tag{2–18}$$

$$w_{kt} = \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{\substack{a=1 \\ a \neq t}}^{T} w_{ka} y_{ai} \right) y_{ti} - \zeta_k}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2}}, \tag{2–19}$$

where

$$\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i \right)^{2}, \tag{2–20}$$

$$\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^{m} d_{kil}^{2}, \tag{2–21}$$

$$\zeta_k = \frac{\sum_{a=1}^{T} \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti} \right) y_{ai}}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2}}}{\sum_{a=1}^{T} \frac{1}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2}}}. \tag{2–22}$$

Equations (2–16) and (2–17) are similar to the update equations of FCM except that the distance between a data point and a center is calculated based on the Euclidean distance between them and aggregated result (Equation (2–20)). The feature weight also can be calculated in a similar way to the calculation of memberships (Equation (2–18)). The numerator of Equation (2–19) is composed of two terms. The first term is the actual weight decision term and the second term is added due to the sum-to-one constraint of w_{kt}. A weight for the tth algorithm in the kth cluster is calculated based on the difference between desired output and calculated output using $(T-1)$ algorithms except the tth algorithm. The tth algorithm is excluded because the difference is the portion of the tth algorithm in the kth cluster. The detailed derivation can be found in Appendix A[1] .

[1] Appendix A concerns the derivation of the update equations of *context-dependent fusion with regularization*. The update equations for *context-dependent fusion* can be obtained by equating the regularization parameters, β, γ, and δ, to zero.

Algorithm 2-1 Context-dependent fusion

Input: X : data set $(N \times L)$, Y : classifier output $(N \times T)$, O : target value $(N \times 1)$

1: Initialize randomly U and W.
2: Initialize R to $1/L$.
3: **repeat**
4: Update cluster centers V using (2–17).
5: Update feature weights R using (2–18).
6: Update classifier weights W using (2–19).
7: Update memberships U using (2–16).
8: **until** V, R, W, and U satisfy the convergence criteria
9: Update memberships U using (2–7).
10: **repeat**
11: Update classifier weight W using (2–19).
12: **until** W satisfies the convergence criterion
13: **return**

In Equation (2–20), as the distance is decided based on the weighted Euclidean distance and combined algorithm outputs, the boundary of the cluster is generally not convex in the feature space. Although the cluster boundaries need not to be convex, noise also can cause the non-convex boundaries, i.e., over-fitting. To reduce the effect of noise and make the cluster boundaries smooth, a post-processing routine is employed where the membership matrix is re-calculated only using the Euclidean distance (line 9 in Algorithm 2-1). In the test phase, the membership values for test data are also calculated using Equation (2–7) because the target values of test data are not available. Finally, with the fixed U, R, and V, the classifier weight matrix is updated until convergence (line 10 $\sim$ 12 in Algorithm 2-1). The training phase of the *context-dependent fusion* algorithm is summarized in Algorithm 2-1.

2.4 Context-dependent fusion with regularization

In CDF, the post-processing routine is adopted to reduce the effect of noise. In this section, instead of post-processing, *regularization* is introduced to enhance the performance of CDF, which results in *context-dependent fusion with regularization* (CDF-R).

Regularization is a way to obtain solutions using prior requirements [39] and has been applied to various problems to make algorithms noise-robust. Regularization also has been applied to fuzzy clustering and some objective functions have been formulated as follows [40]–[43]:

$$J_{Entropy} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}||x_i - v_k||^2 + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki} \log u_{ki}, \tag{2–23}$$

$$J_{Quadratic} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}||x_i - v_k||^2 + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^2, \tag{2–24}$$

$$J_{Polynomial} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m||x_i - v_k||^2 + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m. \tag{2–25}$$

Although each regularization term is slightly different from the others, all the regularization terms are minimized when all the u_{ki} have the same value, $u_{ki} = 1/K$ for all $1 \leq i \leq N$ and $1 \leq k \leq K$, which means that the regularization terms prevent u_{ki} from having extreme values, 0 and 1. In this chapter, the polynomial regularization term is used because only that makes it possible to obtain a closed-form solution for the update equations without modifying the objective function of CDF.

In CDF-R, three polynomial regularization terms, for memberships, feature weights, and classifier weights, are added. The objective function for CDF-R is defined as

$$\begin{aligned} J_{CDF-R} = &\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i\right)^2 \\ &+ \beta \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^q + \delta \sum_{k=1}^{K}\sum_{t=1}^{T} w_{kt}^2, \end{aligned} \tag{2–26}$$

where β, γ, and δ are regularization parameters. The update equations for CDF-R using the method of Lagrange multipliers results in the same equations as CDF, except for the update equation for classifier weight. The update equation for the classifier weight can

be written as

$$w_{kt} = \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{\substack{a=1 \\ a \neq t}}^{T} w_{ka} y_{ai} \right) y_{ti} - \zeta_k}{\sum_{i=1}^{N} u_{iki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}}, \tag{2–27}$$

and the terms in Equations (2–20), (2–21), and (2–22) should be modified as

$$\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i \right)^{2} + \beta, \tag{2–28}$$

$$\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^{m} d_{kil}^{2} + \gamma, \tag{2–29}$$

$$\zeta_k = \frac{\sum_{a=1}^{T} \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti} \right) y_{ai} - w_{ka} \frac{\delta}{\alpha}}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}}}{\sum_{a=1}^{T} \frac{1}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}}}. \tag{2–30}$$

The detailed derivation can be found in Appendix A and the training phase of CDF-R can be summarized as Algorithm 2-2. Table 2-1 compares the update equations of CDF and CDF-R. As is shown in Table 2-1, in CDF-R, each update equation except v_{kl} has a regularization parameter. The regularization parameter puts a limit on the minimum value. For example, $\overline{D}_{kl}$ in CDF can be zero but in CDF-R, it should be greater than or equal to γ. After normalization, therefore, r_{kl} in CDF-R will have more similar values each others than in CDF, which means that regularization prevents the values from having extreme values. This is the way that regularization in fuzzy clustering and CDF-R works to make algorithms noise-robust. Although there is no regularization parameter in the update equation for cluster centers, they are also affected indirectly through memberships.

Table 2-1. Update equations of CDF and CDF-R

Context-dependent fusion	Context-dependent fusion with regularization
$J_{CDF} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i\right)^2$	$J_{CDF-R} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i\right)^2 + \beta \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^q + \delta \sum_{k=1}^{K}\sum_{t=1}^{T} w_{kt}^2$
$u_{ki} = \left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}} \Big/ \sum_{a=1}^{K} \left(\frac{1}{\widetilde{D}_{ai}}\right)^{\frac{1}{m-1}}$	$u_{ki} = \left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}} \Big/ \sum_{a=1}^{K} \left(\frac{1}{\widetilde{D}_{ai}}\right)^{\frac{1}{m-1}}$
where $\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i\right)^2$	where $\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i\right)^2 + \boldsymbol{\beta}$
$v_{kl} = \sum_{i=1}^{N} u_{ki}^m x_{il} \Big/ \sum_{i=1}^{N} u_{ki}^m$	$v_{kl} = \sum_{i=1}^{N} u_{ki}^m x_{il} \Big/ \sum_{i=1}^{N} u_{ki}^m$
$r_{kl} = \left(\frac{1}{\overline{D}_{kl}}\right)^{\frac{1}{q-1}} \Big/ \sum_{a=1}^{L} \left(\frac{1}{\overline{D}_{ka}}\right)^{\frac{1}{q-1}}$	$r_{kl} = \left(\frac{1}{\overline{D}_{kl}}\right)^{\frac{1}{q-1}} \Big/ \sum_{a=1}^{L} \left(\frac{1}{\overline{D}_{ka}}\right)^{\frac{1}{q-1}}$
where $\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^m d_{kil}^2$	where $\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^m d_{kil}^2 + \boldsymbol{\gamma}$
$w_{kt} = \dfrac{\sum_{i=1}^{N} u_{ki}^m \left(o_i - \sum_{\substack{a=1 \\ a\neq t}}^{T} w_{ka} y_{ai}\right) y_{ti} - \zeta_k}{\sum_{i=1}^{N} u_{iki}^m y_{ai}^2}$	$w_{kt} = \dfrac{\sum_{i=1}^{N} u_{ki}^m \left(o_i - \sum_{\substack{a=1 \\ a\neq t}}^{T} w_{ka} y_{ai}\right) y_{ti} - \zeta_k}{\sum_{i=1}^{N} u_{iki}^m y_{ai}^2 + \frac{\boldsymbol{\delta}}{\boldsymbol{\alpha}}}$
where $\zeta_k = \dfrac{\sum_{a=1}^{K} \dfrac{\sum_{i=1}^{N} u_{ki}^m \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti}\right) y_{ai}}{\sum_{i=1}^{N} u_{ki}^m y_{ai}^2}}{\sum_{a=1}^{K} \dfrac{1}{\sum_{i=1}^{N} u_{ki}^m y_{ai}^2}}$	where $\zeta_k = \dfrac{\sum_{a=1}^{T} \dfrac{\sum_{i=1}^{N} u_{ki}^m \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti}\right) y_{ai} - \boldsymbol{w_{ka}} \frac{\boldsymbol{\delta}}{\boldsymbol{\alpha}}}{\sum_{i=1}^{N} u_{ki}^m y_{ai}^2 + \frac{\boldsymbol{\delta}}{\boldsymbol{\alpha}}}}{\sum_{a=1}^{T} \dfrac{1}{\sum_{i=1}^{N} u_{ki}^m y_{ai}^2 + \frac{\boldsymbol{\delta}}{\boldsymbol{\alpha}}}}$

Algorithm 2-2 Context-dependent fusion with regularization

Input: X : data set $(N \times L)$, Y : classifier output $(N \times T)$, O : target value $(N \times 1)$

1: Initialize randomly U and W.
2: Initialize R to $1/L$.
3: **repeat**
4: Update cluster centers V using (2–17).
5: Update feature weights R using (2–18).
6: Update classifier weight W using (2–27).
7: Update memberships U using (2–16).
8: **until** V, R, W, and U satisfy the convergence criteria
9: **return**

Table 2-2. Landmine data set (N = number of data points, K = number of clusters, d = feature dimension)

Data set	N	K	Classifier	Sensor	d	Comment
Set I	1000 (266 mines, 734 non-mines)	10	EHD	GPR	40	Low-metal mine vs. non-mine
			HMM	GPR	20	
			SPECT	GPR	20	
Set II	875 (311 mines, 564 non-mines)	8	EHD	GPR	40	Mine vs. non-mine
			SPECT	GPR	18	
			WEMI	EMI	4	

2.5 Experimental results

To investigate the effectiveness of the proposed methods, the *context-dependent fusion* (CDF) and *context-dependent fusion with regularization* (CDF-R) algorithms were applied to landmine detection problem. Two data sets were collected by NIITEK Inc. using two sensors, ground penetrating radar (GPR) and electromagnetic induction (EMI) sensors. Both of the sensors have been widely used in subsurface imaging and landmine detection. Information about the sensors and their applications in landmine detection can be found in [44]. Four classification systems, called HMM [45]–[47], SPECT [48], EHD [49], [50], and WEMI [51], [52], respectively, were trained using the data. These classification systems are up-to-date landmine detection systems developed over the years independently of this work. Each system extracted its own feature set using the subset of data and generated confidence values, both of

which served as inputs to the fusion algorithms. (See the references for more detailed information about the classifier systems and their feature sets.)

Data sets used in this chapter are summarized in Table 2-2. In each data set, a sample corresponds to a landmine, a buried object that is not a landmine, or a blank piece of ground. As the EMI data were not available for data set I, three classifiers using the GPR data were used for test, whereas two classifiers using the GPR data and one using the EMI data were used for data set II. Landmines can be roughly divided into two groups according to the amount of metallic content, high-metal and low-metal mines. High-metal mines are relatively easy to detect using EMI sensors. However low-metal mines are not and even if they are detectable, it is difficult to distinguish them from non-mine objects. Even worse, some mines classified as low-metal mines do not have any metallic content at all. In other words, detecting low-metal mines is a real problem in landmine detection even with up-to-date classification systems. Data set I contains only low-metal mines, whereas data set II contains all types of mines. Data set I is, therefore, made up to show performance improvement for a problem in which the performance improvement using a single classifier is not easy. Data set II is composed of two data subsets from two sites whose environments are quite different from each other. As shown in Table 2-4, any single classifier cannot accommodate two data subsets from different environments. The second experiment is, therefore, designed to show the effectiveness of the proposed methods on the data set composed of several subsets with different contexts. In these experiments, ten-fold cross-validation is used and the numbers are averaged over 10 runs. In each run, the data set is divided into 10 folds randomly. Figures 2-4 and 2-5 show the receiver operating characteristic (ROC) curves of a set of classifiers and fused results for data set I and II respectively. Tables 2-3 and 2-4 summarize the results. In the figures and tables, probability of detection (PD) is the number of mines detected divided by the total number of mines; probability of false alarm (PFA) is the number of non-mines declared to be mines divided by the

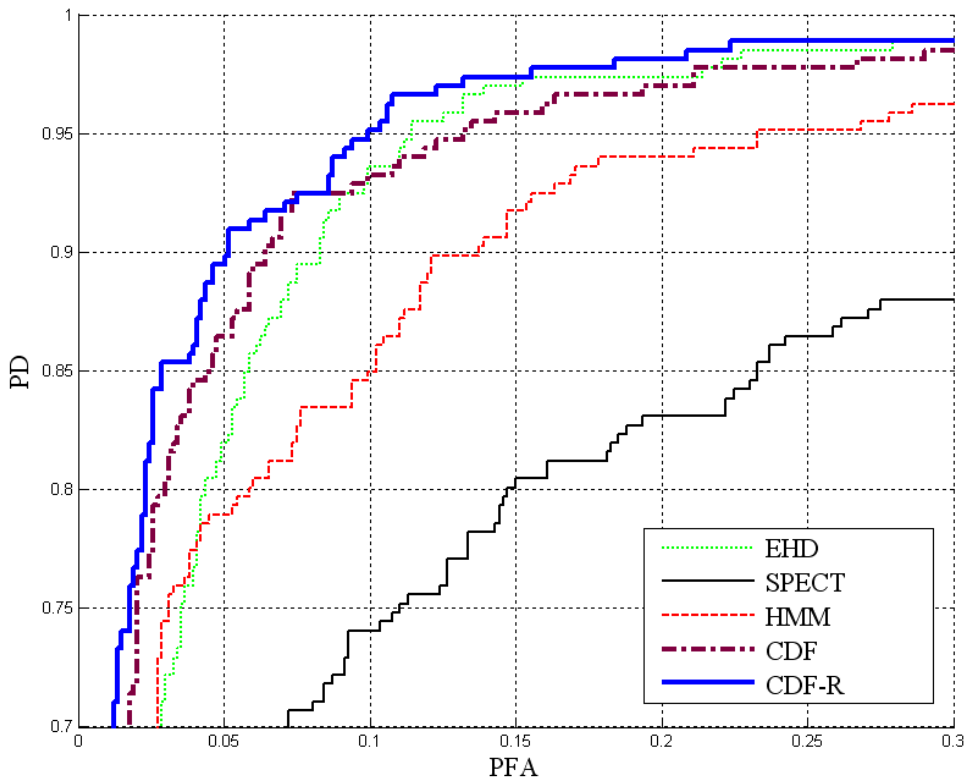


Figure 2-4. ROC curves for a set of classifiers from data set I

total number of non-mines. In Tables 2-3 and 2-4, the original values were multiplied by 100 to clearly show the difference. As can be seen from the results, CDF showed good performance on average for both data sets, although in data set I, at 95% PD, CDF was worse than the single best classifier. On the other hand, CDF-R showed better performance than any single classifier and CDF. The performance degradation of CDF may come from the post-processing routine. The post-processing was applied to make the cluster boundaries smooth and to make CDF noise-robust, which results in convex clusters in the feature space. However, CDF-R does not force the clusters to be convex. The fact that CDF-R is better than CDF in classification suggests that the clusters in the feature space may not be convex.

2.6 Discussion

In this chapter, a new fusion framework, *context-dependent fusion* (CDF), which is a generalization of the previous fusion methods, is described. This is a combination of

Table 2-3. Experimental results from data set I

PD (%)	PFA (%)				
	EHD	SPECT	HMM	CDF	CDF-R
85	5.858	23.297	10.218	4.632	2.861
90	8.311	32.153	13.760	6.403	5.177
95	11.444	48.229	23.297	13.215	9.946

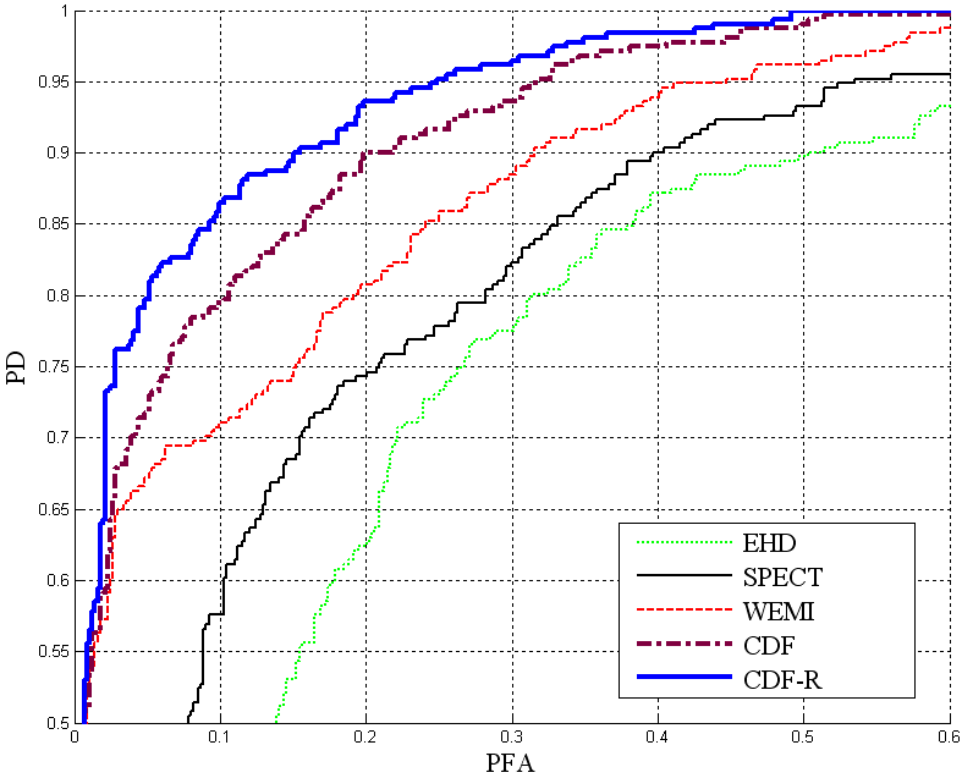


Figure 2-5. ROC curves for a set of classifiers from data set II

traditional fusion method and feature space clustering. The feature space information is referred to as *context* in CDF and helps the fuser to select locally skilled classifiers, which results in improved performance. To extend the performance of CDF, a variant of CDF using regularization, called *context-dependent fusion with regularization* (CDF-R), is proposed. Regularization is a well-known method to make an algorithm noise-robust. Experimental results using landmine data sets proved the effectiveness of the two methods and, especially, the superiority of CDF-R to CDF.

Although both proposed fusion methods were efficient and effective in real world problems, CDF sometimes showed performance degradation. This degradation is mainly

Table 2-4. Experimental results from data set II

PD (%)	PFA (%)				
	EHD	SPECT	WEMI	CDF	CDF-R
85	37.057	33.156	24.113	15.780	9.220
90	49.113	39.362	31.738	19.858	15.071
95	71.631	54.965	43.972	31.915	24.823

caused by the convex-cluster assumption, which is one important difference between CDF and CDF-R. As CDF finalizes the cluster structure based on Euclidean distance, the resulting clusters in the feature space becomes convex. CDF-R does not have that assumption, so showed better results than CDF. These results lead one to the conjecture that contexts cannot be represented as convex clusters in the feature space. CDF-R also has a problem in the linear combination method for fusion. The linearity prevents CDF-R from handling non-linear classes in each cluster. To remove the linear restriction, some non-linear fusion method can be adopted, which are discussed in the following Chapter 6.

CHAPTER 3
RKF-PCA: ROBUST KERNEL FUZZY PCA

Principal component analysis (PCA) is a widely used statistical method that attempts to explain the structure of the data by means of a small number of components. PCA has been applied to statistical analysis, pattern recognition, and image processing [53]. Although PCA has been applied successfully in many areas, it has some problems, such as sensitivity to noise and limitation to Gaussian distributions.

It is well known that the principal components are often affected by outliers, thus may not capture the true structure of the data. Therefore data reduction based on PCA becomes unreliable if outliers are present in the data. To relieve the sensitivity, several robust variants of PCA have been proposed. The goal of robust PCA (R-PCA) is to obtain accurate principal components of a sample set from a distribution when the sample set has been contaminated by samples from a different distribution. Points from the different distribution are referred to here as noise. Most methods achieve robustness by replacing the classical estimate of a covariance matrix by a robust estimate. There are several groups of methods to robustly estimate covariance matrices. The first group of methods is based on subset selection [54], [55]. Several subsets are randomly selected from the whole data set and the standard principal components are calculated for each subset. The principal components from the subsets are aggregated by a robust aggregation method. The second group is based on outlier rejection [56], [57] in which some data points far from most of the data are considered outliers and are removed. The third group consists of methods that assign a weight to each data point representing the contribution of that point to the estimation of the principal components [58]–[60]. The proposed method, robust fuzzy PCA (RF-PCA), belongs to this group.

Another shortcoming of PCA is its inability to accurately represent non-Gaussian distributions. Traditionally, principal components are determined exclusively by the second-order statistics of the data, which are sufficient for Gaussian distribution but

cannot be used to determine other common distributions. In order to describe data with a non-Gaussian distribution, many researchers have introduced appropriate nonlinear processes into PCA and developed nonlinear PCA algorithms [6], [61], [62]. Among these modified PCA methods, kernel PCA (K-PCA) [6] proposed by Schölkopf et al. is the most well known and widely adopted. Although K-PCA is widely used as a nonlinear feature extraction and dimensionality reduction method [63]–[65], it retains the sensitivity to noise found in PCA. To remedy this problem, the proposed RF-PCA is transformed into a kernel-based method, called *robust kernel fuzzy PCA* (RKF-PCA).

In the next section, previous approaches for making PCA robust and their shortcomings are described. RF-PCA, robust PCA using fuzzy membership, is introduced in section 3.2. In section 3.3, K-PCA is presented, and RF-PCA is reformulated as a kernel-based method in section 3.4. Experimental results are given in section 3.5, and discussion in section 3.6.

3.1 Robust PCA (R-PCA)

It is well known that PCA constructs the rank k subspace approximation to training data of $d(> k)$ dimensions that is optimal in a least-squares sense [53]. It is also well known that least-squares techniques are not robust in the sense that outliers can arbitrarily skew an estimate from the true value [66]. Several methods have been proposed to make PCA robust by robustly estimating a covariance matrix, and they can be categorized into several groups according to the covariance estimation method used.

The first group is based on subset selection. Rousseeuw [54] proposed a *minimum covariance determinant* (MCD) method in which one considers some subsets of size h out of a set with $N(> h)$ elements. The covariance matrix of each subset is computed and the robust estimate of the covariance matrix is given by the covariance matrix computed over the subset with the smallest determinant. Although the result is robust, it is not applicable to a small data set because the estimated covariance matrix is close to singular, which causes instability in the calculation of principal components. In addition,

choosing the appropriate number and size of subsets can be difficult. Gabrys et al. [55] proposed another method based on subset selection in which principal components from each subset are aggregated with voting and averaging. This aggregation method, however, with the MCD method shares the two problems described above and there is no guarantee of orthogonality among the resulting principal components.

The second group is based on outlier rejection, in which a data point assumed to be most atypical is removed iteratively. The two methods in [56] and [57] basically share the same structure except the measure of atypicality. In [56], the reconstruction error, defined as Equation (3–1), is used as an atypicality measure,

$$e(x) = ||x - WW^T x||^2, \tag{3–1}$$

where W is a matrix having the top k principal components of the covariance matrix of data set $X = \{x_1, \cdots, x_N\}$ as columns and k is assumed to be known. In [57], the distance from the origin is used together with one class support vector machine [67]. These measures can have any non-negative values, which is different from fuzzy memberships that take on values between zero and one inclusively. The basic limitation of these methods is the pre-defined noise level, which cannot be decided easily in real world problems. The second group of methods is also subject to the singular problem when using small data sets.

The third group uses membership to control the effect of each data point on a covariance matrix. The membership can be applied in two ways: directly on the data [58] and indirectly in the calculation of a covariance matrix [59], [60]. In [58], the data are clustered into a pre-defined number of clusters (M), and the weight (z_i, $1 \leq i \leq N$) is calculated as a function of the membership values in M clusters. With these weights, a

diagonal weight matrix Z is constructed as

$$Z = \begin{bmatrix} z_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & z_N \end{bmatrix} \tag{3–2}$$

and then the traditional PCA is conducted on ZX instead of X; this is called the weighted PCA (W-PCA) method. W-PCA is different from the proposed method and indirect weighting methods in that the data should be clustered into M clusters beforehand, whereas only one cluster is defined in other methods. One shortcoming in W-PCA is the pre-clustering, which is not a simple problem even when the number of clusters, M, is given.

Indirect weighting methods use weights in the calculation of fuzzy covariance matrices defined as

$$C_{fuzzy} = \frac{\sum_{i=1}^{N} u_i^m (x_i - \bar{x})(x_i - \bar{x})^T}{\sum_{i=1}^{N} u_i^m}, \tag{3–3}$$

where $\bar{x}$ is the mean of data, u_i is the membership of x_i (the degree of belonging to the cluster), and m is a fuzzifier constant. Indirect weighting methods consider only one cluster, and that cluster corresponds to the first principal component. The key point in these methods lies in deciding the membership u_i, which represents the degree of importance in the calculation of the fuzzy covariance matrix. In indirect weighting methods, the first principal component is considered to be a line prototype [68] and an additional term corresponding to a noise cluster [69] is added to decide the membership in a robust way. The objective function can be written as

$$J = \sum_{i=1}^{N} u_i^m d^2(x_i, ww^T x_i) + \eta \sum_{i=1}^{N} (1 - u_i)^m, \tag{3–4}$$

where w represents the line prototype and η the noise distance. In Equation (3–4), the first term represents the weighted sum of projection distance (or reconstruction error)

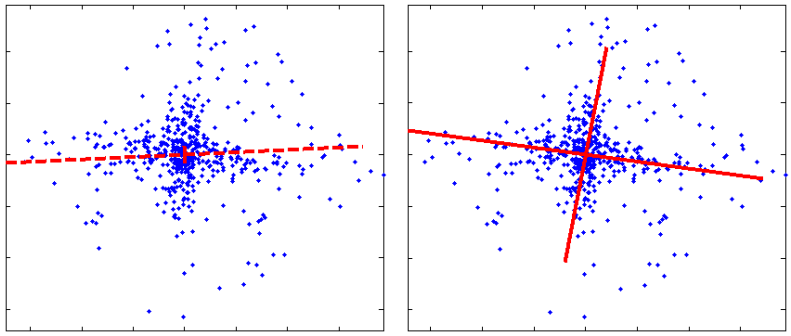
to the line prototype (it corresponds to the first principal component), and the second term computes the sum of noise distance. This formulation was first introduced by Xu et al. [70], in which u_i can only have discrete values, $u_i \in \{0, 1\}$. In this case, it is not easy to minimize Equation (3–4) with respect to $U = \{u_1, \ldots, u_N\}$ and w simultaneously because it is a mixture of discrete and continuous optimization. The constraint on U is relaxed to have continuous values in [59] and [60], which makes it easy to optimize the objective function. After the membership is decided, the robust principal components can be found by analyzing the fuzzy covariance matrix [59] or through the Gram-Schmidt orthogonalization procedure [60]. One problem in the indirect weighting methods is that the membership value is decided based only on the first principal component, which may not reflect the true structure of the data.

3.2 Robust Fuzzy PCA (RF-PCA)

Given the considerations discussed above, RF-PCA is based on two assumptions: (i) all the available data should be used to calculate principal components, and (ii) k principal components should be calculated simultaneously.

Sampling and rejection methods are simple and accurate, but they may suffer from computational problems due to the small size of the data set. Another possible problem is information loss due to the removal of data points, although they might not be important in calculating the principal components. Therefore it is desirable to use all the available data, and weighting data points is better than simply discarding some of them.

