

University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

12-2010

A generic framework for context-dependent fusion with application to landmine detection.

Ahmed Chamseddine Ben Abdallah
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>

Recommended Citation

Ben Abdallah, Ahmed Chamseddine, "A generic framework for context-dependent fusion with application to landmine detection." (2010). *Electronic Theses and Dissertations*. Paper 99.
<https://doi.org/10.18297/etd/99>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

**A GENERIC FRAMEWORK FOR CONTEXT-DEPENDENT
FUSION WITH APPLICATION TO LANDMINE
DETECTION**

By

AHMED CHAMSEDDINE BEN ABDALLAH
M.S., Tunisia Polytechnic School, Tunisia, 2005

A Dissertation

Submitted to the Faculty of the
Graduate School of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science
University Of Louisville
Louisville, Kentucky

December 2010

Copyright © 2010 by Ahmed Chamseddine BEN ABDALLAH

All rights reserved

A GENERIC FRAMEWORK FOR CONTEXT-DEPENDENT FUSION WITH APPLICATION TO LANDMINE DETECTION

By

AHMED CHAMSEDDINE BEN ABDALLAH
M.S., Tunisia Polytechnic School, Tunisia, 2005

A Dissertation Approved on

November 22, 2010

by the Following Reading and Examination Committee:

Hichem Frigui, Ph.D., Dissertation Director

Amir Amini, Ph.D.

Ayman El-Baz, Ph.D.

Ming Ouyang, Ph.D.

Roman V. Yampolskiy, Ph.D.

“ *Whoever slays a soul, ..., it is as though he slew all men; and whoever keeps it alive, it is as though he kept alive all men*

”

Quran 5:32

“ *As a footballer I can't imagine life without the use of one of my legs... Sadly this is exactly what happens to thousands of children every year when they accidentally step on a landmine.*

”

Ryan Giggs

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Hichem Frigui, for giving me the opportunity to be a member in his research group and for his support over the course of this work. He provided a very rich working environment with many opportunities to develop new ideas, work in promising applications and get experience in diverse areas. I am indebted to his support and help.

I would like to thank Dr. Amir Amini, Dr. Ayman El-Baz, Dr. Ming Ouyang, and Dr. Roman V. Yampolskiy for agreeing to serve on my dissertation committee and being a part of this special milestone.

Last but certainly not least, I would like to thank all the members of the Multimedia Research Laboratory at University of Louisville, both past and present, for their support and sympathy.

ABSTRACT

A GENERIC FRAMEWORK FOR CONTEXT-DEPENDENT FUSION WITH APPLICATION TO LANDMINE DETECTION

AHMED CHAMSEDDINE BEN ABDALLAH

NOVEMBER 22, 2010

For complex detection and classification problems, involving data with large intra-class variations and noisy inputs, no single source of information can provide a satisfactory solution. As a result, combination of multiple classifiers is playing an increasing role in solving these complex pattern recognition problems, and has proven to be a viable alternative to using a single classifier.

Over the past few years, a variety of schemes have been proposed for combining multiple classifiers. Most of these were *global* as they assign a degree of worthiness to each classifier, that is averaged over the entire training data. This may not be the optimal way to combine the different experts since the behavior of each one may not be uniform over the different regions of the feature space. To overcome this issue, few *local* methods have been proposed in the last few years. Local fusion methods aim to adapt the classifiers' worthiness to different regions of the feature space. First, they partition the input samples. Then, they identify the best classifier for each partition and designate it as the expert for that partition. Unfortunately, current local methods are either computationally expensive and/or perform these two tasks independently of each other. However, feature space partition and algorithm selection are not independent and their optimization should be simultaneous.

In this dissertation, we introduce a new local fusion approach, called Context Extraction for Local Fusion (CELF). CELF was designed to adapt the fusion to different regions of the feature space. It takes advantage of the strength of the different experts and overcome their limitations. First, we describe the baseline CELF algorithm. We formulate a novel objective function that *combines* context identification and multi-algorithm fusion criteria into a joint objective function. The context identification component thrives to partition the input feature space into different clusters (called *contexts*), while the fusion component thrives to learn the optimal fusion parameters within each cluster. Second, we propose several variations of CELF to deal with different applications scenario. In particular, we propose an extension that includes a feature discrimination component (CELF-FD). This version is advantageous when dealing with high dimensional feature spaces and/or when the number of features extracted by the individual algorithms varies significantly. CELF-CA is another extension of CELF that adds a regularization term to the objective function to introduce competition among the clusters and to find the optimal number of clusters in an unsupervised way. CELF-CA starts by partitioning the data into a large number of small clusters. As the algorithm progresses, adjacent clusters compete for data points, and clusters that lose the competition gradually become depleted and vanish. Third, we propose CELF-M that generalizes CELF to support multiple classes data sets.

The baseline CELF and its extensions were formulated to use linear aggregation to combine the output of the different algorithms within each context. For some applications, this can be too restrictive and non-linear fusion may be needed. To address this potential drawback, we propose two other variations of CELF that use non-linear aggregation. The first one is based on Neural Networks (CELF-NN) and the second one is based on Fuzzy Integrals (CELF-FI). The latter one has the desirable property of assigning weights to subsets of classifiers to take into account the interaction between them.

To test a new signature using CELF (or its variants), each algorithm would extract its set of features and assigns a confidence value. Then, the features are used to identify the best context, and the fusion parameters of this context are used to fuse the individual confidence values.

For each variation of CELF, we formulate an objective function, derive the necessary conditions to optimize it, and construct an iterative algorithm. Then we use examples to illustrate the behavior of the algorithm, compare it to global fusion, and highlight its advantages.

We apply our proposed fusion methods to the problem of landmine detection. We use data collected using Ground Penetration Radar (GPR) and Wideband Electromagnetic Induction (WEMI) sensors. We show that CELF (and its variants) can identify meaningful and coherent contexts (e.g. mines of same type, mines buried at the same site, etc.) and that different expert algorithms can be identified for the different contexts. In addition to the landmine detection application, we apply our approaches to semantic video indexing, image database categorization, and phoneme recognition. In all applications, we compare the performance of CELF with standard fusion methods, and show that our approach outperforms all these methods.

TABLE OF CONTENTS

Acknowledgements	iv
Abstract	v
List of Figures	xv
List of Tables	xvii
List of Algorithms	xviii
1 Introduction	1
I Literature Review	
2 Classifier Fusion Methods	8
2.1 Bayesian Fusion	11
2.2 Artificial Neural Networks Fusion	12
2.3 Borda Count Fusion	12
2.3.1 General Approach	13
2.3.2 Weighted Borda Count Approach	14
2.4 Dempster-Shafer Fusion	15
2.4.1 Basic Probability Assignment	15
2.4.2 Belief	16

2.4.3	Plausibility	16
2.4.4	Combination Rule	17
2.5	Decision Template Fusion	18
2.5.1	General Model for DT Classifier Fusion	19
2.5.2	Decision Templates (DTs)	20
2.6	Fuzzy Integral	22
2.7	Local Fusion methods	25
3	Prototype-Based Clustering	28
3.1	The K –Means Algorithm	29
3.2	The Fuzzy C –Means Algorithm	29
3.3	Competitive Agglomeration Algorithm	31
3.4	The Simultaneous Clustering and Attribute Discrimination Algorithm	34
4	Landmine Detection	37
4.1	Sensors for Landmine Detection	38
4.1.1	Ground Penetrating Radar (GPR)	38
4.1.2	Metal Detectors (MD)	40
4.1.3	Electromagnetic Induction (EMI)	41
4.1.4	Infrared Imaging (IR)	43
4.1.5	Bulk Explosive Detection	44
4.2	Landmine Detection Algorithms	44
4.2.1	Landmine Detection using GPR	45
4.2.1.1	Data Preprocessing	45
4.2.1.2	The Edge Histogram Descriptor (EHD) Algorithm . .	46
4.2.1.3	Hidden Markov Model (HMM) Algorithm	49
4.2.1.4	Spectral Correlation Feature (SCF) Algorithm	51
4.2.2	Landmine Detection using WEMI	52

II Proposed Work

5	Context Extraction for Local Fusion	56
5.1	CELF	57
5.2	CELF with Feature Discrimination	67
5.3	CELF with Competitive Agglomeration	76
5.4	CELF for Multi-Class Data	82
6	Non-Linear Local Fusion	89
6.1	Local Fusion with Neural Networks	89
6.2	Local Fusion with Fuzzy Integrals	98
 III Experimental Results		
7	Application to Landmine Detection	108
7.1	Landmine Detection Using a Vehicle Mounted GPR System	109
7.1.1	Data Collection	109
7.1.2	Evaluation Method	111
7.1.3	Motivation for Multi-Algorithm Fusion	113
7.1.4	Multi-Algorithm Fusion	115
7.2	Landmine Detection Using AMDS System	120
7.2.1	Data Collection	120
7.2.2	Motivation for Multi-Sensor Multi-Algorithm Fusion	122
7.2.3	Multi-sensor Multi-algorithm Fusion	125
8	Other Applications of CELF	131
8.1	Semantic Video Indexing	131
8.1.1	Data Collection	131
8.1.2	Low-Level Descriptors and Classifiers	133
8.1.3	Results	134
8.2	Image Database Categorization	136
8.2.1	Features Descriptors	137

8.2.2 Results	138
8.3 Phoneme Recognition	142
9 Conclusions and Future Work	145
9.1 Conclusions	145
9.2 Future Work	146
References	147
A Abbreviations	163
B Symbols	166
Curriculum Vitae	168

LIST OF FIGURES

1.1	Illustration of the proposed Context Extraction for Local Fusion method.	4
2.1	A general architecture for information fusion	9
2.2	Architecture of the decision templates classifier fusion scheme	21
2.3	Architecture of the Context-Dependent Fusion	27
4.1	Wichmann/Niitek vehicle-mounted GPR.	39
4.2	Sample of GPR responses	40
4.3	Measurement cart based EMI sensor.	43
4.4	Extraction of the EHD for a 3-D mine signature.	48
4.5	HMM Feature of a mine signature	50
4.6	Illustration of the HMM-based model architecture.	51
4.7	WEMI response curves and their curve fits.	53
5.1	Architecture of the proposed Context Extraction for Local Fusion method.	64
5.2	Feature and confidence distribution of a sample data.	65
5.3	Clustered samples in the feature space using CELF. The fused confi- dences are shown above each sample.	65
5.4	Cumulative histograms of the confidence values fused using CELF . .	67
5.5	Synthetic data in the combined 2-D feature space.	70
5.6	Classification results of two independent classifiers on the synthetic data.	71

5.7	Fusion results using a global approach.	72
5.8	Local fusion results using CELF-FD.	73
5.9	Local fusion results using a small value of β ($\beta = 5$).	75
5.10	Local fusion results using a large value of β ($\beta = 40$).	76
5.11	Clustered samples in the feature space using CELF-CA	80
5.12	Local fusion results using CELF-CA.	81
5.13	Synthetic data of a 3-class problem in the 2-D feature space.	86
5.14	Classification results of two independent classifiers on the synthetic data.	87
5.15	Local fusion results using CELF-M.	88
6.1	Architecture of the proposed CELF-NN	90
6.2	Block diagram illustrating signal flow for the error back-propagation algorithm.	94
6.3	Synthetic data in the 2-D feature space.	95
6.4	Classification results of two independent classifiers on the synthetic data.	96
6.5	Fusion results using a global Neural Network approach.	96
6.6	Fusion results using the baseline CELF approach.	97
6.7	Local fusion results using CELF-NN.	97
6.8	Synthetic data in the 2-D feature space.	102
6.9	Classification results of three independent classifiers on the synthetic data.	103
6.10	Fusion results using a global fuzzy integral approach.	104
6.11	Fusion results using CELF.	104
6.12	Local fusion results using CELF-FI.	105
6.13	Shapley values of the different classifiers assigned by the global fu- sion and the local fusion (within each cluster).	106
6.14	Interactions indices of the different pair of classifiers assigned by the global fusion and the local fusion (within each cluster).	106

7.1	GPR data collection	109
7.2	NIITEK Radar down-track and cross-track B-scans pairs for 3 alarms	110
7.3	Performance of the different detectors on the entire data collection .	113
7.4	Performance of the detectors on two different sites	114
7.5	Performance of the detectors on mines buried at different depths . . .	114
7.6	Distribution of the confidence values assigned by the different detectors for alarms assigned to context 2	117
7.7	Performance of the different detectors for alarms assigned to context 9	118
7.8	Distribution of the confidence values assigned by the different detectors for alarms assigned to context 10	119
7.9	Performance of the individual detectors and the different fusion methods on (a) site A, (b) site B, (c) site C, (d) site D.	121
7.10	Performance of the individual detectors and the different fusion methods on the entire collection using lane-based cross-validation	122
7.11	NIITEK Autonomous Mine Detection System.	123
7.12	Performance of the individual detectors for different types of mines when: (a) only anti-tank (AT) mines are considered, (b) only anti-personal (AP) mines are considered, (c) only high-metal (HM) mines are considered, (d) only low-metal (LM) mines are considered.	125
7.13	Cumulative histograms of the confidence values assigned by the different detectors to samples within context 1.	127
7.14	ROC curves generated by the different detectors for samples assigned to context 3.	128
7.15	Probability of detection (PD) vs. probability of false alarms (PFA) of (a) average and standard deviation of CELF over 100 runs, (b) the individual detectors and the different fusion methods on the entire collection using 6 fold cross-validation.	129
8.1	Sample keyframes from 6 of the contexts identified by CELF	135
8.2	Weights assigned to each classifier in each context	139

8.3	Representative images from context 2	140
8.4	Representative images from context 6	140
8.5	Comparison of the three individual classifiers that use subsets of the features, with the global and local fusion for the phoneme data set. .	143
8.6	Performance of the three individual k -NN classifiers within each of the 4 contexts generated by CELF-CA.	144

LIST OF TABLES

5.1	Performance of the two classifiers and assigned aggregation weights to each classifier within each cluster	66
5.2	Feature Weights assigned by CELF-FD	74
5.3	Accuracy of each classifier in each cluster and assigned weights by CELF-FD	74
5.4	Accuracy of each classifier in each cluster and assigned weights by CELF-CA	81
7.1	Statistics of dataset 1	111
7.2	Burial depth of mines in dataset 1	111
7.3	Distribution of the alarms among the 10 clusters for one cross validation set	116
7.4	Weights assigned to each classifier in each cluster	116
7.5	Representative alarms from context 2	117
7.6	Representative mines and false alarms from context 9	118
7.7	Representative alarms from context 10	119
7.8	Weights assigned by the global fusion approach	120
7.9	Statistics of the data collection used in our experiment	123
7.10	Burial depth of all objects in the data collection	124
7.11	Distribution of the alarms among the 13 contexts identified by CELF	127

7.12 Weights assigned to each classifier in each context for the entire training data	127
8.1 Feature development and test sets of the TRECVID-2002 collection used in our experiment	132
8.2 Number of shots per semantic concept used in our experiment	132
8.3 MAP values for the individual classifier and the fusion algorithms averaged over the test data	136
8.4 Distribution of the images among the 20 contexts	138
8.5 Accuracy of the individual classifiers and the fusion algorithms	141
8.6 Assigned weights to each classifier in each cluster.	144

LIST OF ALGORITHMS

2.1	Decision Template(Training)	20
2.2	Decision Template(Operation)	21
3.1	K -means	30
3.2	Fuzzy C -Means	31
3.3	Competitive Agglomeration	33
3.4	Simultaneous Clustering and Attribute Discrimination	36
5.1	Context Extraction for Local Fusion (CELF)	61
5.2	CELF with Feature Discrimination (CELF-FD)	69
5.3	CELF with Competitive Agglomeration (CELF-CA)	80
5.4	CELF for Multi-class data (CELF-M)	85
6.1	CELF with Neural Networks (CELF-NN)	94
6.2	CELF with Fuzzy Integrals (CELF-FI)	102

INTRODUCTION

Traditional machine learning and pattern recognition systems use features to describe sensor data and a classifier (also called "expert" or "learner") to determine the true class of a given pattern. However, for complex detection and classification problems involving data with large intra-class variations and noisy inputs, perfect solutions are difficult to achieve, and no single source of information can provide a satisfactory solution. As a result, *combination* of multiple classifiers (or multiple experts) is playing an increasing role in solving these complex pattern recognition problems, and has proven to be a viable alternative to using a single classifier. Classifier combination is mostly a heuristic approach and is based on the idea that classifiers with different methodologies or different features can have complementary information. Thus, if these classifiers cooperate, group decisions should be able to take advantages of the strengths of the individual classifiers, overcome their weaknesses, and achieve a higher accuracy than any individual's.

Over the past few years, a variety of schemes have been proposed for combining multiple classifiers. The most representative approaches include majority vote [61],

Borda count [56], average [96], weighted average [53], Bayesian [77], probabilistic [67], polling methods [66, 100], logistic regression [56], and combination by neural networks [16, 57]. Most of the above approaches assume that the classifier decisions are independent. However, in practice, the outputs of multiple classifiers are usually highly correlated. Therefore, in addition to assigning fusion weights to the individual classifiers, it is desirable to assign weights to subsets of classifiers to take into account the interaction between them. Fusion methods based on the fuzzy integral [114, 42] and Dempster-Shafer theory [79] have this desirable property.

Methods for combining multiple classifiers can be classified into two main categories: *global methods* and *local methods*. Global methods assign a degree of worthiness, that is averaged over the entire training data, to each classifier. Local methods, on the other hand, adapt the classifiers' worthiness to different data subspaces. Intuitively, the use of data-dependent weights, when learned properly, provides higher classification accuracy. This approach requires partitioning the input samples into regions during the training phase. The partition can be defined from the space of individual classifier decisions [83], according to which classifiers agree with each other [56], or by features of the input space [74]. Then, the best classifier for each region is identified and is designated as the expert for this region [122]. Conversely, the partitioning can be defined such that each classifier is an expert in one region [104]. This approach may be more efficient, however, its implementation is not trivial. In the classification phase, the region of an unknown sample is identified, and the output of the classifier responsible for this region is used to make the final decision. Data partition and classifier selection could also be made dynamic during the testing phase [72, 126]. In this case, the accuracy of each classifier (with respect to the training samples) is estimated in local regions of the feature space in the vicinity of the test sample. The most accurate classifier is selected to classify the test sample.