The second assumption comes from the defect of indirect weighting methods. When using indirect weighting methods, only the first PC is considered as a line prototype and the objective function is optimized iteratively to decide the membership values. However the line prototype may not contain enough information to represent complex data structures, which results in the inaccurate representation of the membership values, covariance matrix, and finally PCs. Figure 3-1 shows the estimated PCs using reconstruction error in Equation (3–1). Figure 3-1(a) uses the first one PC to calculate



(a) (b)

Figure 3-1. Estimated principal components using reconstruction error with (a) one and (b) two principal components.

the error and Figure 3-1(b) uses the first two. The dimensionality of the data is three and the data points and PCs are projected onto two-dimensional space using PCA. As is clear from Figure 3-1, the indirect methods that only use the first PC sometimes fail to correctly estimate the PCs when the data is noisy. Another problem is the eigenvalue to which the length of a line is proportional in Figure 3-1. Ideally the second vertical line has half the length of the first one, but the second largest eigenvalue in Figure 3-1(a) is much smaller than the first one and its value is almost zero. By using k PCs simultaneously, however, one can get a more stable result, as shown in the figure. The value of k is decided through cross-validation.

The proposed method is based on fuzzy mean and fuzzy covariance and has a connection with M-estimators in robust statistics. In robust statistics [66], M-estimators for mean and covariance can be defined as

$$\mu_R = \frac{\sum_{i=1}^{N} f_1(g(x_i|\mu_R, C_R))x_i}{\sum_{i=1}^{N} f_1(g(x_i|\mu_R, C_R))}, \tag{3–5}$$

$$C_R = \frac{1}{N}\sum_{i=1}^{N} f_2(g^2(x_i|\mu_R, C_R))(x_i - \mu_R)(x_i - \mu_R)^T, \tag{3–6}$$

where f_1 and f_2 are monotonically decreasing weight functions, and g is any distance measure. If f_1 and f_2 are defined using the reconstruction error in Equation (3–1) and an exponential function,

$$\begin{aligned} f_1(x_i) &= \exp\left(-\frac{e(x_i)}{\sigma^2}\right) \\ &= \exp\left(-\frac{||x_i - WW^T x_i||^2}{\sigma^2}\right) = u_i, \end{aligned} \tag{3–7}$$

$$\begin{aligned} f_2(x_i) &= \left(\exp\left(-\frac{e(x_i)}{\sigma^2}\right)\right)^2 \\ &= \left(\exp\left(-\frac{||x_i - WW^T x_i||^2}{\sigma^2}\right)\right)^2 = u_i^2, \end{aligned} \tag{3–8}$$

where σ is a constant, then f_1 is a fuzzy membership function whose range is between zero and one inclusively. The membership function also can be derived from an objective function similar to Equation (3–4). An objective function with entropy regularization [71], which is widely used to mitigate the effect of noise, can be represented as

$$J = \sum_{i=1}^{N} u_i e(x_i) + \sigma^2 \sum_{i=1}^{N} (u_i \log u_i - u_i), \tag{3–9}$$

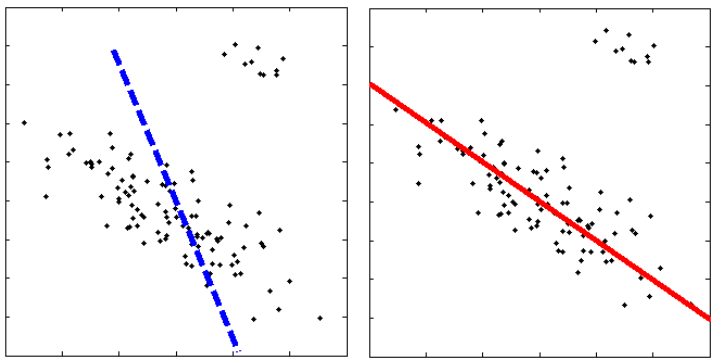
where the first term represents the weighted sum of reconstruction error and the second term does the regularization term, which is a little different from the one in fuzzy clustering. In fuzzy clustering, there is a sum-to-one constraint over $M(> 1)$ clusters and the constraint prevents the objective function from having a trivial solution, $u_i = 0$ for all $i = 1, \ldots, N$. In RF-PCA, however, there is only one cluster and no constraint. So the term $-u_i$ is added to prevent u_i from having a trivial solution. By taking a partial derivative of J with respect to u_i, one can obtain

$$\frac{\partial J}{\partial u_i} = e(x_i) + \sigma^2 \log u_i = 0, \tag{3–10a}$$

$$u_i = \exp\left(-\frac{e(x_i)}{\sigma^2}\right), \tag{3–10b}$$

Algorithm 3-1 RF-PCA

1: Initialize a membership vector $U_0 = [u_1, \ldots, u_N] = [1, 1, \ldots, 1]$ and a principal component matrix W_0 using PCA.
2: **repeat**
3: Calculate mean (μ_R) and covariance (C_R) using (3–11) and (3–12).
4: Calculate principal component matrix W using C_R.
5: Calculate a new membership vector U using (3–7).
6: **until** U or iteration counter t satisfy the convergence criteria
7: **return**



(a) (b)

Figure 3-2. The first principal component using (a) PCA and (b) RF-PCA.

which is equal to Equation (3–7). Using Equations (3–7) and (3–8), Equations (3–5) and (3–6) can be re-written as

$$\mu_R = \frac{\sum_{i=1}^{N} u_i x_i}{\sum_{i=1}^{N} u_i}, \tag{3–11}$$

$$C_R = \frac{1}{N}\sum_{i=1}^{N} u_i^2 (x_i - \mu_R)(x_i - \mu_R)^T, \tag{3–12}$$

which correspond to the robust estimates of mean and covariance. Robust k principal components can be obtained from C_R. The overall RF-PCA algorithm can be written as in Algorithm 3-1. This algorithm iterates until the maximum difference of the previous and current membership values is not greater than the pre-specified constant, $\max_{i=1,\ldots,N} ||u_i^{t-1} - u_i^t|| < \varepsilon$, or the counter reaches its maximum value, $t > t_{\max}$.

Figure 3-2 shows the first principal component found by PCA and RF-PCA. The data set consists of 110 data points, 100 points from a Gaussian distribution and ten noise points. PCA found a skewed principal component due to noise, but RF-PCA found an unskewed one because the noise points have small membership values and consequently are negligible in the calculation of a covariance matrix.

3.3 Kernel PCA (K-PCA)

Another shortcoming of PCA originates from the fact that it only finds the independent axes of the data under the Gaussian assumption. To remedy the inability to accurately represent non-Gaussian distributions, several nonlinear PCA methods have been proposed [6], [61], [62]. Among them, Schölkopf et al.'s kernel PCA (K-PCA) [6] is considered here. In K-PCA, the input vector x is mapped into a high dimensional feature space F via a mapping function Φ

$$\Phi : R^d \rightarrow F, x \rightarrow \phi(x), \tag{3–13}$$

where d is the dimensionality of an input vector. Assuming the mapped data has zero mean in the feature space F, the covariance matrix in F is

$$C_F = \frac{1}{N}\sum_{i=1}^{N} \phi(x_i)\phi(x_i)^T, \tag{3–14}$$

where N is the number of data points. We have to find eigenvalues $\lambda \geq 0$ and eigenvectors $w \in F\backslash\{0\}$ satisfying

$$C_F w = \lambda w. \tag{3–15}$$

All solutions w with $\lambda \geq 0$ lie in the span of $\phi(x_1), \phi(x_2), \ldots, \phi(x_N)$. Thus there exist coefficients α_i $(i = 1, \ldots, N)$ such that

$$w = \sum_{i=1}^{N} \alpha_i \phi(x_i). \tag{3–16}$$

By defining an $N \times N$ kernel matrix K by $K_{ij} = \phi(x_i) \cdot \phi(x_j)$, one can obtain

$$K\alpha = N\lambda\alpha = \lambda'\alpha, \tag{3–17}$$

where α denotes the column vector with entries $\alpha_1, \alpha_2, \ldots, \alpha_N$. Let $\lambda'_1 \geq \lambda'_2 \geq \ldots \geq \lambda'_N$ denote the eigenvalues and $\alpha^1, \alpha^2, \ldots, \alpha^N$ the corresponding eigenvectors, $\alpha^k = [\alpha^k_1, \ldots, \alpha^k_N]^T$ $(k = 1, \ldots, N)$, of Equation (3–17). Each eigenvector should be normalized so that the corresponding principal components in F have unit length, i.e., $w^k \cdot w^k = 1$. This can be done by taking α^k to have a length of $1/\sqrt{\lambda'_k}$. For a test point x, the image of a test point $\phi(x)$ can be calculated by projecting x onto the kth eigenvector w^k in F according to

$$w^k \cdot \phi(x) = \sum_{i=1}^{N} \alpha^k_i (\phi(x_i) \cdot \phi(x)) = \sum_{i=1}^{N} \alpha^k_i \kappa(x_i, x). \tag{3–18}$$

Using Equation (3–18), we can calculate the projected data in F without the explicit calculation of the covariance matrix in F, which is often impossible. When the data are not centered, the following transformation transforms the data to have zero mean in the feature space F,

$$\widetilde{K}_{ij} = K_{ij} - \frac{1}{N}\sum_{n=1}^{N} K_{in} - \frac{1}{N}\sum_{m=1}^{N} K_{mj} + \frac{1}{N^2}\sum_{m,n=1}^{N} K_{mn}, \tag{3–19}$$

where $\widetilde{K}$ is a zero mean kernel matrix in F. A test point also should be transformed as

$$\widetilde{K}(x, x_i) = K(x, x_i) - \frac{1}{N}\sum_{n=1}^{N} K(x, x_n) - \frac{1}{N}\sum_{m=1}^{N} K(x_m, x) + \frac{1}{N^2}\sum_{m,n=1}^{N} K(x_m, x_n). \tag{3–20}$$

Refer to [6] for a more detailed derivation.

3.4 Robust Kernel Fuzzy PCA (RKF-PCA)

In this section, RF-PCA developed in section 3.2 is reformulated as a kernel-based method. First of all, Equations (3–11) and (3–12) are kernelized as

$$\mu_\phi = \frac{\sum_{i=1}^{N} u_i \phi(x_i)}{\sum_{i=1}^{N} u_i}, \tag{3–21}$$

$$C_\phi = \frac{1}{N} \sum_{i=1}^{N} u_i^2 (\phi(x_i) - \mu_\phi)(\phi(x_i) - \mu_\phi)^T. \tag{3–22}$$

Assume that the mean in the feature space (μ_ϕ) is zero. Then Equation (3–22) can be simplified as

$$\begin{aligned} C_\phi &= \frac{1}{N} \sum_{i=1}^{N} (u_i \phi(x_i))(u_i \phi(x_i))^T \\ &= \frac{1}{N} \sum_{i=1}^{N} \psi(x_i) \psi(x_i)^T, \end{aligned} \tag{3–23}$$

where $\psi(x_i)$ is defined as $\psi(x_i) = u_i \phi(x_i)$. The problem then is to find eigenvalues $\lambda \geq 0$ and eigenvectors $w \in F \backslash \{0\}$ satisfying

$$C_\phi w = \lambda w. \tag{3–24}$$

As with K-PCA, all solutions w with $\lambda \geq 0$ lie in the span of $\psi(x_1), \psi(x_2), \ldots, \psi(x_N)$ so there exist coefficients α_i (i = 1, . . . , N) such that

$$w = \sum_{i=1}^{N} \alpha_i \psi(x_i) = \sum_{i=1}^{N} \alpha_i u_i \phi(x_i), \tag{3–25}$$

which is a weighted combination of mapped data in F. By defining an $N \times N$ kernel matrix K' as $K'_{ij} = \psi(x_i) \cdot \psi(x_j)$, one arrives at the equation

$$K' \alpha = N \lambda \alpha, \tag{3–26}$$

which should be solved for eigenvectors with non-zero eigenvalues. The eigenvector α_k also should be normalized to ensure that the corresponding principal component (w^k) has unit length in F, which results in the same normalization as that in K-PCA. The projection of a test point onto the principal components w^k in F can be evaluated as

$$\begin{aligned} w^k \cdot \phi(x) &= \sum_{i=1}^{N} \alpha_i^k u_i \phi(x_i) \cdot \phi(x) \\ &= \sum_{i=1}^{N} \alpha_i^k u_i \kappa(x_i, x). \end{aligned} \tag{3–27}$$

If the data are not centered, the data should be centered using a transformation similar to that given in Equation (3–19).

$$\begin{aligned} \widetilde{K'}_{ij} =& (\phi(x_i) - \mu_\phi) \cdot (\phi(x_j) - \mu_\phi) \\ =& K_{ij} - \frac{1}{N'} \sum_{n=1}^{N} u_n K_{in} - \frac{1}{N'} \sum_{m=1}^{N} u_m K_{mj} + \frac{1}{N'^2} \sum_{m,n=1}^{N} u_m u_n K_{mn}, \end{aligned} \tag{3–28}$$

where $N' = \sum_{i=1}^{N} u_i$. A test point, x, can be transformed in a similar way to have zero mean.

$$\begin{aligned} \widetilde{K'}(x, x_i) =& (\phi(x) - \mu_\phi) \cdot (\phi(x_i) - \mu_\phi) \\ =& K(x, x_i) - \frac{1}{N'} \sum_{n=1}^{N} u_n K(x, x_n) - \frac{1}{N'} \sum_{m=1}^{N} u_m K(x_m, x) + \frac{1}{N'^2} \sum_{m,n=1}^{N} u_m u_n K_{mn}. \end{aligned} \tag{3–29}$$

The remaining problem is the calculation of membership values. Using Equation (3–1), the reconstruction error in F can be represented as

$$\begin{aligned} e(\phi(x)) =& ||(\phi(x) - \mu_\phi) - WW^T(\phi(x) - \mu_\phi)||^2 \\ =& \kappa(x, x) - \frac{2}{N'} \sum_{i=1}^{N} u_i \kappa(x_i, x) + \frac{1}{N'^2} \sum_{i,j=1}^{N} u_i u_j \kappa(x_i, x_j) \\ & - 2 \sum_{i=1}^{k} \left(\left(\beta^i\right)^2 - 2\beta^i \gamma^i + \left(\gamma^i\right)^2 \right) \\ & + \sum_{l,m=1}^{k} \sum_{i,j=1}^{N} (\beta^l \beta^m - 2\beta^l \gamma^m + \gamma^l \gamma^m) \xi_i^l \xi_j^m \kappa(x_i, x_j), \end{aligned} \tag{3–30}$$

Algorithm 3-2 RKF-PCA

1: Initialize a membership vector $U_0 = [u_1, \dots, u_N] = [1, 1, \dots, 1]$ and a kernel matrix K.
2: **repeat**
3: Center the kernel matrix K and calculate a kernel matrix K'.
4: Solve Equation (3–26) for eigenvalues (λ_k) and eigenvectors (α^k).
5: Calculate a new membership vector, U, using (3–34).
6: **until** U or iteration counter t satisfy the convergence criterion
7: **return**

where β^j, γ^l, and ξ_i^l are defined as

$$\beta^j = \sum_{i=1}^{N} \alpha_i^j u_i \kappa(x_i, x), \tag{3–31}$$

$$\gamma^l = \frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i^l u_i u_j \kappa(x_i, x_j), \tag{3–32}$$

$$\xi_i^l = \alpha_i^l u_i, \tag{3–33}$$

and $k(< d)$ is the reduced dimensionality. The detailed derivation of Equation (3–30) can be found in Appendix B. Using Equation (3–30), the reconstruction error in F can be calculated without an explicit evaluation of Φ and the membership value can be calculated using Equations (3–7) and (3–30) as

$$u_i = \exp\left(-\frac{e(\phi(x_i))}{\sigma^2}\right), \tag{3–34}$$

where σ is a constant. The RKF-PCA algorithm is basically the same as RF-PCA except that the mean and covariance (Equations (3–21) and (3–22), respectively) are not evaluated explicitly. The RKF-PCA algorithm can be written as in Algorithm 3-2. This algorithm uses the same stopping criterion as Algorithm 3-1.

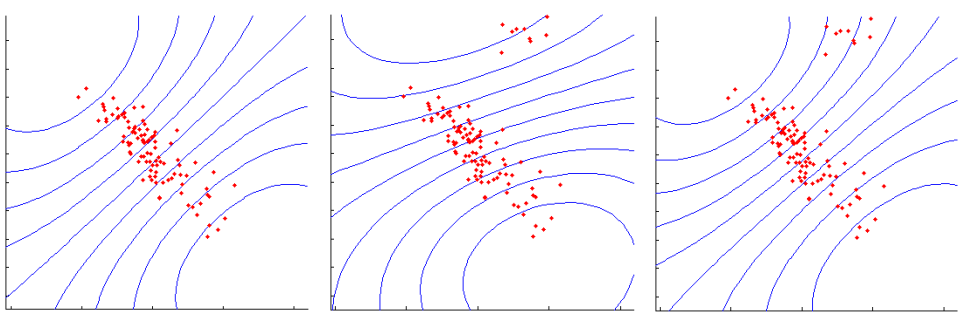
Figure 3-3 shows the first principal component found by K-PCA and RKF-PCA on clean and noisy data. The curves represent the points that have the same projected value on an eigenvector. It is clear that the results are almost the same as those in Figure 3-2.

3.5 Experimental results

To investigate the effectiveness of the proposed algorithm, RKF-PCA was first applied to the data set used in [6] to show the effectiveness of K-PCA. The data set is composed of three Gaussian clusters, each of which has 30 points. Figure 3-4(a) shows the results using K-PCA. In Figures 3-4(b) and 3-4(c), ten noise points are added, and K-PCA and RKF-PCA are applied. In Figure 3-4(b), it is clear that the noise points tend to skew the original principal components toward themselves. In Figure 3-4(c), however, the principal components found are almost the same as those in Figure 3-4(a), because the noise points have small membership values and have little effect on the calculation of a covariance matrix. Figure 3-5 shows the error histograms between the eigen-system of clean data using K-PCA and that of noisy data using K-PCA and RKF-PCA. Two hundred experiments were conducted and each of the experiments used the data in Figure 3-4(a) with ten randomly generated noise points. The error is defined as the weighted sum of angles between two eigenvectors,

$$E = \sum_{i=1}^{N_1} \lambda_i^C \theta(w_C^i, w_N^i), \tag{3–35}$$

where $N_1(= 90)$ is the number of points in the clean data set, λ_i^C is the ith eigenvalue from clean data using K-PCA, w_C^i and w_N^i are the ith eigenvector from clean and



(a) (b) (c)

Figure 3-3. The first principal component from (a) K-PCA on clean data, (b) K-PCA on noisy data, and (c) RKF-PCA on noisy data.

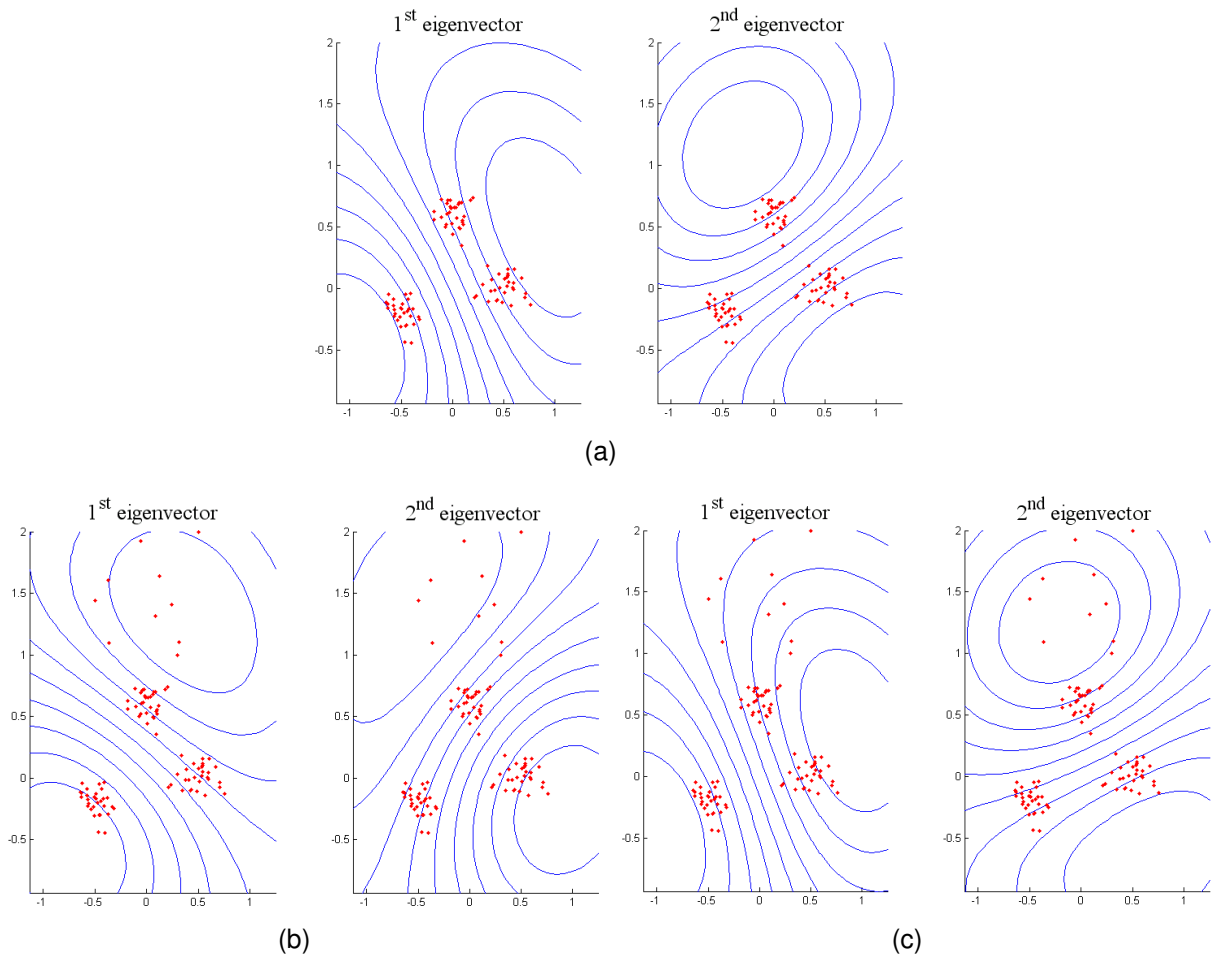


Figure 3-4. (a) K-PCA on clean data, (b) K-PCA on noisy data, and (c) RKF-PCA on noisy data.

noisy data. The eigenvalues are assumed to be sorted in descending order and the eigenvectors are rearranged according to their eigenvalues. The value $\theta(w_C^i, w_N^i)$ represents the angle between two eigenvectors and can be calculated as

$$\begin{aligned}\cos(\theta(w^1, w^2)) &= w^1 \cdot w^2 \\ &= \sum_{i=1}^{N_1} \alpha_i^1 \phi(x_i) \cdot \sum_{j=1}^{N_2} \alpha_j^2 u_j \phi(x_j) \\ &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \alpha_i^1 \alpha_j^2 u_j \kappa(x_i, x_j),\end{aligned} \tag{3–36a}$$

$$\theta(w^1, w^2) = \cos^{-1}\left(\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \alpha_i^1 \alpha_j^2 u_j \kappa(x_i, x_j)\right), \tag{3–36b}$$

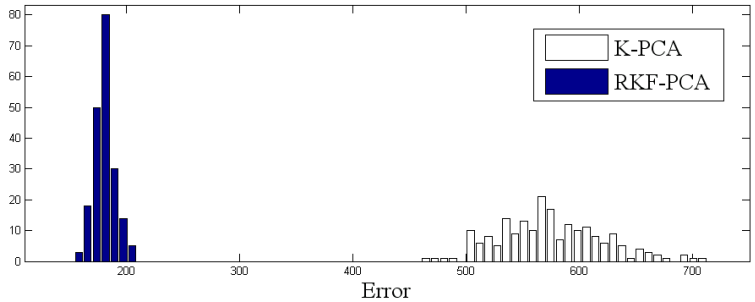


Figure 3-5. Error histogram between eigen-systems of clean and noisy data using K-PCA and RKF-PCA.

where $N_2(= 100)$ is the number of points in a noisy data set. The membership values are $u_j = 1(1 \leq j \leq N)$ for K-PCA and from Equation (3–34) for RKF-PCA. As is clear from Figure 3-5, the error of RKF-PCA is much smaller than that of K-PCA and the variance, too, which means that RKF-PCA is better than K-PCA in noisy condition.

The proposed algorithm was also tested as a feature extraction method on the *letter recognition data set* that is available from the UCI machine learning repository [72]. The data set consists of 26 classes, each of which corresponds to a character in the English alphabet. There are approximately 700 samples per class. In this experiment, each pair of classes were classified using Fisher's linear discriminant [2], and the three pairs having the largest error rate were selected for further experiment: namely, H-O, H-R, and S-Z pairs. Ten-fold cross-validation was conducted on each character-pair data and Table 3-1 summarizes the error rate on each pair. The parameters, σ of the kernel functions, were set using cross-validation. As is clear from Table 3-1, none of the three character pairs is linearly separable in the original feature space and the kernel-based non-linear feature extraction method reduces the error significantly. One thing that should be noted is that PCA in the input space shows the best result with the original dimensionality (which is 16) and both kernel-based PCAs show the best result with a dimensionality of around 50. This is much higher than the original one and suggests that the high order features are helpful in this case. Although it appears that K-PCA and RKF-PCA achieve about the same error rate, the difference is statistically significant,

Table 3-1. Classification error on each set of character-pair data

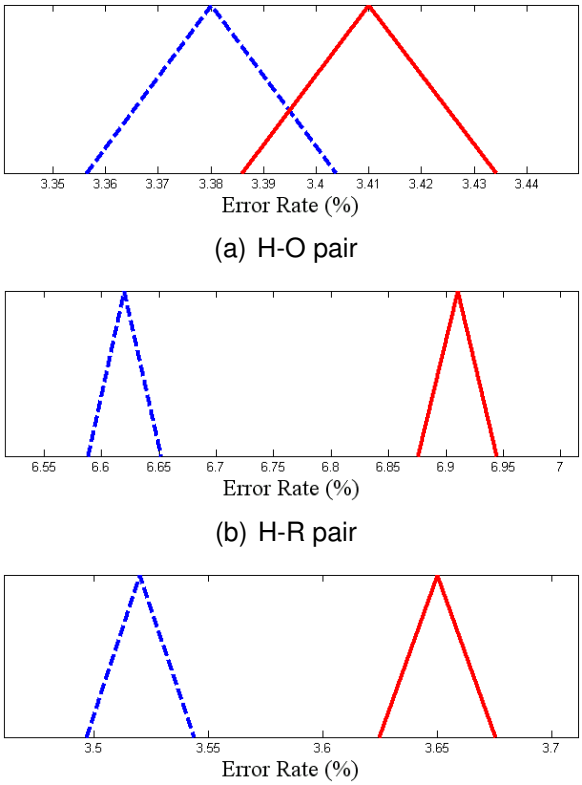
Feature extraction method	Error rate (%)		
	H-O	H-R	S-Z
PCA	21.44	11.80	15.91
K-PCA	3.41	6.91	3.65
RKF-PCA	3.38	6.62	3.52

except for H-O pair. In Table 3-1, each error rate is the probability of an error sample (p). Assume each error sample is independent. Then the number of error samples (N_F) out of the number of samples (N_{total}) is binomially distributed, which can be further approximated with a Gaussian distribution with $\mu = N_{total} \cdot p$ and $\sigma^2 = N_{total} \cdot p(1-p)$. The 100(1 – α)% confidence interval (CI) for N_F can be formulated as [73]

$$\left[\mu - \frac{\sigma \cdot t_{\alpha/2}}{\sqrt{N_{total}}}, \mu + \frac{\sigma \cdot t_{\alpha/2}}{\sqrt{N_{total}}}\right], \tag{3–37}$$

where $t_{\alpha/2}$ is the cutoff value of a student-t distribution with (N_{total} – 1) degrees of freedom. Figure 3-6 shows the CIs with a 95% confidence level (CL). In Figure 3-6, dotted lines represent confidence intervals of RKF-PCA and solid lines represent those of K-PCA. For H-R and S-Z pair, there is no overlap between the CIs, which means that the error rate of K-PCA and RKF-PCA is statistically significant and that RKF-PCA performed better classification than K-PCA in this experiment at a 95% CL. For H-O pair, the difference is not significantly different at a 95% CL.

To investigate the robustness of RKF-PCA, the previous experiments were conducted again with some noise points. Noise points are selected randomly from other classes of characters except the two real training classes, then added to the training set with randomly generated labels. The noise points can be considered as mis-labeled points and play the role of outliers. The number of noise points is decided as the number of real training points multiplied by noise ratio. Figure 3-7 summarizes the classification error using Fisher's linear discriminant with respect to noise ratio. The values plotted in Figure 3-7 are averaged over ten runs.



(a) H-O pair

(b) H-R pair

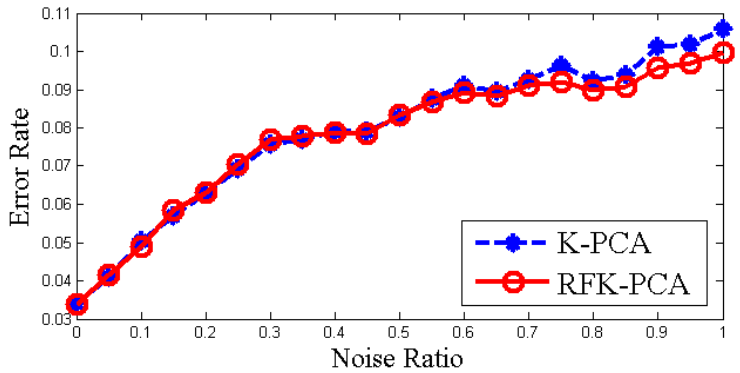
(c) S-Z pair

Figure 3-6. Confidence intervals of the error rate

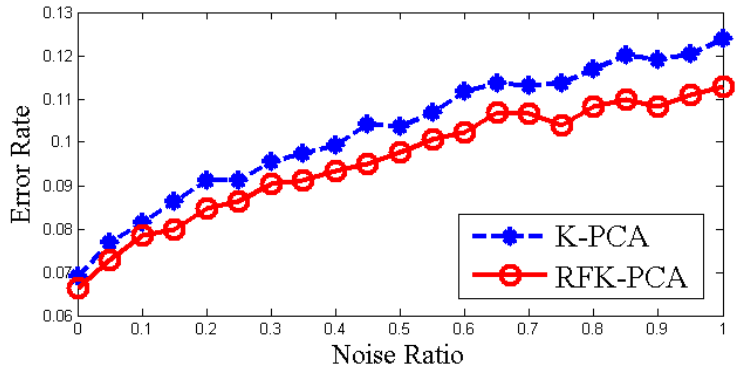
Kernel-based methods are known to be less sensitive to outliers than the corresponding input space methods including K-PCA. From Figure 3-7, however, we can see that RKF-PCA is less sensitive to outliers than K-PCA and the difference gets bigger as the noise ratio increases. In the case of H-O pair, the difference between K-PCA and RKF-PCA is not statistically significant until the noise ratio is bigger than 0.7 under a 95% CL. However, RKF-PCA converges to K-PCA as σ in Equation (3–34) goes infinity, which implies that the performance of K-PCA may be a lower bound of RKF-PCA performance.

3.6 Discussion

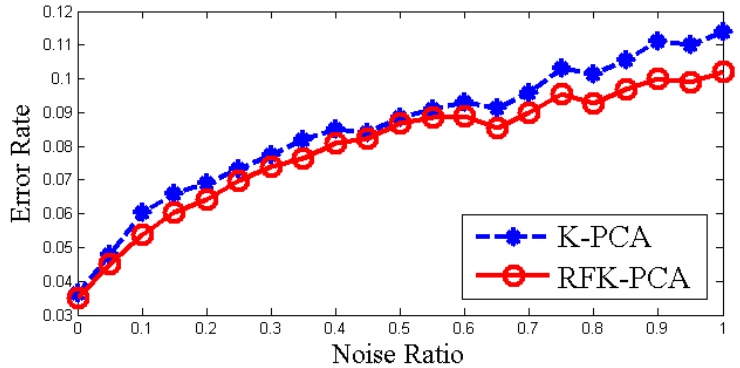
A new variant of PCA, *robust kernel fuzzy PCA* (RKF-PCA), is proposed as a solution to the noise sensitivity and linearity restriction problems of PCA. RKF-PCA is an extension of kernel PCA (K-PCA) with the help of a robust estimation technique.



(a) H-O pair



(b) H-R pair



(c) S-Z pair

Figure 3-7. Error rate with respect to noise ratio.

The proposed algorithm outperformed K-PCA as demonstrated using an artificial and character recognition data sets.

Although RKF-PCA performed better than K-PCA, there are also some drawbacks. RKF-PCA is an iterative method, which means that it requires more computation than K-PCA. The reconstruction error defined in Equation (3–30) also requires $O(k^2N^2)$

computation, where k is the number of principal components and N is the number of data points. An approximated version of Equation (3–30) or a different measure might reduce the computational complexity without sacrificing the performance, which is under investigation. Another problem is determining the extra membership parameter σ in Equation (3–34). With the proper setting of σ, RKF-PCA outperforms K-PCA, but the cross-validation method used here is time consuming. If the distribution of membership values is known, σ might be decided without cross-validation, which is left for further research.

CHAPTER 4
KG-FCM: KERNELIZED GLOBAL FUZZY *C*-MEANS

Clustering has formed an important area in pattern recognition, image processing, and, most recently, data mining. Clustering, also known as cluster analysis, assigns an unlabeled data set $X = \{x_1, \ldots, x_N\}$ into $K(1 < K < N)$ homogeneous clusters based on a similarity measure [74]. A variety of clustering algorithms have been proposed, including fuzzy c-means (FCM), which is the main concern of this chapter. Although FCM is a simple but efficient algorithm, there are several difficulties with it: (1) determining the optimal number of clusters; (2) choosing the initial cluster seeds; (3) suppressing noise; and (4) handling clusters with arbitrarily shapes. In this chapter, we try to solve the problems, except the first problem, while the number of clusters, K, is assumed to be known. To address the remaining problems, global fuzzy c-means, which uses a deterministic approach to select a set of initial seeds, is extended using the kernel method. Specifically, the Cauchy kernel is used to achieve noise robustness and the random walk kernel is adopted to deal with non-convex clusters.

It has long been known that FCM is only guaranteed to converge on a locally optimal solution, which is sensitive to initial state. Since the problems of obtaining a globally optimal initial state and a globally optimal solution have been shown to be NP-hard, the initialization method for a sub-optimal solution is more practical and of great value.

There exist various initialization methods for clustering, which can be categorized into three groups [75]: random sampling methods, distance optimization methods, and density estimation methods. Random sampling methods are probably the most widely adopted ones, in which multiple runs of clustering with random initial seeds are conducted, and the best or aggregated one is selected as the final clustering result [76]. Although the random selection methods are simple and statistically sound, they depend heavily on the number of runs, and the computational cost is very high. Distance

optimization methods are used to maximize the distance among the seeds. As many clustering methods, including FCM, try to minimize the intra-cluster variance without optimizing the inter-cluster separation, it is natural to choose to optimize the inter-cluster distance beforehand. The *MaxMin procedure* [77] and its variants, in which a data point having the largest distance to the existing set of seeds is iteratively added, are the most widely used ones. Although they are simple and efficient, some of them also require multiple runs and the resulting set of seeds tends to be placed on the boundary of the data. Density estimation methods select a set of seeds based on local density, the estimation of which is crucial in these methods. The k-nearest neighbor or ε-radius ball is generally used to decide neighboring points [78], and each requires one additional parameter. Global fuzzy c-means [79] also belongs to this class, which iteratively adds a data point optimizing the original objective function for clustering as a seed. Due to its deterministic property, it does not require multiple runs and does not need any extra parameter.

The global clustering method was first formulated for hard clustering [80], [81] and extended to soft clustering [79]. These global methods can determine the initial seeds efficiently in computation and performance, but are still affected by noise and cannot manage non-convex clusters. The sensitivity to noise can be alleviated in several ways and the kernel-based method is one of them. The kernel-based method was originally proposed to convert a linear method into a non-linear one [5], but it is also well-known that some kernels used in kernel-based clustering have outlier robustness together with their non-linear property [82]. Another reason for adopting kernel-based methods is that the kernel-based methods can be extended in several ways using different kernels.

In this chapter, global fuzzy c-means (G-FCM) is converted into a kernel-based method, called *kernel-based global fuzzy* c*-means* (KG-FCM) algorithm, and realized using two different kernels. First of all, KG-FCM is implemented as a noise resistant method using the Cauchy kernel, called KG-FCM with Cauchy kernel (KG-FCM-C).

KG-FCM-C can be used to determine the initial seeds efficiently and treat outliers in a robust way. KG-FCM is also implemented in another way using the random walk kernel [83], [84] to manage non-convex clusters, called KG-FCM with random walk kernel (KG-FCM-RW). KG-FCM-C is more noise resistant than existing methods, and KG-FCM-RW is the most efficient one for clustering non-convex clusters.

In the next two sections, the hard and soft global clustering algorithms – global k-means and global fuzzy c-means, respectively – are briefly described. In section 4.3, global fuzzy c-means is re-formulated as kernel-based global fuzzy c-means. Experimental results are given in section 4.4 and discussion and further researches are given in section 4.5.

4.1 Global *k*-means (GKM)

k-means algorithm [85] is an iterative algorithm that finds K crisp and hyper-spherical clusters in the data such that a measure defined in Equation (4–1) is minimized.

$$E_{k-means}(V|X) = \sum_{i=1}^{N}\sum_{k=1}^{K} I(x_i \in C_k)||x_i - v_k||^2, \tag{4–1}$$

where $I(\cdot)$ is an indicator function having a value of 1 if a data point x_i belongs to the kth cluster (C_k) and 0 otherwise. Commonly, the K cluster centers, $V = \{v_1, v_2, \ldots, v_K\}$, are initialized randomly, and the data set $X = \{x_1, x_2, \ldots, x_N\}$ is partitioned based on the minimum squared distance criterion. Although k-means is one of the simplest learning algorithms and is still in use, one of its major problems is its sensitivity to initialization. To resolve this problem, several methods for initialization have been proposed and global k-means is one of them. Global k-means [80], [81] is an incremental approach to clustering that dynamically adds one cluster at a time through a deterministic search procedure. The assumption on which the method is based is: an optimal clustering solution to the k–clustering problem can be obtained through N local searches using k-means starting from an initial state with

Algorithm 4-1 Fast global k-means

Input: K : number of clusters, X : data set $(N \times d)$

1: $V_1 = \left\{ \frac{1}{N} \sum_{i=1}^{N} x_i \right\}$
2: **for** $k = 2$ to K **do**
3: **for** $l = 1$ to N **do**
4: $J_k^l = \sum_{j=1}^{N} \max\left(\left(d_{k-1}^j \right)^2 - ||x_l - x_j||^2, 0 \right)$
5: **end for**
6: $\alpha = \arg\max_{1 \le l \le N} J_k^l$
7: $V_k = V_{k-1} \cup \{x_\alpha\}$
8: $[V_k] \leftarrow k$-means(X, V_k)
9: **end for**
10: **return**

- the $(k-1)$ centers placed at the optimal positions for the $(k-1)$–clustering problem and
- the remaining k^{th} center placed at a data point x_i $(1 \le i \le N)$.

The original version of GKM requires $O(N)$ executions of k-means for one new seed, which results in $O(KN)$ executions of k-means and is not feasible with a large N. Thus a fast GKM method was proposed, in which only one data point maximizing an objective function is considered as a new seed. The objective function is defined as

$$b_i = \sum_{j=1}^{N} \max\left(\left(d_{k-1}^j \right)^2 - ||x_i - x_j||^2, 0 \right), \tag{4–2}$$

where d_{k-1}^j is the distance between x_j and the closest center among the $(k-1)$ cluster centers obtained so far. The quantity b_i measures the guaranteed reduction in the objective function obtained by inserting a new cluster center at x_i. The modified algorithm significantly reduces the number of k-means executions from $O(KN)$ to $O(K)$. The fast GKM algorithm can be summarized as Algorithm 4-1.

4.2 Global fuzzy *c*-means (G-FCM)

With the introduction of fuzzy memberships, global fuzzy c-means (G-FCM) is a direct extension of global k-means (GKM) as FCM is of k-means. k-means is used to minimize the sum of squared distance in Equation (4–1), which can be extended to

FCM by relaxing the indicator function, $I(\cdot)$, to have continuous values with appropriate constraints. The objective function can be written as

$$E_{FCM}(U, V|X) = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m}||x_i - v_k||^2, \tag{4–3}$$

where u_{ki} is the membership of the ith point to the kth cluster, and m is a fuzzifier constant. G-FCM uses the same algorithm as GKM, except for two things. First of all, FCM is used instead of k-means. Equations (4–4) and (4–5) show the update equations for FCM [86]:

$$u_{ki} = \frac{||x_i - v_k||^{-2/(m-1)}}{\sum_{j=1}^{K}||x_i - v_j||^{-2/(m-1)}}, \tag{4–4}$$

$$v_k = \frac{\sum_{i=1}^{N} u_{ki}^{m} x_i}{\sum_{i=1}^{N} u_{ki}^{m}}. \tag{4–5}$$

Another difference is the objective function for the selection of a new seed. Using Equation (4–2), GKM tries to find a data point minimizing Equation (4–1). Likewise, in G-FCM, one should find a data point minimizing Equation (4–3). Equation (4–3) can be re-formulated using Equation (4–4) as follows [87]:

$$E_{FCM}(U, V|X) = \sum_{i=1}^{N}\left(\sum_{k=1}^{K}||x_i - v_k||^{2/(1-m)}\right)^{1-m}. \tag{4–6}$$

The objective function for seed selection then becomes

$$J_k^l = E_{FCM}(U, V_{k-1} \cup \{x_l\}|X) \tag{4–7}$$

and data point x_α minimizing Equation (4–7) is selected as the initial position of a new cluster center.

4.3 Kernel-based global fuzzy *c*-means (KG-FCM)

Although global fuzzy c-means can be used to decide initial seeds efficiently, it still has drawbacks in that it is sensitive to outliers due to the Euclidean distance measure

and it is limited to linearly separable clusters. Several approaches have been adopted to alleviate these problems and kernel-based clustering is one of them.

Kernel-based methods were firstly introduced to clustering by Girolami [88] who proposed kernel k-means. Other kernel-based fuzzy clustering algorithms were proposed after that [89]. Kernel FCM (K-FCM) is used to minimize the following objective function, which is a direct extension of Equation (4–3):

$$E_{K-FCM}(U, V|X) = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m}||\phi(x_i) - \phi(v_k)||^2, \tag{4–8}$$

where $\phi(\cdot)$ is a mapping function. The update equations can be written as follows:

$$u_{ki} = \frac{||\phi(x_i) - \phi(v_k)||^{-2/(m-1)}}{\sum_{j=1}^{K} ||\phi(x_i) - \phi(v_j)||^{-2/(m-1)}}, \tag{4–9}$$

$$\phi(v_k) = \frac{\sum_{i=1}^{N} u_{ki}^{m}\phi(x_i)}{\sum_{i=1}^{N} u_{ki}^{m}}. \tag{4–10}$$

To make KG-FCM noise-robust, the Cauchy kernel is used instead of the commonly used Gaussian kernel. The Cauchy kernel has been shown to be outlier-robust and able to handle clusters with different densities effectively compared to the Gaussian kernel [82]. The Cauchy kernel can be defined as

$$\kappa_C(x, y) = \frac{1}{1 + \beta||x - y||^2}, \tag{4–11}$$

where β is a kernel parameter. Using Equation (4–11), the update equations can be written as

$$u_{ki} = \frac{(1 - \kappa_C(x_i, v_k))^{-1/(m-1)}}{\sum_{j=1}^{K} (1 - \kappa_C(x_i, v_j))^{-1/(m-1)}}, \tag{4–12}$$

$$v_k = \frac{\sum_{i=1}^{N} u_{ki}^m \kappa_C(x_i, v_k)^2 x_i}{\sum_{i=1}^{N} u_{ki}^m \kappa_C(x_i, v_k)^2}. \tag{4–13}$$

Although the coordinates of centers in the feature space cannot be evaluated due to dimensionality, the corresponding ones in the input space (Equation (4–13)) can be calculated when the Cauchy kernel is used. In Equation (4–13), the kernel function $\kappa_C(x_i, v_k)$ works as a weighting function. The function $\kappa_C(x_i, v_k)$ weights a data point based on the similarity between x_i and v_j, which results in noise robustness. To do a global search, Equation (4–6) also should be kernelized as follows:

$$J_k^l = \sum_{i=1}^{N} \left(\sum_{j=1}^{k} \left(||\phi(x_i) - \phi(v_j)||^2 \right)^{1/(1-m)} \right)^{1-m}. \tag{4–14}$$

In Equation (4–14) the distance term between a data point and a center, $||\phi(x_i) - \phi(v_j)||$, should be divided in two according to the type of a center: one from the previous $(k-1)$ clustering problem and the other from a data point,

$$J_k^l = \sum_{i=1}^{N} \left(\sum_{j=1}^{k-1} \left(||\phi(x_i) - \phi(v_j)||^2 \right)^{1/(1-m)} + \left(||\phi(x_i) - \phi(x_l)||^2 \right)^{1/(1-m)} \right)^{1-m}, \tag{4–15}$$

which can be simplified as

$$J_k^l = \sum_{i=1}^{N} \left(\sum_{j=1}^{k-1} (2(1 - \kappa_C(x_i, v_j))^{1/(1-m)} + (2(1 - \kappa_C(x_i, x_l))^{1/(1-m)} \right)^{1-m}. \tag{4–16}$$

The final thing that should be considered is the evaluation order of the update equations. At the beginning of a $k-$clustering problem, the membership in Equation (4–12) must be calculated first to incorporate the current problem into the prior information – the optimal clustering result from $(k-1)$ clustering problem and the data point minimizing Equation (4–16). Algorithm 4-2 summarizes the kernel FCM with Cauchy kernel (K-FCM-C) algorithm with initialization. Kernel-based global FCM with Cauchy kernel (KG-FCM-C) can be summarized as Algorithm 4-3. The proposed algorithm, KG-FCM-C, can be used to decide initial seeds efficiently and is more

Algorithm 4-2 K-FCM with Cauchy kernel

Input: X : data set $(N \times d)$, V_{k-1} : initial center $((k-1) \times d)$, α : data point index, K : number of clusters

1: $V_k = V_{k-1} \cup x_\alpha$
2: **repeat**
3: **for** $k = 1$ to K **do**
4: **for** $i = 1$ to N **do**
5: $$u_{ki} = \frac{(1 - \kappa_C(x_i, v_k))^{-1/(m-1)}}{\sum_{j=1}^{K} (1 - \kappa_C(x_i, v_j))^{-1/(m-1)}}$$
6: **end for**
7: $$v_k = \frac{\sum_{i=1}^{N} u_{ki}^m \kappa_C(x_i, v_k)^2 x_i}{\sum_{i=1}^{N} u_{ki}^m \kappa_C(x_i, v_k)^2}$$
8: **end for**
9: **until** U satisfies the convergence criterion
10: **return**

Algorithm 4-3 KG-FCM with Cauchy kernel

Input: K : number of clusters, X : data set $(N \times d)$

1: $V_1 = \left\{ \frac{1}{N} \sum_{i=1}^{N} x_i \right\}$
2: **for** $k = 2$ to K **do**
3: **for** $l = 1$ to N **do**
4: $$J_k^l = \sum_{i=1}^{N} \left(\sum_{j=1}^{k} \left(||\phi(x_i) - \phi(v_j)||^2 \right)^{1/(1-m)} \right)^{1-m}$$
5: **end for**
6: $\alpha = \arg\min_{1 \le l \le N} J_k^l$
7: $[V_k, U_k] \leftarrow$ K-FCM-C(X, V_{k-1}, α, k)
8: **end for**
9: **return**

robust to outliers than the existing methods. However, it still cannot efficiently manage non-convex clusters due to its noise suppressing property. This problem can be tackled by the introduction of a random walk kernel using the random walk distance.

The random walk distance is a graph theoretic distance, in which each data point is considered as a node and the distance between two points is defined based on Markov property. The random walk distance was first proposed by Klein and Randic [83] and is also known as resistance distance following the electronic circuit analogy. Of the two distance measures that can be computed following the property of Markov chain, the *average commute time* is used here. The average commute time is defined

Algorithm 4-4 Average commute time

1: Build an affinity matrix A,
$A_{ij} = A(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{\sigma_i \sigma_j}\right)$.
2: Calculate a diagonal degree matrix D,
$D_{ij} = \begin{cases} \sum_{k=1}^{N} A_{ik}, & \text{if } i = j, \\ 0, & otherwise. \end{cases}$
3: Calculate a Laplacian matrix $L = D - A$ and its pseudo-inverse L^+.
4: Build an average commute time matrix D_C,
$D_{C,ij} = V_G(L_{ii}^+ + L_{jj}^+ - 2L_{ij}^+)$, where $V_G = \sum_{k=1}^{N} D_{kk}$.
5: **return**

as the average time that a random walker, starting at node i, will take to enter node $j(\neq i)$ for the first time and go back to node i [84]. The traditional shortest path distance does not consider the number of paths connecting two nodes, but the random walk distance decreases as the number of paths increases or the length of a path decreases. The average commute time can be calculated in terms of the pseudo-inverse of the Laplacian matrix, L^+, as in Algorithm 4-4. In Algorithm 4-4, σ_i denotes the distance to the $(2d + 1)^{th}$ nearest neighbor from x_i. By deciding the parameter in this way, we can efficiently reduce both the effect of noise and clusters with different densities [90]. To obtain more detailed information, refer to [84] and the references therein. Using the average commute time, the random walk kernel can be defined as

$$\kappa_{RW}(x, y) = \exp\left(-\frac{D_C(x, y)}{\sigma^2}\right), \tag{4–17}$$

where σ is a kernel parameter. The update equation of the memberships for K-FCM with random walk kernel (K-FCM-RW) can be written in a way similar to Equation (4–12) as

$$u_{ki} = \frac{\left(1 - \kappa_{RW}(x_i, v_k)\right)^{-1/(m-1)}}{\sum_{j=1}^{K} \left(1 - \kappa_{RW}(x_i, v_j)\right)^{-1/(m-1)}}. \tag{4–18}$$

The cluster center cannot be evaluated in the feature space or in the input space, however, by dot-product $\phi(x_i)$ on both side of Equation (4–10), $\kappa_{RW}(x_i, v_k)$ can be

Algorithm 4-5 K-FCM with random walk kernel

Input: X : data set $(N \times d)$, U_{k-1} : membership matrix $(N \times (k-1))$, α : data point index, K : number of clusters

```
repeat
    for k = 1 to K do
        for i = 1 to N do
            u_ki = (1 − κ_C(x_i, v_k))^{−1/(m−1)} / Σ_{j=1}^{K} (1 − κ_C(x_i, v_j))^{−1/(m−1)}
        end for
    end for
until U satisfies the convergence criterion
return
```

evaluated as a weighted sum of dot-products between data points,

$$\kappa_{RW}(x_i, v_j) = \frac{\sum_{a=1}^{N} u_{ja}^m \kappa_{RW}(x_a, x_i)}{\sum_{a=1}^{N} u_{ja}^m}. \tag{4–19}$$

Equation (4–19) can be used to update membership values in Equation (4–18) and to calculate the objective function for seed selection in Equation (4–16). Algorithm 4-2 also should be modified, because the coordinates of centers cannot be evaluated in K-FCM-RW and the membership values describe the centers indirectly. Thus, the set of centers, V, in Algorithm 4-2 should be replaced with the membership matrix U, whose k^{th} column represents the membership vector to the kth cluster. The initial value U_1 in KG-FCM-RW can be represented as a $(N \times 1)$ vector of ones. Algorithm 4-5 summarizes K-FCM-RW with initialization and Algorithm 4-6 does KG-FCM-RW.

4.4 Experimental results

4.4.1 Experiments on artificial data sets

To investigate the effectiveness of the proposed method, the existing methods and proposed methods are implemented and tested using Matlab. In the first experiment, the usefulness of the global search technique is shown. Figure 4-1 shows a clustering result using FCM. The test data (D_7) consists of seven well-separated circular clusters, each of which has 50 points. The large dots represent the random initial seeds used in FCM.

Algorithm 4-6 KG-FCM with random walk kernel

Input: K : number of clusters, X : data set $(N \times d)$

1: $U_1 = [\,1\ 1\ \ldots\ 1]^T$
2: **for** $k = 2$ to K **do**
3: **for** $l = 1$ to N **do**
4: $$J_k^l = \sum_{i=1}^{N} \left(\sum_{j=1}^{k} \left(||\phi(x_i) - \phi(v_j)||^2 \right)^{1/(1-m)} \right)^{1-m}$$
5: **end for**
6: $\alpha = \arg\min_{1 \le l \le N} J_k^l$
7: $[U_k] \leftarrow$ K-FCM-RW(X,U_{k-1}, α, k)
8: **end for**
9: **return**

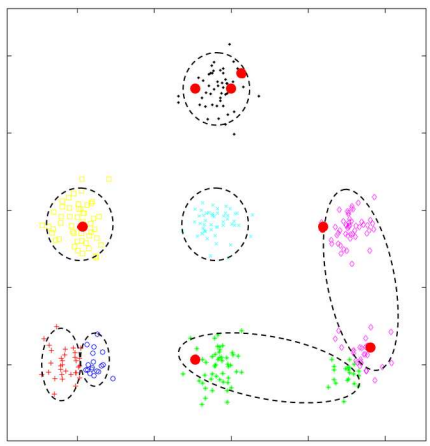
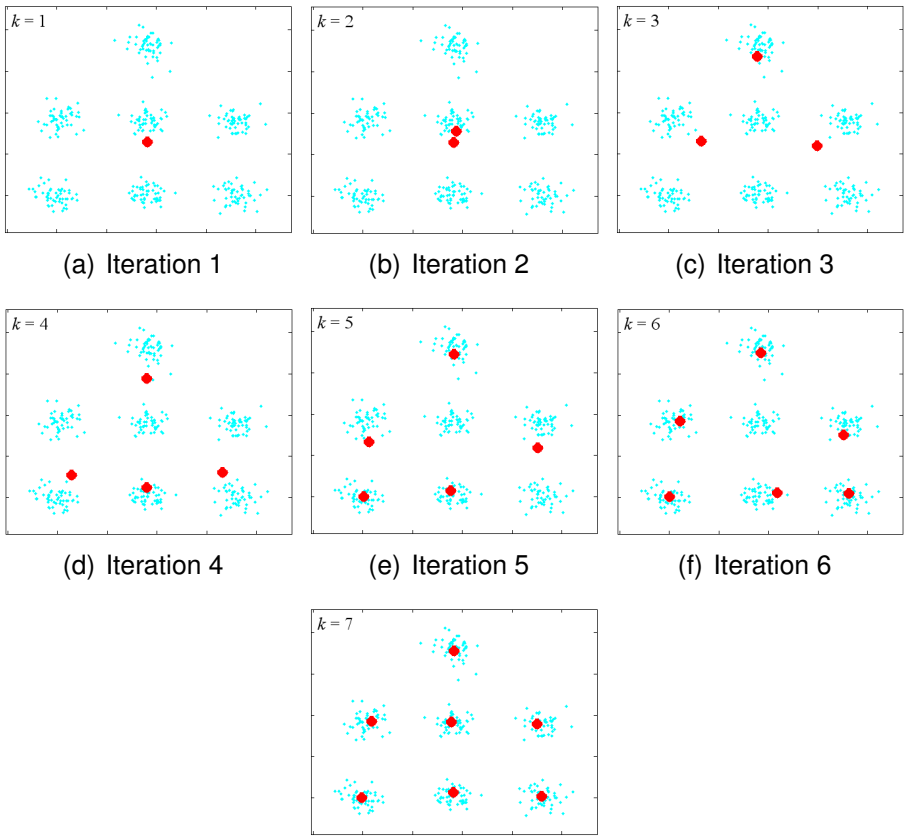


Figure 4-1. Sensitivity to initialization in FCM.

Although FCM finds the correct structure most of the time, it sometimes fails due to the convergence to a local optimum even in a data set with well-separated clusters. Only three clusters out of seven are correctly identified in Figure 4-1. Figure 4-2 shows the initial seeds in G-FCM, which are decided in a deterministic way. Conceptually, KG-FCM works exactly the same way as that in Figure 4-2, but is hard to visualize due to the dimensionality.