Another approach for building multiple classifiers is based on bagging and boosting. Each classifier is trained using a different subset of the training set. The different subsets are obtained from the original using sampling. The final output is obtained by voting. Bagging specifically refers to the process of generating training subsets by sampling with replacement multiple times. A classifier is trained on each subset. All classifiers are used to classify a test sample. The outputs are combined via voting. Boosting generally refers to a more sequential process of building multiple classifiers on a training set. The general idea is that an initial classifier is trained on the training set. Points for which the initial classifier performs poorly are weighted more strongly in training a different classifier. The process is repeated multiple times in order to try and build a multi-classifier system consisting of classifiers that perform well on subsets of the training set. Boosting can cause problems by over-fitting classifiers on subsets of the training data [41].

This thesis was motivated by the development of a generic framework for context-dependent fusion. Our proposed approach, called Context Extraction for Local Fusion (CELF), is local and thrives to partition the input feature space into different clusters (called *contexts*) and identifies the relevant classifiers for each cluster. Figure 1.1 displays the architecture of the proposed approach. It is composed of two *interactive* components: *context extraction* and *decision fusion*. The context extraction component uses features extracted by the various algorithms (from one or different sensors) and their confidences to partition the training input samples into different contexts. The decision fusion component uses confidence values, assigned by the individual algorithms, to learn the optimal fusion parameters for the different algorithms within each context, based on their relative performance within that context.

The main contribution of this dissertation consists of the development of a novel approach to local fusion that combines context identification and multi-algorithm

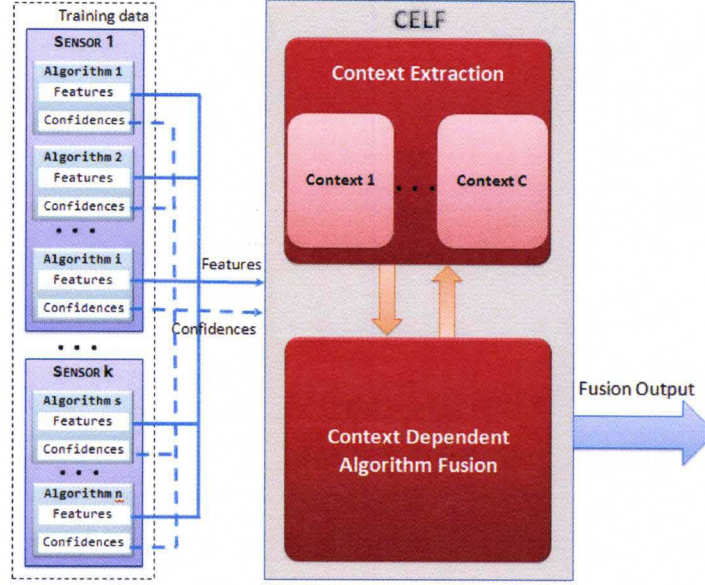


Figure 1.1: Illustration of the proposed Context Extraction for Local Fusion method.

fusion. Our approach is based on formulating a joint objective function that optimizes both criteria simultaneously. We propose several variations that address different practical scenarios. For each variation, we formulate the objective function, derive the necessary conditions to optimize it, and construct an iterative algorithm. In particular, we propose:

1. **CELLF:** This is the baseline algorithm and is based on optimizing an objective function that has two components. The first one is used to partition the feature space into clusters that share similar features and similar response to the different classification algorithms. The second component is used to learn the optimal cluster-dependent aggregation weights to combine the multiple algorithms in a linear way.
2. **CELLF-FD:** The baseline CELF treats all features equally important. This may not be the optimal way especially when working in a high dimensional feature space, or when the algorithms to be combined extract different number of features. To overcome this drawback, we extended the objective function of CELF to include a feature discrimination component. The resulting algorithm,

called CELF with Features Discrimination (CELF-FD), treats the features extracted by each algorithm as one set, and assigns a relevance weight to each one, within each context. This extension allows finding clusters in subspaces of the original sparse and high dimensional feature space.

3. **CELF-CA:** The baseline CELF requires the specification of the number of clusters. However, in most applications, this parameter may be hard to fix. In fact, the optimal number of clusters depends on the distribution in the feature space as well as the performance of the algorithms in the different regions. To address this issue, we extended CELF by adding a regularization term to the objective function. The resulting algorithm, called CELF with Competitive Agglomeration (CELF-CA), starts by partitioning the data into a large number of small clusters. As the algorithm progresses, adjacent clusters compete for data points, and clusters that lose the competition gradually become depleted and vanish.
4. **CELF-M:** The baseline CELF was developed for a two class problem. In order to apply our approach to different applications that involve multiple classes, we generalized CELF to support this aspect. The resulting algorithm, called CELF for Multi-class data (CELF-M), is an extension of CELF that can be used with any number of classes.
5. **CELF-NN:** The baseline CELF algorithm uses a simple linear aggregation to assign weights to the individual classifiers. This may not be the optimal way to combine the algorithms within each context. To make the fusion of the algorithms' decisions more effective, we extended CELF to support non-linear fusion using Neural Networks. The resulting algorithm, called CELF with Neural Networks (CELF-NN), adapts a Neural Network for each context. This is done by optimizing a novel joint objective function that combines context identification and Neural Networks learning.

6. **CELF-FI:** CELF-FI is another extension of CELF to support non-linear aggregation using fuzzy integrals. This extension has the additional desirable property of assigning weights to subsets of classifiers to take into account the interaction between them.
7. **Application to landmine detection:** Recently, a variety of sensors and detection algorithms have been proposed for landmine detection. Extensive testing of these methods has shown that the relative performance of different detectors/sensors can vary significantly depending on the mine type, geographical site, soil and weather conditions, and burial depth. In this thesis, we report results of the application of our proposed fusion methods to data collected using GPR and WEMI sensors. We also provide a comparison of the results of our algorithms with those obtained using common fusion approaches.
8. **Application to other data sets:** Even though our approaches were mainly designed and developed for the landmine detection problem, we have applied them to the problems of semantic video indexing, image database categorization, and phoneme recognition. For each application, we compared their results with other well-known fusion methods.

The rest of this thesis is organized in three main parts. The first part, consisting of Chapters 2, 3, and 4, gives a literature review of related work. Specifically, Chapter 2 gives an overview of some relevant fusion methods. Chapter 3 describes prototype-based clustering, widely used in local fusion approaches, and enumerates some representative clustering algorithms. Chapter 4 provides motivations and the background needed to apply the proposed fusion to the problem of landmine detection. The second part of this thesis consists of Chapters 5, and 6. It describes the proposed CELF approach and its variations and illustrates them with various examples. The last part of this thesis illustrates the experimental results of the proposed fusion methods. Finally, Chapter 9 summarizes the contributions and outlines potential future work.



LITERATURE REVIEW

CLASSIFIER FUSION METHODS

Classification techniques have been successfully applied to many real world problems. It is 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 [93]. Therefore, many methods to combine the decisions of several classifiers were initiated in order to increase the performance of traditional single classifier systems. Classifier combination has produced promising results and research in this domain has increased significantly [75], partly as a result of advances in the classification technology itself. Classifier combination have been applied to various fields of pattern recognition, including character recognition [110], speech recognition [59], and text categorization [9], and have been proved to be superior to single classifier systems both theoretically and experimentally.

Motivated by the classifiers' complementary characteristics, classifier combination can achieve a higher accuracy than individual algorithms by taking advantages of the strengths of the individual classifiers and overcoming their weaknesses. Nonetheless, a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any individual classifier is that the classifiers are both *accurate* and *diverse* [52]. 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, these conditions are assumed to be satisfied and the superiority of classifier combination over a single classifier has been demonstrated experimentally.

Fusion of data/information can be carried out on three levels of abstraction closely connected with the flow of the classification process: *data level fusion*, *feature level fusion*, and *decision level fusion*. *Data level fusion*, also called low level fusion, combines several sources of raw data to produce new raw data that is expected to be more informative and synthetic than any of the single sources. *Feature level fusion*, also called intermediate level fusion, combines various features. These features may come from different raw data sources (e.g. sensors) or from the same raw data. In the latter case, the objective is to find relevant features among available features that might come from several feature extraction methods. *Decision level fusion*, also called high level fusion, combines decisions coming from several experts.

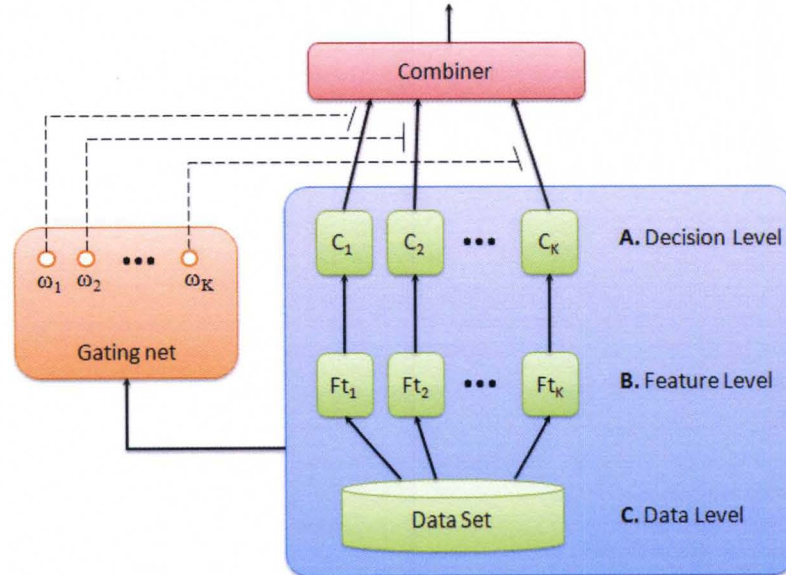


Figure 2.1: A general architecture for information fusion

Figure 2.1 shows a generic architecture for the different levels of information fusion. It illustrates three basic ingredients of fusion (fusion level, gating net, and

combiner). Different combinations of these different ingredients lead to different specific models for expert combination [127].

Methods for combining multiple classifiers can be classified into two main categories: *classifier selection* and *classifier fusion*. Classifier selection methods put an emphasis on the development of the classifier structure. First, these methods identify the single best classifier or a selected group of classifiers and then only their outputs are considered for the final decision or for further processing. This approach assumes that the classifiers are complementary, and that their expertise varies according to the different areas of the feature space. For a given test sample, these methods attempt to predict which classifiers are more likely to be correct. Some of these methods consider the output of only a single classifier to make the final decision [104]. Others, combine the output of multiple "local expert" classifiers [60]. *Classifier fusion* methods operate mainly on the classifiers outputs, and strive to combine the classifiers outputs effectively. This approach assumes that the classifiers are competitive and equally experienced over the entire feature space. For a given test sample, the individual classifiers are applied in parallel, and their outputs are combined in some manner to take a group decision.

Another way to categorize classifier combination methods is based on the way they select or assign weights to the individual classifiers. Some methods are *global* and assign a degree of worthiness, that is averaged over the entire training data, to each classifier. Other methods are *local* and adapt the classifiers' worthiness to different data subspaces. Intuitively, the use of data-dependent weights, when learned properly, provides higher classification accuracy.

In the rest of this chapter, several classifier fusion methods are briefly overviewed. Global fusion method are described first. Then, some local fusion are introduced in Section 2.7.

Let $\mathcal{X} = \{x_j | j = 1, \dots, N\}$ be a set of N training observations to be classified into one of the M classes: c_1, \dots, c_M , and Let e_1, \dots, e_K denote K classifiers. Each classifier k

generates confidence values, $\mathcal{Y}^k = \{y_j^k | j = 1, \dots, N\}$.

2.1 Bayesian Fusion

The Bayesian methods can be applied to the classifier fusion under the condition that the classifiers' outputs are expressed by posterior probabilities. Effective combination of given likelihoods is also a probability of the same type, which is expected to be higher than the probability of the best individual classifier for the correct class [106].

Let x be an input sample that has been processed by K classifiers: e_1, \dots, e_K to be classified into one of the M classes: c_1, \dots, c_M . Let $P_k(x \in c_i | x)$ be the posterior probability given by classifier k , $k = 1, \dots, K$, that x comes from class i . One simple way to fuse the outputs of the K classifiers is to compute the average of their posterior probability. That is,

$$P_E(x \in c_i | x) = \frac{1}{K} \sum_{k=1}^K P_k(x \in c_i | x), \quad i = 1, \dots, M. \quad (2.1)$$

Such decision, based on the newly estimated posterior probabilities, is called average Bayes classifier. This approach can be applied for the Bayes classifiers. For other non-Bayesian classifiers, several other methods to estimate the posterior probability could be used. For example, for the k -NN classifier the transformation can be computed using

$$P_k(x \in c_i | x) = \frac{k_i}{k_{NN}}, \quad (2.2)$$

where k_i denotes the number of prototype samples from class c_i out of all k_{NN} nearest prototype samples.

Bayesian fusion methods have been used in various applications [106]. Although they have proven to be effective in some cases, Bayesian models cannot handle correlated information coming from multiple sources because this approach assumes

that the classifier decisions are independent, which is generally not true.

2.2 Artificial Neural Networks Fusion

Artificial Neural Networks (ANN) have been applied successfully to many pattern classification problems. They have also shown promise to the classifier fusion problem. A neural network designed for the purpose of classifier fusion should have one crisp output or alternatively a number of soft outputs equal to the number of classes if there is a need to produce qualitative assignment values to each class. The input of such a network should be associated with the individual classifier outputs [16, 57].

Given a neural network that performs a mapping of K individual classifiers outputs (taken as input) into M outputs corresponding to the level of assignment to each of the M classes. If a crisp decision is required, the output with the highest value is chosen. The input-output mapping in ANN is determined via an iterative learning process. During the learning stage, weights between each pair of connected nodes of the network are adapted in such a way as to minimize the difference between the actual network output and the desired output.

It is quite common for the output of a set of ANNs to be combined using another ANN. Following this approach neural networks working as a mixture can be expanded to a higher dimension by fusing several neural networks [17] or arranging them in an efficient ANN-like structure [62].

2.3 Borda Count Fusion

The Borda Count is a single-winner election method in which voters rank candidates in order of preference [19]. The Borda Count determines the winner of an election by giving each candidate a certain number of points corresponding to the position in which he or she is ranked by each voter. Once all votes have been counted the

candidate with the most points is the winner. Because it sometimes elects broadly acceptable candidates, rather than those preferred by the majority, the Borda Count is often described as a consensus-based electoral system, rather than a majoritarian one.

The Borda Count has been used for fusing the results of classifiers for the task of handwriting recognition [56, 43, 120]. In particular, Ho et al. [56] presented a weighted Borda Count technique for this application that uses logistic regression to identify classifier weights by comparing the ranking results of each classifier with a best ranking derived by applying several different independent classification algorithms. Gader et al. [43] employed a method in which the Borda weights are determined dynamically based on a match confidence between the object and a lexicon string. Van Erp and Schomaker [120] compare the performance of Median Borda, a variant of the Borda Count in which the median rank (rather than sum or average) is used, and Nanson's [87] election procedure (an iterative Borda scheme that deletes the candidate ranked lowest in each successive iteration).

2.3.1 General Approach

One approach to combine multiple classifiers with a supervised learning system using rank weighting is to consider each discrimination algorithm to be a voter, and each observation in the training set to be a candidate. Given K algorithms e_1, \dots, e_K and N training samples x_1, \dots, x_N , each algorithm maps samples to their confidence values, elements of \mathbb{R} . The number of points given to candidates for each ranking is determined by the number of candidates standing in the voting. For each algorithm e_i and for each candidate x_j , a rank $r_i(x_j)$ is assigned to x_j if $e_i(x_j)$ has a confidence value greater than exactly $r_i(x_j) - 1$ other candidate alarms. In other words, a candidate will receive N points for a first preference, $N - 1$ points for a second preference, $N - 2$ for a third, and so on. Thus, r_i is a map from the confidence values assigned by algorithm e_i into the set $\{1, \dots, N\}$. The final result of applying

the Borda Count to x_j is expressed by the following expression:

$$r(x_j) = \frac{1}{KN} \sum_{i=1}^K r_i(x_j) \quad (2.3)$$

Note that this result is normalized to yield a value in the range $[0, 1]$.

2.3.2 Weighted Borda Count Approach

If there are evidences that algorithms e_i and e_j have differing predictive abilities, say e_i is more likely to be correct than e_j , then one should use this prior information and assigns weights w_i and w_j to these algorithms, such that $w_i > w_j$. In general, a weighted Borda scheme assigns a weight w_k to each algorithm e_k such that

$$\sum_{k=1}^K w_k = 1, \quad (2.4)$$

and the weighted Borda Count assigns confidence r to x_j as follows:

$$r(x_j) = \frac{1}{KN} \sum_{k=1}^K w_k r_i(x_j), \quad (2.5)$$

Borda fusion has been applied to landmine detection [123], and (in a different way) to handwriting recognition [42], and fusion of social choices (voting, evaluation, etc.).

The main advantages of the Borda based fusion is that it makes no assumptions about the underlying distributions of the confidence value assignments. In addition, it maps each of the confidence distribution to a uniform distribution, thus providing a reasonable method for combining decision statistics.

2.4 Dempster-Shafer Fusion

Dempster-Shafer theory (DST) is a mathematical theory of evidence, based on belief functions and plausible reasoning, and is used to combine separate pieces of information (evidence) to calculate the probability of an event [108]. In a finite discrete space, DST can be interpreted as a generalization of probability theory where probabilities are assigned to sets as opposed to mutually exclusive singletons. In traditional probability theory, evidence is associated with only one possible event. In DST, evidence can be associated with multiple possible events, e.g., sets of events. As a result, evidence in DST can be meaningful at a higher level of abstraction without having to resort to assumptions about the events within the evidential set. One of the most important features of Dempster-Shafer theory is that the model is designed to cope with varying levels of precision regarding the information and no further assumptions are needed to represent the information.

Let $\Theta = \{\theta_1, \dots, \theta_K\}$ be a finite set of possible hypotheses. This set is referred to as *the frame of discernment*, and its power set¹ is denoted by $\mathbb{P}(\Theta)$. There are three important functions in Dempster-Shafer theory: the *basic belief assignment function* (BBA or m), the *Belief function* (Bel), and the *Plausibility function* (Pl).