Table 4-1 summarizes the clustering results for D_7. If not stated otherwise, all the numbers reported are numbers averaged over 100 runs. FCM occasionally falls into a local optimum. K-FCM-C is a little better than FCM, but sometimes it also can fall into a local optimum. K-FCM-RW is the least satisfactory because the random walk distance is



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

(d) Iteration 4 (e) Iteration 5 (f) Iteration 6

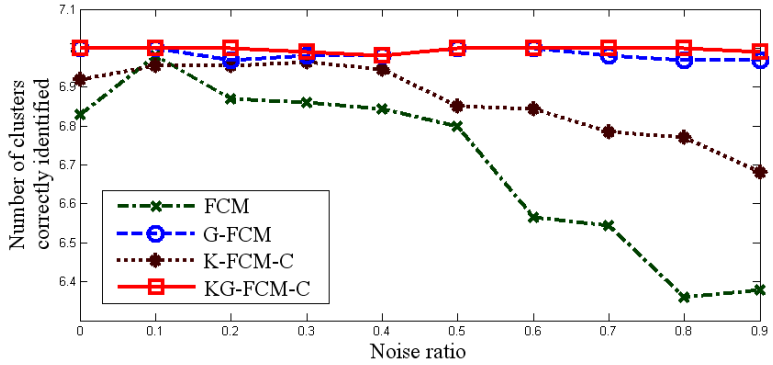
(g) Iteration 7

Figure 4-2. Incremental seed selection in G-FCM

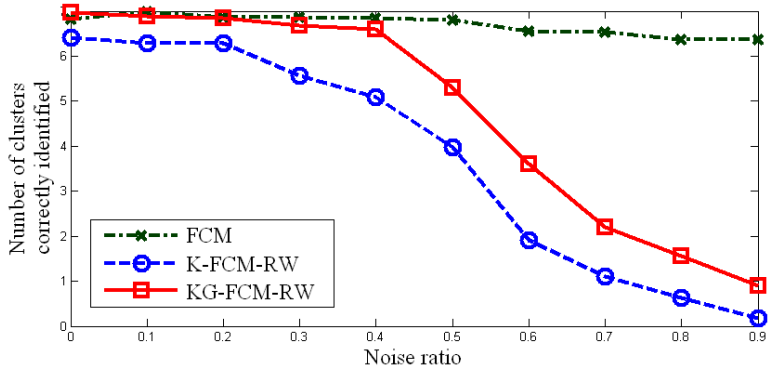
Table 4-1. Experimental results on D_7 (N = number of clusters correctly identified, $var(N)$ = variance of N)

Method	N	$var(N)$
FCM	6.82	0.4925
G-FCM	7.00	0.0000
K-FCM-C	6.92	0.3168
KG-FCM-C	7.00	0.0000
K-FCM-RW	6.42	3.1164
KG-FCM-RW	6.96	0.0788

based on the path connectivity between points, which is sensitive to outliers. The large variance for K-FCM-RW also corroborates this. The two global methods, G-FCM and KG-FCM-C, always find the correct structure of D_7. KG-FCM-RW is much better than K-FCM-RW, but not as good as other global methods.



(a)



(b)

Figure 4-3. Number of clusters correctly identified with respect to noise ratio.

To test robustness to noise, some noise points are added to D_7 and the same experiments are repeated. Noise points are sampled from a uniform distribution and the noise ratio is given as the ratio of the number of noise points to the number of data points. Figure 4-3 shows the number of correctly identified clusters with respect to the noise ratio. For a clear comparison, two subsets out of six methods are plotted on different scales. From Figure 4-3(a), we can conclude that the global search methods are more robust to noise than the random initialization methods, and KG-FCM-C is best. As the noise ratio increases, the random initialization methods tend to fall into a local optimum, but the global search methods do not suffer noise problems and find the right structure. In Figure 4-3(b), we can see that K-FCM-RW collapses as the noise ratio increases. Although global search technique helps, KG-FCM-RW also fails to find the right structure in noisy conditions.

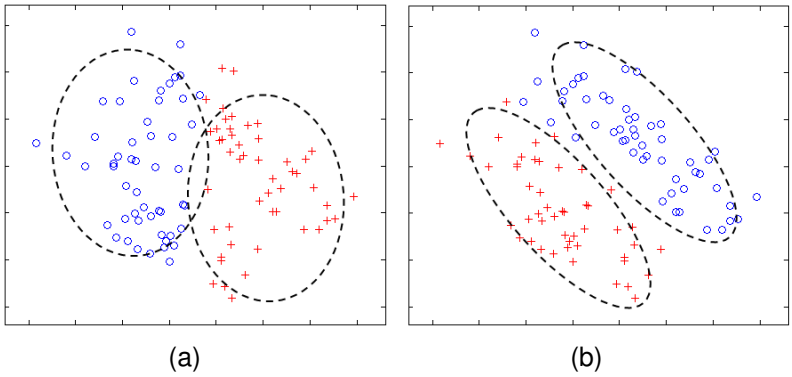


Figure 4-4. Clustering results using (a) FCM and (b) K-FCM-C with random initialization.

Another merit of the kernel-based methods is that they can cluster irregularly shaped clusters. Figure 4-4 shows clustering results on a sample data set consisting of two elongated clusters ($D_{parallel}$), each of which has 50 data points generated from a Gaussian distribution. By the introduction of another distance measure, for example, Mahalanobis distance instead of Euclidean distance, FCM can also cluster the elongated clusters correctly. However the kernel-based methods have another merit, outlier robustness, in addition to that. Figure 4-5 summarizes the error rate for $D_{parallel}$ with respect to the distance between two cluster centers ($d_{between}$) and Figure 4-6 does the same for the variance of the error rate. The error rate is defined as the number of mis-clustered points divided by the number of data points. In this experiment, the methods using the global search technique showed almost the same result to the corresponding methods with random initialization, so the results of the global methods are not plotted. As is clear from Figure 4-5, FCM with Mahalanobis distance achieves the best result when $d_{between}$ is small because the measure is specialized for Gaussian distributions. However it sometimes fails to separate the clusters even when the $d_{between}$ is large enough, because the size of the data is not large enough to estimate parameters, including the covariance matrix. The overall large variance also supports that. Another interesting point in Figure 4-6 is that the variance for K-FCM-C decreases as $d_{between}$ increases. On the other hand, FCM and K-FCM-RW have a peak, which

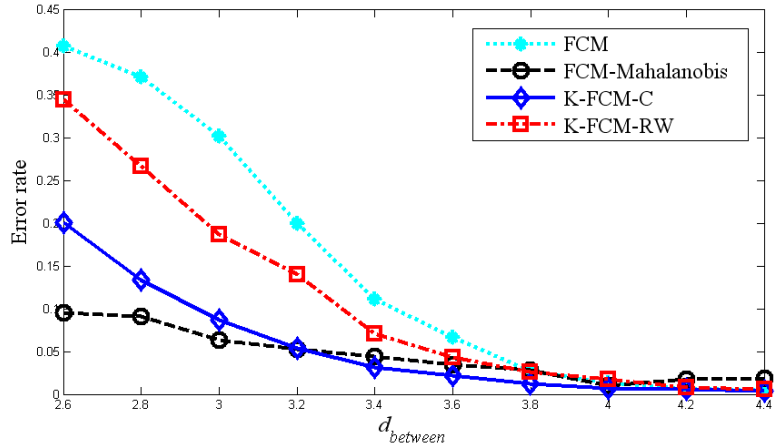


Figure 4-5. Error rate with respect to $d_{between}$ in $D_{parallel}$

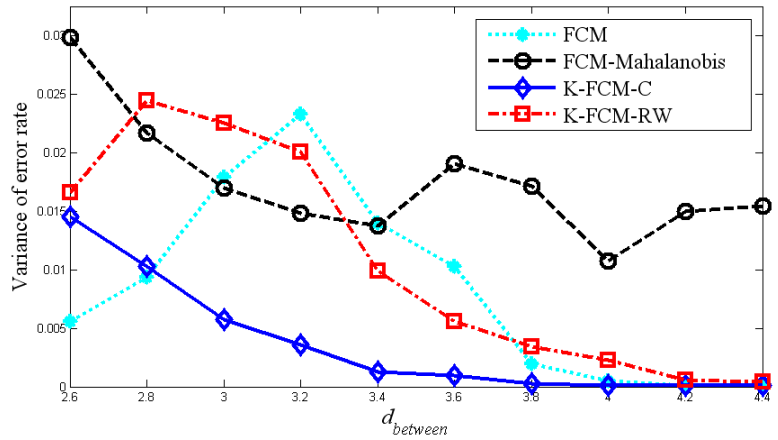


Figure 4-6. Variance of error rate with respect to $d_{between}$ in $D_{parallel}$

can be considered as a threshold that determines marginal performance. FCM with Mahalanobis distance shows a relatively large variance for all values of $d_{between}$ and does not show a clear relationship between $d_{between}$ and the variance of the error rate.

Although the random walk kernel is noise-sensitive, it is suitable for non-convex clusters. Figure 4-7 shows a sample data set that is not linearly separable ($D_{non-linear}$). FCM and K-FCM-C cannot separate the two clusters in Figure 4-7 for two different reasons. First, FCM fails to separate the clusters mainly due to the mean squares measure, so FCM tends to separate $D_{non-linear}$ into two clusters in a vertical direction. Second, K-FCM-C fails to separate the clusters due to its noise suppressing property. As a result, K-FCM-C considers the points at the end of semi-circles as outliers and

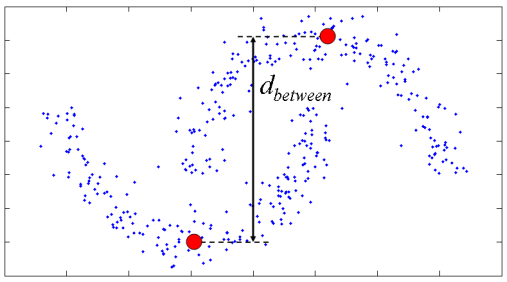


Figure 4-7. Non-linearly separable clusters.

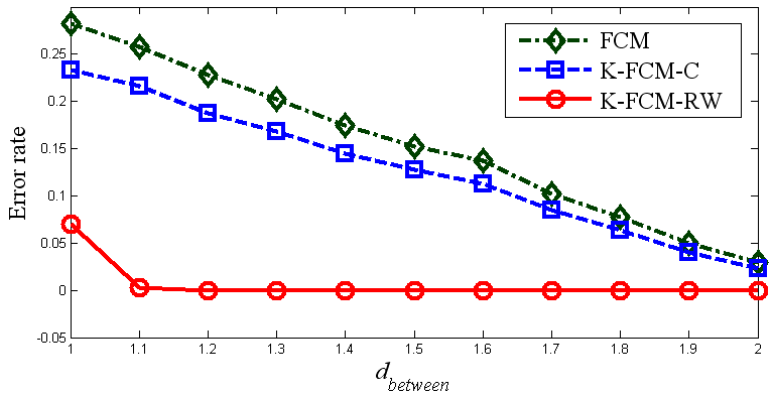


Figure 4-8. Error rate with respect to $d_{between}$ in $D_{non-linear}$.

tends to separate $D_{non-linear}$ into two clusters in a horizontal direction. Figure 4-8 shows the error rate with respect to the distance, $d_{between}$, depicted in Figure 4-7. As the distance decreases, the error rate of FCM increases. K-FCM-C is slightly better than FCM, but K-FCM-C appears to be a more noise-robust method than a non-linear method, as described above, although the cluster boundaries in K-FCM-C are not linear. Only K-FCM-RW can separate the non-convex clusters, but it also fails when two clusters are too close.

4.4.2 Experiments on real world data sets

In this section, the proposed methods are applied to some real world data sets. All the data sets are available from the UCI Machine Learning Repository [72]. To compare the results using different clustering methods, an information-based distance, a modified *variation of information* [91], is used. Suppose two different clusterings, $C = \{C_1, C_2, \ldots, C_K\}$ and $C' = \{C'_1, C'_2, \ldots, C'_{K'}\}$ (corresponds to the set of known labels), which cluster

the data into K and K' clusters. The probability of an outcome being in cluster C_k equals $p(k) = N_k/N$, where N_k is the number of points belonging to the kth cluster. N is the number of data points and satisfies $N = \sum_{k=1}^{K} N_k = \sum_{k=1}^{K'} N'_k$. The information contained in the clustering C' can be represented as the entropy of the random variable $p(k')$,

$$H(C') = -\sum_{k=1}^{K'} p(k') \log p(k'), \tag{4–20}$$

and the amount of information common to both clusterings can be represented as the mutual information of the two random variables, $p(k)$ and $p(k')$,

$$I(C, C') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} p(k, k') \log \frac{p(k, k')}{p(k)p(k')}, \tag{4–21}$$

where $p(k, k') = |C_k \cap C'_{k'}|/N$ is the probability that a point belongs to C_k in clustering C and to $C'_{k'}$ in C'. Then the difference between the two terms represents the amount of information that is not described by a clustering C:

$$D_I(C, C') = H(C') - I(C, C'). \tag{4–22}$$

Equation (4–22) is zero if and only if the two clusterings are identical when $K = K'$, and can be considered as distance of a clustering (C) from the ground truth (C'). As the number of clusters in C increases, the mutual information increases and D_I decreases. Experimentally, clustering with large number of clusters, $K > K'$, showed a smaller D_I than that with the same number of clusters, $K = K'$, but the relative performance did not change. Therefore, the number of clusters (K) is set to be equal to the number of classes (K') in the data set. Table 4-2 summarizes the experimental results using UCI data sets and all the values in Table 4-2 are D_I values averaged over 100 runs.

***Iris data*:** On average, the global methods are better than the corresponding methods using random initialization, and the kernel-based methods are better than the input space method. KG-FCM-RW is the best because iris data is relatively noise-free and one of the three clusters is clearly separated from the others. One interesting thing is

that K-FCM-RW is likely to fall into a local optimum, which is much worse than that of KG-FCM-RW. This means that the solution space of K-FCM-RW has many local optima and is different from that of KG-FCM-RW. Conversely, FCM and K-FCM-C rarely fall into a local minimum and each has a smooth solution space.

***Wine data*:** As before, the kernel-based methods show better results, and KG-FCM-C shows the best result. Two things that should be noted are that K-FCM-RW with random initialization hits the minimum and that, on average, K-FCM-RW is better than KG-FCM-RW.

***Breast cancer data*:** Although the number of classes is only two, the dimensionality of this data set is much higher than those in previous data sets. Again, the global methods are better than the random methods, but K-FCM-RW performs worst. This is mostly due to the small size of the data, which distributes the data points sparsely in high dimensional space and makes it difficult to estimate the connectivity information. KG-FCM-C also fails to achieve the minimum of the random method, although on average it is better.

***Yeast data*:** Only 5 of the 10 original classes are used in this experiment. The results are almost the same as the previous ones on breast cancer data. As the number of clusters increases, K-FCM-RW becomes more unstable and K-FCM-RW performs the worst in this experiment. The values for K-FCM-C also suggest the different structures of solution space between K-FCM-C and KG-FCM-C.

From these experiments, we can conclude that:

1. The global search technique is useful for deciding initial seeds, although the global methods sometimes fail to achieve the minimum of the corresponding random methods. This failure is mainly due to the different structures of the solution space. The random methods can only have data points as their initial positions, but the global methods can have non-preoccupied positions, which is another reason for the difference.

Table 4-2. Experimental results using UCI data sets (N = number of samples, d = data dimension, K' = number of classes)

Data set	N	d	K'	Method	D_I value			
					Global method	Random method		
						Min.	Average	Max.
Iris	150	4	3	FCM	0.4041	0.4041	0.4042	0.4045
				K-FCM-C	0.3898	0.3898	0.3898	0.3898
				K-FCM-RW	0.2663	0.6057	0.6164	0.6365
Wine	178	13	3	FCM	1.2944	1.2944	1.3088	1.3386
				K-FCM-C	1.2284	1.2284	1.2368	1.3128
				K-FCM-RW	1.2683	1.1542	1.2351	1.4986
Breast cancer	569	30	2	FCM	0.9329	0.9329	0.9339	0.9535
				K-FCM-C	0.8712	0.8243	0.9224	0.9535
				K-FCM-RW	0.9381	0.9202	0.9393	0.9535
Yeast	180	8	5	FCM	1.6129	1.6043	1.6139	1.7376
				K-FCM-C	1.5855	1.8001	1.8001	1.8001
				K-FCM-RW	1.8021	1.6755	1.843	1.9138

2. KG-FCM-C is more robust to outliers than other methods and shows better and more stable results on test data sets.
3. KG-FCM-RW shows the best result on *iris data*, which is clean and has a relatively well separated cluster structure. However, as the dimensionality or the number of clusters increases, it is more likely to fall into a local optimum.

4.5 Discussion

Fuzzy c-means (FCM) is a simple and efficient method for clustering. However there are also several well-known shortcomings to FCM. In this chapter, we developed kernel-based global fuzzy c-means (KG-FCM) to solve the sensitivity to initialization, sensitivity to noise, and inability to handle non-convex clusters. KG-FCM is based on global fuzzy c-means (G-FCM), which is a variant of FCM with a deterministic initialization, and is extended with the help of kernel method on it. KG-FCM is implemented using two different kernels: the Cauchy kernel to suppress noise and the random walk kernel to cluster non-convex clusters. Experimental results show that, on average, the global methods are better than the random methods, and KG-FCM-C gave the best

results on both artificial and real world data sets most of the time. KG-FCM-RW showed good results only when the cluster structure is simple and well-separated, for example, on *iris data*.

Although the proposed algorithm, KG-FCM, is better than existing methods, it still has some problems. Initially, the kernel method is adopted because of the various effects obtained using different kernels. As is clear from the experiments, the Cauchy kernel is good at suppressing outliers and the random walk kernel is efficient at separating non-convex clusters. However, each kernel has the other's strength as its weakness. Still another kernel may combine the strengths of the two; this is under investigation. Another weakness of KG-FCM is that the global method sometimes shows poorer results than the corresponding random method. This is mainly due to the difference in solution space. A different seed selection function may solve this problem, which is left for further research. The last thing that should be mentioned concerns the selection of kernel parameters. The kernel parameters, β in the Cauchy kernel and σ in the random walk kernel, are decided using cross-validation here. There is on going research to efficiently determine the kernel parameters, which is also an important further research area.

CHAPTER 5
FSVM-N: FUZZY SUPPORT VECTOR MACHINE FOR NOISY DATA

Support vector machine (SVM) is a set of supervised machine learning methods based on structural risk minimization. After its introduction by Vapnik [92], it has been successfully used in many applications, such as handwriting recognition [92], image retrieval [93], and text categorization [94]. In spite of the successes, SVM still has a *noise sensitivity problem*[1] . The problem originates from the assumption that each training point has equal importance or weight in training. In many real world problems, however, there are cases where some training points are corrupted by noise and they should be treated as less important. Although SVM generalizes well due to regularization by balancing the margin and error, the error term based on sum of squares still can skew the decision boundary toward outliers. To relax the noise sensitivity problem, several SVM variants have been proposed, includingly fuzzy SVM (FSVM) [95], new fuzzy SVM [96], prior knowledge SVM [97], posterior probability SVM [98], and soft SVM [99]. Each is a little different from the others in formulation, but they share the basic idea of different weights for different data points. Although the SVM variants are useful in noisy conditions, their basic limitations are that they assume some domain-specific knowledge and that the weights are assumed to be known or to be calculated easily using that knowledge. There are also several methods for estimating the weights based solely on the data [95], [100]–[102]. All of them are based on FSVM and introduce their own measures of outlier-ness. One problem in these methods is that most of the measures are based on the Euclidean distance, which results in the assumption of circular data distribution.

[1] Noise sensitivity problem is not unique to SVM and all the classifiers suffer from the problem in applying themselves to real world problems. Introducing fuzzy memberships is one of the most common approaches to mitigate the problem and fuzzy SVM also adopts this strategy.

There is another group of SVM variants to accommodate imbalanced data sets which are quite common in recent applications. The SVM variants for imbalanced data can be formulated in a similar way to FSVM. However, as none of the previous membership calculation methods considered the imbalance in data and the performance of a classifier on an imbalanced data set cannot be correctly estimated with classification accuracy, all the data sets used in this chapter are assumed to be balanced.

In this chapter, we develop a new membership calculation method based on the reconstruction error, which is based on the statistics of data and does not require any domain knowledge. The method assumes that data follows an Gaussian distribution in the feature space, which can accommodate more distributions than circular assumption. The reconstruction error measures the degree of conformity of a point to overall data structure using principal component analysis (PCA). Using the reconstruction error, the membership values can be decided in a robust way and the robust memberships result in more accurate classification than SVM and its variants. Experimental results with synthetic and real data sets also support this.

In the next section, SVM is briefly overviewed. In section 5.2 the previous membership calculation methods and their weaknesses are described in detail. The proposed method is introduced in section 5.3. Experimental results are given in section 5.4 and the discussion section concludes this chapter.

5.1 Support vector machine

Suppose a set of labeled training points $\{(x_i, y_i)|1 \leq i \leq N\}$. Each training point $x_i \in R^d$ belongs to either of two classes and given a label $y_i \in \{-1, 1\}$. One wishes to find the hyper-plane $w \cdot x + b = 0$ defined by the weight vector and the offset pair (w, b), such that the inequality

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \forall i = 1, \ldots, N \tag{5–1}$$

holds, where $\xi_i \geq 0$ is an error. The optimal hyper-plane problem is regarded as the solution to the problem [5]

$$\min \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i, \tag{5–2a}$$

$$\text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, N, \tag{5–2b}$$

$$\xi_i \geq 0, \quad \forall i = 1, \dots, N, \tag{5–2c}$$

where C is a regularization constant, which is the only free parameter in SVM formulation. Searching for the optimal solution of Equation (5–2) can be solved by constructing a Lagrange equation and transforming it into its dual problem

$$\max \quad \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j), \tag{5–3a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0, \tag{5–3b}$$

$$0 \leq \alpha_i \leq C, \quad \forall i = 1, \dots, N, \tag{5–3c}$$

where $\alpha = (\alpha_1, \dots, \alpha_N)$ is the vector of non-negative Lagrange multipliers associated with the constraints in Equation (5–2). According to the Karush-Kuhn-Tucker (KKT) condition, the solution α_i of Equation (5–3) satisfies the following equalities:

$$\alpha_i(y_i(w \cdot x_i + b) - 1 + \xi_i) = 0, \quad \forall i = 1, \dots, N, \tag{5–4}$$

$$(C - \alpha_i)\xi_i = 0, \quad \forall i = 1, \dots, N. \tag{5–5}$$

From these equalities, it comes that the only non-zero values α_i in Equation (5–4) correspond to the points for which the constraints in Equation (5–2b) are satisfied with equality. The point x_i corresponding with $\alpha_i > 0$ is called a *support vector*, which is divided into two types. In the case $0 < \alpha_i < C$, the corresponding support vector x_i satisfies the equalities $y_i(w \cdot x_i + b) = 1$ and $\xi_i = 0$. In the case $\alpha_i = C$, the corresponding ξ_i is not zero and the corresponding x_i is considered as mis-classified.

The points x_i corresponding with $\alpha_i = 0$ are classified correctly and clearly away from the decision boundary. These points have no influence on the formation of the decision boundary. To construct the optimal hyper-plane, it follows that

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{5–6}$$

and the scalar b can be determined from Equation (5–4). The decision function, thus, can be written as

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x) + b\right). \tag{5–7}$$

This formulation can be easily extended into non-linear SVM, in which the data points are first mapped into a high dimensional space using a mapping function Φ and the linear SVM is applied. By replacing each data point x_i with the mapped one $\phi(x_i)$ and the product with kernel function $\phi(x_i) \cdot \phi(x_j) = \kappa(x_i, x_j)$, one can get the equations of non-linear SVM.

5.2 Previous approaches to membership calculation

Given a set of labeled training points with fuzzy memberships, $\{(x_i, y_i, u_i) | 1 \leq i \leq N\}$. Each training point $x_i \in R^d$ is given a label $y_i \in \{-1, 1\}$ and a fuzzy membership $0 < u_i \leq 1$. The optimal hyper-plane problem is then regarded as the solution to a modified version of SVM, called fuzzy SVM (FSVM):

$$\min \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} u_i \xi_i, \tag{5–8a}$$

$$\text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \forall i = 1, \ldots, N, \tag{5–8b}$$

$$\xi_i \geq 0, \quad \forall i = 1, \ldots, N. \tag{5–8c}$$

It is noted that a small u_i reduces the effect of the parameter ξ_i such that the corresponding point x_i is treated as less important. The problem in Equation (5–8) can be transformed

into its dual

$$\max \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \kappa(x_i, x_j), \tag{5–9a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \tag{5–9b}$$

$$0 \leq \alpha_i \leq u_i C, \quad \forall i = 1, \dots, N, \tag{5–9c}$$

and the corresponding KKT conditions can be written as

$$\alpha_i (y_i (w \cdot z_i + b) - 1 + \xi_i) = 0, \quad \forall i = 1, \dots, N, \tag{5–10}$$

$$(u_i C - \alpha_i) \xi_i = 0, \quad \forall i = 1, \dots, N. \tag{5–11}$$

The point x_i corresponding with $\alpha_i > 0$ is called a *support vector* and there are also two types of support vectors as in SVM. The data point corresponding $0 < \alpha_i < u_i C$ lies on the margin of the hyper-plane and the one with $\alpha_i = u_i C$ is a mis-classified one. In FSVM, the points with the same value of α_i may indicate a different type of support vectors due to the different upper bound $(u_i C)$. Another difference is that there is only one free parameter C in SVM while the number of free parameters in FSVM is equal to the number of training points.

FSVM was designed to improve the performance of SVM by reducing the effect of noise. There are also some FSVM variants to improve the performance further, which can be divided into two groups. One group of methods is to formulate the objective function in a different way to incorporate domain-specific knowledge. The other is to estimate memberships using the statistics of data [95], [100]–[102]. The proposed method belongs to this latter group.

Lin et al. [95] proposed a membership calculation method together with the original FSVM, in which membership is given as a function of the distance between a data point and its corresponding class center, referred to as *FSVM in the input space* or FSVM-I.

More precisely, the membership is given as

$$u_i = \begin{cases} 1 - \dfrac{||\bar{x}_+ - x_i||}{r_+ + \delta}, & \text{if } \ i \in I_+, \\ 1 - \dfrac{||\bar{x}_- - x_i||}{r_- + \delta}, & \text{if } \ i \in I_-. \end{cases} \tag{5–12}$$

where $I_+ = \{i | y_i = +1\}$ and $I_- = \{i | y_i = -1\}$ are the index sets of positive and negative samples, $\bar{x}_\pm$ are class means,

$$\bar{x}_\pm = \frac{1}{|I_\pm|} \sum_{i \in I_\pm} x_i, \tag{5–13}$$

$r_\pm$ are class radii,

$$r_\pm = \max_{i \in I_\pm} ||\bar{x}_\pm - x_i||, \tag{5–14}$$

and $\delta > 0$ is a constant to avoid the case $u_i = 0$. In Equation (5–13), $|I|$ represents the cardinality of the set. One problem in this method is that the membership values are calculated based on the Euclidean distance. When the distribution of data points is not spherical, the contribution of each point to the decision boundary may not be represented properly. To mitigate this problem, Jiang et al. [100] proposed a kernelized version of FSVM-I, called *FSVM in the feature space* or FSVM-F. In the feature space, the center of a cluster can be represented as

$$\phi(\bar{x}_\pm) = \frac{1}{|I_\pm|} \sum_{i \in I_\pm} \phi(x_i). \tag{5–15}$$

Using Equation (5–15), the distance between a mapped point $\phi(x_i)$ and a center $\phi(\bar{x}_\pm)$ can be represented as

$$
\begin{aligned}
&D^2(\phi(\bar{x}_\pm), \phi(x_i)) \\
&\quad = ||\phi(\bar{x}_\pm) - \phi(x_i)||^2, \\
&\quad = \phi(\bar{x}_\pm) \cdot \phi(\bar{x}_\pm) - 2\phi(\bar{x}_\pm) \cdot \phi(x_i) + \phi(x_i) \cdot \phi(x_i), \\
&\quad = \frac{1}{|I_\pm|^2} \sum_{j \in I_\pm} \phi(x_j) \cdot \sum_{k \in I_\pm} \phi(x_k) - \frac{2}{|I_\pm|} \sum_{j \in I_\pm} \phi(x_j) \cdot \phi(x_i) + \phi(x_i) \cdot \phi(x_i), \\
&\quad = \frac{1}{|I_\pm|^2} \sum_{j \in I_\pm} \sum_{k \in I_\pm} \kappa(x_j, x_k) - \frac{2}{|I_\pm|} \sum_{j \in I_\pm} \kappa(x_j, x_i) + \kappa(x_i, x_i),
\end{aligned}
\tag{5–16}
$$

where $\kappa(\cdot, \cdot)$ represents the kernel function. Using Equation (5–16), the class radii in the feature space can be written as

$$
\phi(r_\pm) = \max_{i \in I_\pm} ||\phi(\bar{x}_\pm) - \phi(x_i)||, \tag{5–17}
$$

and the membership of FSVM-F can be written as

$$
u_i = \begin{cases} 1 - \dfrac{||\phi(\bar{x}_+) - \phi(x_i)||}{\phi(r_+) + \delta}, & \text{if} \quad i \in I_+, \\ 1 - \dfrac{||\phi(\bar{x}_-) - \phi(x_i)||}{\phi(r_-) + \delta}, & \text{if} \quad i \in I_-. \end{cases} \tag{5–18}
$$

The basic limitation of these two methods comes from the assumption of a circular data distribution. As outliers are detected based solely on the distance from corresponding class mean, these methods may produce good results only when the distribution of data points is spherical in the input or feature space.