2.4.1 Basic Probability Assignment

The basic probability assignment, represented by m , defines a mapping of the power set to the interval between 0 and 1, such that

$$m : \mathbb{P}(\Theta) \rightarrow [0, 1] \quad (2.6)$$

$$m(\emptyset) = 0 \quad (2.7)$$

$$\sum_{A \in \mathbb{P}(\Theta)} m(A) = 1 \quad (2.8)$$

¹The power set, $\mathbb{P}(\Theta)$, is the set of all possible sub-sets of Θ , including the empty set.

The value of the basic probability assignment for a given set \mathcal{A} expresses the proportion of all relevant and available evidence that supports the claim that a particular element of Θ belongs to the set \mathcal{A} but to no particular subset of \mathcal{A} . Any further evidence on the subsets of \mathcal{A} would be presented by another basic probability assignment.

From the basic probability assignment, the upper and lower bounds of an interval can be defined. This interval contains the precise probability of a set of interest (in the classical sense) and is bounded by two nonadditive continuous measures called *Belief* and *Plausibility*.

2.4.2 Belief

The Belief of a set \mathcal{A} is defined as the sum of all the basic probability assignments of all its subsets. It is interpreted as a measure of the total belief committed to \mathcal{A} , and is defined by:

$$Bel(\mathcal{A}) = \sum_{\mathcal{B} \subseteq \mathcal{A}} m(\mathcal{B}) \quad (2.9)$$

We can consider a basic belief assignment as a generalization of a probability density function whereas a belief function is a generalization of a probability function. It can be easily verified that the belief in some hypothesis \mathcal{A} and the belief in its negation $\overline{\mathcal{A}}$ do not necessarily sum to 1, which is a major difference with probability theory.

2.4.3 Plausibility

The Plausibility of a set \mathcal{A} is defined as the sum of all the basic probability assignments of the sets that intersect \mathcal{A} [69]. It defines the extent one fails to doubt \mathcal{A} .

$$Pl(\mathcal{A}) = \sum_{\mathcal{B} \cap \mathcal{A} \neq \emptyset} m(\mathcal{B}) \quad (2.10)$$

$$= 1 - Bel(\overline{\mathcal{A}}) \quad (2.11)$$

The two measures, Belief and Plausibility are nonadditive. This can be interpreted as it is not required for the sum of all the Belief measures to be 1 and similarly for the sum of the Plausibility measures.

2.4.4 Combination Rule

Dempster's combination rule combines multiple belief functions through their basic probability assignments. These belief functions are defined on the same frame of discernment, but are based on independent arguments or bodies of evidence. The issue of independence is a critical factor when combining evidence and is an important research subject in Dempster-Shafer theory. The Dempster rule of combination is purely a conjunctive operation (AND). Specifically, the combination (called the joint m_{12}) is calculated from the aggregation of two basic probability assignment's m_1 and m_2 using

$$m_{12} = \frac{\sum_{\mathcal{B} \cap \mathcal{C} = \mathcal{A}} m_1(\mathcal{B})m_2(\mathcal{C})}{1 - K}, \quad (2.12)$$

where K represents basic probability mass associated with conflict. This is determined by summing the products of the basic probability assignment's of all sets where the intersection is null. That is,

$$K = \sum_{\mathcal{B} \cap \mathcal{C} = \emptyset} m_1(\mathcal{B})m_2(\mathcal{C}). \quad (2.13)$$

The above rule is commutative, associative, but not idempotent or continuous.

The denominator in (2.12) is a normalization factor. It has the effect of completely ignoring conflict and attributing any probability mass associated with conflict to the null set. Consequently, this operation will yield counterintuitive results in the face of significant conflict in certain contexts.

One of the computational advantages of the Dempster-Shafer framework is that priors and conditionals need not be specified, unlike Bayesian methods which often use a symmetry (minimax error) argument to assign prior probabilities to random variables. However, any information contained in the missing priors and conditionals is not used in the Dempster-Shafer framework unless it can be obtained indirectly and arguably is then available for calculation using Bayes equations. Finally, DST allows one to specify a degree of ignorance in this situation instead of being forced to supply prior probabilities which add to unity.

2.5 Decision Template Fusion

Decision Template (DT) [74] is a robust classifier fusion scheme that combines classifier outputs by comparing them to a characteristic template for each class. DT fusion uses all classifier outputs to calculate the final support for each class, which is in sharp contrast to most other fusion methods which use only the support for that particular class to make their decision.

In many cases, the classifier output is a M -dimensional vector with support to the M classes, i.e.,

$$e_i(x) = [d_{i,1}(x), \dots, d_{i,m}(x)]^T, \quad i = 1, \dots, K, \quad (2.14)$$

where $\mathcal{E} = \{e_1, e_2, \dots, e_K\}$ is a set of classifiers and $\mathcal{C} = \{c_1, \dots, c_M\}$ is a set of class labels. Without loss of generality, for $i = 1, \dots, K$ and $j = 1, \dots, M$, $d_{i,j}(x)$ are restricted to the interval $[0,1]$, and the classifier outputs are called 'soft labels'. Thus, $d_{i,j}(x)$ is the degree of 'support' given by classifier i to the hypothesis that x comes from class c_j (most often an estimate of the posterior probability $P(c_j|x)$). Classifiers

combination can be defined as a function of the K classifier outputs $e_1(x), \dots, e_K(x)$, i.e.:

$$\hat{e}(x) = f(e_1(x), \dots, e_K(x)) \quad (2.15)$$

DT generates a vector with final degrees of support for the M classes as a soft label for x , denoted

$$\hat{e}(x) = [\mu_1(x), \dots, \mu_M(x)]^T, \quad (2.16)$$

If a crisp class label is needed, it can use the maximum membership rule; i.e assign x to class c_s if $\mu_s(x) \geq \mu_t(x)$, for all $t = 1, \dots, M$.

2.5.1 General Model for DT Classifier Fusion

The DT Classifier fusion assumes that all classifiers are trained over the whole feature space, and are thereby considered as competitive rather than complementary [128]. This approach treats the classifiers' outputs as input to a second-level classifier in some intermediate feature space, and designs a new classifier for the second (combination) level. In particular, the classifier outputs can be organized in a *decision profile* [76] matrix as

$$DP(x) = \begin{bmatrix} d_{1,1}(x) & \cdots & d_{1,j}(x) & \cdots & d_{1,M}(x) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{i,1}(x) & \cdots & d_{i,j}(x) & \cdots & d_{i,M}(x) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{K,1}(x) & \cdots & d_{K,j}(x) & \cdots & d_{K,M}(x) \end{bmatrix} \quad (2.17)$$

The entries in $DP(x)$ are the *intermediate features space*. The DT method can build a minimum-error classifier by replacing the problem of estimating $P(w_i|x)$ with one of estimating $P(w_i|e_1(x), \dots, e_K(x))$, or more compactly, $P(w_i|DP(x))$. Thus, the

initial feature space with n features, \mathbb{R}^n , is transformed into a new space with $K \times M$ features. This treatment of the combination problem underpins the schemes in [56, 58, 63]. In a way, this idea is akin to support vector machines approach where the initial feature space is transformed in a new (generally higher dimensional) space and the classifier is built in that new space [121]. However, in the model here, the intermediate feature space has a special context-related structure on which the combination model is based [76].

2.5.2 Decision Templates (DTs)

The decision template for class c_i , denoted DT_i is the centroid of this class in the intermediate feature space. DT_i can be regarded as the expected support for class c_i . The support for class c_i offered by the combination of the K classifiers, $\mu_i(x)$, is then found by measuring the similarity between the current $DP(x)$ and DT_i . In [76], the authors treat $DP(x)$ and DT_i as two fuzzy sets, defined over the set of intermediate features, and use measures of similarity from fuzzy set theory. The following algorithms describes the main steps of the DT training and testing phases.

Algorithm 2.1 Decision Template(Training)

- 1: **for** $i = 1, \dots, M$ **do**
- 2: Calculate the mean of the decision profiles $DP(x_j)$ of all member of c_i from the data set \mathcal{X} .
- 3: Let DT_i be the mean of decision template :

$$DT_i = \frac{1}{N_i} \sum_{\substack{x_j \in c_i \\ x_j \in \mathcal{X}}} DP(x_j), \quad (2.18)$$

where N_i is the number of elements of \mathcal{X} from c_i .

- 4: **end for**
 - 5: **return** DT_1, \dots, DT_M
-

If the classifier outputs are some estimates of the posterior probabilities $P(c_j|x)$, $j = 1, \dots, M$, the decision template is an unbiased estimate of the expectation of the

Algorithm 2.2 Decision Template(Operation)

- 1: Given the input $x \in \mathbb{R}^n$ construct $DP(x)$ as in (2.17).
- 2: Calculate the distance between $DP(x)$ and each DT_i , $i = 1, \dots, M$.

$$d_E(DP(x), DT_i) = \sum_{j=1}^M \sum_{k=1}^K (d_{k,j}(x) - dt_i(k, j))^2, \quad (2.19)$$

where $dt_i(k, j)$ is the $(k, j)^{th}$ entry in decision template

- 3: Calculate the components of the soft label of x by:

$$\mu_i(x) = 1 - \frac{1}{M \cdot K} d_E(DP(x), DT_i), \quad (2.20)$$

$K \times M$ dimensional random variable $DP(x)$ given that the true class is c_i . Therefore, assessing the similarity between the actually occurred matrix of outputs $DP(x)$ and the expected one for c_i is a reasonable classification strategy.

Figure 2.2 shows the architecture of the DT approach. The decision templates are calculated in advance using (2.18).

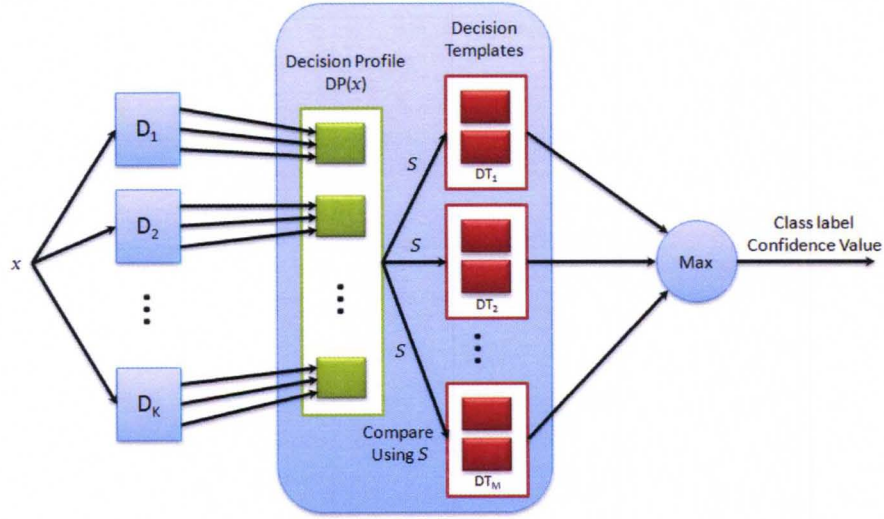


Figure 2.2: Architecture of the decision templates classifier fusion scheme

2.6 Fuzzy Integral

The fuzzy integral has been investigated extensively for information fusion [114, 18, 45, 5]. This integral defines a family of generally nonlinear aggregation operators on some function of the algorithm confidence values. The aggregation operator is defined by the fuzzy integral with respect to a non-additive fuzzy measure. As used here, fuzzy measures are real-valued functions defined on sets of algorithms. Thus, the fuzzy integral is a mathematical construct that can be used to optimize the aggregation operator for a specific fusion application.

Definition 2.1 (Fuzzy measure). Let $\mathcal{A} = \{a_1, \dots, a_K\}$ be a finite set. A fuzzy measure, g , is a real valued function defined on the power set of \mathcal{A} , $\mathbb{P}(\mathcal{A})$, with range $[0,1]$, satisfying the following properties:

1. $g(\emptyset) = 0$ and $g(\mathcal{A}) = 1$.
2. given $A, B \in \mathcal{A}$, if $A \subseteq B$ then $g(A) \leq g(B)$.

For the purpose of fusion, the set \mathcal{A} is considered to contain the names of different information sources (algorithms), and for a subset $A \subseteq \mathcal{A}$, $g(A)$ is considered to be the degree of *worthiness* of this subset of information. Many fuzzy measures were introduced in the literature [80, 68, 47, 20, 103]. In this work, we limit our study to the Sugeno measures which are a special class of fuzzy measures [113].

Definition 2.2 (Sugeno measure). A fuzzy measure g is called a Sugeno measure if it satisfies the following additional property: for all $A, B \subseteq \mathcal{A}$ with $A \cap B = \emptyset$, there exists $\lambda > -1$ such that

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B). \quad (2.21)$$

It can be shown that a set function satisfying the conditions in Definition 2.2 is a fuzzy measure. In particular, equation (2.21) implicitly imposes the monotonicity

constraints on the Sugeno measures. The value of λ can be determined for a finite set \mathcal{A} using (2.21) and the facts that $\mathcal{A} = \bigcup_{i=1}^K \{a_k\}$ and $g(\mathcal{A}) = 1$, which leads to solving the following equation for λ :

$$1 + \lambda = \prod_{k=1}^K (1 + \lambda g(\{a_k\})), \text{ and } \lambda > -1. \quad (2.22)$$

Equation (2.22) is a polynomial in λ of degree $K - 1$, and can be easily solved numerically [80, 68].

The discrete Choquet integral [99] has proved to be useful tool to fuse evidence supplied by different information sources.

Definition 2.3 (Choquet integral). Let $e : \mathcal{A} \rightarrow [0, 1]$. Let $\{a_{\sigma(1)}, \dots, a_{\sigma(K)}\}$ denote the reordering of the set \mathcal{A} such that $e(a_{\sigma(1)}) \leq \dots \leq e(a_{\sigma(K)})$, and let A_k be a collection of subsets defined by $A_k = \{a_{\sigma(k)}, \dots, a_{\sigma(K)}\}$. The discrete Choquet integral of e with respect to g on \mathcal{A} is defined as

$$C_g(e) = \sum_{k=1}^K [e(a_{\sigma(k)}) - e(a_{\sigma(k-1)})] \cdot g(A_k), \quad (2.23)$$

or

$$C_g(e) = \sum_{k=1}^K [g(A_k) - g(A_{k+1})] \cdot e(a_{\sigma(k)}), \quad (2.24)$$

where $e(a_{\sigma(0)}) = 0$ and $A_{k+1} \equiv \emptyset$.

The function e is a particular instance of the partial support (evidence) supplied by each information source in determining the confidence in an underlying hypothesis. The integral fuses this objective support with the degree of *worthiness* of the various subsets of the information sources. The analysis of the coefficients of the fuzzy measure can be performed by the calculation of the Shapley values [97].

Definition 2.4 (Shapley value). The Shapley value of g is a K -dimensional vector $\phi_g = [\phi_g(a_1), \dots, \phi_g(a_K)]$, defined by

$$\phi_g(a_k) = \sum_{A \subseteq \mathcal{A} \setminus \{a_k\}} \gamma_{\mathcal{A}}(A) (g(A \cup \{a_k\}) - g(A)) \quad (2.25)$$

with

$$\gamma_{\mathcal{A}}(A) = \frac{(|\mathcal{A}| - |A| - 1)! \times |A|!}{|\mathcal{A}|!}, \quad (2.26)$$

where $|A|$ indicates the cardinality of A .

The Shapley value, $\phi_g(a_k)$, with respect to a fuzzy measure g , represents the global importance of each source a_k with respect to any subset A not containing a_k . It is confined to the interval $[0, 1]$. A value close to zero indicates that the k^{th} algorithm is not relevant for the given data, while a value close to 1 indicates that the given algorithm is highly relevant for the given data. It can be proven that $\sum_{k=1}^K \phi_g(a_k) = 1$.

Another way to analyze the coefficient of the fuzzy measure is to compute the interaction index $I_g(a_k, a_l)$ [98, 46] between pairs of information sources.

Definition 2.5 (Interaction index). The mean interaction index between 2 sources k and l with respect to g is defined by

$$I_g(a_k, a_l) = \sum_{A \subseteq \mathcal{A} \setminus \{a_k, a_l\}} \xi_{\mathcal{A}}(A) (g(A \cup \{a_k, a_l\}) - g(A \cup \{a_k\}) - g(A \cup \{a_l\}) + g(A)) \quad (2.27)$$

with

$$\xi_{\mathcal{A}}(A) = \frac{(|\mathcal{A}| - |A| - 2)! \times |A|!}{(|\mathcal{A}| - 1)!}. \quad (2.28)$$

A positive value of the interaction index ($I_g(a_k, a_l) > 0$) induces a *conjunctive* behavior in aggregation. That is, algorithms k and l have to be both satisfied in order to have a good global score. On the other hand, a negative value of the interaction index ($I_g(a_k, a_l) < 0$) induces a *disjunctive* behavior in aggregation. That is, it suffices to satisfy one of the two algorithms, k or l , to have a good global score. A null

value of the interaction index ($I_g(a_k, a_l) = 0$) induces no interaction. In this case, a linear aggregation is sufficient to have a good global score.

2.7 Local Fusion methods

Global fusion methods outlined in the previous sections assign a degree of worthiness, that is averaged over the entire training data, to each classifier. An alternative approach, that is local, adapts the classifiers' worthiness to different data subspaces. Intuitively, the use of data-dependent weights, when learned properly, provides higher classification accuracy.

In [126], Woods et al. proposed a method called *dynamic classifier selection by local accuracy*. The basic concept of this method is to estimate each classifier's accuracy in local regions of the 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* [73], 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 [81] proposed a modified version of the clustering-and-selection method, that tried to take advantage of the class labels. For 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 drawback of this method is that each classifier should maintain its own partition, which makes the decision process memory and computational time-intensive.

Frigui et al. [50] proposed a local fusion method called *Context-Dependent Fusion* (CDF). CDF attempts to partition the feature space into regions that share common attributes and adapts the fusion to the different regions. The training part of CDF has two main components: *Context Extraction* and *Algorithm Fusion*. In Context Extraction, the features used by the different classifiers are combined, and a clustering algorithm is used to partition the training signature into groups of similar signatures, or contexts, and learn the relevant features within each context. Here, it is assumed that signatures that have similar response to different algorithms share some common features, and would be assigned to the same cluster. The Algorithm Fusion component assigns an aggregation weight to each detector in each context based on its relative performance within the context. To test a new signature using CDF, each detector would extract its set of features and assigns a confidence value. Then, the features are used to identify the best context, and the aggregation weights of this context are used to fuse the individual confidence values. Figure 2.3 displays the architecture of the training and testing phases of the CDF scheme. This figure highlights the two main components of the training phase, namely, context extraction and algorithm fusion. In context extraction, the features extracted by the different algorithms (from different sensors) are combined, and a clustering algorithm is used to partition the training signatures into groups of similar signatures, or contexts, and learn the relevant features within each context. The algorithm fusion component assigns an aggregation weight to each detector in each context based on its relative performance within the context.

Local fusion method requires partitioning the input samples into regions during the training phase. Then, the best classifier for each region is identified and is designated as the expert for this region. These two processes are often performed independently of each other. However, these two tasks are not independent, and their optimization should be combined.

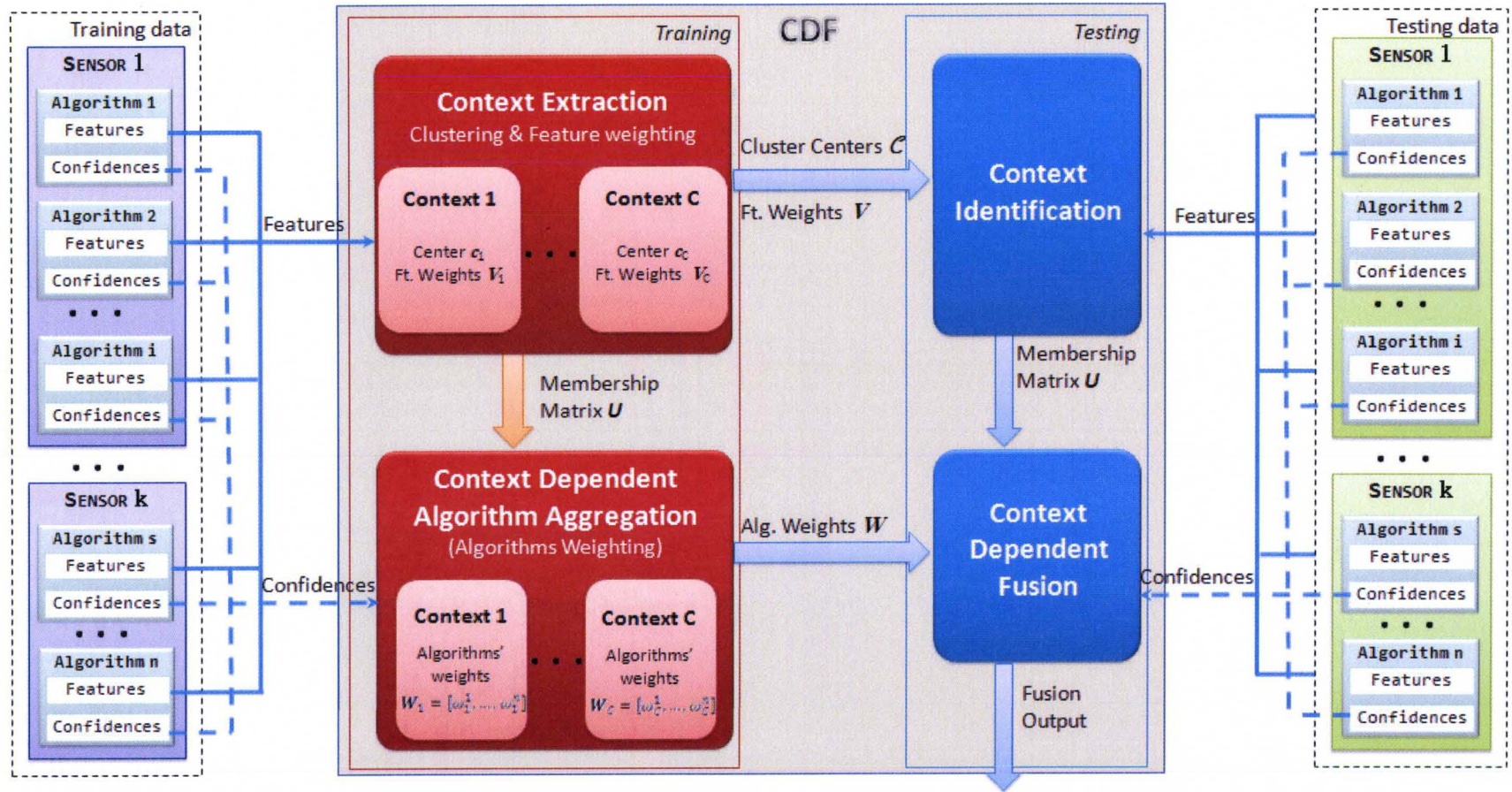


Figure 2.3: Architecture of the Context-Dependent Fusion

PROTOTYPE-BASED CLUSTERING

Clustering is an effective technique for exploratory data analysis, and has been studied extensively in statistics [64], pattern recognition [21, 35], and machine learning [90, 22]. Clustering aims at partitioning unlabeled data set into different groups or clusters, such that members of the same cluster are as similar as possible, while members of different clusters are as dissimilar as possible. In local fusion work, clustering methods are needed to partition the input feature space so that the multiple classifiers' worthiness could be adapted to different data regions.

In this chapter, we focus on prototype-based clustering methods (also called objective function-driven). Prototype-based clustering methods build partitions (clusters) of data sets and extract a prototype for each cluster by optimizing an *objective function*. These methods have the advantage of being able to incorporate knowledge about the global shape or size of clusters by using appropriate prototypes and distance measures in the objective function [21, 8, 71, 70].

In the following, we outline four prototype-based clustering algorithms that are highly relevant to our research area; namely, the K -Means [82], the Fuzzy C -Means (FCM) [8], the Competitive Agglomeration(CA) [30], and the Simultaneous Clustering and Attribute Discrimination (SCAD) [32] algorithms.

Let $\mathcal{X} = \{\mathbf{x}_j \in \mathbb{R}^n | j = 1, \dots, N\}$ be a set of feature vectors. Let $\mathcal{C} = (\mathbf{c}_1, \dots, \mathbf{c}_C)$ represents a set of C prototypes; each of which characterizes one of the C clusters.

3.1 The K –Means Algorithm

The K –Means [82] algorithm, is one of the oldest and most widely used clustering algorithms. It minimizes the following objective function

$$J_{K-Means} = \sum_{i=1}^C \sum_{\mathbf{x}_j \in \mathcal{X}_i} d^2(\mathbf{x}_j, \mathbf{c}_i), \quad (3.1)$$

where $d^2(\mathbf{x}_j, \mathbf{c}_i) = d_{ij}^2$ represents the distance from a feature point \mathbf{x}_j to the prototype \mathbf{c}_i , and \mathcal{X}_i is the set of points assigned to the i^{th} cluster and is given by

$$\mathcal{X}_i = \{\mathbf{x}_j \in \mathcal{X} | d_{ij}^2 = \min_{k=1}^C d_{kj}^2\} \quad (3.2)$$

The optimal prototype parameters \mathbf{c}_i of the i^{th} cluster are derived by setting $\frac{\partial J}{\partial \mathbf{c}_i} = \mathbf{0}$. For instance, if d_{ij}^2 is the squared Euclidean distance, $d_{ij}^2 = \|\mathbf{x}_j - \mathbf{c}_i\|^2$, then the center \mathbf{c}_i is given by

$$\mathbf{c}_i = \frac{\sum_{\mathbf{x}_j \in \mathcal{X}_i} \mathbf{x}_j}{N}. \quad (3.3)$$

The K –Means algorithm consists of alternating updates of the centers using (3.3) and the partition using (3.2), until convergence or when a maximum number of iterations is reached. The K –means is formally described by Algorithm 3.1.

3.2 The Fuzzy C –Means Algorithm

Since Zadeh [131] proposed fuzzy sets that produced the idea of partial membership functions, fuzzy clustering has been widely studied and applied to various

Algorithm 3.1 *K*–means

Inputs: \mathcal{X} : the features of the data samples.

C : the number of clusters.

Outputs: \mathbf{c} : the cluster centers.

- 1: Select C points as initial centroids.
 - 2: **repeat**
 - 3: Form C clusters by assigning each point to its closest centroid using (3.2).
 - 4: Recompute the centroid of each cluster using (3.3).
 - 5: **until** centroids stabilize
 - 6: **return** \mathbf{c}
-

areas. The Fuzzy C –Means (FCM) [8] is a simple but powerful clustering method that uses the concept of fuzzy sets. The FCM optimizes the following objective function:

$$J_{FCM} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2, \quad (3.4)$$

subject to

$$u_{ij} \in [0, 1] \quad \forall i, j \text{ and } \sum_{i=1}^C u_{ij} = 1 \quad \forall j. \quad (3.5)$$

In (3.4), N is the number of data points, C is the number of clusters, \mathbf{c}_i is the center of the i^{th} cluster, u_{ij} is the membership of the j^{th} point, \mathbf{x}_j , in the i^{th} cluster, and m is a constant called the fuzzifier.

Minimization of J_{FCM} with respect to the centers, \mathbf{c}_i , yields

$$\mathbf{c}_i = \frac{\sum_{j=1}^N u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^m}. \quad (3.6)$$

Minimization of J_{FCM} with respect to the membership degree, u_{ij} , yields

$$u_{ij} = \frac{1}{\sum_{l=1}^C (d_{ij}/d_{lj})^{1/(m-1)}}, \quad (3.7)$$

where

$$d_{ij} = \|\mathbf{x}_j - \mathbf{c}_i\|^2. \quad (3.8)$$

The FCM algorithm is formally described by Algorithm 3.2.

Algorithm 3.2 Fuzzy C –Means

Inputs: \mathcal{X} : the features of the data samples.

C : the number of clusters.

m : the fuzzifier, $m \in (1, +\infty)$.

Outputs: U : the fuzzy membership matrix of the data samples.

c : the cluster centers.

- 1: Initialize U .
 - 2: **repeat**
 - 3: Update c using (3.6).
 - 4: Update U using (3.7).
 - 5: **until** centers stabilize
 - 6: **return** c, U .
-

3.3 Competitive Agglomeration Algorithm

The objective function in (3.4), which is essentially the sum of (fuzzy) intra-cluster distances, has a monotonic tendency with respect to the number of clusters, C , and has the minimum value of zero when $C = N$. Therefore, it is not useful for the automatic determination of the "optimum" number of clusters, and C has to be specified a priori. The Competitive Agglomeration (CA) algorithm [30] overcomes this drawback by adding a second regularization term to prevent over fitting the data set with too many prototypes. The CA algorithm starts by partitioning the data set into a large number of small clusters. As the algorithm progresses, adjacent clusters compete for data points, and clusters that lose the competition gradually vanish. The CA algorithm minimizes the following objective function

$$J_{CA} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^2 d_{ij}^2 - \alpha \sum_{i=1}^C \left(\sum_{j=1}^N u_{ij} \right)^2, \quad (3.9)$$

subject to

$$\sum_{i=1}^C u_{ij} = 1, \quad \forall j. \quad (3.10)$$

It should be noted that the number of clusters, C , in (3.9) is dynamically updated in the CA algorithm.

The first term in (3.9) controls the shape and size of the clusters and encourages partitions with many clusters. The second term, on the other hand, penalizes solutions with a large number of clusters and encourages the agglomeration of clusters. When both components are combined and α is chosen properly, the final partition will minimize the sum of intra-cluster distances, while partitioning the data set into the optimal number of clusters [30].

Minimization of J_{CA} with respect to the membership degree, u_{ij} , yields

$$u_{ij} = u_{ij}^{FCM} + u_{ij}^{Bias}, \quad (3.11)$$

where

$$u_{ij}^{FCM} = \left(\sum_{k=1}^C \frac{d_{ij}^2}{d_{kj}^2} \right)^{-1}, \quad (3.12)$$

and

$$u_{ij}^{Bias} = \frac{\alpha}{d_{ij}^2} (N_i - \bar{N}_j). \quad (3.13)$$

In (3.13),

$$N_i = \sum_{j=1}^N u_{ij} \quad (3.14)$$

is the cardinality of cluster i , and

$$\bar{N}_j = \left(\sum_{k=1}^C \frac{N_k}{d_{kj}^2} \right) / \left(\sum_{k=1}^C \frac{1}{d_{kj}^2} \right) \quad (3.15)$$

is a weighted average of the cardinalities of all clusters. The first term in (3.11) is the membership term in the FCM algorithm (see equation (3.7)) which takes into account only the relative distances of the feature point to all clusters. The second term is a signed bias term which allows good clusters to agglomerate and spurious clusters to disintegrate.

Minimization of J_{CA} with respect to the prototypes leads to

$$\mathbf{c}_i = \frac{\sum_{j=1}^N u_{ij}^2 \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^2}. \quad (3.16)$$

The value of the agglomeration constant α in (3.9) needs to be initially small to encourage the formation of small clusters. Then, it should be increased gradually to promote agglomeration. After a few iterations, when the number of clusters becomes close to the "optimum", the value of α should again decay slowly to allow the algorithm to converge.

The Competitive Agglomeration is formally described by Algorithm 3.3.

Algorithm 3.3 Competitive Agglomeration

Inputs: \mathcal{X} : the features of the data samples.

C_{max} : the maximum number of clusters.

ϵ : a given threshold.

Outputs: \mathbf{c} : the cluster centers.

- 1: Fix the maximum number of clusters $C = C_{max}$;
 - 2: Initialize \mathbf{U} .
 - 3: Compute the initial cardinalities N_i for $1 \leq i \leq C$ using (3.14);
 - 4: **repeat**
 - 5: Update the partition matrix \mathbf{U} using (3.11);
 - 6: Compute the cardinalities N_i for $1 \leq i \leq C$ using (3.14);
 - 7: **if** $N_i < \epsilon$ **then**
 - 8: discard cluster i ;
 - 9: **end if**
 - 10: Update the number of clusters C ;
 - 11: Update the centers using (3.16);
 - 12: **until** centers stabilize
 - 13: **return** \mathbf{c}
-

3.4 The Simultaneous Clustering and Attribute Discrimination Algorithm

Feature weighting is useful in clustering high-dimensional data as this can reduce the effect of irrelevant features. The Simultaneous Clustering and Attribute Discrimination (SCAD) [32] performs clustering and feature weighting simultaneously and has several advantages. First, its continuous feature weighting provides a much richer feature relevance representation than binary feature selection. Second, the SCAD learns a cluster-dependent feature relevance weight in an unsupervised manner. The objective function of SCAD is defined as

$$J = \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \sum_{k=1}^n v_{ik}^q d_{ijk}^2, \quad (3.17)$$

subject to the constraint in (3.5) and

$$\sum_{k=1}^n v_{ik} = 1 \quad \forall i, \quad \text{and} \quad v_{ik} \in [0, 1] \quad \forall i, k. \quad (3.18)$$

In (3.17), n is the feature dimension, v_{ik} is the feature relevance weight for the k^{th} feature in the i^{th} cluster, $q \in (1, +\infty)$ is an exponent that controls the features discrimination rate, and $d_{ijk} = \|x_{jk} - c_{ik}\|$ is the euclidian distance between the j^{th} observation and the i^{th} cluster center taking into account the k^{th} feature only.

Minimizing J with respect to the centers \mathbf{c}_i yields

$$\mathbf{c}_i = \frac{\sum_{j=1}^N u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^m}. \quad (3.19)$$

Minimizing J with respect to the membership degree u_{ij} yields

$$u_{ij} = \frac{1}{\sum_{l=1}^C (D_{ij}/D_{lj})^{1/(m-1)}}, \quad (3.20)$$

where

$$D_{ij} = \sum_{k=1}^K v_{ik}^q d_{ijk}^2. \quad (3.21)$$

Minimizing J with respect to the feature weight v_{ik} yields

$$v_{ik} = \frac{1}{\sum_{l=1}^K (\tilde{D}_{ik}/\tilde{D}_{il})^{1/(q-1)}}, \quad (3.22)$$

where

$$\tilde{D}_{il} = \sum_{j=1}^N u_{ij}^m d_{ijl}^2. \quad (3.23)$$

The role of the feature weight exponent, q , can be deduced from equation (3.22).

It can be shown that as q approaches 1, v_{ik} tends to take binary values, i.e.,

$$v_{ik} = \begin{cases} 1 & \text{if } \tilde{D}_{ik} = \min_{t=1}^n \tilde{D}_{it} \\ 0 & \text{otherwise.} \end{cases}$$

This case is analogous to the winner-take-all situation where the feature along which the i^{th} cluster is the most compact gets all the relevancy ($v_{ik} = 1$), while all other attributes get assigned zero relevance, and hence do not contribute to the distance or center computations. On the other hand, when q approaches infinity, it can easily be shown that

$$v_{ik} = 1/n.$$

This means that all attributes share the relevancy equally. This is equivalent to the situation where no feature selection/weighting takes place. As it can be expected, for the case where q takes finite values in $(1, \infty)$, we obtain weights that provide a moderate level of feature discrimination. For this reason, q is referred to as a "discrimination exponent".

The SCAD algorithm is summarized by Algorithm 3.1.

Algorithm 3.4 Simultaneous Clustering and Attribute Discrimination

Inputs: \mathcal{X} : the features of the data samples.

C : the number of clusters.

m : the fuzzifier, $m \in (1, +\infty)$.

q : the exponent of the feature weights, $q \in (1, +\infty)$.

Outputs: U : the fuzzy membership matrix of the data samples.

c : the cluster centers.

V : the feature weights in each cluster.

- 1: Initialize U and V .
 - 2: **repeat**
 - 3: Update c using (3.19).
 - 4: Update V using (3.22).
 - 5: Update U using (3.20).
 - 6: **until** centers stabilize
 - 7: **return** c , U , V .
-

LANDMINE DETECTION

Detection and removal of landmines is a serious problem affecting human beings worldwide. The world is now littered with an estimated 80-110 million landmines in 64 countries, which maim or kill an estimated 500 people every week, mostly innocent civilians. Since the 1940s, many countries have worked on the solution to the problem of detecting nonmetallic landmines. The research has encompassed an extremely wide range of technologies and hundreds of millions of dollars have been spent. Despite these efforts, there is still no operational satisfactory detection solution. This lack of success is attributable to the extreme difficulty of the problem, such as: the large variety of landmine types, differing soil type and compaction, temperature, moisture, shadow, time of day, weather conditions, and varying terrain, to name a few.