Lin et al. [101] proposed another membership calculation method using *kernel target alignment* [103]. Kernel target alignment is a method for measuring the degree of agreement between a kernel and target values. The inner product between kernel matrices can be defined as

$$
< K_1, K_2 >= \sum_{i=1}^{N} \sum_{j=1}^{N} \kappa_1(x_i, x_j) \kappa_2(x_i, x_j). \tag{5–19}
$$

If we consider $K_2 = yy^T$, where y is the vector of data labels, then Equation (5–19) can be re-written as

$$< K_1, yy^T >= \sum_{i=1}^{N}\sum_{j=1}^{N} \kappa_1(x_i, x_j) y_i y_j. \tag{5–20}$$

In this method, a heuristic function derived from Equation (5–20) is evaluated as the *alignment* between a data vector and the label for that vector. Therefore, this method is referred to as *heuristic function FSVM* or H-FSVM. From Equation (5–20), the heuristic function is defined as

$$\begin{aligned} f(x_i, y_i) &= \sum_{j=1}^{N} y_i y_j \kappa(x_i, x_j), \\ &= \sum_{y_i = y_j} \kappa(x_i, x_j) - \sum_{y_i \neq y_j} \kappa(x_i, x_j). \end{aligned} \tag{5–21}$$

This method assumes that a data point can be considered as noise with high probability if this data point is more similar to the other class than its class. The value defined in Equation (5–21) is proportional to the difference of the average square distance to the points belonging to the same class and the average square distance to the points belonging to the other class when the data set is balanced and $\kappa(x, x) = 1$. Let $\overline{D}_{\pm}(x_k)$ be the mean square distances between a data point x_k and a set of data points belong to each class $\{x_i | i \in I_{\pm}\}$ which is defined as

$$\begin{aligned} \overline{D}_{\pm}(x_k) &= \frac{1}{|I_{\pm}|} \sum_{i \in I_{\pm}} ||x_k - x_i||^2, \\ &= \frac{1}{|I_{\pm}|} \sum_{i \in I_{\pm}} (\kappa(x_k, x_k) + \kappa(x_i, x_i) - 2\kappa(x_k, x_i)), \\ &= \kappa(x_k, x_k) + \frac{1}{|I_{\pm}|} \sum_{i \in I_{\pm}} (\kappa(x_i, x_i) - 2\kappa(x_k, x_i)). \end{aligned} \tag{5–22}$$

Then the value $y_k\left(\overline{D}_+(x_k) - \overline{D}_-(x_k)\right)$ is the difference of the mean square distances. Assume that $|I_+| \approx |I_-| \approx N/2$ and $\kappa(x, x) = 1$, then

$$
\begin{aligned}
& y_k(\overline{D}_+(x_k) - \overline{D}_-(x_k)) \\
& \qquad \approx y_k \left(\frac{2}{N} \sum_{i \in I_+} (1 - 2\kappa(x_k, x_i)) - \frac{2}{N} \sum_{i \in I_-} (1 - 2\kappa(x_k, x_i)) \right) \\
& \qquad = \frac{4y_k}{N} \left(-\sum_{i \in I_+} \kappa(x_k, x_i) + \sum_{i \in I_-} \kappa(x_k, x_i) \right) \\
& \qquad = -\frac{4}{N} \left(\sum_{i \in I_+} y_k \kappa(x_k, x_i) - \sum_{i \in I_-} y_k \kappa(x_k, x_i) \right) \\
& \qquad = -\frac{4}{N} f(x_k, y_k),
\end{aligned}
\tag{5–23}
$$

the difference is reduced to the heuristic function. This method also assumes a circular distribution of the data because the outlier-ness is based on the distance in the feature space as shown in Equation (5–23). Another problem in this method is that the value from Equation (5–21) can be positive and negative and the bound is dependent on the number of data points and the kernel function. To obtain the memberships, therefore, the value from Equation (5–21) should be rescaled. The rescaling function was defined as

$$
u_i = \begin{cases} \zeta, & \text{if } \; f(x_i, y_i) < h_T, \\ 1, & \text{if } \; f(x_i, y_i) > h_C, \\ \zeta + (1 - \zeta) \left(\dfrac{f(x_i, y_i) - h_T}{h_C - h_T} \right)^d, & \textit{otherwise}, \end{cases}
\tag{5–24}
$$

where h_C and h_T represent the maximum and minimum value of the heuristic function respectively, ζ is the minimum membership value, and d is the degree of mapping function. Figure 5-1 shows the mapping function with respect to the degree.

Shilton et al. [102] proposed an iterative method using the slack variable ξ_i, which represents the distance from a data point to the decision hyper-plane. The membership

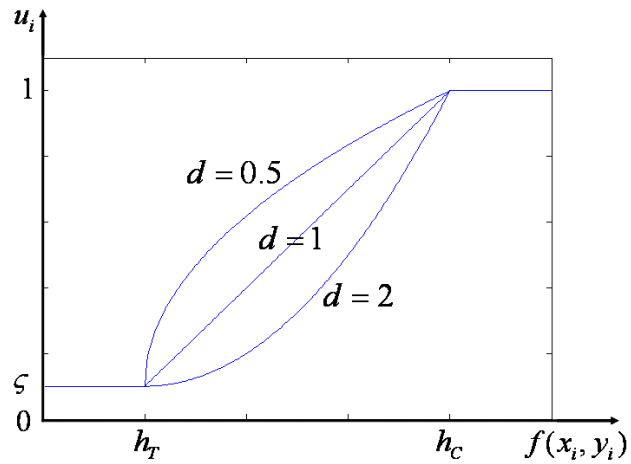


Figure 5-1. Mapping function for H-FSVM.

value is given as

$$u_i = h(\xi_i), \tag{5–25}$$

where h is a non-increasing function satisfying

$$\begin{aligned} &\lim_{\xi \to 0+} h(\xi) = 1, \\ &0 < h(\xi) \leq 1, \quad \forall \xi \geq 0, \end{aligned} \tag{5–26}$$

and $h(\xi) = \text{sech}^2(\xi)$ was used in [102]. Using these membership values, the decision boundary can be re-constructed using FSVM. The two steps – FSVM training and membership calculation – may be iterated using a fixed number of times or until the convergence of the membership vector. This method is referred to as *iterative FSVM* or I-FSVM. This method assumes that the points far from the decision boundary are less important in constructing the boundary and does not assume any distribution of the data. The defect of this method also arises from this assumption. Although the points adjacent to the boundary are important, the large membership values of these points may make the boundary over-fitted to the points and result in poor generalization. The increase in error rate as the number of iterations increases clearly shows the overfitting problem [102].

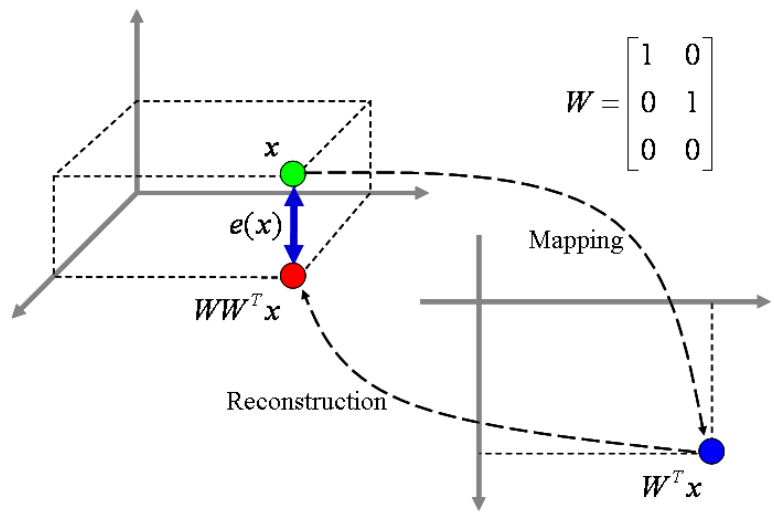


Figure 5-2. Reconstruction error $e(x)$.

5.3 Fuzzy SVM for noisy data

In this section, a new membership calculation method based on the reconstruction error is introduced. The reconstruction using principal component analysis (PCA) was previously used to smooth the data by removing the effect of noise [104], [105], thus the reconstruction error is a promising measure of outlier-ness. The reconstruction error is the distance between a d–dimensional data point x and the principal component reconstruction of x using $k(< d)$ principal components. Figure 5-2 shows an example of the reconstruction error. The dimensionality k is assumed to be known and will be discussed later. Let $X = \{x_1, \cdots, x_N\}$ be a set of samples. The reconstruction error of a centered data point in the input space can be represented as

$$e(x) = ||x - WW^T x||^2, \tag{5–27}$$

where W is a matrix having k principal components of sample covariance matrix as columns. In this chapter, to deal with non-linearly separable data, kernel PCA (KPCA) [106] is considered. KPCA is conceptually composed of two steps, mapping the data into a high-dimensional space and applying PCA to the mapped data. Sometimes the second step cannot be implemented due to the dimensionality of the mapped space.

However, the inner product of two vectors in the mapped space can be evaluated and that makes it possible to perform PCA in the mapped space indirectly, which is commonly called the *kernel trick*. Refer to [106] for a more detailed information about KPCA. Let $X_\phi = \{\phi(x_1), \cdots, \phi(x_N)\}$ be a set of mapped samples. Using KPCA, the reconstruction error in the mapped space can be calculated as

$$\begin{aligned} e(\phi(x)) =& ||(\phi(x) - \mu_\phi) - W_\phi W_\phi^T(\phi(x) - \mu_\phi)||^2, \\ =& \kappa(x, x) - \frac{2}{N}\sum_{i=1}^{N}\kappa(x_i, x) + \frac{1}{N^2}\sum_{i,j=1}^{N}\kappa(x_i, x_j) \\ & - 2\sum_{i=1}^{k}\left(\left(\beta^i\right)^2 - 2\beta^i\gamma^i + \left(\gamma^i\right)^2\right) \\ & + \sum_{l,m=1}^{k}\sum_{i,j=1}^{N}(\beta^l\beta^m - 2\beta^l\gamma^m + \gamma^l\gamma^m)\alpha_i^l\alpha_j^m\kappa(x_i, x_j), \end{aligned} \tag{5–28}$$

where N is the number of data points, μ_ϕ is the mean of the data in the mapped space, and $\kappa(\cdot, \cdot)$ is the kernel function. The matrix W_ϕ has k principal components of covariance matrix of X_ϕ. As the dimensionality of $\phi(x)$ is very high and sometimes infinite, it is impossible to build the covariance matrix explicitly. Therefore, the reconstruction error in the feature space is evaluated indirectly using the eigenvectors of the kernel matrix. The vector $\alpha^k = [\alpha_1^k, \cdots, \alpha_N^k]$ $(1 \leq k \leq N)$ is the eigenvector of the kernel matrix K having the kth largest eigenvalue. The (i, j) element of the kernel matrix is given by

$$K_{ij} = \phi(x_i) \cdot \phi(x_j) = \kappa(x_i, x_j). \tag{5–29}$$

The values β^j and γ^l are defined by

$$\beta^j = \sum_{i=1}^{N}\alpha_i^j\kappa(x_i, x), \tag{5–30}$$

$$\gamma^l = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i^l\kappa(x_i, x_j). \tag{5–31}$$

The detailed derivation of Equation (5–28) can be found in Appendix B[2] . Although the eigenvectors of W cannot be calculated due to the dimensionality, the eigenvalues of K can be calculated, which results in the indirect calculation of the reconstruction error. Using Equation (5–28), the membership value is defined as

$$u_i = \exp\left(-\frac{e'(\phi(x_i))}{\sigma_N^2}\right), \tag{5–32}$$

where σ_N is a free parameter. The function $e'(\cdot)$ performs rescaling on $e(\cdot)$ and is defined as

$$e'(\phi(x_i)) = \max\left(0, \frac{e(\phi(x_i)) - \mu_e}{\sigma_e}\right), \tag{5–33}$$

where σ_e and μ_e are the mean and variance of the reconstruction error, respectively. This rescaling treats data points with reconstruction error smaller than the mean reconstruction error equally by assigning them all a membership value of 1. The rescaling also makes it possible for the algorithm to accommodate classes with different densities by normalizing the reconstruction error having unit variance. Figure 5-3 shows the membership calculation function in FSVM-N. The value u_i represents the degree of typical-ness, and a typical point is defined as a point with small reconstruction error. A large value of σ_N makes u_i = 1 for all $1 \leq i \leq N$, which makes all the sample points have equal weights and reduces the proposed method to SVM. A small value of σ_N, on the other hand, throws most of the points away in the training. Thus, the parameter value of σ_N should be decided carefully and the grid search method [107] is used to set that value.

Still remaining is the decision on the number of principal components k. In this chapter, the *profile likelihood method* (PLM) [108] is used to decide k. The PLM is a

[2] Appendix B concerns the derivation of the reconstruction error with memberships. The equation for the reconstruction error without membership can be obtained by setting the memberships to 1, $u_i = 1$ for all $1 \leq i \leq N$.

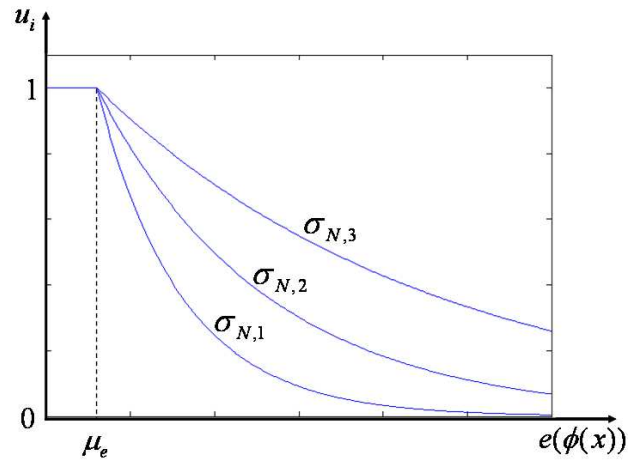


Figure 5-3. Rescaling function for FSVM-N ($\sigma_{N,1} < \sigma_{N,2} < \sigma_{N,3}$).

maximum likelihood method to find an elbow in the ordered sequence of eigenvalues. Elbow-finding methods are the most widely used class of methods to select the number of principal components, i.e., the intrinsic dimensionality of data.

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \geq 0$ be ordered eigenvalues. Given the value q $(1 \leq q < N)$, divide the eigenvalues into two sets, $\Lambda_1 = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$ and $\Lambda_2 = \{\lambda_{q+1}, \lambda_{q+2}, \ldots, \lambda_N\}$. The basic assumption of the PLM is that if an elbow exists at position q, then we can think of Λ_1 and Λ_2 as samples from two different Gaussian distributions. The profile log-likelihood can be written as

$$l(q) = \sum_{i=1}^{q} \log g(\lambda_i | \mu_1, \sigma_\lambda) + \sum_{i=q+1}^{N} \log g(\lambda_i | \mu_2, \sigma_\lambda), \tag{5–34}$$

where g is a Gaussian distribution function,

$$\mu_j = \frac{1}{|\Lambda_j|} \sum_{\lambda_i \in \Lambda_j} \lambda_i, \tag{5–35}$$

$$\sigma_j^2 = \frac{1}{|\Lambda_j| - 1} \sum_{\lambda_i \in \Lambda_j} (\lambda_i - \mu_j)^2, \tag{5–36}$$

$$\sigma_\lambda^2 = \frac{(q-1)\sigma_1^2 + (N-q-1)\sigma_2^2}{N-2}, \tag{5–37}$$

Algorithm 5-1 Fuzzy SVM for noisy data

Input: X : data set $(N \times d)$, Y : data label $(N \times 1)$

1: Divide data set X into two subsets according to their labels, $X_{\pm} = \{x_i | y_i = \pm 1\}$.
2: Do PCA on each subset and decide the number of principal components, k_+ and k_-, using Equation (5–38).
3: Calculate the memberships using Equations (5–28), (5–32), and (5–33).
4: Run the fuzzy SVM with the membership values.
5: **return**

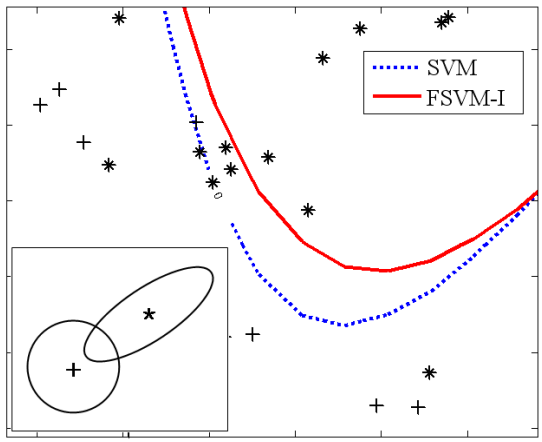


Figure 5-4. Decision boundaries of SVM and FSVM-I.

where $|\Lambda_j|$ is the cardinality of the set Λ_j. The optimal number of eigenvectors is given as

$$k = \arg\max_{1 \le q < N} I(q). \tag{5–38}$$

The overall procedure of FSVM-N can be summarized as Algorithm 5-1. One thing that should be noticed is that the membership values are decided in a supervised manner, i.e., positive and negative samples are treated separately and the membership values are decided using the eigenvectors of the corresponding class to consider the different structure of the data in each class.

5.4 Experimental results

To investigate the effectiveness of the proposed method, the previous methods and the proposed method were implemented and tested using Matlab. First of all, the defects of the previous methods are shown using artificial data sets.

Figure 5-4 shows the decision boundaries of SVM and FSVM-I. The data set in Figure 5-4 consists of two clusters, one circular cluster ("+") and one elliptical cluster ("*"). In the overlapped region, because the distance from the center of the elliptical cluster is larger than that from the center of the circular cluster, the boundary of FSVM-I is skewed toward the center of the elliptical cluster. This shift may make some points from the elliptical cluster become incorrectly classified. Due to this inability to deal with non-circular clusters, FSVM-I is not included in the following experiments. Instead, FSVM-F is included because it can deal with non-convex as well as non-circular clusters, although it also assumes that clusters are circular in the feature space. In FSVM-F, there is another free parameter (δ) that controls the minimum value of the memberships.

H-FSVM also assumes a circular distribution of the data in the feature space. Another problem in H-FSVM is the number of free parameters in Equation (5–21), which makes it difficult to obtain optimal parameter values. Therefore, used is a different rescaling function which can be optimized easily. The new rescaling function is defined as

$$u_i = \exp\left(-\frac{f''(x_i, y_i)}{\sigma_H^2}\right), \tag{5–39}$$

which is similar to Equation (5–32). The function $f''(x_i, y_i)$ is defined as

$$f''(x_i, y_i) = \max\left(0, \frac{f'(x_i, y_i) - \mu_{f'}}{\sigma_{f'}}\right), \tag{5–40}$$

where $\mu_{f'}$ and $\sigma_{f'}$ are mean and variance of $f'(x_i, y_i)$. The function $f'(x_i, y_i)$ is defined as

$$f'(x_i, y_i) = \left(\max_{j=1,\ldots,N} f(x_j, y_j) - f(x_i, y_i)\right). \tag{5–41}$$

Because $e(\phi(x_i))$ measures dissimilarity and $f(x_i, y_i)$ measures similarity, the function $f(x_i, y_i)$ should be inverted first. Using the same function, the measures of outlier-ness – the reconstruction error and the kernel target alignment – can be compared.

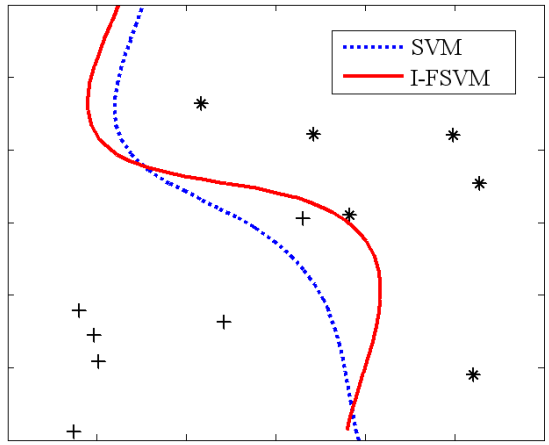


Figure 5-5. Decision boundaries of SVM and I-FSVM.

Figure 5-5 shows the differences between the boundaries of SVM and I-FSVM. In Figure 5-5, I-FSVM iterates until the membership vector converges. Although there is no error under the decision boundary of I-FSVM, it is more complex, i.e., over-fitted to training data and may not be good for generalization. As shown in [102], on average, the best result was obtained after seven iterations, but we did not fix the iteration number (n) and considered it as a free parameter.

Given the considerations discussed above, the four previous methods – SVM, FSVM-F, H-FSVM, and I-FSVM – and the proposed method, FSVM-N, were tested using data sets available from the UCI Machine Learning Repository [72]. The Gaussian kernel,

$$\kappa(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{\sigma^2}\right), \tag{5–42}$$

was used for all the methods. The kernel parameter σ and the regularization parameter C for SVM were set using the grid search method [107]. The kernel parameter used for SVM was also used for all other SVM variants, and the free parameter specific to each method together with the regularization parameter C were also optimized using the grid search method. Table 5-1 sums up the methods used in the experiments.

Table 5-2 summarizes the data sets used and the experimental results using ten-fold cross-validation. As is clear from Table 5-2, all the SVM variants outperformed

Table 5-1. SVM and its variants

	Free parameters	Limitations
SVM	σ, C	Noise sensitivity
FSVM-F	δ, C	Circular data distribution in the feature space
H-FSVM	δ_H, C	Circular data distribution in the feature space
I-FSVM	n, C	Overfitting
FSVM-N	δ_N, C	Gaussian data distribution in the feature space

Table 5-2. Error rates on UCI data sets (N = number of data points, d = data dimension)

Data set	N	d	Error rate (%)				
			SVM	FSVM-F	H-FSVM	I-FSVM	FSVM-N
Breast cancer	569	30	2.109	2.109	1.933	**1.757**	**1.757**
Ionosphere	351	34	5.413	5.128	5.128	5.413	**4.843**
Parkinsons	197	22	5.641	4.615	4.615	**3.590**	4.615
Pima Indians diabetes	768	8	22.396	**21.875**	22.135	22.266	22.005
Yeast	892	8	35.202	34.193	34.753	33.968	**33.857**

SVM for all data sets. The proposed method, FSVM-N, showed the best results for three out of five sets and was second best for the remaining two sets, i.e., FSVM-N is better than SVM and other SVM variants on average. I-FSVM comes next with two best results. I-FSVM, however, performed worst for *Ionosphere* and *Pima Indians diabetes* data sets among SVM variants. FSVM-F and H-FSVM were also better than SVM, but not as much as the other two SVM variants. The main drawback of the two methods is the assumption that the data points have a circular distribution. The other two SVM variants do not have that assumption. The last thing that should be noted is that the error rate of SVM variants might be lowered by adjusting the kernel parameter σ. We assumed that the optimal value of σ for SVM is also optimal for SVM variants to simplify the parameter search. However that assumption may not be true although SVM variants are better in classification than SVM.

5.5 Discussion

In this chapter, we proposed a modified fuzzy SVM, specifically a robust membership calculation method. Although there have been several approaches used to estimate memberships in robust ways, they have an assumption on the data distribution. The

proposed method, fuzzy SVM for noisy data (FSVM-N), is based on the reconstruction error, and the error can be decided based on the statistics of the data. Even more, FSVM-N does not assume circular distribution of the data and is complex enough to capture the data structure. Experimental results with synthetic and real data sets prove the usefulness of the proposed method.

Although the proposed method showed better results than SVM and its variants, there is still room for improvement. The first improvement would be the Gaussian assumption. FSVM-N achieved better performance by relaxing the assumption of the previous methods, therefore, removing the Gaussian assumption may improve the performance further. A non-parametric method that does not assume any data distribution is under investigation. Another improvement would be to the computational complexity of it. When the data set is large, kernel PCA used to calculate the reconstruction error is computationally expensive. Another measure of outlier-ness approximating the reconstruction error with low computational complexity is under investigation. Another topic left for further research is a method to deal with imbalanced data sets, which are quite common in emerging applications. Although SVM variant for imbalanced data is a special case of FSVM, not only FSVM-N, but existing methods cannot deal efficiently with highly imbalanced data. Only FSVM-N opened a way to that through the rescaling of the reconstruction error and supervised membership decision procedure.

CHAPTER 6
KERNEL-BASED CONTEXT-DEPENDENT FUSION

Context-dependent fusion (CDF) is a general framework for fusion that divides the feature space into some number of homogeneous regions, i.e., contexts, and develops different fusion processes for different contexts. CDF consists of two components, context extraction and decision fusion, as depicted in Figure 1-1. CDF uses feature space information in fusion and the information enables it to achieve better performance than previous fusion methods. Although CDF demonstrated promising results, it also had some problems and could be improved further. CDF assumes that each context corresponds to a convex cluster in the feature space. A variant of CDF using regularization, called context-dependent fusion with regularization (CDF-R), mitigated the assumption and achieved better performance than CDF, which suggests that each context might not be represented as a convex cluster in the feature space. Although CDF-R used a different distance function to soften the convex cluster assumption, the distance was still based on the Euclidean distance in the feature space. A non-linear feature mapping can remove the assumption and may improve the performance of CDF. Another problem is the linear combination method used in decision fusion. In each context, multiple algorithm outputs were combined into a cluster output using a linear combination method, which cannot accommodate non-linearly separable classes. The linear restriction can be resolved by using a non-linear combination method.

In this chapter, to solve the CDF problems, CDF is generalized as a robust non-linear method, called *kernel-based context-dependent fusion* (K-CDF) using the robust methods discussed in the previous chapters. As shown in Figure 6-1, K-CDF consists of three components: dimension reduction, context extraction, and decision fusion. Different from CDF, K-CDF requires one additional component for dimension

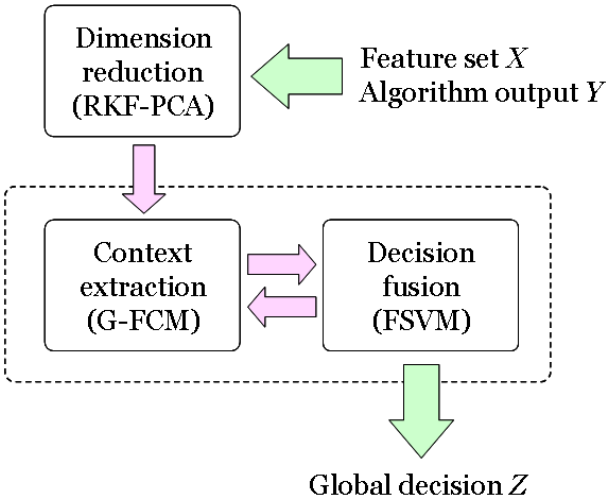


Figure 6-1. Block diagram of kernel-based context-dependent fusion.

reduction. As the dimensionality of the data in the kernel feature space[1] is generally very high and often assumed to be infinite, the feature-wise operation used in CDF is not applicable. Therefore, a dimension reduction method is introduced in K-CDF and robust kernel fuzzy principal component analysis (RKF-PCA) introduced in Chapter 3 is used for that. RKF-PCA is a robust non-linear variant of PCA, efficient in reducing feature dimension. By using RKF-PCA, K-CDF can overcome the convex cluster limitation.
To extract contexts, a modified fuzzy clustering, global fuzzy *c*-means (G-FCM), is adopted. As RKF-PCA maps the original feature in a non-linear way, clusters are no longer convex in the feature space, although clusters may be convex in the kernel feature space. Therefore G-FCM is adopted to overcome the sensitivity to initialization in FCM and its variants. Finally, a robust variant of support vector machine (SVM) using fuzzy memberships, called fuzzy support vector machine (FSVM), is used to fuse classifier outputs in a non-linear manner. Another strength of K-CDF is its noise

[1] The space where the original feature vector resides is generally called *feature space* and the high dimensional space where the mapped feature vector through a mapping function resides is also called *feature space*. Therefore, in this chapter, the latter will be referred to as *kernel feature space* to discriminate between the two spaces.

robustness. Although K-CDF has noise robustness due to the regularization conferred by CDF-R, it achieves more noise robustness by the use of noise-robust algorithms for each component. Noise robustness of the algorithms results from fuzzy memberships and global initialization.