A variety of sensors have been proposed or are under investigation for landmine detection. The research problem for sensor data analysis is to determine how well signatures of landmines can be characterized and distinguished from other objects under the ground using returns from one or more sensors. *Ground Penetrating*

Radar (GPR) offers the promise of detecting landmines with little or no metal content [1, 12]. Unfortunately, landmine detection via GPR has been a difficult problem [124, 40]. Although systems can achieve high detection rates, they have done so at the expense of high false alarm rates. The detection problem is compounded by the large variety of explosive object types, differing soil conditions, temperature, weather conditions, and varying terrain. In particular, many systems can be significantly affected by rapidly changing environmental conditions. Therefore, detection algorithms which can adopt to changing conditions are needed for detecting buried landmines.

The rest of this chapter gives a brief overview of several sensors that have been used to detect landmines and outlines the main landmine detection algorithms that will be used later in our fusion approach.

4.1 Sensors for Landmine Detection

4.1.1 Ground Penetrating Radar (GPR)

Ground penetrating radar (GPR) sensors have been used in a variety of landmine detection systems for quite some time [1, 12] and various algorithms for preprocessing GPR data to detect mines and discriminate between landmines and non-mine clutter objects have been employed [39, 38, 40, 36, 34, 29, 50, 117].

GPR works by emitting an electromagnetic wave covering a large frequency band into the ground through a wide-band antenna. Reflections from the soil caused by dielectric variations such as the presence of an object are measured. By moving the antenna it is possible to reconstruct an image representing a vertical slice of the soil (refer to Figure 4.2). GPR is sensitive to discontinuities in the electrical properties of the interrogated medium, rather than to the presence of metal. Consequently, nonmetallic objects, such as wood, plastic, stone, as well as metallic objects, can be

seen by the radar. Therefore, GPR offers the promise of detecting landmines with little or no metal content. This technology has been used for more than 20 years in civil engineering, geology and archeology for detecting buried objects and studying soil [1]. However these systems usually lack automatic recognition algorithms.

An example of GPR system that has been developed to detect landmines include the Wichmann/Niitek GPR System [54]. This radar is a very-wide bandwidth (200 Mhz - 7 Ghz) bi-static GPR with very low radar cross-section that implicitly solves many of the problems typically associated with shallow-buried object detection utilizing ground penetrating radar phenomenology. This system, shown in Figure 4.1, consists of a vehicle-mounted wide-bandwidth impulse radar integrated with a marking and GPS system. The radar is 1.2 m wide and contains 24 antennae or channels, spaced approximately 5 cm apart. As the vehicle moves in the down-track direction all 24 of the radars channels are sampled once every 5 cm and at each down-track position each channel measures one 416-element time-domain vector.



Figure 4.1: Wichmann/Niitek vehicle-mounted GPR.

The collected input data is represented by a 3-dimensional matrix of sample values, $S(z, x, y)$, $z = 1, \dots, 416$, $x = 1, \dots, 24$, $y = 1, \dots, N_s$, where N_s is the total number of collected scans, and the indices z , x , and y represent depth, cross-track position, and down-track positions respectively. A sample of unprocessed data is shown in

Figure 4.2. This image shows 600 down-track GPR responses from a central antenna channel. Clearly the largest source of GPR response energy is the dielectric discontinuity between the air and ground, seen near time sample 150 in all down-track scans. Despite the ground response, one can still visually identify two subsurface anomalies at scans 90 and 460.

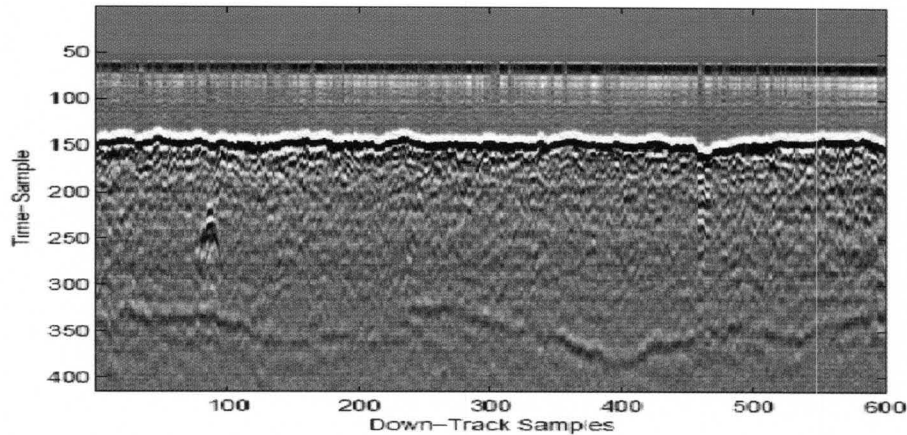


Figure 4.2: Sample of GPR responses. The x-axis represents down-track scan number, y-axis represents time sample. Two anomalies are visible in this data slice one at approximately sample 90, and another near sample 460. Also note the high energy of ground bounce visible in all down-track scans near time sample 150. This data has been clipped to enhance contrast.

4.1.2 Metal Detectors (MD)

Some interesting studies have been and are being carried out to see if it is feasible to discriminate mines from metallic clutter with metal detectors, to reduce the false alarm rate. For example, in [112], the author reported results on using an impulse MD looking for a characteristic decay curve and comparing it to the ones stored in a library. Problems arise from the fact that the response curve depends on several factors, such as the orientation of the metallic object, and the exact metal type. Also, the matching is done only with objects that are known a priori. Nevertheless, this approach could be promising in specific situations.

Somewhat along the same line, in [118], the author studied the possibility of characterizing objects/mines by measuring the frequency response over a large frequency range.

Another interesting and unconventional application is represented by the Meandering Winding Magnetometer (MWM) described in [119]. The device has the characteristic of using a square wave winding conductor in order to generate a spatially periodic electromagnetic field, whose spatial wavelength depends only on the primary winding spatial periodicity. It can, in principle, detect several characteristics of a buried metallic object (size, shape, etc.), and its application to humanitarian demining is currently being investigated.

The idea of using metal detectors to actually locate nonconducting targets, or more generally "cavities" in the soil, is also not new, as a (large) nonconducting target does indeed alter locally the natural ground conductivity, and has led for example to the patent ("cavity detector") described in [92]. The system should probably work best for large objects in soils with high natural conductivity ("background" signal).

Arrays of metal detectors, to quickly scan a large path for example, have also been built, such as the Schiebel VAMIDS system [10].

4.1.3 Electromagnetic Induction (EMI)

Another widely deployed metal detector (MD) for landmine detection is the electromagnetic induction (EMI) device that operates by sensing the metal present in land mines. The metal parts present in a landmine are detected by sensing the secondary magnetic field produced by eddy currents induced in the metal by a time-varying primary magnetic field. The frequency range employed is usually limited to a few tens of kHz. EMI sensors usually consist of a pair of coils, one of which is used to

transmit either a broadband pulse or a continuous wideband electromagnetic waveform. The transmitted field induces a secondary current in the earth as well as in any buried conducting objects. In the case of pulsed excitation, the transmit waveform is quenched quickly and the receiving coil measures the decaying secondary field that has been induced in the earth and subsurface objects [15]. In the case of wideband excitation, the receiving coil is placed within the magnetic cavity so that it senses only the weak secondary field radiated by the earth and buried objects [125]. Present research is investigating replacement of the receive coil with magnetoresistive devices.

The most obvious and serious limitation of metal detectors used to detect landmines is the fact that they are metal detectors. A modern metal detector is very sensitive and can detect tiny metal fragments as small as a couple of millimeters in length and less than a gram in weight. An area to be demined is usually littered with a large number of such metal fragments and other metallic debris of various sizes. This results in a high rate of "nuisance" alarms since a metal detector cannot currently distinguish between the metal in a landmine and that in a harmless fragment. The more sensitive a detector is, the higher the number of nuisance alarms it is likely to produce in a given location. Operating a detector at a lower sensitivity to reduce the number of such nuisance alarms may render it useless for detecting the very targets it was designed to detect, that is, the minimum metal-content landmines buried up to a few centimeters. Electromagnetic properties of certain soils can also limit the performance of metal detectors.

Figure 4.3 shows a cart based EMI sensor. This cart holds two EMI detectors, one with a dipole head and one with a quadrapole head.



Figure 4.3: Measurement cart based EMI sensor.

4.1.4 Infrared Imaging (IR)

Mines retain or release heat at a different rate than their surrounding, and during natural temperature variations of the environment it is possible, using IR cameras, to measure the thermal contrast between the soil over a buried mine and the soil close to it. When this contrast is due solely to the presence of the buried mine (alteration of the heat flow), one speaks of a volume effect. When it is due primarily to the disturbed soil layer above and around the mine (resulting from the burying operation), one speaks of a surface effect, which can be detectable for some time (say weeks) after burial and enhances the mine's signature. A good explanation of the various thermal mechanisms affecting the surface temperature contrast can be found in [109].

Landmine detection with passive infrared images can depend quite heavily on the environmental conditions [105], and there are cross over periods (in the evening and in the morning) when the thermal contrast is negligible and the mines may be undetectable. Foliage is also an additional problem.

4.1.5 Bulk Explosive Detection

Other interesting studies are growing towards techniques that can detect the explosive itself, in bulk form as opposed to trace explosive detection, and which have found application in security (airport luggage [101] or mail screening) or Non Destructive Testing applications. What makes the landmine detection problem formidable are, among others, the need for one-sided sensor configurations, operator security and equipment portability, and the limited soil penetration of particles/radiation.

In addition to the above sensors, there exists several other promising techniques for landmine detection. Examples include neutron activation, X-ray backscatter [44], Nuclear Magnetic or Quadrupole Resonance(NMR/NQR) [94, 95, 86], and Thermal Neutron Activation(TNA) [7].

4.2 Landmine Detection Algorithms

Generally, automated landmine discrimination algorithms consist of three phases: Preprocessing, feature extraction, and confidence assignment. Preprocessing performs tasks such as normalizing data, correcting for variations in height and speed, and removing stationary effects due to the system response. Previous methods include wavelets and Kalman filters [13, 14], subspace methods and polynomial matching [49], and subtracting optimally shifted and scaled reference vectors [11]. Feature extraction reduces the Preprocessed data to a lower-dimensional, salient set of values that represent the data. The principal component transform is a common feature extraction tool [129], as are wavelets [13], image processing based differentiation [38], and Hough and Radon transforms [116]. Confidence assignment

can be performed using methods such as Bayesian [116], hidden Markov Models [38, 29], fuzzy logic [39], rules and order statistics [37], neural networks, or nearest neighbor classifiers [28].

In the following, we outline four distinct feature-based algorithms for landmine detection. Using GPR and WEMI collected data, these algorithms have been applied to the landmine data with promising results.

4.2.1 Landmine Detection using GPR

In this section, we briefly highlight the GPR data preprocessing phase. Then we highlight three algorithms that have performed well in extensive field testing, and are being considered for real-time implementation in hand-held and vehicle-mounted GPR systems.

4.2.1.1 Data Preprocessing

Preprocessing is an important step to enhance the mine signatures for detection. In general, preprocessing includes ground-level alignment and signal and noise background removal. First, we identify the location of the ground bounce as the signal's peak and align the multiple signals with respect to their peaks. This alignment is necessary because the vehicle-mounted system cannot maintain the radar antenna at a fixed distance above the ground. The early time samples of each signal, up to few samples beyond the ground bounce are discarded. The remaining signal samples are divided into N depth bins, and each bin would be processed independently. The reason for this segmentation is to compensate for the high contrast between the responses from deeply buried and shallow anomalies.

Next, the adaptive least mean squares (LMS) pre-screener proposed by Torrione et al. [117] is used to focus attention and identify regions with subsurface anomalies. The goal of a pre-screener algorithm in the framework of vehicle-mounted realtime

landmine detection is to flag locations of interest utilizing a computationally inexpensive algorithm so that more advanced feature-processing approaches can be applied only on the small subsets of data flagged by the pre-screener. The LMS is applied to the energy at each depth bin and assigns a confidence value to each point in the cross-track, down-track plane based on its contrast with a neighboring region. The components that satisfy empirically pre-determined conditions are considered as potential targets. Their cross-track x_s , and down-track y_s positions of the connected component center are reported as alarm positions for further processing by the feature-based discrimination algorithm to attempt to separate mine targets from naturally occurring clutter.

4.2.1.2 The Edge Histogram Descriptor (EHD) Algorithm

The Edge Histogram Descriptor (EHD) algorithm uses translation invariant features, that are based on the Edge Histogram Descriptor (EHD) of the 3-D GPR signatures, and a possibilistic k -Nearest Neighbors (k -NN) rule for confidence assignment [51]. The EHD is an adaptation of the MPEG-7 EHD feature [85] which captures the signature's texture as feature for recognition. For a generic image, the EHD represents the frequency and the directionality of the brightness changes in the image. Simple edge detector operators are used to identify edges and group them into five categories: vertical, horizontal, 45° diagonal, 135° diagonal, and isotropic (non-edges). The EHD would include five bins corresponding to the above categories.

For the GPR data, the EHD has been adapted to capture the spatial distribution of the edges within a 3-D GPR data volume. To keep the computation simple, 2-D edge operators are used, and two types of edge histograms are computed. The first one is obtained by fixing the cross-track dimension and extracting edges in the (depth, down-track) plane. The second edge histogram is obtained by fixing the down-track dimension and extracting edges in the (depth, cross-track) plane.

Let $S_{zy}^{(x)}$ be the x^{th} plane of the 3-D signature $S(x, y, z)$. First, for each $S_{zy}^{(x)}$, four categories of edge strengths are computed: vertical, horizontal, 45° diagonal, and 135° anti-diagonal. If the maximum of the edge strengths exceeds a certain preset threshold, η , the corresponding pixels is considered to be an edge pixel. Otherwise, it is considered a non edge pixel. A global histogram that captures the frequency of the different edge orientations cannot take into account the relative position of the different edges. For instance, it cannot discriminate between mine signatures with a concave down hyperbolas and background signatures with a concave up hyperbolas. To overcome this limitation, each $S_{zy}^{(x)}$ image is vertically subdivided into 7 overlapping sub-images $S_{zy_i}^{(x)}$, $i = 1, \dots, 7$. For each $S_{zy_i}^{(x)}$, a 5 bin edge histogram, $H_{zy_i}^{(x)}$, is computed. The bins correspond to the 4 edge categories, and the non-edge pixels. The overlap is needed to make the sub-images large enough to include sufficient edges, and to reduce the sensitivity of the feature representation to the width and shift variations of the signatures.

The down-track component of the EHD, or EHD^y , is defined as the concatenation of the 7 five-bin histograms. That is,

$$\text{EHD}^y(S_{xyz}) = [\overline{H}_{zy_1} \overline{H}_{zy_2} \overline{H}_{zy_3} \dots \overline{H}_{zy_7}], \quad (4.1)$$

where \overline{H}_{zy_i} is the cross-track average of the edge histograms of sub-image $S_{zy_i}^{(x)}$ over N_C channels, i.e.,

$$\overline{H}_{zy_i} = \frac{1}{N_C} \sum_{x=1}^{N_C} H_{zy_i}^{(x)}. \quad (4.2)$$

To compute the cross-track component of the EHD, or EHD^x , the scans are fixed, and the 4 edge strengths on the $S_{zx}^{(y)}$ are computed, $y = 1, \dots, N_s$ (depth, cross-track) planes. Since these planes do not have enough columns (typically < 7) where the signature is present, they are not divided into sub-images, and only one global

histogram per plane, $H_{zx}^{(y)}$, is computed. That is, EHD^x is computed as the down-track average of the edge histograms over N_s scans

$$\text{EHD}^x(S_{xyz}) = \frac{1}{N_s} \sum_{y=1}^{N_s} H_{zx}^{(y)} \quad (4.3)$$

The EHD of each 3-D GPR alarm is a 40-dimensional histogram that concatenates the down-track and cross-track EHD components, i.e.,

$$\text{EHD}(S_{xyz}) = [\text{EHD}^y(S_{xyz}) \text{EHD}^x(S_{xyz})]. \quad (4.4)$$

The extraction of the EHD is illustrated in Figure 4.4.

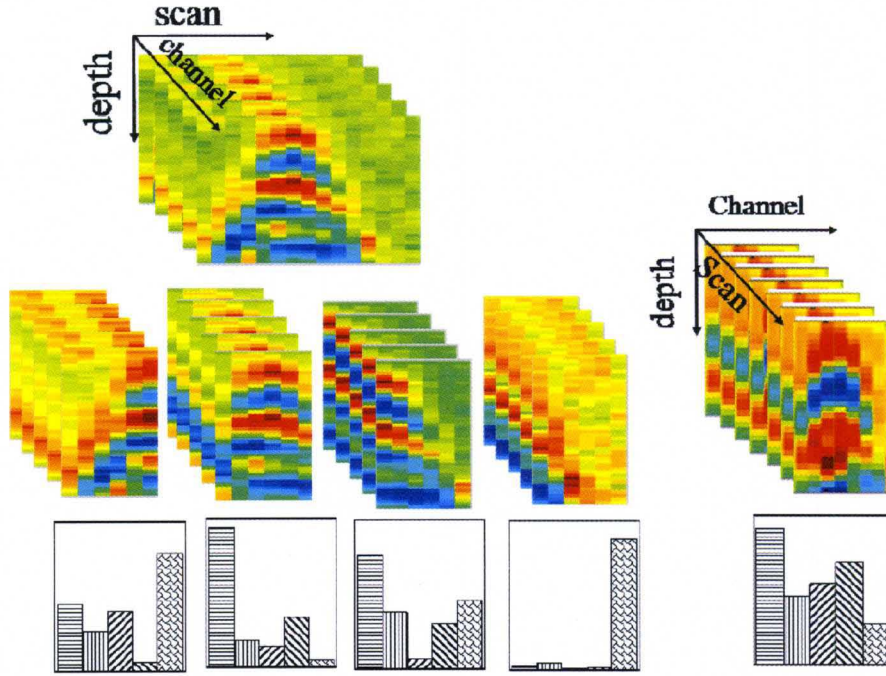


Figure 4.4: Extraction of the EHD for a 3-D mine signature. For clarity, only 4 of the 7 sub-images in the (depth, down-track) plane are shown.

A set of alarms with known ground truth is used to train the decision-making process. These labeled alarms are clustered to identify a small number of representative

prototypes that capture signature variations due to differing soil conditions, mine types, weather conditions, and so forth.

For a given test signature, the EHD histograms is extracted. Then, a possibilistic k -Nearest Neighbors (k -NN) rule is used to assign a confidence value [51].