In the next section, K-CDF is formulated by extending CDF-R. Experimental results using artificial and real data sets are given in section 6.2 and a discussion concludes this chapter.

6.1 Kernel-based context-dependent fusion

The objective function of K-CDF can be written as

$$\begin{aligned} J &= \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha \sum_{k=1}^{K}\left(\frac{1}{2}||\underline{w}_k||^2 + C\sum_{i=1}^{N} u_{ki}^{m}\xi_{ki}\right) \\ &\quad + \beta \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^{q}, \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} |v_{kl} - x_{il}|^2 + \alpha \sum_{k=1}^{K}\left(\frac{1}{2}||\underline{w}_k||^2 + C\sum_{i=1}^{N} u_{ki}^{m}\xi_{ki}\right) \\ &\quad + \beta \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^{q}, \end{aligned} \tag{6–1}$$

where N is the number of data points, K is the number of clusters, and L is the feature dimension in the kernel feature space. The value u_{ki} is the membership of x_i to the kth cluster and r_{kl} is the weight of the lth feature in the kth cluster. The constants m and q represent the fuzzifier constants for membership and feature weight, respectively. The value $d_{kil} = |v_{kl} - x_{il}|$ is a feature-wise distance between the lth feature of data point x_i and the lth feature of the kth cluster center v_k. The second term is the sum of K fuzzy support vector machine (FSVM) objective functions, each of which corresponds to a fusion process for each cluster. All the FSVMs share the regularization constant C. The value $\underline{w}_k$ represents the decision boundary in the kth FSVM and ξ_{ki} is an error of x_i in the kth FSVM. The remaining two terms are regularization terms defined for memberships and feature weights, respectively. The value α is a constant balancing

between clustering and fusion, and the remaining two constants – β and γ – are regularization constants.

The objective function should be minimized under the constraints on membership U and feature weight R. The constraints can be represented as

$$U_{K\times N} = \left\{ u_{ki} \middle| u_{ki} \in [0,1] \ \forall i \text{ and } k; \sum_{k=1}^{K} u_{ki} = 1 \ \forall i \right\}, \tag{6–2}$$

$$R_{K\times L} = \left\{ r_{kl} \middle| r_{kl} \in (0,1] \ \forall k \text{ and } l; \sum_{l=1}^{L} r_{kl} = 1 \ \forall k \right\}, \tag{6–3}$$

which are the same as those in CDF and CDF-R. The objective function of K-CDF is somewhat different from that of CDF, which can be written as

$$J_{CDF} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L'} r_{kl}^{q} d_{kil}^{2} + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i \right)^2, \tag{6–4}$$

where L' is the feature dimension in the original feature space, w_{kt} is a weight of the tth classifier in the kth cluster, y_{ti} is a output value from the tth classifier for x_i, and o_i is a desired output for x_i. The main difference lies in the second term of Equations (6–1) and (6–4). Different from CDF, in K-CDF, there is no explicit mention of the algorithm output (y_{ti}) or target value (o_i). As K-CDF uses FSVM for fusion, algorithm outputs and target values are given as inputs to FSVM. Instead of minimizing the difference between fused outputs and target values, therefore, K-CDF optimizes the margin (indirectly represented by $\underline{w}_k$, $1 \le k \le K$) and errors (ξ_{ki}, $1 \le k \le K$, $1 \le i \le N$). There are K FSVMs in K-CDF, each of which corresponds to a fusion process for a cluster and uses different memberships $\{u_{ki} | i = 1, \cdots, N\}$.

Alternating optimization can be used to optimize Equation (6–1) and the update equations can be written as

$$u_{ki} = \frac{\left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K}\left(\frac{1}{\widetilde{D}_{ai}}\right)^{\frac{1}{m-1}}}, \tag{6–5}$$

$$v_{kl} = \frac{\sum_{i=1}^{N} u_{ki}^{m} x_{il}}{\sum_{i=1}^{N} u_{ki}^{m}}, \tag{6–6}$$

$$r_{kl} = \frac{\left(\frac{1}{\overline{D}_{kl}}\right)^{\frac{1}{q-1}}}{\sum_{a=1}^{L}\left(\frac{1}{\overline{D}_{ka}}\right)^{\frac{1}{q-1}}}, \tag{6–7}$$

where

$$\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha C \xi_{ki} + \beta, \tag{6–8}$$

$$\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^{m} d_{kil}^{2} + \gamma. \tag{6–9}$$

The detailed derivation can be found in Appendix C. The optimal values of the remaining parameters specific to the kth cluster, $\{\underline{w}_k, \xi_{ki} | i = 1, \cdots, N\}$, can be obtained through the optimization of FSVM. The FSVM optimization procedure is a convex optimization guaranteeing that the solution found is a global optimum. The update equations in Equations (6–5), (6–6), and (6–7) are essentially the same as those of CDF and CDF-R. The difference lies in the distance calculation methods. The distances in Equations (6–8) and (6–9) also look similar to those of CDF and CDF-R, but, in K-CDF, d_{kil} is calculated in the kernel feature space, not in the feature space. Table 6-1 compares the update equations of CDF-R and K-CDF[2] .

The last thing that should be mentioned is the mapping of output values from FSVM. As an output value from FSVM is not bounded, it should be mapped into a specific range, typically $[0, 1]$, to be considered as a probability. Although there have been several approaches to the conversion of the output to a posterior probability [109],

[2] In Table 6-1, the update equation for classifier weights in CDF-R is not shown. Refer to Table 2-1 for complete update equations of CDF and CDF-R.

Table 6-1. Update equations of CDF-R and K-CDF

Context-dependent fusion with regularization	Kernel-based context-dependent fusion
$J_{CDF-R} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i\right)^2 + \beta \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^q + \delta \sum_{k=1}^{K}\sum_{t=1}^{T} w_{kt}^2$	$J_{K-CDF} = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha \sum_{k=1}^{K} \left(\frac{1}{2}\|\|w_k\|\|^2 + C \sum_{i=1}^{N} u_{ki}^m \xi_{ki}\right) + \beta \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^m + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^q$
$u_{ki} = \frac{\left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K}\left(\frac{1}{\widetilde{D}_{ai}}\right)^{\frac{1}{m-1}}}$ where $\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha\left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i\right)^2 + \beta$	$u_{ki} = \frac{\left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K}\left(\frac{1}{\widetilde{D}_{ai}}\right)^{\frac{1}{m-1}}}$ where $\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^q d_{kil}^2 + \alpha C \xi_{ki} + \beta$
$v_{kl} = \frac{\sum_{i=1}^{N} u_{ki}^m x_{il}}{\sum_{i=1}^{N} u_{ki}^m}$	$v_{kl} = \frac{\sum_{i=1}^{N} u_{ki}^m x_{il}}{\sum_{i=1}^{N} u_{ki}^m}$
$r_{kl} = \frac{\left(\frac{1}{\overline{D}_{kl}}\right)^{\frac{1}{q-1}}}{\sum_{a=1}^{L}\left(\frac{1}{\overline{D}_{ka}}\right)^{\frac{1}{q-1}}}$ where $\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^m d_{kil}^2 + \gamma$	$r_{kl} = \frac{\left(\frac{1}{\overline{D}_{kl}}\right)^{\frac{1}{q-1}}}{\sum_{a=1}^{L}\left(\frac{1}{\overline{D}_{ka}}\right)^{\frac{1}{q-1}}}$ where $\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^m d_{kil}^2 + \gamma$

Algorithm 6-1 Kernel-based context-dependent fusion

Input: X : data set $(N \times L')$, Y : classifier output $(N \times T)$, O : target value $(N \times 1)$

1: Extract L dimensional non-linear feature vectors using RKF-PCA.
2: Initialize U using global fuzzy c-means.
3: Initialize R to $1/L$.
4: **repeat**
5: Update cluster centers V using (6–6).
6: Update feature weights R using (6–7).
7: Update memberships U using (6–5).
8: Update SVM parameters using the optimization procedures of FSVM.
9: **until** V, R, U, and SVM parameters satisfy the convergence criteria
10: **return**

[110], in K-CDF, a simple sigmoid function was used. The mapping function is defined as

$$y'' = \frac{1}{1 + e^{-\rho y'}}, \tag{6–10}$$

where $y' = \{y'_{ki} | 1 \leq k \leq K, 1 \leq i \leq N\}$ is FSVM output and ρ is a constant. The value of ρ is decided experimentally. Algorithm 6-1 summarizes the training phase of the K-CDF algorithm.

In the test phase, when a data point x is given, it is first mapped into an L dimensional feature x' using RKF-PCA and algorithm outputs $y = [y_1, \cdots, y_T]$ are calculated. For a test point, as the target value is not known, the error ξ_{k*} cannot be calculated, so Equation (6–8) cannot be evaluated. Therefore, the membership of x is calculated using Equation (6–5) with the following distance function:

$$\widetilde{D}_{k*} = \sum_{l=1}^{L} r_{kl}^{q} d_{k*l}^{2} + \beta. \tag{6–11}$$

The K FSVMs take algorithm outputs as inputs and generate SVM outputs that are converted into cluster output using Equation (6–10). The final output is given as a membership-weighted sum of the K cluster outputs:

$$z(x) = \sum_{k=1}^{K} u_k^m y_k''(x). \tag{6–12}$$

Figure 6-2 shows the procedure generating final output from input point x. In Figure 6-2,

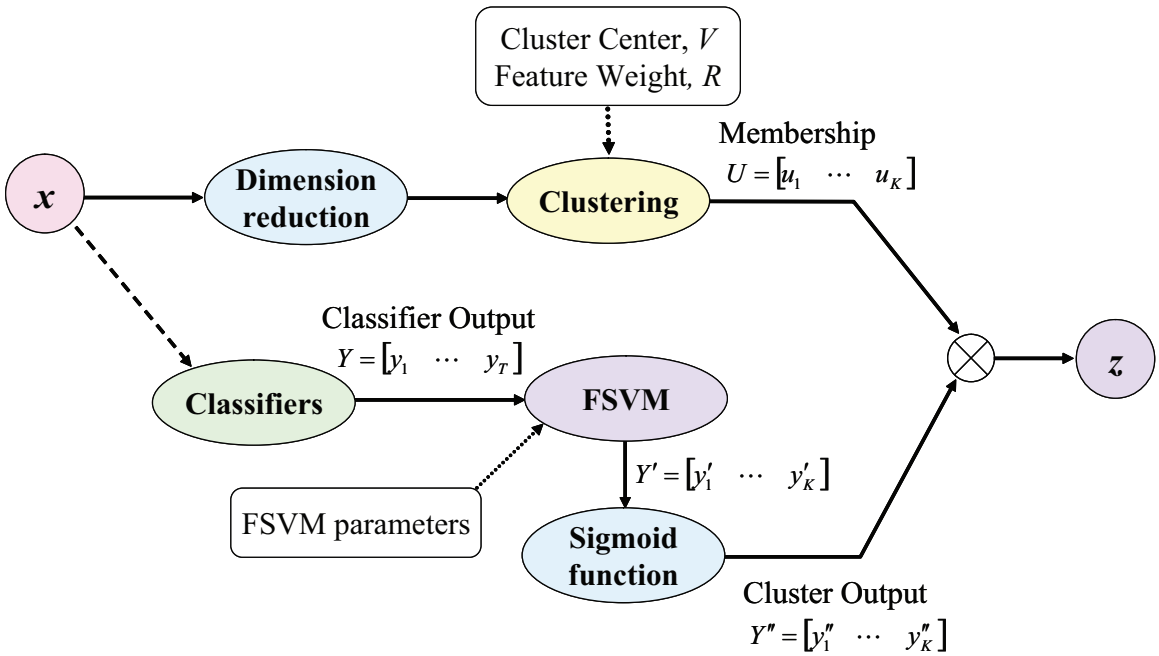


Figure 6-2. Test phase of kernel-based context-dependent fusion.

the matrices V and R and the FSVM parameters are optimized using Algorithm 6-1.

6.2 Experimental results

To investigate the effectiveness of the proposed method, kernel-based context dependent fusion (K-CDF) was first applied to a two dimensional artificial data set shown in Figure 6-3. The data set consists of 400 data points from two classes and naturally forms two clusters. The first cluster is linearly separable, but the second one is not. The confidence value of the first classifier takes the first feature value after normalization. Simple linear rescaling was performed cluster-wise for the confidence values to have values between zero and one. The confidence value of the second classifier was set using the second feature value in a similar way. Figure 6-4 shows the receiver operating characteristic (ROC) curves of the two classifiers and the fused results using CDF with regularization (CDF-R) and K-CDF. As the two clusters are well separated, CDF-R and K-CDF clustered the data, as shown in Figure 6-3, in the feature

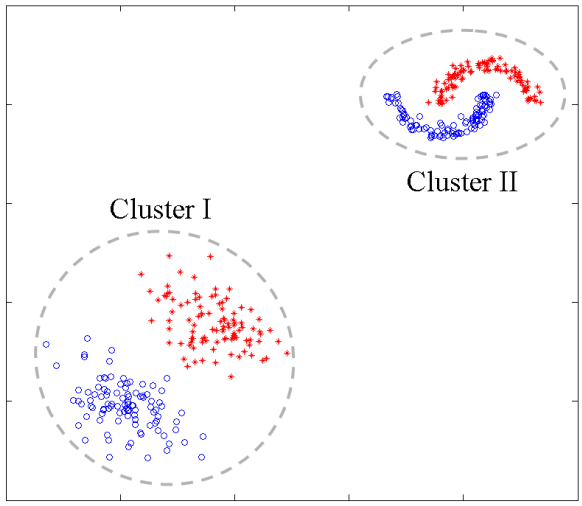


Figure 6-3. Artificial data set.

space and kernel feature space.[3] CDF-R, however, failed to classify the data correctly in cluster II, the cluster that is not linearly separable. K-CDF also made occasional errors due to the random data generation, but K-CDF was much better than CDF-R, sometimes achieving a 100% classification rate for the artificial data set, because SVM can effectively accommodate non-linearly separable classes.

There are several important parameters in K-CDF, including kernel parameters for kernel PCA (σ_{PCA}) and fuzzy SVM (σ_{FSVM}), and a regularization parameter for fuzzy SVM (C). The initial value of σ_{PCA} was set as the value generating minimum classification error using the grid search method [107]. Classification error was calculated using the Fisher linear discriminant as described in Chapter 3. The number of eigenvectors used in reconstruction was set as the minimum number of eigenvectors with which at least 90% of the data variance is described with the given kernel parameter σ_{PCA}. The initial values of σ_{FSVM} and C were also set as the values generating minimum classification error with SVM. Starting from the initial values

[3] CDF was also tested, but the results were virtually the same as or a little worse than those of CDF-R. Therefore, in this chapter, the results of CDF are not presented.

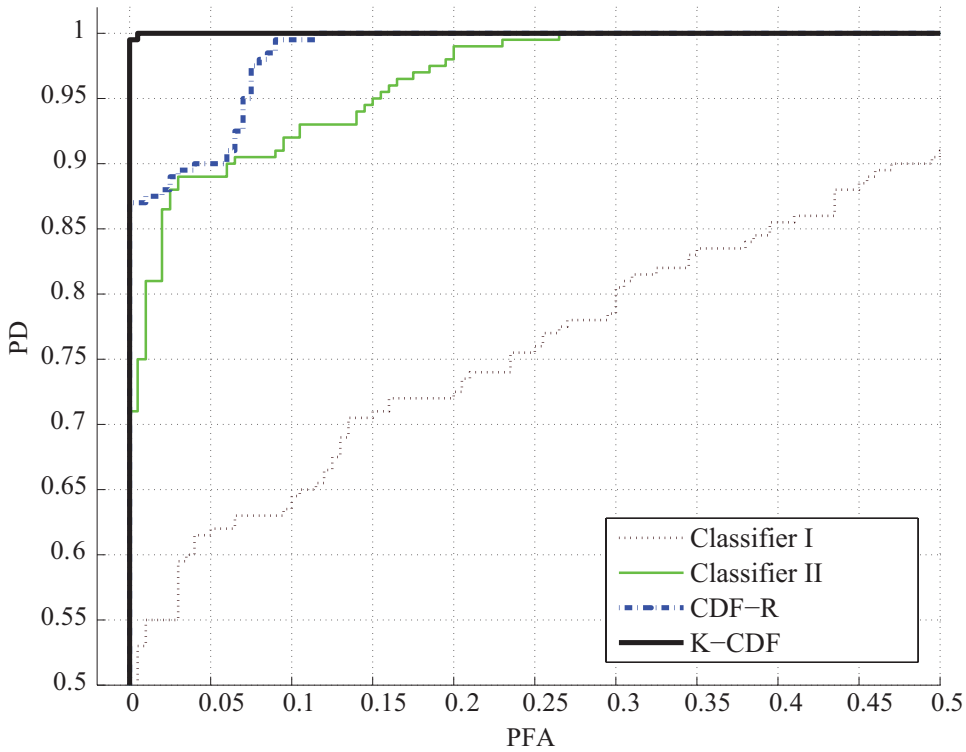


Figure 6-4. ROC curves from an artificial data set.

decided above, the values were adjusted experimentally to obtain the best ROC curve. Although there are several available methods to compare ROC curves [111]–[113], the probability of false alarm (PFA) at 85%, 90%, and 95% of probability of detection (PD) was the first to be minimized in this chapter. The remaining parameters were set equal to those in CDF-R.

K-CDF was also applied to landmine detection problems. When K-CDF was applied to the data used in Chapter 2, it showed almost the same results as CDF-R because all the classifier systems used for fusion were sophisticated enough to generate linear decision boundary in each cluster. Although K-CDF showed somewhat better results than CDF-R, the differences were not statistically significant. As demonstrated in the previous experiment, K-CDF is much better than CDF-R when a linear decision boundary does not work. By randomly sampling with data set II in Chapter 2, we made up another data set, which consisted of 292 samples, with 113 mine samples and

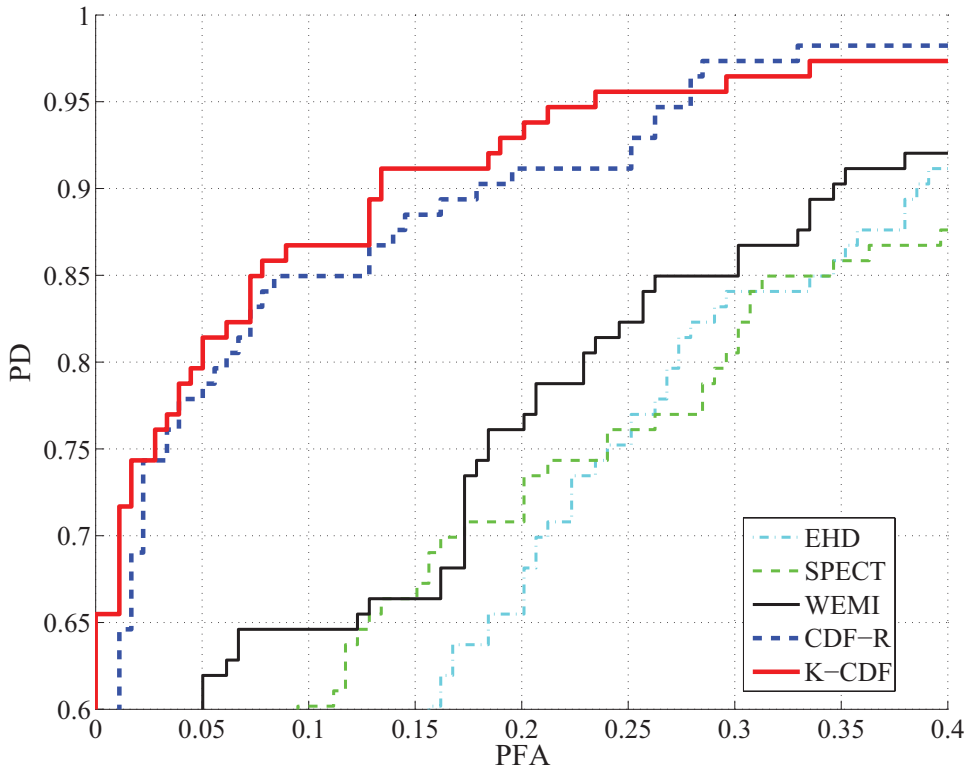


Figure 6-5. ROC curves for a set of classifiers from the subset of data set II.

Table 6-2. Experimental results from the subset of data set II

PD (%)	PFA (%)				
	EHD	SPECT	WEMI	CDF-R	K-CDF
85	34.64	34.64	30.17	12.85	7.82
90	38.55	41.90	34.64	17.88	13.41
95	58.10	50.84	41.34	27.93	23.46

179 non-mine samples. Figure 6-5 shows the ROC curves and Table 6-2 summarizes the results. As is clear from Figure 6-5, K-CDF showed better results than CDF-R, especially at 85%, 90%, and 95% PD. Although it looks as if K-CDF and CDF-R produced almost the same results, the difference between them are statistically significant. Table 6-3 shows the confidence intervals (CIs) at the 90% confidence level (CL) using Equation (3–37). There is no overlap between the CIs of CDF-R and K-CDF, which means that the differences are statistically significant and K-CDF is better than CDF-R at each PD.

Table 6-3. Confidence interval of the number of false alarms with a 90% confidence level

PD (%)	CI	
	CDF-R	K-CDF
85	[22.45 23.55]	[13.56 14.44]
90	[31.37 32.64]	[23.44 24.56]
95	[49.26 50.73]	[41.30 42.69]

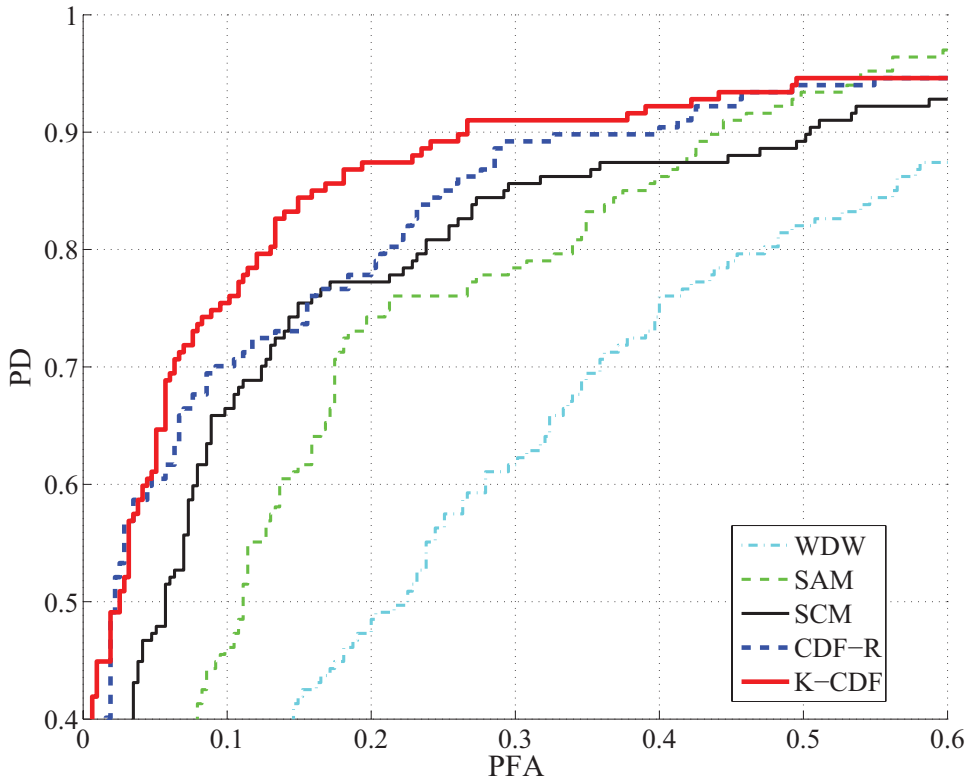


Figure 6-6. ROC curves for a set of classifiers from hyperspectral data.

Another data set was hyperspectral imagery collected from an airborne hyperspectral imager [114]. The data set consisted of 482 samples, with 167 mine samples and 315 non-mine samples, each of which had 70 spectral bands after trimming and binning. Three classifier systems called WDW (whiten and de-whiten) [115], SAM (spectral angle mapper) [116], and SCM (spectral correlation mapper) [117] were used to generate confidence values. Figure 6-6 shows the ROC curves and Table 6-4 summarizes the results. As is clear from Figure 6-6, K-CDF is better than CDF-R or any single classifier at 85% and 90% PD. SAM is better than K-CDF at 95% PD, but K-CDF is still better than

Table 6-4. Experimental results from hyperspectral data set

PD (%)	PFA (%)				
	WDW	SAM	SCM	CDF-R	K-CDF
85	56.19	37.46	29.21	24.76	15.87
90	61.59	44.44	50.48	40.00	26.67
95	73.97	53.97	100.00	64.13	60.63

Table 6-5. Confidence interval of the number of false alarms with a 90% confidence level

PD (%)	CI	
	CDF-R	K-CDF
85	[77.28 78.70]	[49.39 50.59]
90	[125.19 126.81]	[83.28 84.74]
95	[201.22 202.80]	[190.18 191.79]

CDF-R. Table 6-5 shows the CIs of the number of false alarms with a 90% CL, which also supports the superiority of K-CDF to CDF-R.

From the above experiments using artificial and landmine data sets, K-CDF showed better results than CDF-R. The strength of K-CDF comes from its non-linear and robust characteristics. The non-linearity comes from the kernel methods adopted in K-CDF and the robustness mainly originates from regularization and fuzzy memberships. K-CDF requires considerable time to be trained due to the repeated training of FSVM, which also makes it difficult to tune the parameter values in K-CDF. K-CDF is, however, a good choice for a noisy real world problem due to the strengthes mentioned above.

6.3 Discussion

In this chapter, the context-dependent fusion (CDF) framework introduced in Chapter 2 is extended to address and solve solve its limitations. CDF has two limitations: the clusters are convex in the feature space and the classes in each cluster are linearly separable. The extended framework, K-CDF, uses kernel PCA to remove the convex cluster assumption and FSVM to separate non-linearly separable classes. In addition, regularization and the robust methods adopted for each component in K-CDF make it possible to achieve even better performance than previous methods.

Experimental results with artificial and real data sets proved the superiority of K-CDF to CDF or CDF-R.

Although K-CDF was an effective framework for fusion, it can be improved further. The first improvement concerns time complexity. Because K-CDF uses SVM as a fusion method, it requires considerable time for optimization. A simple non-linear classifier might significantly reduce the complexity without sacrificing the performance of K-CDF. This is left for further research. Another problem is the decision of parameter values. K-CDF requires two kernel parameters for kernel PCA and fuzzy SVM, and another regularization parameter for fuzzy SVM. Although the parameter initialization method used in this chapter showed reasonable results, it is time-consuming and may not be optimal. It might be possible to decide the parameter values based on data, which is an open problem to be studied further.

CHAPTER 7
CONCLUSIONS

Combining several classifiers has become an important task in recent years to overcome the limitations of a single classifier. In this dissertation, a generalization of context-dependent fusion (CDF) using robust non-linear methods was investigated.

CDF is a generalization of the previous fusion framework executed by dividing the feature space into a number of homogeneous regions and developing a different fusion process for each of the regions. CDF, however, does not explicitly address the noise sensitivity problem and it cannot accommodate non-linear cases. Regularization was adopted to enhance the noise robustness, which resulted in a robust variant of CDF, called context-dependent fusion with regularization (CDF-R). CDF-R was further extended using robust kernel methods, named kernel-based context-dependent fusion (K-CDF). K-CDF has three main components – dimension reduction, context extraction, and decision fusion – and robust kernel methods were developed for each component. Specifically, robust kernel fuzzy principal component analysis (RKF-PCA), kernel-based global fuzzy *c*-means (KG-FCM), and fuzzy support vector machine for noisy data (FSVM-N) were investigated.

RKF-PCA was used as a dimension reduction method in K-CDF because it overcomes the noise sensitivity and linear restriction problems in PCA, by the introduction of fuzzy memberships and kernel methods, respectively. Iterative optimization enables RKF-PCA to generate more noise-resistant results than previous methods. Global fuzzy c-means was used as a context extraction method due to the deterministic and noise-resistant characteristics of global clustering. Initialization is one of the major difficulties of clustering algorithms and the global method can effectively mitigate the problems in performance and speed. For the decision fusion, a variant of support vector machine using fuzzy memberships was used. As the clustering algorithm is included as

a context extraction method in K-CDF, the memberships can be directly calculated by optimizing the objective function of K-CDF and they can be used in fuzzy SVM.

By combining the three fuzzy kernel methods, K-CDF can accommodate non-linear cases in a noise resistant way, which is the main result of this dissertation. Another contribution is that the three robust kernel methods developed can be used in other pattern recognition problems, as demonstrated in Chapters 3, 4, and 5.

APPENDIX A
UPDATE EQUATIONS FOR CONTEXT-DEPENDENT FUSION WITH REGULARIZATION

The objective function for *context-dependent fusion with regularization* can be written as

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i \right)^2 + \beta \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^{q} + \delta \sum_{k=1}^{K}\sum_{t=1}^{T} w_{kt}^{2}, \tag{A–1}$$

where N is the number of data points, K is the number of clusters, L is the dimension of data, and T is the number of algorithms. The value u_{ki} is the membership of x_i to the kth cluster, r_{kl} is the weight of the lth feature in the kth cluster, and w_{kt} is the weight of the tth algorithm in the kth cluster. The constants m and q represent the fuzzifier constants for membership and feature weight respectively. For a data point x_i, the output value y_{ti} is generated from the tth algorithm and the weighted sum of the algorithm outputs is compared to the desired output o_i. The value $d_{kil} = |v_{kl} - x_{il}|$ is a feature-wise distance which measures the difference between the lth feature of a data point x_i and the lth feature of the kth cluster center v_k. The value α is a constant balancing between clustering and fusion. The remaining three constants – β, γ, and δ – are regularization constants. The objective function should be minimized under the constraints on membership U, feature weight R, and classifier weight W. The constraints can be represented as

$$U_{K\times N} = \left\{ u_{ki} \,\middle|\, u_{ki} \in [0,1] \ \forall i \text{ and } k; \sum_{k=1}^{K} u_{ki} = 1 \ \forall i \right\}, \tag{A–2}$$

$$R_{K\times L} = \left\{ r_{kl} \,\middle|\, r_{kl} \in (0,1] \ \forall k \text{ and } l; \sum_{l=1}^{L} r_{kl} = 1 \ \forall k \right\}, \tag{A–3}$$

$$W_{K\times T} = \left\{ w_{kt} \,\middle|\, w_{kt} \in \mathcal{R} \ \forall k \text{ and } t; \sum_{t=1}^{T} w_{kt} = 1 \ \forall k \right\}. \tag{A–4}$$

From the objective function in Equation (A–1) and the constraints, the Lagrange equation can be written as

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m}\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m}\left(\sum_{t=1}^{T} w_{kt}y_{ti} - o_i\right)^2 \\ &+ \beta\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} + \gamma\sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^{q} + \delta\sum_{k=1}^{K}\sum_{t=1}^{T} w_{kt}^{2} \\ &- \sum_{i=1}^{N}\lambda_{1,i}\left(\sum_{k=1}^{K} u_{ki} - 1\right) - \sum_{k=1}^{K}\lambda_{2,k}\left(\sum_{l=1}^{L} r_{kl} - 1\right) - \sum_{k=1}^{K}\lambda_{3,k}\left(\sum_{t=1}^{T} w_{kt} - 1\right),\end{aligned} \tag{A–5}$$

where $\lambda_{1,i}$, $\lambda_{2,k}$, and $\lambda_{3,k}$ are Lagrange multipliers. By collecting all the terms related to u_{ki} from the Lagrange equation, one can obtain

$$\begin{aligned}\mathcal{L}_1 = &\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m}\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m}\left(\sum_{t=1}^{T} w_{kt}y_{ti} - o_i\right)^2 \\ &+ \beta\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} - \sum_{i=1}^{N}\lambda_{1,i}\left(\sum_{k=1}^{K} u_{ki} - 1\right).\end{aligned} \tag{A–6}$$

By taking a partial derivative with respect to u_{ki} and equating to zero, one can obtain

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial u_{ki}} &= m u_{ki}^{m-1}\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha m u_{ki}^{m-1}\left(\sum_{t=1}^{T} w_{kt}y_{ti} - o_i\right)^2 + \beta m u_{ki}^{m-1} - \lambda_{1,i} \\ &= m u_{ki}^{m-1}\left(\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha\left(\sum_{t=1}^{T} w_{kt}y_{ti} - o_i\right)^2 + \beta\right) - \lambda_{1,i} = 0.\end{aligned} \tag{A–7}$$

Let $\widetilde{D}_{ki}$ be a new distance measure defined as

$$\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha\left(\sum_{t=1}^{T} w_{kt}y_{ti} - o_i\right)^2 + \beta, \tag{A–8}$$

then Equation (A–7) can be re-written as

$$m u_{ki}^{m-1}\widetilde{D}_{ki} = \lambda_{1,i}, \tag{A–9a}$$

$$u_{ki}^{m-1} = \frac{\lambda_{1,i}/m}{\widetilde{D}_{ki}} = \frac{\lambda'_{1,i}}{\widetilde{D}_{ki}}, \tag{A–9b}$$

and the membership value u_{ki} can be calculated as

$$u_{ki} = \left(\frac{\lambda'_{1,i}}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}. \tag{A–10}$$

The membership should satisfy the constraint in Equation (A–2). Using the constraint, the Lagrange multiplier $\lambda'_{1,i}$ can be calculated as

$$\sum_{k=1}^{K} u_{ki} = \sum_{k=1}^{K} \left(\frac{\lambda'_{1,i}}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}} = {\lambda'_{1,i}}^{\frac{1}{m-1}} \sum_{k=1}^{K} \left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}} = 1, \tag{A–11a}$$

$${\lambda'_{1,i}}^{\frac{1}{m-1}} = \frac{1}{\sum_{k=1}^{K} \left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}}. \tag{A–11b}$$

Using Equations (A–10) and (A–11b), the update equation for the membership u_{ki} can be written as

$$u_{ki} = \frac{\left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K} \left(\frac{1}{\widetilde{D}_{ai}}\right)^{\frac{1}{m-1}}}. \tag{A–12}$$

By collecting all the terms related to v_{kl} from the Lagrange equation, one can obtain

$$\mathcal{L}_2 = \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ki}^m \sum_{l=1}^{L} r_{kl}^q |x_{il} - v_{kl}|^2. \tag{A–13}$$

By taking a partial derivative with respect to v_{kl} and equating to zero, one can obtain

$$\frac{\partial \mathcal{L}_2}{\partial v_{kl}} = -2 \sum_{i=1}^{N} u_{ki}^m r_{kl}^q (x_{il} - v_{kl}) = 0. \tag{A–14}$$

Because the feature weight $r_{kl} > 0$, i.e., all the features are assumed to have effects on the distance calculation, Equation (A–14) can be re-written as

$$r_{kl}^q v_{kl} \sum_{i=1}^{N} u_{ki}^m = r_{kl}^q \sum_{i=1}^{N} u_{ki}^m x_{il}, \tag{A–15a}$$

$$v_{kl} \sum_{i=1}^{N} u_{ki}^m = \sum_{i=1}^{N} u_{ki}^m x_{il}, \tag{A–15b}$$

and the update equation of the cluster center v_{kl} can be written as

$$v_{kl} = \frac{\sum_{i=1}^{N} u_{ki}^{m} x_{il}}{\sum_{i=1}^{N} u_{ki}^{m}}. \tag{A–16}$$

By collecting all the terms related to r_{kl} from the Lagrange equation, one can obtain

$$\mathcal{L}_3 = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^{q} - \sum_{k=1}^{K} \lambda_{2,k} \left(\sum_{l=1}^{L} r_{kl} - 1 \right). \tag{A–17}$$

By taking a partial derivative with respect to r_{kl} and equating to zero, one can obtain

$$\begin{aligned} \frac{\partial \mathcal{L}_3}{\partial r_{kl}} &= \sum_{i=1}^{N} u_{ki}^{m} q r_{kl}^{q-1} d_{kil}^{2} + \gamma q r_{kl}^{q-1} - \lambda_{2,k} \\ &= q r_{kl}^{q-1} \left(\sum_{i=1}^{N} u_{ki}^{m} d_{kil}^{2} + \gamma \right) - \lambda_{2,k} = 0. \end{aligned} \tag{A–18}$$

Let $\overline{D}_{kl}$ be another distance measure defined in a similar way to Equation (A–8) as

$$\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^{m} d_{kil}^{2} + \gamma, \tag{A–19}$$

then Equation (A–18) can be re-written as

$$q r_{kl}^{q-1} \overline{D}_{kl} = \lambda_{2,k}, \tag{A–20a}$$

$$r_{kl}^{q-1} = \frac{\lambda_{2,k}/q}{\overline{D}_{kl}} = \frac{\lambda'_{2,k}}{\overline{D}_{kl}}, \tag{A–20b}$$

and the feature weight r_{kl} can be calculated as

$$r_{kl} = \left(\frac{\lambda'_{2,k}}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}}. \tag{A–21}$$

The feature weight r_{kl} should satisfy the constraint in Equation (A–3). Using the constraint, the Lagrange multiplier $\lambda'_{2,k}$ can be calculated as

$$\sum_{l=1}^{L} r_{kl} = \sum_{l=1}^{L} \left(\frac{\lambda'_{2,k}}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}} = {\lambda'_{2,k}}^{\frac{1}{q-1}} \sum_{l=1}^{L} \left(\frac{1}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}} = 1, \tag{A–22a}$$

$${\lambda'_{2,k}}^{\frac{1}{q-1}} = \frac{1}{\sum_{l=1}^{L} \left(\frac{1}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}}}. \tag{A–22b}$$

Using Equations (A–21) and (A–22b), the update equation for the feature weight r_{kl} can be written as

$$r_{kl} = \frac{\left(\frac{1}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}}}{\sum_{a=1}^{L} \left(\frac{1}{\overline{D}_{ka}} \right)^{\frac{1}{q-1}}}. \tag{A–23}$$

By collecting all the terms related to w_{km} from the Lagrange equation, one can obtain

$$\mathcal{L}_4 = \alpha \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ki}^m \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i \right)^2 + \delta \sum_{k=1}^{K} \sum_{t=1}^{T} w_{kt}^2 - \sum_{k=1}^{K} \lambda_{3,k} \left(\sum_{t=1}^{T} w_{kt} - 1 \right). \tag{A–24}$$

By taking a partial derivative with respect to w_{ka} and equating to zero, one can obtain

$$\begin{aligned}
\frac{\partial \mathcal{L}_4}{\partial w_{ia}} &= 2\alpha \sum_{i=1}^{N} u_{ki}^m \left(\sum_{t=1}^{T} w_{kt} y_{ti} - o_i \right) y_{ai} + 2\delta w_{ka} - \lambda_{3,k} \\
&= 2\alpha \sum_{i=1}^{N} u_{ki}^m \left(\sum_{\substack{t=1 \\ t \neq a}}^{T} w_{kt} y_{ti} - o_i + w_{ka} y_{ai} \right) y_{ai} + 2\delta w_{ka} - \lambda_{3,k} \\
&= 2\alpha \sum_{i=1}^{N} u_{ki}^m \left(\sum_{\substack{t=1 \\ t \neq a}}^{T} w_{kt} y_{ti} - o_i \right) y_{ai} + 2\alpha \sum_{i=1}^{N} u_{ki}^m w_{ka} y_{ai}^2 + 2\delta w_{ka} - \lambda_{3,k} \\
&= 2\alpha \sum_{i=1}^{N} u_{ki}^m \left(\sum_{\substack{t=1 \\ t \neq a}}^{T} w_{kt} y_{ti} - o_i \right) y_{ai} + 2 w_{ka} \left(\alpha \sum_{i=1}^{N} u_{ki}^m y_{ai}^2 + \delta \right) - \lambda_{3,k} \\
&= 0.
\end{aligned} \tag{A–25}$$

Equation (A–25) can be further simplified as

$$
\begin{aligned}
2w_{ka}\left(\alpha\sum_{i=1}^{N}u_{ki}^{m}y_{ai}^{2}+\delta\right) &= 2\alpha w_{ka}\left(\sum_{i=1}^{N}u_{ki}^{m}y_{ai}^{2}+\frac{\delta}{\alpha}\right) \\
&= 2\alpha\sum_{i=1}^{N}u_{ki}^{m}\left(o_i-\sum_{\substack{t=1\\t\neq a}}^{T}w_{kt}y_{ti}\right)y_{ai}+\lambda_{3,k},
\end{aligned}
\tag{A–26a}
$$

$$
\begin{aligned}
w_{ka}\left(\sum_{i=1}^{N}u_{ki}^{m}y_{ai}^{2}+\frac{\delta}{\alpha}\right) &= \sum_{i=1}^{N}u_{ki}^{m}\left(o_i-\sum_{\substack{t=1\\t\neq a}}^{T}w_{kt}y_{ti}\right)y_{ai}+\frac{\lambda_{3,k}}{2\alpha} \\
&= \sum_{i=1}^{N}u_{ki}^{m}\left(o_i-\sum_{\substack{t=1\\t\neq a}}^{T}w_{kt}y_{ti}\right)y_{ai}+\lambda'_{3,k},
\end{aligned}
\tag{A–26b}
$$

and the update equation for the classifier weight w_{ka} can be written as

$$
w_{ka}=\frac{\sum_{i=1}^{N}u_{ki}^{m}\left(o_i-\sum_{\substack{t=1\\t\neq a}}^{T}w_{kt}y_{ti}\right)y_{ai}+\lambda'_{3,k}}{\sum_{i=1}^{N}u_{ki}^{m}y_{ai}^{2}+\frac{\delta}{\alpha}}. \tag{A–27}
$$

The algorithm weight should satisfy the constraint in Equation (A–4). Using Equation (A–27), the constraint can be re-written as

$$
\begin{aligned}
\sum_{a=1}^{T} w_{ka} &= \sum_{a=1}^{T} \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{\substack{t=1 \\ t \neq a}}^{T} w_{kt} y_{ti} \right) y_{ai} + \lambda'_{3,k}}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}} \\
&= \sum_{a=1}^{T} \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti} + w_{ka} y_{ai} \right) y_{ai} + \lambda'_{3,k}}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}} \\
&= \sum_{a=1}^{T} \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti} \right) y_{ai} + w_{ka} \sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \lambda'_{3,k}}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}} \\
&= \sum_{a=1}^{T} \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti} \right) y_{ai} + w_{ka} \left(\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha} \right) - w_{ka} \frac{\delta}{\alpha} + \lambda'_{3,k}}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}} \\
&= \sum_{a=1}^{T} \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti} \right) y_{ai} - w_{ka} \frac{\delta}{\alpha}}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}} + \sum_{a=1}^{T} w_{ka} + \lambda'_{3,k} \sum_{a=1}^{T} \frac{1}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}} \\
&= \sum_{a=1}^{T} \frac{\sum_{i=1}^{N} u_{ki}^{m} \left(o_i - \sum_{t=1}^{T} w_{kt} y_{ti} \right) y_{ai} - w_{ka} \frac{\delta}{\alpha}}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}} + 1 + \lambda'_{3,k} \sum_{a=1}^{T} \frac{1}{\sum_{i=1}^{N} u_{ki}^{m} y_{ai}^{2} + \frac{\delta}{\alpha}} \\
&= 1,
\end{aligned}
\tag{A–28}
$$

and the Lagrange constant $\lambda'_{3,k}$ can be calculated as

$$-\lambda'_{3,k}\sum_{a=1}^{T}\frac{1}{\sum\limits_{i=1}^{N}u_{ki}^{m}y_{ai}^{2}+\dfrac{\delta}{\alpha}}=\sum_{a=1}^{T}\frac{\sum\limits_{i=1}^{N}u_{ki}^{m}\left(o_i-\sum\limits_{t=1}^{T}w_{kt}y_{ti}\right)y_{ai}-w_{ka}\dfrac{\delta}{\alpha}}{\sum\limits_{i=1}^{N}u_{ki}^{m}y_{ai}^{2}+\dfrac{\delta}{\alpha}}, \tag{A–29a}$$

$$-\lambda'_{3,k}=\frac{\sum\limits_{a=1}^{T}\dfrac{\sum\limits_{i=1}^{N}u_{ki}^{m}\left(o_i-\sum\limits_{t=1}^{T}w_{kt}y_{ti}\right)y_{ai}-w_{ka}\dfrac{\delta}{\alpha}}{\sum\limits_{i=1}^{N}u_{ki}^{m}y_{ai}^{2}+\dfrac{\delta}{\alpha}}}{\sum\limits_{a=1}^{T}\dfrac{1}{\sum\limits_{i=1}^{N}u_{ki}^{m}y_{ai}^{2}+\dfrac{\delta}{\alpha}}}. \tag{A–29b}$$

APPENDIX B
RECONSTRUCTION ERROR WITH THE MEMBERSHIPS IN THE FEATURE SPACE

The reconstruction error in the feature space can be defined as

$$\begin{aligned} e(\phi(x)) =& ||(\phi(x) - \mu_\phi) - WW^T(\phi(x) - \mu_\phi)||^2 \\ =& (\phi(x) - \mu_\phi) \cdot (\phi(x) - \mu_\phi) - (\phi(x) - \mu_\phi) \cdot (WW^T(\phi(x) - \mu_\phi)) \\ &+ (WW^T(\phi(x) - \mu_\phi)) \cdot (WW^T(\phi(x) - \mu_\phi)), \end{aligned} \tag{B–1}$$

where W is an eigenvector matrix composed of k eigenvectors as columns:

$$W = \left[w^1 w^2 \dots w^k \right]. \tag{B–2}$$

Each eigenvector can be represented as a weighted combination of the mapped data points x_i ($1 \leq i \leq N$) as

$$w^j = \sum_{i=1}^{N} \alpha_i^j u_i \phi(x_i), \tag{B–3}$$

where N is the number of data points. The mean data μ_ϕ also can be represented as a combination of the projected data points as

$$\mu_\phi = \frac{\sum_{i=1}^{N} u_i \phi(x_i)}{\sum_{i=1}^{N} u_i} = \frac{1}{N'} \sum_{i=1}^{N} u_i \phi(x_i), \tag{B–4}$$

where $N' = \sum_{i=1}^{N} u_i$. Using Equation (B–3) and (B–4), the first term of Equation (B–1) can be written as

$$\begin{aligned} &(\phi(x) - \mu_\phi) \cdot (\phi(x) - \mu_\phi) \\ &\quad = \phi(x) \cdot \phi(x) - 2\phi(x) \cdot \mu_\phi + \mu_\phi \cdot \mu_\phi \\ &\quad = \kappa(x, x) - \frac{2}{N'} \sum_{i=1}^{N} u_i \phi(x_i) \cdot \phi(x) + \frac{1}{N'^2} \sum_{i=1}^{N} u_i \phi(x_i) \cdot \sum_{j=1}^{N} u_j \phi(x_j) \\ &\quad = \kappa(x, x) - \frac{2}{N'} \sum_{i=1}^{N} u_i \kappa(x_i, x) + \frac{1}{N'^2} \sum_{i=1}^{N} \sum_{j=1}^{N} u_i u_j \kappa(x_i, x_j). \end{aligned} \tag{B–5}$$

The second term of Equation (B–1) can be written as

$$
\begin{aligned}
&(\phi(x)-\mu_\phi)\cdot(WW^T(\phi(x)-\mu_\phi)) \\
&\quad =(\phi(x)-\mu_\phi)\cdot(WW^T\phi(x)-WW^T\mu_\phi) \\
&\quad =\phi(x)\cdot(WW^T\phi(x))-\phi(x)\cdot(WW^T\mu_\phi)-\mu_\phi\cdot(WW^T\phi(x))+\mu_\phi\cdot(WW^T\mu_\phi) \\
&\quad =(W^T\phi(x))\cdot(W^T\phi(x))-2(W^T\phi(x))\cdot(W^T\mu_\phi)+(W^T\mu_\phi)\cdot(W^T\mu_\phi).
\end{aligned}
\tag{B–6}
$$

In Equation (B–6), $W^T\phi(x)$ can be calculated as

$$
\begin{aligned}
W^T\phi(x) &= \begin{bmatrix} \sum_{i=1}^N \alpha_i^1 u_i \phi(x_i)^T \\ \vdots \\ \sum_{i=1}^N \alpha_i^k u_i \phi(x_i)^T \end{bmatrix} \phi(x) \\
&= \begin{bmatrix} \sum_{i=1}^N \alpha_i^1 u_i \phi(x_i)^T \phi(x) \\ \vdots \\ \sum_{i=1}^N \alpha_i^k u_i \phi(x_i)^T \phi(x) \end{bmatrix} \\
&= \begin{bmatrix} \sum_{i=1}^N \alpha_i^1 u_i \kappa(x_i, x) \\ \vdots \\ \sum_{i=1}^N \alpha_i^k u_i \kappa(x_i, x) \end{bmatrix} \\
&= \begin{bmatrix} \beta^1 & \cdots & \beta^k \end{bmatrix}^T,
\end{aligned}
\tag{B–7}
$$

where β^j is defined as

$$
\beta^j = \sum_{i=1}^N \alpha_i^j u_i \kappa(x_i, x). \tag{B–8}
$$

Similarly, $W^T\mu_\phi$ can be calculated as

$$
\begin{aligned}
W^T\mu_\phi &= \begin{bmatrix} \sum_{i=1}^N \alpha_i^1 u_i \phi(x_i)^T \\ \vdots \\ \sum_{i=1}^N \alpha_i^k u_i \phi(x_i)^T \end{bmatrix} \frac{1}{N'} \sum_{j=1}^N u_j \phi(x_j) \\
&= \frac{1}{N'} \begin{bmatrix} \sum_{i=1}^N \alpha_i^1 u_i \phi(x_i)^T \sum_{j=1}^N u_j \phi(x_j) \\ \vdots \\ \sum_{i=1}^N \alpha_i^k u_i \phi(x_i)^T \sum_{j=1}^N u_j \phi(x_j) \end{bmatrix} \\
&= \frac{1}{N'} \begin{bmatrix} \sum_{i=1}^N \sum_{j=1}^N \alpha_i^1 u_i u_j \kappa(x_i, x_j) \\ \vdots \\ \sum_{i=1}^N \sum_{j=1}^N \alpha_i^k u_i u_j \kappa(x_i, x_j) \end{bmatrix} \\
&= \begin{bmatrix} \gamma^1 & \cdots & \gamma^k \end{bmatrix}^T,
\end{aligned}
\tag{B–9}
$$

where γ^l is defined as

$$
\gamma^l = \frac{1}{N'} \sum_{i=1}^N \sum_{j=1}^N \alpha_i^l u_i u_j \kappa(x_i, x_j). \tag{B–10}
$$

Using Equations (B–7) and (B–9), Equation (B–6) can be represented as

$$
(\phi(x) - \mu_\phi) \cdot (WW^T(\phi(x) - \mu_\phi)) = \sum_{i=1}^k \left(\left(\beta^i\right)^2 - 2\beta^i\gamma^i + \left(\gamma^i\right)^2 \right). \tag{B–11}
$$

The third term of Equation (B–1) can be written as

$$
\begin{aligned}
&(WW^T(\phi(x) - \mu_\phi)) \cdot (WW^T(\phi(x) - \mu_\phi)) \\
&= (WW^T\phi(x) - WW^T\mu_\phi) \cdot (WW^T\phi(x) - WW^T\mu_\phi) \\
&= (WW^T\phi(x)) \cdot (WW^T\phi(x)) - 2(WW^T\phi(x)) \cdot (WW^T\mu_\phi) + (WW^T\mu_\phi) \cdot (WW^T\mu_\phi).
\end{aligned}
\tag{B–12}
$$

In Equation (B–12), $WW^T\phi(x)$ can be calculated using Equation (B–7) as

$$
\begin{aligned}
WW^T\phi(x) &= \begin{bmatrix} \sum_{i=1}^N \alpha_i^1 u_i \phi(x_i)^T \\ \vdots \\ \sum_{i=1}^N \alpha_i^k u_i \phi(x_i)^T \end{bmatrix}^T \begin{bmatrix} \beta^1 \\ \vdots \\ \beta^k \end{bmatrix} \\
&= \sum_{l=1}^k \sum_{i=1}^N \beta^l \alpha_i^l u_i \phi(x_i) \\
&= \sum_{l=1}^k \sum_{i=1}^N \beta^l \xi_i^l \phi(x_i),
\end{aligned}
\tag{B–13}
$$

where ξ_i^l is defined as

$$
\xi_i^l = \alpha_i^l u_i. \tag{B–14}
$$

Similarly, $WW^T\mu_\phi$ can be calculated using Equation (B–9) as

$$
\begin{aligned}
WW^T\mu_\phi &= \begin{bmatrix} \sum_{i=1}^N \alpha_i^1 u_i \phi(x_i)^T \\ \vdots \\ \sum_{i=1}^N \alpha_i^k u_i \phi(x_i)^T \end{bmatrix}^T \begin{bmatrix} \gamma^1 \\ \vdots \\ \gamma^k \end{bmatrix} \\
&= \sum_{l=1}^k \sum_{i=1}^N \gamma^l \alpha_i^l u_i \phi(x_i) \\
&= \sum_{l=1}^k \sum_{i=1}^N \gamma^l \xi_i^l \phi(x_i).
\end{aligned}
\tag{B–15}
$$

Using Equations (B–13) and (B–15), the terms in Equation (B–12) can be represented as

$$
\begin{aligned}
(WW^T\phi(x)) \cdot (WW^T\phi(x)) &= \left(\sum_{l=1}^k \sum_{i=1}^N \beta^l \xi_i^l \phi(x_i) \right) \cdot \left(\sum_{m=1}^k \sum_{j=1}^N \beta^m \xi_j^m \phi(x_j) \right) \\
&= \sum_{l=1}^k \sum_{m=1}^k \sum_{i=1}^N \sum_{j=1}^N \beta^l \beta^m \xi_i^l \xi_j^m \phi(x_i) \cdot \phi(x_j) \\
&= \sum_{l=1}^k \sum_{m=1}^k \sum_{i=1}^N \sum_{j=1}^N \beta^l \beta^m \xi_i^l \xi_j^m \kappa(x_i, x_j),
\end{aligned}
\tag{B–16}
$$

$$(WW^T\phi(x)) \cdot (WW^T\mu_\phi) = \sum_{l=1}^{k}\sum_{m=1}^{k}\sum_{i=1}^{N}\sum_{j=1}^{N}\beta^l\gamma^m\xi_i^l\xi_j^m\kappa(x_i, x_j), \tag{B–17}$$

$$(WW^T\mu_\phi) \cdot (WW^T\mu_\phi) = \sum_{l=1}^{k}\sum_{m=1}^{k}\sum_{i=1}^{N}\sum_{j=1}^{N}\gamma^l\gamma^m\xi_i^l\xi_j^m\kappa(x_i, x_j), \tag{B–18}$$

and Equation (B–12) can be written as

$$\begin{aligned}(WW^T(\phi(x) - \mu_\phi)) \cdot (WW^T(\phi(x) - \mu_\phi)) \\ = \sum_{l,m=1}^{k}\sum_{i,j=1}^{N}(\beta^l\beta^m - 2\beta^l\gamma^m + \gamma^l\gamma^m)\xi_i^l\xi_j^m\kappa(x_i, x_j).\end{aligned} \tag{B–19}$$

Finally, Equation (B–1) can be written using Equations (B–5), (B–11), and (B–19) as

$$\begin{aligned}e(\phi(x)) =& ||(\phi(x) - \mu_\phi) - WW^T(\phi(x) - \mu_\phi)||^2 \\ =& \kappa(x, x) - \frac{2}{N'}\sum_{i=1}^{N}u_i\kappa(x_i, x) + \frac{1}{N'^2}\sum_{i,j=1}^{N}u_iu_j\kappa(x_i, x_j) \\ & - 2\sum_{i=1}^{k}\left(\left(\beta^i\right)^2 - 2\beta^i\gamma^i + \left(\gamma^i\right)^2\right) \\ & + \sum_{l,m=1}^{k}\sum_{i,j=1}^{N}(\beta^l\beta^m - 2\beta^l\gamma^m + \gamma^l\gamma^m)\xi_i^l\xi_j^m\kappa(x_i, x_j).\end{aligned} \tag{B–20}$$