4.2.1.3 Hidden Markov Model (HMM) Algorithm

Hidden Markov Model (HMM) is a model of a doubly stochastic process that produces a sequence of random observation vectors at discrete times according to an underlying Markov chain. At each observation time, the Markov chain may be in one of N_s states $\{s_1, \dots, s_N\}$ and, given that the chain is in a certain state, there are probabilities of moving to other states. These probabilities are called the transition probabilities. An HMM is characterized by three sets of probability density functions, the transition probabilities (\mathcal{A}), the state probability density functions (\mathcal{B}), and the initial probabilities (π). Let T be the length of the observation sequence (i.e., number of time steps), let $O = \{O_1, \dots, O_T\}$ be the observation sequence, and let $Q = \{q_1, \dots, q_T\}$ be the state sequence. The compact notation is generally used to indicate the complete parameter set of the HMM model.

$$\lambda = (\mathcal{A}, \mathcal{B}, \pi) \quad (4.5)$$

In Equation (4.5), $A = [a_{ij}]$ is the state transition probability matrix, where $a_{ij} = Pr(q_t = j | q_{t-1} = i)$ for $i, j = 1, \dots, N_s$; $\pi = \{\pi_i\}$, where $\pi_i = Pr(q_1 = s_i)$ are the initial state probabilities; and $B = \{b_i(O_t), i = 1, \dots, N\}$, where $b_i(O_t) = Pr(O_t | q_t = i)$ is the set of observation probability distribution in state i .

The HMM algorithm for landmine detection using GPR [38, 29] treats the down-track dimension as the time variable and produces a confidence that a mine is present at various positions, (x, y) , on the surface being traversed. In particular, a sequence of observation vectors is produced for each point. These observation

vectors encode the degree to which edges occur in the diagonal and anti-diagonal directions. In particular, for every point (x_s, y_s) , the strengths for the positive/negative diagonal/anti-diagonal edges is computed. Then, the observation vector at a point (x_s, y_s) consists of a set of features that encode the maximum edge magnitude over multiple depth values around (x_s, y_s) . Figure 4.5 displays a hyperbolic curve superimposed on a preprocessed metal mine signature to illustrate the features of a typical mine signature.

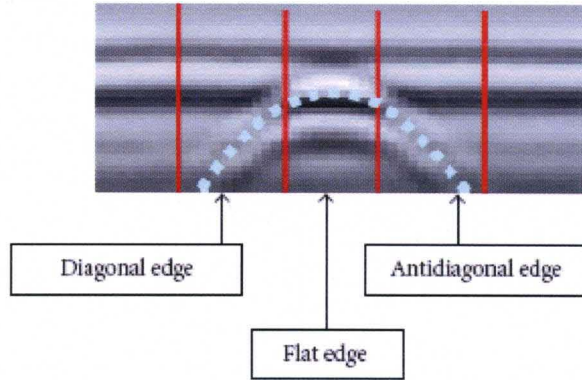


Figure 4.5: HMM Feature of a mine signature

The HMM classifier for landmine detection consists of two HMM models, one for mine and one for background. Each model has three states and produces a probability value by backtracking through model states using the Viterbi algorithm [24]. The mine model, λ_m , is designed to capture the hyperbolic spatial distribution of the features. λ_m has 3 states which correspond to the rising edge, flat, and decreasing edge. Each state is represented by 3 Gaussian components. The mine model is left to right model in that states are ordered and the transition probabilities for moving to a lower numbered state are zero. The background model is needed to capture the background characteristics and to reject false alarms. Each of the 24 channels is treated independently from the others, and has its own background model, λ^{b_c} . In addition to allowing each channel to have a model that reflects its own data, this

decoupling allows the channels to be processed in parallel, and thus facilitating real-time operation. All λ^{b_c} (for $c = 1, \dots, 24$) have 3 states and 3 Gaussian components per state. The probability value produced by the mine (background) model can be thought of as an estimate of the probability of the observation sequence given that there is a mine (background) present. The model architecture of the HMM classifier is illustrated in Figure 4.6.

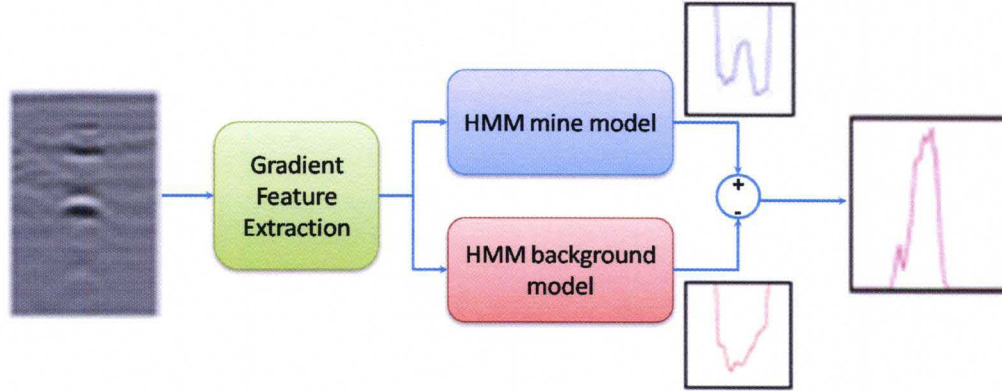


Figure 4.6: Illustration of the HMM-based model architecture.

4.2.1.4 Spectral Correlation Feature (SCF) Algorithm

Spectral feature (SPECT) algorithm aims at capturing the characteristics of a target in the frequency domain. It extracts the alarm Spectral Correlation Feature (SCF), and formulates a confidence value based on similarity to prototypes that characterize mine objects [55].

The spectral features are derived from the Energy Density Spectrum (EDS) of an alarm declared by the pre-screener. The estimation of EDS involves three main steps: pre-processing, whitening, and averaging. Pre-processing estimates the ground level, aligns the data from each scan with respect to ground level, and removes the 60 data above and near the ground surface. This step is needed to avoid an EDS that is dominated by the response of the ground bounce. The whitening step performs

equalization on the spectrum from the background so that the estimated EDS reflects the actual spectral characteristics of an alarm. Averaging reduces the variance in the EDS.

4.2.2 Landmine Detection using WEMI

The Wideband Electro-Magnetic Induction (WEMI) sensor was developed by W. Scott [107]. The sensor measures the response of an object at 21 logarithmically spaced frequencies over the range 330 Hz to 90 KHz. The goal is to obtain characteristic spectral shapes that can help discriminate objects of interest from false alarms.

The response of the system can be modeled as

$$S(w) = A[I(w) + iQ(w)]; \quad (4.6)$$

where w is the frequency, A is the magnitude and $I(w) + iQ(w)$ describes the shape of the response as a function of frequency. An input data point is composed of 21 complex responses at the following measured frequencies (in Hz.): 330, 390, 510, 690, 930, 1230, 1650, 2190, 2910, 3930, 5190, 6930, 9210, 12210, 16230, 21630, 28770, 38250, 50850, 67650, and 90030.

Before feature extraction, the I and Q values are normalized between 0 and 1. This eliminates variation in magnitude due several factors - such as the depth of the buried object to be detected as well as metal mass and content - that do not affect the shape of the response curve. The magnitude can always be measured separately. After normalization, the response models proposed by Miller et al. [91] are used to fit the curve. The 3-parameter model is given by

$$I + iQ = q \left(s + \frac{(iw\tau)^{1/2} - 2}{(iw\tau)^{1/2} + 1} \right) \quad (4.7)$$

where q , s , and τ are the three parameters describing the shape of the response curve. The value q represents the magnitude of the response curve after normalization, s does the shift in the frequency axis, and τ controls the rate of shape change.

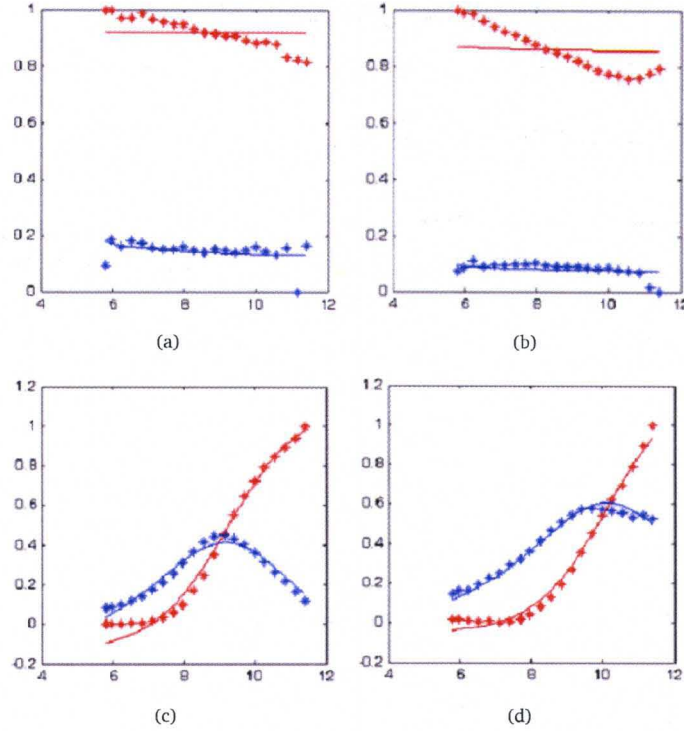


Figure 4.7: Response curves (sequences of dots) and their curve fits (smooth curves) from (a) blank, (b) non-metallic clutter item, (c) metallic clutter item, and (d) low-metal mine

The parameters resulting from this curve fit plus the error in the fit provide 4 features. Figure 4.7 displays the response curves and their curve fits of metallic and non-metallic objects. We note that other researchers, such as Torrione [117] and Yuksel [130] have also used these model parameters as features. In addition to the 4 features provided by the model, 3 spread features [91] are used. These are defined by the following equations in which I and Q represent the In-phase (Real) and Quadrature (Imaginary) values at each frequency and N is the number

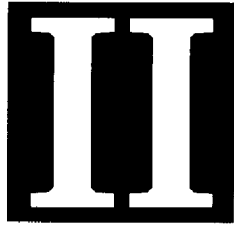
of frequencies.

$$Q_{sum} = \sum_{i=1}^N Q_i \quad (4.8)$$

$$Q_{spread} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N |Q_i - Q_j| \quad (4.9)$$

$$T_{spread} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N |Q_i - Q_j| + \sum_{i=1}^{N-1} \sum_{j=i+1}^N |I_i - I_j| \quad (4.10)$$

Together, these make up the 7 features used to describe a WEMI signal. Feature selection was performed using the well-known divergence measure. Four features were selected: τ , the fitting error, Q_{spread} , and T_{spread} . A Multi-Layer Perceptron (MLP) classifier was built from these features. We will refer to this classifier as the Model Fitting (MFIT) detector.



PROPOSED WORK

CONTEXT EXTRACTION FOR LOCAL FUSION

In this Chapter, we present our novel fusion method, called *Context Extraction for Local Fusion* (CELF). CELF is a local approach that adapts the fusion process to different regions of the feature space, referred to as *contexts*. It takes advantages of the strengths of few algorithms in different contexts without being affected by the weaknesses of the other algorithms.

Existing local classifier fusion methods treat the partitioning of the feature space and the selection/fusion of local expert classifiers as two independent processes that are performed sequentially (refer to section 2.7). However, these two tasks are not independent, and their optimization should be combined. CELF is a generic framework that optimizes these two tasks simultaneously. It is based on a novel objective function that combines context identification and multi-algorithm fusion criteria into a joint objective function. This objective function is defined and optimized to produce contexts as compact clusters via unsupervised clustering. Optimization of the objective function also provide optimal fusion parameters for each context.

CELF has mainly two advantages over the existing local fusion methods. First, the extraction of the different contexts and the optimization of the decision fusion are done in parallel. In that way, we can obtain more *robust contexts* where the different

experts behave consistently. As a result, the fuser is able to make more robust decisions. Second, rather than dealing with hard clusters like most of the other local fusion methods, CELF generates fuzzy clusters, which makes it *robust* to noise. A test point can be assigned to one cluster or several clusters with different membership degrees. These membership degrees are used to combine the local decisions and generate the final decision.

In this chapter, we present several variants of the proposed approach. First, in Section 5.1, the basic form of CELF is fully developed. Extensions of CELF are introduced in Sections 5.2, 5.3, and 5.4. To explain the behavior of our approach, experimental results on synthetic data are given within each section.

5.1 Context Extraction for Local Fusion (CELF)

In the following, we assume that we have N training observations with desired output $\mathcal{T} = \{t_j | j = 1, \dots, N\}$ that were processed by K algorithms. These algorithms could process data from different sensors, and/or use different feature extraction, and/or classification algorithms. Each algorithm k extracts its own feature set, $\mathcal{X}_k = \{\mathbf{x}_{kj} | j = 1, \dots, N\}$, and generates confidence values, $\mathcal{Y}_k = \{y_{kj} | j = 1, \dots, N\}$. The K feature sets are then concatenated to generate one global descriptor for each observation:

$$\mathcal{X} = \bigcup_{k=1}^K \mathcal{X}_k = \{\mathbf{x}_j = [\mathbf{x}_{1j}, \dots, \mathbf{x}_{Kj}] | j = 1, \dots, N\}. \quad (5.1)$$

The Context Extraction for Local Fusion (CELF) is designed to

- (i) to partition the feature space into groups of homogeneous samples; and
- (ii) to learn the optimal classification method within each group.

The first task can be achieved by an unsupervised learning or clustering of the observations in the aggregate feature space. The second task is a supervised learning

problem that uses the observation labels to minimize the overall classification error. CELF achieves these two tasks by minimizing the following objective function:

$$J_1 = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2, \quad (5.2)$$

subject to

$$u_{ij} \in [0, 1] \quad \forall i, j, \quad \sum_{i=1}^C u_{ij} = 1 \quad \forall j, \quad \text{and} \quad \sum_{k=1}^K \omega_{ik} = 1 \quad \forall i. \quad (5.3)$$

The first term in (5.2) is the unsupervised learning component. It is the sum of intra-cluster distances and is the objective function used in the Fuzzy C-Means (FCM) algorithm [8]. It seeks to partition the N samples into C clusters, and represent each cluster by a center \mathbf{c}_i . Each sample \mathbf{x}_j will be assigned to each cluster i with a membership degree u_{ij} . In this term, $m \in (1, \infty)$ is a constant called the *fuzzifier* and is used to control the degree of fuzziness [8]. The second term in (5.2) is the supervised learning component. It attempts to learn cluster-dependent aggregation weights of the K algorithm outputs. In this term, w_{ik} is the aggregation weight assigned to algorithm k within cluster i . This term is minimized when the aggregated partial output values match the desired output. When both terms are combined and α is chosen properly, the algorithm seeks to partition the data into compact and homogeneous clusters while learning optimal aggregation weights for each algorithm within each cluster.

To optimize J_1 with respect to $\mathbf{W} = [\omega_{ik}]$, we incorporate the constraints using Lagrange multipliers and obtain

$$L^\omega = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2 + \sum_{i=1}^C \lambda_i \left(\sum_{k=1}^K \omega_{ik} - 1 \right), \quad (5.4)$$

where $\Lambda = [\lambda_1, \dots, \lambda_C]^t$ is a vector of Lagrange multipliers corresponding to the C constraints on \mathbf{W} in (5.3). Since the set of weights within each cluster are independent of each other, the optimization problem in (5.4) could be reduced to C simpler

independent problems. In particular, for $i = 1, \dots, C$, we minimize

$$L_i^\omega = \sum_{j=1}^N u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N u_{ij}^m \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2 + \lambda_i \left(\sum_{k=1}^K \omega_{ik} - 1 \right). \quad (5.5)$$

To obtain the optimal \mathbf{W} , we compute the derivative of L_i^ω with respect to ω_{ik} and set it to 0, i.e.,

$$\frac{\partial L_i^\omega}{\partial \omega_{ik}} = 2\alpha \sum_{j=1}^N u_{ij}^m y_{kj} \left(\sum_{l=1}^K \omega_{il} y_{lj} - t_j \right) + \lambda_i = 0. \quad (5.6)$$

Solving (5.6), we obtain

$$\omega_{ik} = \left[\sum_{j=1}^N u_{ij}^m y_{kj} \left(t_j - \sum_{\substack{l=1 \\ l \neq k}}^K \omega_{il} y_{lj} \right) - \frac{\lambda_i}{2\alpha} \right] / \left[\sum_{j=1}^N u_{ij}^m y_{kj}^2 \right]. \quad (5.7)$$

The Lagrange constant λ_i could be solved using the constraint that $\sum_{l=1}^K \omega_{il} = 1$. Doing so, we obtain

$$\frac{\lambda_i}{2\alpha} = \left[\sum_{l=1}^K \frac{\sum_{j=1}^N u_{ij}^m y_{lj} \left(t_j - \sum_{k=1}^K \omega_{ik} y_{kj} \right)}{\sum_{j=1}^N u_{ij}^m y_{lj}^2} \right] / \left[\sum_{l=1}^K \frac{1}{\sum_{j=1}^N u_{ij}^m y_{lj}^2} \right]. \quad (5.8)$$

From equation (5.7), we can see that algorithm k will be assigned the highest weight, ω_{ik} , in cluster i if it is the most relevant classifier within this cluster. That is, its exclusion (in the K summation in the numerator) will result in the largest deviation from the desired output for samples with high memberships in this cluster.

To derive the update equation of the cluster centers, we set the derivative of J_1 with respect to \mathbf{c}_i to zero and solve

$$\frac{\partial J_1}{\partial \mathbf{c}_i} = 2 \sum_{j=1}^N u_{ij}^m (\mathbf{x}_j - \mathbf{c}_i) = \mathbf{0}. \quad (5.9)$$

We obtain

$$\mathbf{c}_i = \left[\sum_{j=1}^N u_{ij}^m \mathbf{x}_j \right] / \left[\sum_{j=1}^N u_{ij}^m \right]. \quad (5.10)$$

That is, \mathbf{c}_i is the centroid of each cluster in the aggregated feature space.