APPENDIX C
UPDATE EQUATIONS FOR KERNEL-BASED CONTEXT-DEPENDENT FUSION

The objective function for *kernel-based context-dependent fusion* can be written as

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha \sum_{k=1}^{K}\left(\frac{1}{2}||w_k||^2 + C\sum_{i=1}^{N} u_{ki}^{m}\xi_{ki}\right) + \beta \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^{q}, \tag{C–1}$$

where N is the number of data points, K is the number of clusters, and L is the dimension of data. The value u_{ki} is the membership of x_i to the kth cluster and r_{kl} is the weight of the lth feature in the kth cluster. The constants m and q represent the fuzzifier constants for membership and feature weight respectively. The value $d_{kil} = |v_{kl} - x_{il}|$ is a feature-wise distance which measures the difference between the lth feature of a data point x_i and the lth feature of the kth cluster center v_k. The second term is a sum of K objective functions of fuzzy support vector machine (FSVM), each of which corresponds to a different fusion process in a specific cluster. All the FSVMs share the regularization constant C. The value w_k represents the decision surface in the kth FSVM and ξ_{ki} is an error of x_i in the kth FSVM. As the parameter set for the kth FSVM $\{w_k, \xi_{ki} | i = 1, \cdots, N\}$ is independent from other parameters and can be optimized using the optimization procedure of FSVM when $\{u_{ki} | i = 1, \cdots, N\}$ is known, the update equations for U, V, and R are discussed here. Refer to chapter 5 for more detailed information about SVM. The value α is a constant balancing between clustering and fusion and the remaining two constants – β and γ – are regularization constants. The objective function should be minimized under the constraints on membership U and feature weight R. The constraints can be represented as

$$U_{K\times N} = \left\{ u_{ki} \middle| u_{ki} \in [0,1]\ \forall i \text{ and } k; \sum_{k=1}^{K} u_{ki} = 1\ \forall i \right\}, \tag{C–2}$$

$$R_{K\times L} = \left\{ r_{kl} \middle| r_{kl} \in (0,1]\ \forall k \text{ and } l; \sum_{l=1}^{L} r_{kl} = 1\ \forall k \right\}. \tag{C–3}$$

From the objective function in Equation (C–1) and the constraints, the Lagrange equation can be written as

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m}\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha\sum_{k=1}^{K}\left(\frac{1}{2}||w_k||^2 + C\sum_{i=1}^{N} u_{ki}^{m}\xi_{ki}\right)\\ &+ \beta\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} + \gamma\sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^{q}\\ &- \sum_{i=1}^{N}\lambda_{1,i}\left(\sum_{k=1}^{K} u_{ki} - 1\right) - \sum_{k=1}^{K}\lambda_{2,k}\left(\sum_{l=1}^{L} r_{kl} - 1\right),\end{aligned} \tag{C–4}$$

where $\lambda_{1,i}$ and $\lambda_{2,k}$ are Lagrange multipliers. By collecting all the terms related to u_{ki} from the Lagrange equation, one can obtain

$$\begin{aligned}\mathcal{L}_1 = &\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m}\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha\sum_{k=1}^{K}\left(\frac{1}{2}||w_k||^2 + C\sum_{i=1}^{N} u_{ki}^{m}\xi_{ki}\right)\\ &+ \beta\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} - \sum_{i=1}^{N}\lambda_{1,i}\left(\sum_{k=1}^{K} u_{ki} - 1\right).\end{aligned} \tag{C–5}$$

By taking a partial derivative with respect to u_{ki} and equating to zero, one can obtain

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial u_{ki}} &= m u_{ki}^{m-1}\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha C m u_{ki}^{m-1}\xi_{ki} + \beta m u_{ki}^{m-1} - \lambda_{1,i}\\ &= m u_{ki}^{m-1}\left(\sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha C\xi_{ki} + \beta\right) - \lambda_{1,i} = 0.\end{aligned} \tag{C–6}$$

Let $\widetilde{D}_{ki}$ be a new distance measure defined as

$$\widetilde{D}_{ki} = \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \alpha C\xi_{ki} + \beta, \tag{C–7}$$

then Equation (C–6) can be re-written as

$$m u_{ki}^{m-1}\widetilde{D}_{ki} = \lambda_{1,i}, \tag{C–8a}$$

$$u_{ki}^{m-1} = \frac{\lambda_{1,i}/m}{\widetilde{D}_{ki}} = \frac{\lambda'_{1,i}}{\widetilde{D}_{ki}}, \tag{C–8b}$$

and the membership value u_{ki} can be calculated as

$$u_{ki} = \left(\frac{\lambda'_{1,i}}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}. \tag{C–9}$$

The membership should satisfy the constraint in Equation (C–2). Using the constraint, the Lagrange multiplier $\lambda'_{1,i}$ can be calculated as

$$\sum_{k=1}^{K} u_{ki} = \sum_{k=1}^{K} \left(\frac{\lambda'_{1,i}}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}} = {\lambda'_{1,i}}^{\frac{1}{m-1}} \sum_{k=1}^{K} \left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}} = 1, \tag{C–10a}$$

$${\lambda'_{1,i}}^{\frac{1}{m-1}} = \frac{1}{\sum_{k=1}^{K} \left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}}. \tag{C–10b}$$

Using Equations (C–9) and (C–10b), the update equation for the membership u_{ki} can be written as

$$u_{ki} = \frac{\left(\frac{1}{\widetilde{D}_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K} \left(\frac{1}{\widetilde{D}_{ai}}\right)^{\frac{1}{m-1}}}. \tag{C–11}$$

By collecting all the terms related to v_{kl} from the Lagrange equation, one can obtain

$$\mathcal{L}_2 = \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ki}^m \sum_{l=1}^{L} r_{kl}^q |x_{il} - v_{kl}|^2. \tag{C–12}$$

By taking a partial derivative with respect to v_{kl} and equating to zero, one can obtain

$$\frac{\partial \mathcal{L}_2}{\partial v_{kl}} = -2 \sum_{i=1}^{N} u_{ki}^m r_{kl}^q (x_{il} - v_{kl}) = 0. \tag{C–13}$$

Because the feature weight $r_{kl} > 0$, i.e., all the features are assumed to have effects on the distance calculation, Equation (C–13) can be re-written as

$$r_{kl}^q v_{kl} \sum_{i=1}^{N} u_{ki}^m = r_{kl}^q \sum_{i=1}^{N} u_{ki}^m x_{il}, \tag{C–14a}$$

$$v_{kl} \sum_{i=1}^{N} u_{ki}^m = \sum_{i=1}^{N} u_{ki}^m x_{il}, \tag{C–14b}$$

and the update equation of the cluster center v_{kl} can be written as

$$v_{kl} = \frac{\sum_{i=1}^{N} u_{ki}^{m} x_{il}}{\sum_{i=1}^{N} u_{ki}^{m}}. \tag{C–15}$$

By collecting all the terms related to r_{kl} from the Lagrange equation, one can obtain

$$\mathcal{L}_3 = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ki}^{m} \sum_{l=1}^{L} r_{kl}^{q} d_{kil}^{2} + \gamma \sum_{k=1}^{K}\sum_{l=1}^{L} r_{kl}^{q} - \sum_{k=1}^{K} \lambda_{2,k} \left(\sum_{l=1}^{L} r_{kl} - 1 \right). \tag{C–16}$$

By taking a partial derivative with respect to r_{kl} and equating to zero, one can obtain

$$\begin{aligned} \frac{\partial \mathcal{L}_3}{\partial r_{kl}} &= \sum_{i=1}^{N} u_{ki}^{m} q r_{kl}^{q-1} d_{kil}^{2} + \gamma q r_{kl}^{q-1} - \lambda_{2,k} \\ &= q r_{kl}^{q-1} \left(\sum_{i=1}^{N} u_{ki}^{m} d_{kil}^{2} + \gamma \right) - \lambda_{2,k} = 0. \end{aligned} \tag{C–17}$$

Let $\overline{D}_{kl}$ be another distance measure defined in a similar way to Equation (C–7) as

$$\overline{D}_{kl} = \sum_{i=1}^{N} u_{ki}^{m} d_{kil}^{2} + \gamma, \tag{C–18}$$

then Equation (C–17) can be re-written as

$$q r_{kl}^{q-1} \overline{D}_{kl} = \lambda_{2,k}, \tag{C–19a}$$

$$r_{kl}^{q-1} = \frac{\lambda_{2,k}/q}{\overline{D}_{kl}} = \frac{\lambda'_{2,k}}{\overline{D}_{kl}}, \tag{C–19b}$$

and the feature weight r_{kl} can be calculated as

$$r_{kl} = \left(\frac{\lambda'_{2,k}}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}}. \tag{C–20}$$

The feature weight r_{kl} should satisfy the constraint in Equation (C–3). Using the constraint, the Lagrange multiplier $\lambda'_{2,k}$ can be calculated as

$$\sum_{l=1}^{L} r_{kl} = \sum_{l=1}^{L} \left(\frac{\lambda'_{2,k}}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}} = {\lambda'_{2,k}}^{\frac{1}{q-1}} \sum_{l=1}^{L} \left(\frac{1}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}} = 1, \tag{C–21a}$$

$${\lambda'_{2,k}}^{\frac{1}{q-1}} = \frac{1}{\sum_{l=1}^{L} \left(\frac{1}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}}}. \tag{C–21b}$$

Using Equations (C–20) and (C–21b), the update equation for the feature weight r_{kl} can be written as

$$r_{kl} = \frac{\left(\frac{1}{\overline{D}_{kl}} \right)^{\frac{1}{q-1}}}{\sum_{a=1}^{L} \left(\frac{1}{\overline{D}_{ka}} \right)^{\frac{1}{q-1}}}. \tag{C–22}$$

REFERENCES

[1] R. A. Fisher, "The use of multiple measurements in taxonomic problem," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, Singapore, 2007.

[3] H. Frigui, P. Gader, and A. C. B. Abdallah, "A generic framework for context-dependent fusion with application to landmine detection," in *Proceedings of the SPIE Conference on Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIII*, vol. 6953, Orlando FL, March 2008.

[4] H. Frigui, L. Zhang, P. Gader, and D. Ho, "Context dependent fusion for landmine detection with ground penetrating radar," in *Proceedings of the SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets XII*, vol. 6553, Orlando FL, April 2007.

[5] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2001.

[6] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[7] G. Heo, P. Gader, and H. Frigui, "Robust kernel PCA using fuzzy membership," in *Proceedings of the 2009 International Joint Conference on Neural Networks*, Atlanta, GA, June 2009.

[8] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. John Wiley, New York, 1973.

[9] M. Minsky, "Logical versus analogical or symbolic versus connectionist or neat versus scruffy," *AI Magazine*, vol. 12, no. 2, pp. 34–51, 1991.

[10] L. I. Kuncheva, *Combining Pattern Classifiers*. Wiley-Interscience, New York, 2004.

[11] K. Sirlantzis, S. Hoque, and M. Fairhurst, "Trainable multiple classifier schemes for handwritten character recognition," in *Proceedings of the 3rd International Workshop on Multiple Classifier Systems*, Cagliari, Italy, June 2002, pp. 319–322.

[12] F. Huenupan, N. B. Yoma, C. Molina, and C. Garreton, "Confidence based multiple classifier fusion in speaker verification," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 957–966, 2008.

[13] Y. Bi, D. Bell, H. Wang, G. Guo, and J. Guan, "Combining multiple classifiers using Dempster's rule for text categorization," *Applied Artificial Intelligence*, vol. 21, no. 3, pp. 211–239, 2007.

[14] L. K. Hansen and P. Salamon, “Neural network ensembles,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[15] T. G. Dietterich, “Ensemble methods in machine learning,” in *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, June 2000, pp. 1–15.

[16] S. E. Shimony, “Finding MAPs for belief networks is NP-hard,” *Artificial Intelligence*, vol. 68, no. 2, pp. 399–410, 1994.

[17] A. M. Abdelbar and S. M. Hedetniemi, “Approximating MAPs for belief networks is NP-hard and other theorems,” *Artificial Intelligence*, vol. 102, no. 1, pp. 21–38, 1998.

[18] J. Kittler and F. M. Alkoot, “Sum versus vote fusion in multiple classifier systems,” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 110–115, 2003.

[19] D. Ruta and B. Gabrys, “Classifier selection for majority voting,” *Information Fusion*, vol. 6, no. 1, pp. 63–81, 2005.

[20] L. S. Kennedy and S.-F. Chang, “A reranking approach for context-based concept fusion in video indexing and retrieval,” in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, Amsterdam, The Netherlands, July 2007, pp. 330–340.

[21] J. Feng, S. Yoon, and A. K. Jain, “Latent fingerprint matching: Fusion of rolled and plain fingerprints,” in *Proceedings of the 3rd IAPR/IEEE International Conference on Biometrics*, Alghero, Italy, June 2009.

[22] T.-Y. Kim and H. Ko, “Bayesian fusion of confidence measures for speech recognition,” *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 871–874, 2005.

[23] P. de Oude, G. Pavlin, and T. Hood, “A modular approach to adaptive Bayesian information fusion,” in *Proceedings of the 10th International Conference on Information Fusion*, Quebec, Canada, July 2007, pp. 1–8.

[24] N. Farah, M. T. Khadir, and M. Sellami, “Artificial neural network fusion: Application to Arabic words recognition,” in *Proceedings of 2005 European Symposium on Artificial Neural Networks*, Bruges, Belgium, April 2005, pp. 151–156.

[25] S. Kang and S. Park, “A fusion neural network classifier for image classification,” *Pattern Recognition Letters*, 2008.

[26] M. Arif, T. Brouard, and N. Vincent, “A fusion methodology based on Dempster-Shafer evidence theory for two biometric applications,” in *Proceedings of the 18th International Conference on Pattern Recognition*, Hong Kong, August 2006, pp. 590–593.

[27] S. L. Hegarat-Mascle, D. Richard, and C. Ottle, “Multi-scale data fusion using Dempster-Shafer evidence theory,” *Integrated Computer-Aided Engineering*, vol. 10, no. 1, pp. 9–22, 2003.

[28] M. M. Islam, X. Yao, S. M. S. Nirjon, M. A. Islam, and K. Murase, “Bagging and boosting negatively correlated neural networks,” *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 38, no. 3, pp. 771–784, 2008.

[29] S. B. Kotsiantis and P. E. Pintelas, “Combining bagging and boosting,” *International Journal of Computational Intelligence*, vol. 1, no. 4, pp. 324–333, 2004.

[30] A. Temko, D. Macho, and C. Nadeu, “Fuzzy integral based information fusion for classification of highly confusable non-speech sounds,” *Pattern Recognition*, vol. 41, no. 5, pp. 1831–1840, 2008.

[31] A. Mendez-Vazquez, P. Gader, J. M. Keller, and K. Chamberlin, “Minimum classification error training for Choquet integrals with application to landmine detection,” *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 225–238, 2008.

[32] K. Woods, W. P. K. Jr., and K. Bowyer, “Combination of multiple classifiers using local accuracy estimates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405–410, 1997.

[33] L. I. Kuncheva, “Clustering-and-selection model for classifier combination,” in *Proceedings of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Brighton, UK, August 2000, pp. 185–188.

[34] R. Liu and B. Yuan, “Multiple classifiers combination by clustering and selection,” *Information Fusion*, vol. 2, pp. 163–168, 2001.

[35] H. Frigui and O. Nasraoui, “Simultaneous clustering and attribute discrimination,” in *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, San Antonio, TX, May 2000, pp. 158–163.

[36] A. Keller and F. Klawonn, “Fuzzy clustering with weighting of data variables,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 8, no. 6, pp. 735–746, 2000.

[37] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

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

[39] A. N. Tikhonov, "On solving incorrectly posed problems and method of regularization," *Dokl. Acad. Nauk USSR*, vol. 151, pp. 501–504, 1963.

[40] R.-P. Li and M. Mukaidono, "A maximum entropy approach to fuzzy clustering," in *Proceeding of the 4th IEEE International Conference on Fuzzy Systems*, Yokohama, Japan, March 1995, pp. 2227–2232.

[41] S. Miyamoto and K. Umayahara, "Fuzzy clustering by quadratic regularization," in *Proceeding of the 1998 IEEE International Conference on Fuzzy Systems*, Anchorage, AK, May 1998, pp. 1394–1399.

[42] D. Ozdemir and L. Akarun, "A fuzzy algorithm for color quantization of images," *Pattern Recognition*, vol. 35, pp. 1785–1791, 2002.

[43] J. Yu and M.-S. Yang, "A generalized fuzzy clustering regularization model with optimality tests and model complexity analysis," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 5, pp. 904–915, 2007.

[44] J. A. MacDonald, *Alternatives for Landmine Detection*. RAND, 2003.

[45] H. Frigui, K. C. Ho, and P. Gader, "Real-time landmine detection with ground-penetrating radar using discriminative and adaptive hidden Markov models," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 12, pp. 1867–1885, 2005.

[46] Y. Zhao, P. Gader, P. Chen, and Y. Zhang, "Training DHMMs of mine and clutter to minimize landmine detection errors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 5, pp. 1016–1024, 2003.

[47] G. Heo and P. Gader, "Prior-updating ensemble learning for discrete HMM," in *Proceedings of the 19th International Conference on Pattern Recognition*, Tampa FL, December 2008, pp. 1–4.

[48] K. C. Ho, L. Carin, P. Gader, and J. N. Wilson, "An investigation of using the spectral characteristics from ground penetrating radar for landmine/clustter discrimination," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 4, pp. 1177–1191, 2008.

[49] H. Frigui and P. Gader, "Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic k-nearest neighbor classifier," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 185–199, 2009.

[50] ——, "Detection and discrimination of landmines based on edge histogram descriptors and fuzzy k-nearest neighbors," in *Proceedings of IEEE International Conference on Fuzzy Systems*, Vancouver Canada, July 2006, pp. 1494–1499.

[51] G. Heo and P. Gader, “Shape-based landmine detection using wideband electromagnetic induction method,” Technical Report, 2006, Computer and Information Science and Engineering, University of Florida.

[52] S. E. Yuksel, G. Ramachandran, P. Gader, J. Wilson, D. Ho, and G. Heo, “Hierarchical methods for landmine detection with wideband electro-magnetic induction and ground penetrating radar multi-sensor systems,” in *Proceedings of IEEE International Conference on Geoscience and Remote Sensing Symposium*, Boston MA, July 2008, pp. II–177–II–180.

[53] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, New York, 2002.

[54] P. Rousseeuw, “Multivariate estimation with high breakdown point,” *Mathematical Statistics and Applications*, vol. B, pp. 283–297, 1985.

[55] B. Gabrys, B. Baruque, and E. Corchado, “Outlier resistant PCA ensembles,” in *Proceedings of the 10th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, Bournemouth, UK, October 2006, pp. 432–440.

[56] C.-D. Lu, T.-Y. Zhang, X.-Z. Du, and C.-P. Li, “A robust kernel PCA algorithm,” in *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, Shanghai, China, August 2004, pp. 3084–3087.

[57] C. Lu, T. Zhang, R. Zhang, and C. Zhang, “Adaptive robust kernel PCA algorithm,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, April 2003, pp. VI 621–624.

[58] M. Sato-Ilic and L. C. Jain, “Fuzzy clustering based principal component analysis,” in *Innovations in Fuzzy Clustering*. Springer, 2006, pp. 9–44.

[59] T. R. Cundari, C. Sarbu, and H. F. Pop, “Robust fuzzy principal component analysis (FPCA). A comparative study concerning interaction of carbon-hydrogen bonds with molybdenum-oxo bonds,” *Journal of Chemical Information and Computer Sciences*, vol. 42, no. 6, pp. 1363–1369, 2002.

[60] T.-N. Yang and S.-D. Wang, “Fuzzy auto-associative neural networks for principal component extraction of noisy data,” *IEEE Transaction on Neural Networks*, vol. 11, no. 3, pp. 808–810, 2000.

[61] M. Scholz, M. Fraunholz, and J. Selbig, “Nonlinear principal component analysis: Neural network models and applications,” in *Principal Manifolds for Data Visualization and Dimension Reduction*, A. N. Gorban, B. Kegl, D. C. Wunsch, and A. Zinovyev, Eds. Springer, Berlin, 2007, pp. 44–67.

[62] M. Harkat, G. Mourot, and J. Ragot, “Nonlinear PCA combining principal curves and RBF-networks for process monitoring,” in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 1956–1961.

[63] J. Li, X. Li, and D. Tao, "KPCA for semantic object extraction in images," *Pattern Recognition*, vol. 41, no. 10, pp. 3244–3250, 2008.

[64] G.-H. Cha, "Kernel principal component analysis for content based image retrieval," in *Proceedings of the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Hanoi, Vietnam, May 2005, pp. 844–849.

[65] A. Jade, B. Srikanth, V. Jayaraman, B. Kulkarni, J. Jog, and L. Priya, "Feature extraction and denoising using kernel PCA," *Chemical Engineering Science*, vol. 58, pp. 4441–4448, 2003.

[66] P. J. Huber, *Robust Statistics*. Wiley-Interscience, New York, 1981.

[67] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, 2001.

[68] J. M. Leski, "Fuzzy *c*-varieties/elliptotypes clustering in reproducing kernel Hilbert space," *Fuzzy Sets and Systems*, vol. 141, no. 2, pp. 259–280, 2004.

[69] R. N. Dave, "Characterization and detection of noise in clustering," *Pattern Recognition Letters*, vol. 12, pp. 657–664, 1991.

[70] L. Xu and A. L. Yuille, "Robust principal component analysis by self-organizing rules based on statistical physics approach," *IEEE Transaction on Neural Networks*, vol. 6, no. 1, pp. 131–143, 1995.

[71] H. Ichihashi, K. Honda, and N. Tani, "Gaussian mixture PDF approximation and fuzzy *c*-means clustering with entropy regularization," in *Proceedings of the 4th Asian Fuzzy Systems Symposium*, Tsukuba, Japan, May 2000, pp. 217–221.

[72] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[73] J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*. Wiley-Interscience, New York, 2000.

[74] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, pp. 645–678, 2005.

[75] J. He, M. Lan, C.-L. Tan, S.-Y. Sung, and H.-B. Low, "Initialization of cluster refinement algorithms: A review and comparative study," in *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*, Budapest, Hungary, July 2004, pp. 297–302.

[76] D. Steinley and M. J. Brusco, "Initializing k-means batch clustering: A critical evaluation of several techniques," *Journal of Classification*, vol. 24, pp. 99–121, 2007.

[77] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*. Chapman & Hall/CRC, 2005.

[78] S.-Z. Yang and S.-W. Luo, "A novel algorithm for initializing clustering centers," in *Proceedings of the 2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China, August 2005, pp. 5579–5583.

[79] W. Wang, Y. Zhang, Y. Li, and X. Zhang, "The global fuzzy *c*-means clustering algorithm," in *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, June 2006, pp. 3604–3607.

[80] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, pp. 451–461, 2003.

[81] A. M. Bagirov, "Modified global k-means algorithm for minimum sum-of-squares clustering problems," *Pattern Recognition*, vol. 41, pp. 3192–3199, 2008.

[82] H.-S. Tsai, "A study on kernel-based clustering algorithms," Ph.D. dissertation, Department of Applied Mathematics, Chung Yuan Christian University, Chung Li, Taiwan, 2007.

[83] D. J. Klein and M. Randic, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, pp. 81–85, 1993.

[84] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[85] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, Berkeley, CA, 1967, pp. 281–297.

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

[87] R. J. Hathaway and J. C. Bezdek, "Optimization of clustering criteria by reformulation," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, pp. 241–245, 1995.

[88] M. Girolami, "Mercer kernel-based clustering in feature space," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 780–784, 2002.

[89] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[90] I. Fischer and J. Poland, “Amplifying the block matrix structure for spectral clustering,” in *Proceedings of the 14th Annual Machine Learning Conference of Belgium and the Netherlands*, Enschede, The Netherlands, February 2005, pp. 21–28.

[91] M. Meila, “Comparing clusterings – an information based distance,” *Journal of Multivariate Analysis*, vol. 98, no. 5, pp. 873–895, 2007.

[92] V. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

[93] K. Wu and K.-H. Yap, “Fuzzy SVM for content-based image retrieval,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 2, pp. 10–16, 2006.

[94] B. Lim, M. Tsui, V. Charastrakul, and D. Shi, “Web search with text categorization using probabilistic framework of SVM,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Taipei, Taiwan, October 2006, pp. 2950–2955.

[95] C.-F. Lin and S.-D. Wang, “Fuzzy support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 464–471, 2002.

[96] Y. Wang, S. Wang, and K. K. Lai, “A new fuzzy support vector machine to evaluate credit risk,” *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 820–831, 2005.

[97] L. Wang, P. Xue, and K. L. Chan, “Incorporating prior knowledge into SVM for image retrieval,” in *Proceedings of the 17th International Conference on Pattern Recognition*, Cambridge, UK, August 2004, pp. 981–984.

[98] Q. Tao, G.-W. Wu, F.-Y. Wang, and J. Wang, “Posterior probability support vector machines for unbalanced data,” *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1561–1573, 2005.

[99] Y. Liu and Y. F. Zheng, “Soft SVM and its application in video-object extraction,” *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3272–3282, 2007.

[100] X. Jiang, Z. Yi, and J. C. Lv, “Fuzzy SVM with a new fuzzy membership function,” *Neural Computing and Applications*, vol. 15, pp. 268–276, 2006.

[101] C. fu Lin and S. de Wang, “Training algorithms for fuzzy support vector machines with noisy data,” *Pattern Recognition Letters*, vol. 25, pp. 1647–1656, 2004.

[102] A. Shilton and D. T. H. Lai, “Iterative fuzzy support vector machine classification,” in *Proceedings of the IEEE International Conference on Fuzzy Systems*, London, UK, July 2007, pp. 1–6.

[103] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, “On kernel-target alignment,” *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[104] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," in *Proceedings of Neural Information Processing Systems*, Denver, CO, November-December 1999, pp. 536–542.

[105] T. Takahashi and T. Kurita, "Robust de-noising by kernel PCA," in *Proceedings of the 2002 International Conference on Artificial Neural Networks*, Madrid, Spain, August 2002, pp. 739–744.

[106] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[107] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Technical Report, 2003, Department of Computer Science, National Taiwan University, Taipei, Taiwan.

[108] M. Zhu and A. Ghodsi, "Automatic dimensionality selection from the scree plot via the use of profile likelihood," *Computational Statistics and Data Analysis*, vol. 51, pp. 918–930, 2006.

[109] J. C. Platt, *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. MIT Press, 2004.

[110] A. Madevska-Bogdanova, D. Nikolik, and L. Curfs, "Probabilistic SVM outputs for pattern recognition using analytical geometry," *Neurocomputing*, vol. 62, pp. 293–303, 2004.

[111] C. M. Schubert, M. E. Oxley, and K. W. Bauer, "A comparison of ROC curves for label-fused within and across classifier systems," in *Proceedings of the 7th International Conference on Information Fusion*, 2005, pp. 415–422.

[112] C. Cortes and M. Mohri, "AUC optimization vs. error rate minimization," in *Advances in Neural Information Processing Systems*, 2003, pp. 313–320.

[113] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[114] P. G. Lucey, T. J. Williams, M. Mignard, J. Julian, D. Kobubun, G. Allen, D. Hampton, W. Schaff, M. J. Schlangen, E. M. Winter, W. B. Kendall, A. D. Stocker, K. A. Horton, and A. P. Bowman, "AHI: an airborne long-wave infrared hyperspectral imager," in *Proceedings of SPIE*, vol. 3431, 1998, pp. 36–43.

[115] R. Mayer, F. Bucholtz, and D. Scribner, "Object detection by using whitening/dewhitening to transform target signatures in multitemporal hyperspectral and multispectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 5, pp. 1136–1142, 2003.

[116] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, “Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm,” in *Summaries of the 3rd Annual JPL Airborne Geoscience Workshop*, Pasadena, CA, 1992, pp. 147–149.

[117] O. A. de Carvalho and P. R. Meneses, “Spectral correlation mapper (SCM): An improvement on the spectral angle mapper (SAM),” in *Summaries of the 9th Annual JPL Airborne Earth Science Workshop*, Pasadena, CA, 2000, pp. 65–74.

BIOGRAPHICAL SKETCH

Gyeongyong Heo received his Bachelor of Science and Master of Engineering in electronic engineering at Yonsei University in Korea in 1994 and 1996 respectively. Since 2004, he continued his studies in the department of Computer and Information Science and Engineering at University of Florida. His research interests include landmine detection, kernel methods, image analysis, and pattern recognition.