To optimize J_1 with respect to the memberships $\mathbf{U} = [u_{ij}]$, we incorporate the constraints using Lagrange multipliers and obtain

$$L^u = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2 - \sum_{j=1}^N \xi_j \left(\sum_{i=1}^C u_{ij} - 1 \right), \quad (5.11)$$

where $\Xi = [\xi_1, \dots, \xi_N]^t$ is a vector of Lagrange multipliers corresponding to the N constraints in (5.3). Since the memberships of the different observations are independent of each other, the above optimization problem can be reduced to N simpler independent problems. In particular, for $j = 1, \dots, N$, we minimize

$$L_j^u = \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{i=1}^C u_{ij}^m \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2 - \xi_j \left(\sum_{i=1}^C u_{ij} - 1 \right). \quad (5.12)$$

To derive the necessary condition to optimize u_{ij} , we compute the derivative of L_j^u with respect to u_{ij} and set it to zero, i.e.,

$$\frac{\partial L_j}{\partial u_{ij}} = m u_{ij}^{m-1} D_{ij} - \xi_j = 0, \quad (5.13)$$

where

$$D_{ij} = \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2. \quad (5.14)$$

Solving (5.13) for u_{ij} , we obtain

$$u_{ij} = \left(\frac{\xi_j}{m} \right)^{1/(m-1)} \frac{1}{(D_{ij})^{1/(m-1)}}. \quad (5.15)$$

The Lagrange constant ξ_j could be solved using the constraint that $\sum_{l=1}^C u_{lj} = 1$. We obtain

$$\left(\frac{\xi_j}{m} \right)^{1/(m-1)} = \left[\sum_{l=1}^C (1/D_{lj})^{1/(m-1)} \right]^{-1}. \quad (5.16)$$

Substituting (5.16) into (5.15), we obtain the following update equation for the membership degree of observation j in cluster i .

$$u_{ij} = \left[\sum_{l=1}^C (D_{ij}/D_{lj})^{\frac{1}{m-1}} \right]^{-1}. \quad (5.17)$$

In (5.14), D_{ij} can be viewed as the total cost when considering point \mathbf{x}_j in cluster i . This cost depends on: (i) the distance between the considered point and the cluster's centroid \mathbf{c}_i ; and (ii) the deviation of the combined algorithms' decision from the desired output (weighted by α). In other words, in (5.17), points will be assigned high membership degree in the same cluster, i.e., clustered together if:

- (i) they are close to each other in the feature space, i.e. small $\|\mathbf{x}_j - \mathbf{c}_i\|^2$, and
- (ii) their confidence values could be combined linearly with the same coefficients to match the desired output, i.e. small $\left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2$.

The CELF algorithm is an iterative process involving successive updates of the clusters' prototypes (\mathcal{C}), the partition matrix (\mathbf{U}), and the aggregation weights (\mathbf{W}). It is summarized in Algorithm 5.1.

Algorithm 5.1 Context Extraction for Local Fusion (CELF)

Inputs: \mathcal{X} : the features of the training data samples.

\mathcal{Y} : the confidences given to the data samples by the different classifiers.

\mathcal{T} : the labels of the data samples.

C : the number of clusters.

m : the fuzzifier, $m \in (1, +\infty)$.

α , the weight of the second term in the objective function.

Outputs: \mathbf{U} : the fuzzy membership matrix of the data samples.

\mathcal{C} : the cluster centers.

\mathbf{W} : the confidence weights within each cluster.

- 1: Initialize \mathbf{U} and \mathbf{W} .
 - 2: **repeat**
 - 3: Update \mathcal{C} using (5.10).
 - 4: Update \mathbf{W} using (5.7).
 - 5: Update \mathbf{U} using (5.17).
 - 6: **until** parameters do not change significantly
 - 7: **return** \mathcal{C} , \mathbf{U} , \mathbf{W}
-

We should note here that using (5.14) to compute the membership of a given sample j to cluster i , its desired output t_j needs to be given. This information is available and is necessary to build the clusters in the training phase. However, during testing, for an unlabeled test sample, it is not possible to assign it to a cluster using (5.14). Instead, we identify the nearest training sample and use its label. Given an unlabeled test sample j , we apply the following steps to generate the final fusion decision.

1. Process the sample by the different algorithms to generate a set of features, \mathbf{x}_j , and decision values, $\mathbf{y}_j = [y_{1j}, \dots, y_{Kj}]$.
2. Identify the nearest training sample and use its label to assign a temporary label to the test sample.
3. Assign a membership degree to sample j in each cluster i , u_{ij} , using (5.14).
4. Combine the output of the different classifiers within each cluster using

$$\hat{y}_{ij} = \sum_{k=1}^K \omega_{ik} y_{kj}. \quad (5.18)$$

5. Generate the final fusion decision confidence using

$$\hat{y}_j = \sum_{i=1}^C u_{ij} \hat{y}_{ij}. \quad (5.19)$$

Figure 5.1 displays the architecture of the training and the testing phases of our approach. The training phase is composed of two *interactive* components: *context extraction* and *decision fusion*. The context extraction step uses both the features extracted by various algorithms (indicated by solid lines in the figure) and their confidences (indicated by dotted lines) to partition the training input samples into different contexts. The decision fusion step uses the confidence values assigned by the individual algorithms (indicated by dotted lines in the figure) to assign aggregation weights to the different algorithms within each context based on their relative

performance within that context. To test a new alarm, each algorithm extracts its set of features and assigns a confidence value. Then, as shown in the right part of Figure 5.1, the features are used to assign the test sample to the closest context. The aggregation weights of this context are then used to fuse the individual confidence values.

Illustrative example

To illustrate the behavior of the proposed dynamic fusion approach, we use it to partition and fuse a toy data with 30 samples that belong to two classes. Suppose that each sample has been processed by two algorithms. Each algorithm, k , extracts one feature, x_k , and assigns one output value, y_k . Figure 5.2(a) displays these samples in the 2-D feature space. For each sample point, we display the output of the two algorithms on the top of each sample. Figures 5.2(b) and 5.2(c) display the cumulative histograms of the confidences assigned by algorithm 1 and 2 respectively. As it can be seen, none of the two algorithms can separate the two classes perfectly. In particular, by examining the labels of the samples in Figure 5.2(a), we notice that for the 10 samples on the right side (last 2 columns) of the feature space, the second algorithm performs better than the first one (lower confidence and desired output is 0). However, for the 4 samples on the bottom left corner of the feature space, the first algorithm outperforms the second one. Thus, to take advantage of the complementary nature of these algorithms, we need a local fusion approach that partitions the feature space into *coherent* contexts and adapts the fusion to each context.

Figure 5.3 displays the result of applying CELF when the number of clusters C is set to 5 and the coefficient α is set to 10. In this figure, for visualization purposes, we map the fuzzy partition generated by CELF into a crisp one using the maximum membership assignment. First, we note that the 5 clusters include points that are spatially close to each other. Second, each cluster includes samples that have consistent algorithm outputs. For instance, the first cluster (blue circles) includes

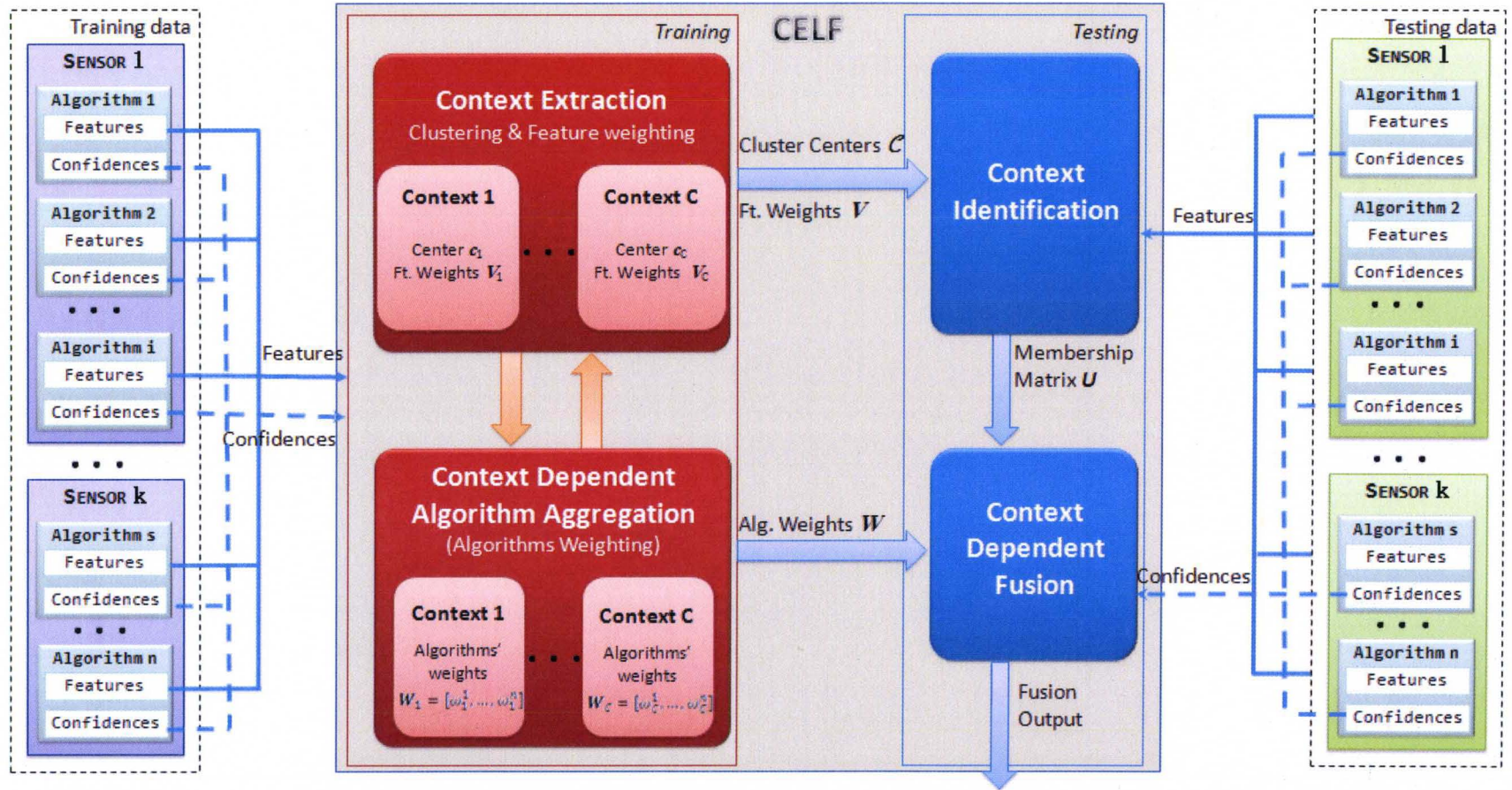


Figure 5.1: Architecture of the proposed Context Extraction for Local Fusion method. The left part highlight the training phase and the the right side highlight the testing phase.

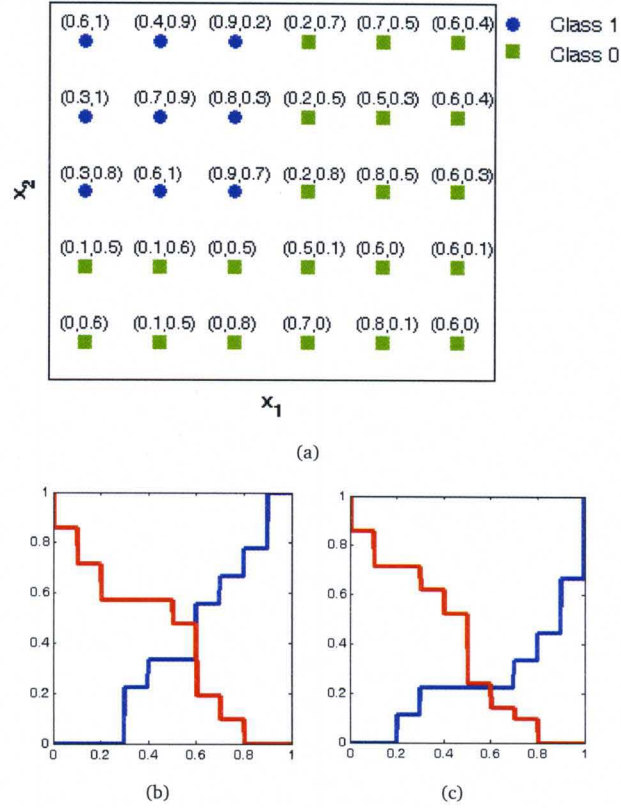


Figure 5.2: Feature and confidence distribution of a sample data. (a) 30 samples in the 2-D feature space. The two algorithm outputs are shown above each sample. (b) Cumulative histogram of the confidences assigned by the first algorithm. (c) Cumulative histogram of the confidences assigned by the second algorithm.

samples where algorithm 2 outputs are more reliable in predicting the desired output. On the other hand, the second cluster (green squares) includes samples where algorithm 1 outputs are more reliable in predicting the desired output.

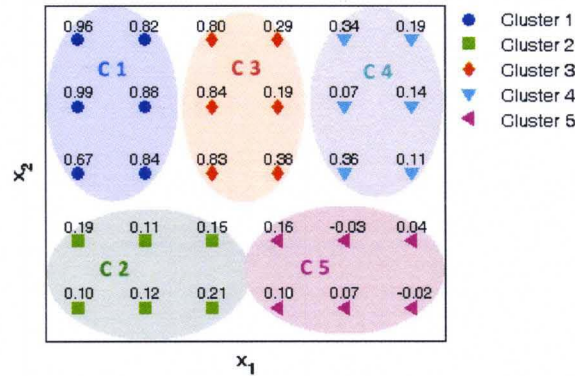
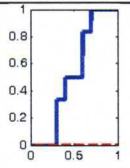
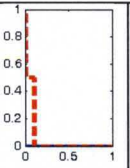
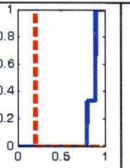
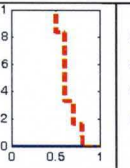
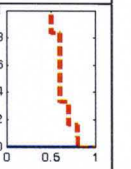
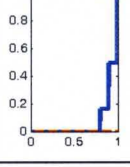
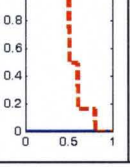
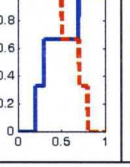
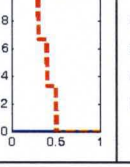
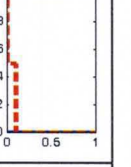


Figure 5.3: Clustered samples in the feature space using CELF. The fused confidences are shown above each sample.

The cumulative histograms of the two algorithms and the algorithm fusion weights assigned by CELF to each algorithm within each cluster are displayed in Table 5.1. As it can be seen, CELF assigns the highest weight to the most reliable classifier in each cluster. For instance, for samples assigned to cluster 3, algorithm 1 can easily discriminate between the two classes (non-overlapping confidence distribution), while algorithm 2 confuses the two classes (large overlap). Consequently, CELF assigns a high aggregation weight to algorithm 1 (1.26) and low weight to algorithm 2 (-0.26). In the proposed approach, no constraint was imposed on the range of values of the classifier weights, \mathbf{W} , as long as they add up to 1. Thus, the weights of the classifiers can take any real values, including negative ones. In fact, as shown in Table 5.1, the weights of the classifiers in the 4th cluster are -1.5 for the first algorithm and 2.5 for the second algorithm. The cumulative histogram of the two algorithms within this cluster shows that the second algorithm is better than the first one. However, the assigned confidences are far from the desired output. CELF tries to assign the appropriate aggregation weights to have a confidence as close as possible to the desired output (which is 0 in this case).

Table 5.1: Performance of the two classifiers (dashed line for class 0 and solid line for class 1) and assigned aggregation weights to each classifier within each cluster

Cluster #	1	2	3	4	5
Cumulative Histogram					
Algorithm 1					
Algorithm 2					
Assigned Weights					
ω_1	-0.05	1.04	1.26	-1.5	-0.06
ω_2	1.05	-0.04	-0.26	2.5	1.06

Using the learned context dependent aggregation weights, the final decision is computed using equation (5.19). Figure 5.4 displays the cumulative histogram of the assigned confidences. As it can be seen, the two distributions become separable and any threshold between 0.4 and 0.6 would result in an accuracy of 100%.

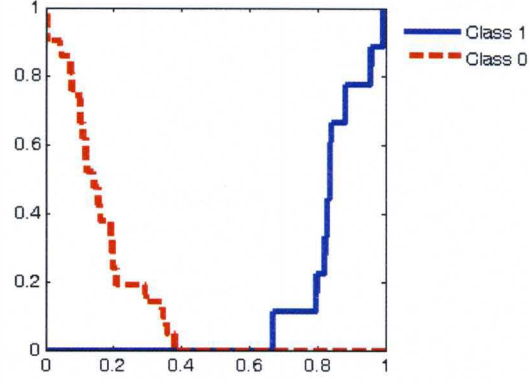


Figure 5.4: Cumulative histograms of the confidence values fused using CELF.

5.2 CELF with Feature Discrimination

For complex classification problems, multiple sources of information and multiple classifiers for each source may be needed to obtain satisfactory results. In this case, the composite feature space can be high dimensional and standard clustering algorithms may not generate meaningful partitions. This is because clusters tend to form in sub-spaces of the original feature space, and the influence of the features is generally not equally important for the different clusters. Moreover, the number of features extracted by each algorithm can vary significantly. This could lead to a partition that is biased by the algorithm that has the largest number of features. To alleviate this drawback, we propose generalizing the objective function in (5.2) to allow finding clusters in subspaces of the original feature space. In particular, instead of treating all individual features equally, we treat them as subsets (one subset per algorithm) and learn one optimal feature relevance weight for each subset within each cluster. The resulting algorithm, called Context Extraction for Local Fusion with Feature Discrimination (CELF-FD), combines clustering, feature

discrimination, and multi-algorithm fusion. It minimizes

$$J_2 = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \sum_{k=1}^K v_{ik}^q d_{ijk}^2 + \sum_{i=1}^C \sum_{j=1}^N \alpha_i u_{ij}^m \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2, \quad (5.20)$$

subject to the constraints in (5.3) and

$$\sum_{k=1}^K v_{ik} = 1 \quad \forall i, \quad \text{and} \quad v_{ik} \in [0, 1] \quad \forall i, k. \quad (5.21)$$

In (5.20), v_{ik} is the feature relevance weight for feature subset k (extracted by algorithm k) in cluster i , $q \in (1, +\infty)$ is an exponent that controls the features discrimination rate, and $d_{ijk} = \|x_{jk} - c_{ik}\|$ is the Euclidian distance between the j^{th} observation and the i^{th} cluster center taking into account feature subset k only. Rather than using a constant value for α as in (5.2), we use a cluster dependent α_i to balance the feature relevance weights. In particular, we let $\alpha_i = \beta \sum_{l=1}^K v_{il}^q$ where β is a constant.

Minimizing J_2 with respect to $\mathbf{U} = [u_{ij}]$ yields

$$u_{ij} = \left[\sum_{l=1}^C (D_{ij}/D_{lj})^{\frac{1}{m-1}} \right]^{-1} \quad (5.22)$$

where

$$D_{ij} = \sum_{k=1}^K v_{ik}^q d_{ijk}^2 + \beta \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2 \left(\sum_{k=1}^K v_{ik}^q \right). \quad (5.23)$$

Minimizing J_2 with respect to $\mathbf{V} = [v_{ik}]$ yields

$$v_{ik} = \left[\sum_{s=1}^K (D_{ik}/D_{is})^{\frac{1}{q-1}} \right]^{-1}, \quad (5.24)$$

where

$$\tilde{D}_{ik} = \sum_{j=1}^N u_{ij}^m d_{ijk}^2 + \beta \sum_{j=1}^N u_{ij}^m \left(\sum_{l=1}^K \omega_{il} y_{lj} - t_j \right)^2. \quad (5.25)$$

Minimization of J_2 with respect to \mathbf{W} and \mathcal{C} yields the same equations as in CELF

(i.e. (5.7) and (5.10) respectively). The CELF-FD algorithm is summarized in Algorithm 5.2.

Algorithm 5.2 CELF with Feature Discrimination (CELF-FD)

Inputs: \mathcal{X} : the features of the data samples.

\mathcal{Y} : the confidences given to the data samples by the different classifiers.

\mathcal{T} : the labels of the data samples.

C : the number of clusters.

m : the fuzzifier, $m \in (1, +\infty)$.

β : the weight of the second term in the objective function.

q : the exponent of the feature weights, $q \in (1, +\infty)$.

Outputs: U : the fuzzy membership matrix of the data samples.

\mathcal{C} : the cluster centers.

W : the confidence weights in each cluster.

V : the feature weights in each cluster.

- 1: Initialize U , W and V .
 - 2: **repeat**
 - 3: Update \mathcal{C} using (5.10).
 - 4: Update V using (5.24).
 - 5: Update W using (5.7).
 - 6: Update U using (5.22).
 - 7: **until** parameters do not change significantly
 - 8: **return** \mathcal{C} , U , V , W
-

Illustrative example

To illustrate the behavior of CELF-FD, we use it to partition a synthetic data. This data set has two classes and is designed to illustrate the need for local fusion. Suppose that each sample has been processed by two algorithms. Each algorithm, k , extracts one feature (x_k) and assigns one output value (y_k). Figure 5.5(a) displays this data in the combined 2-D feature space; samples from `class 0` are represented by blue dots and samples from `class 1` are represented by black dots.

As it can be seen, the data form 4 clusters in the aggregate feature space, and each cluster has samples from both classes. Figure 5.5(b) displays the clustering result of these samples using SCAD [33]. We should emphasize here that the ground truth labels of the samples are not used in the clustering step. As it can be seen, SCAD (like most other clustering algorithms) succeeds in identifying the four

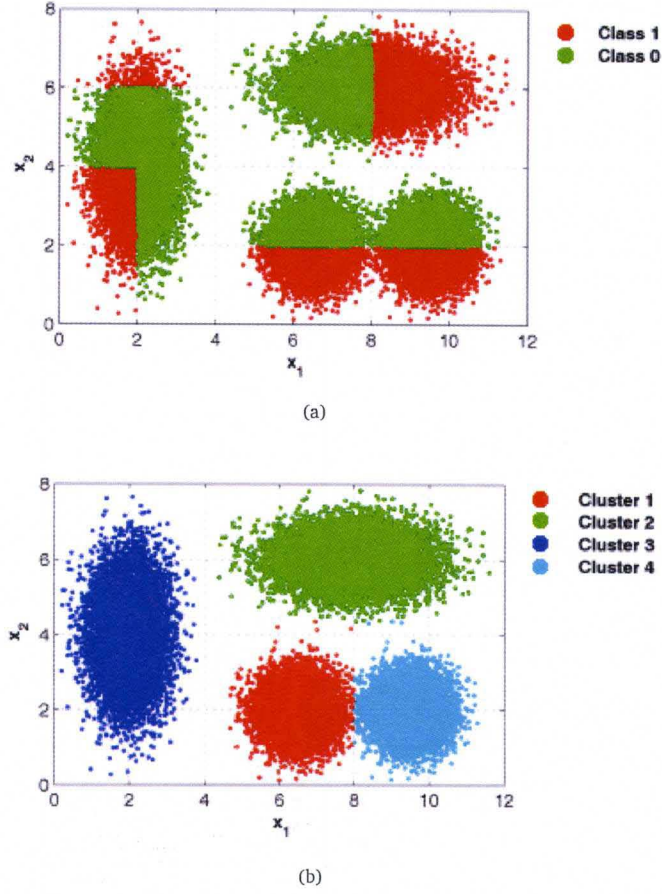


Figure 5.5: Synthetic data in the combined 2-D feature space. (a) Class 0 samples are shown as blue dots and class 1 samples are shown as black dots. (b) Typical clustering results when the problem is treated as unsupervised learning and the true labels are not used.

intuitive clusters. If the objective of this task is clustering, this would be the optimal solution. However, our objective in this fusion application is to identify compact clusters where the classifiers have similar behavior. To illustrate this, we display the classification results of the two classifiers in Figure 5.6. As it can be seen, none of the two classifiers classify this data perfectly as both figures include many misclassified samples. In fact, the accuracy of classifier 1 is 69% and for classifier 2 is 81%. More importantly, the performance of each classifier varies in different regions of the feature space. For instance, in Figure 5.6(a), we observe that classifier one classifies all samples located on the top right ellipsoidal cluster correctly, but has only a 50% correct classification rate for the two spherical clusters. On the other

hand, classifier two, in Figure 5.6(b), has a 100% correct classification rate for the two spherical clusters and 50% classification rate for the top right ellipsoidal cluster. This synthetic example illustrates the need for local fusion to take advantages of the strengths of the classifiers in different regions of the feature space.

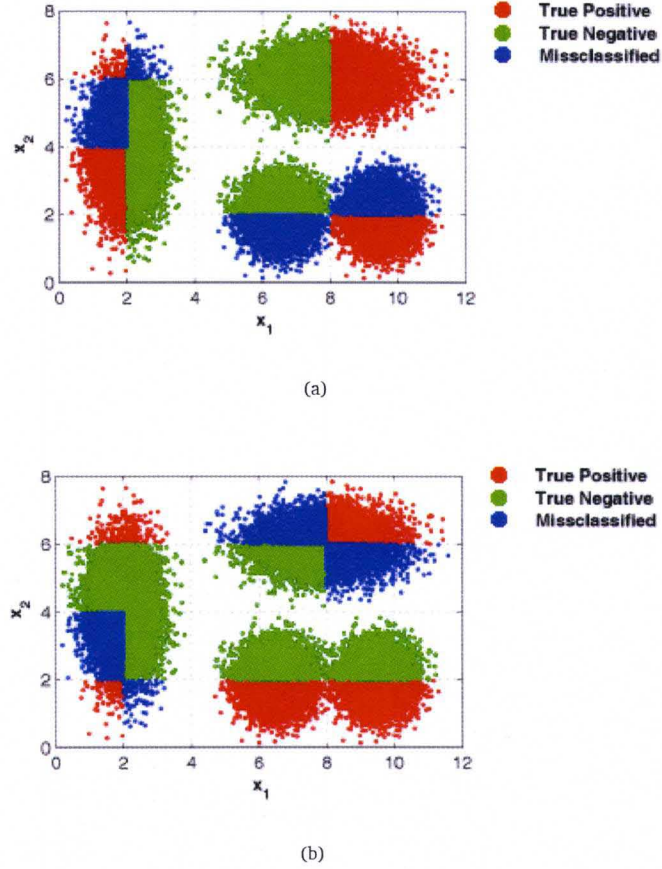


Figure 5.6: Classification result of (a) the first algorithm (based on feature x_1), and (b) the second algorithm (based on feature x_2).

Classification results with global fusion The two classifiers could be fused in many different ways. Similarly, several fusion methods could be integrated into CELF-FD objective function. However, in this example, the goal is to illustrate the local fusion approach with a simple linear aggregation. Thus, we compare the results of CELF-FD with global fusion that uses the same aggregation method. We do this by simply setting the number of clusters to 1. In the subsequent experiments,

we will compare the performance of CELF to other state-of-the-art fusion methods. Using this setting, the global fusion assigns a 0.38 weight to classifier 1 and a 0.62 weight to classifier 2. This result seems to be logical since the overall accuracy of the second classifier is higher than that of the first one. Figure 5.7(a) displays the cumulative histogram of the confidences assigned by the global fusion algorithm. As it can be seen, the fusion cannot achieve perfect classification as the distribution of the two classes overlap. In fact, for a threshold of 0.5, the accuracy of the fusion is 81% which is not any better than the best individual classifier. These results, shown in Figure 5.7(b), are similar to those obtained by classifier 2 only.

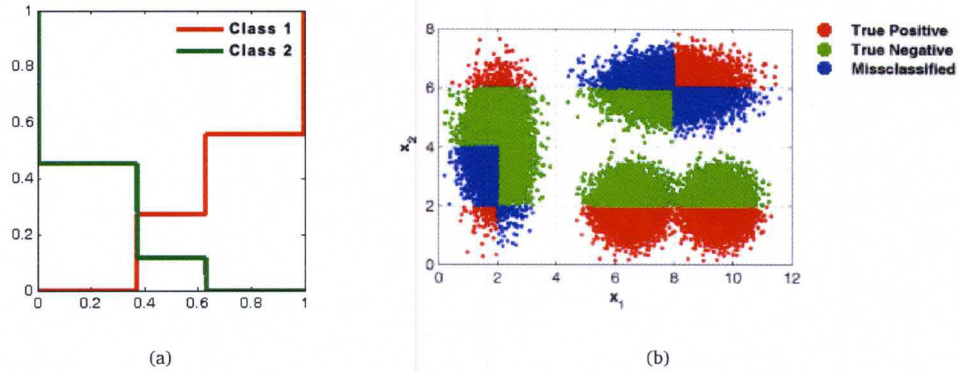


Figure 5.7: Fusion results using a global approach. (a) Cumulative histograms of the confidences assigned by the global fusion. (b) Assigned label when the threshold is fixed to 0.5.

Classification results with local fusion For this experiment, we report the results using CELF-FD with the number of clusters set to 4 and β set to 10. The four clusters are displayed in Figure 5.8(a). Initially, these clusters may appear incorrect as the ellipsoidal cluster in the left is split into two groups (clusters 2 and 4), and the two spherical clusters are merged into one. However, careful investigation of the classifiers performance in Figure 5.6 would explain this behavior. For instance, for the samples in the top part of the left cluster, classifier 1 has several misclassified samples while classifier 2 has none. For the bottom part, we have the opposite behavior. Thus, for the purpose of fusion, these two regions should be fused in

different ways. On the other hand, for the two spherical clusters, classifier 1 has a 50% correct classification rate; and classifier 2 has a 100% correct classification rate. Since the behavior of the two classifiers is consistent across this region, these two clusters are sufficiently close to each other, and the number of clusters was limited to 4, CELF-FD merges the two clusters.

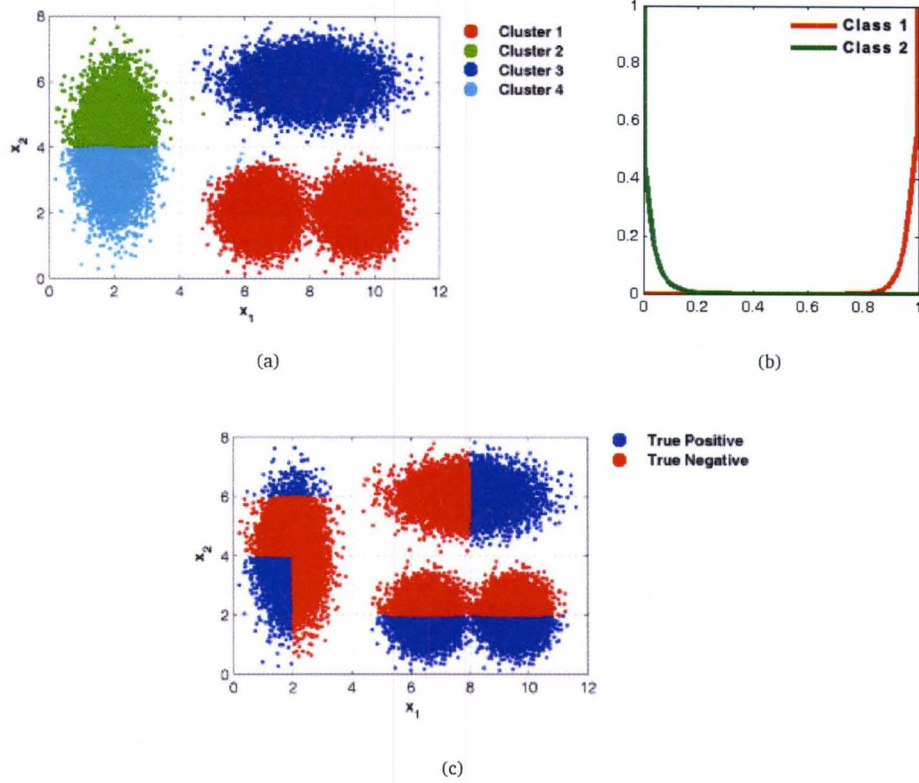


Figure 5.8: Local fusion results using CELF-FD. (a) Clustered samples in the feature space. A different color is used for each of the 4 clusters. (b) Cumulative histogram of the confidences assigned by the fusion algorithm. (c) fusion results (using 0.5 as threshold).

Table 5.2 displays the feature weights assigned by CELF-FD to each cluster. As it can be seen, the second feature is more reliable than the first one for cluster 1 and cluster 3. This can be explained by the shape of these two clusters; they are more stretched on the first feature space.

Table 5.3 shows the accuracy and the aggregation weights assigned to the two classifiers in each cluster. As it can be seen, algorithm 1 is more reliable for clusters 3

Table 5.2: Feature Weights assigned by CELF-FD

Clusters	1	2	3	4
Feature 1	0.31	0.50	0.33	0.49
Feature 2	0.69	0.50	0.67	0.51

and 4 and is selected as the dominant algorithm for these regions. Similarly, algorithm 2 is more reliable for the other two clusters, and is selected as the dominant algorithm.

Table 5.3: Accuracy of each classifier in each cluster and assigned weights by CELF-FD

Clusters		1	2	3	4
Accuracy	Algorithm 1	50.78%	59.81%	99.78%	85.90%
	Algorithm 2	99.01%	90.37%	54.17%	64.75%
Weights	Algorithm 1	0.0026	0.0002	0.9942	0.9978
	Algorithm 2	0.9974	0.9998	0.0058	0.0022

Figure 5.8(b) displays the histograms of the confidences generated by CELF-FD. As it can be seen, the two distributions are almost separable and any threshold in the $[0.3, 0.7]$ range would result in an accuracy of 99.7%. The classification results, using a 0.5 threshold, is shown in Figure 5.8(c).

Effect of the parameter β The performance of CELF depends on the chosen value of β . In the following, we investigate this dependency by varying the value of β and checking the effect on the fusion results. First, we pick a small value of β ($\beta = 5$). Figure 5.9(a) shows the obtained clusters. Comparing these results to those shown in Figure 5.5(b), we can see that we obtain almost the same clusters as those obtained by SCAD. In fact, when β is too small, the multi-algorithm fusion criteria (term 2 in (5.20)) is negligible compared to the clustering criteria (term 1 in (5.20)). As result, the optimal solution was not reached, and the fusion results, shown in Figures 5.9(b) and 5.9(c), confirm that using a small value of β , CELF is not able to achieve the optimal classification results.

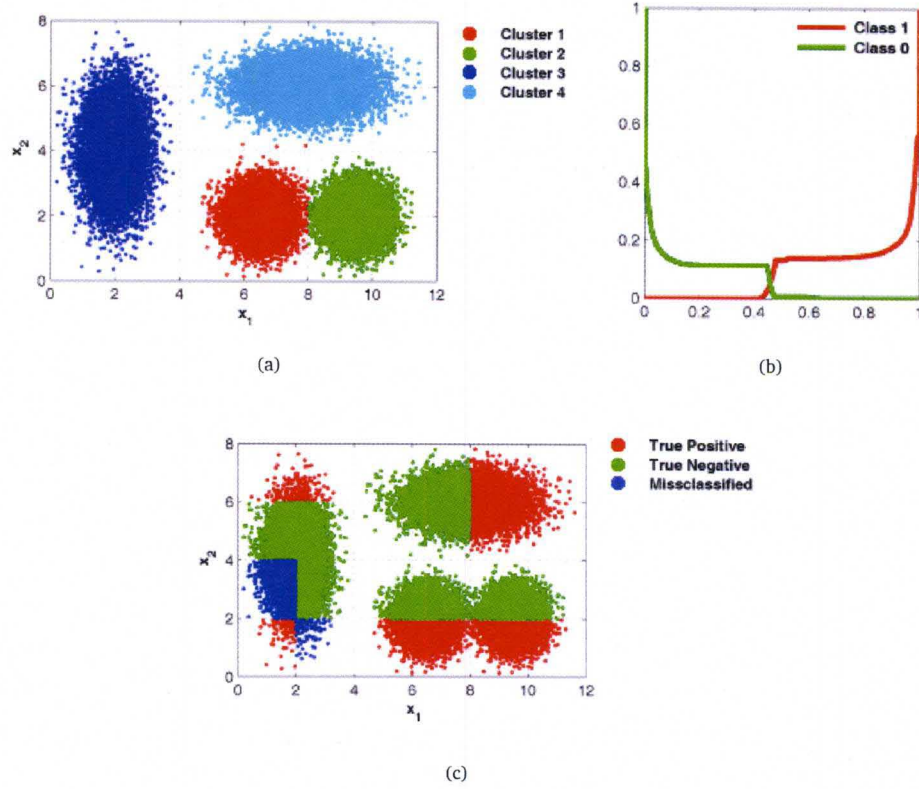


Figure 5.9: Local fusion results using a small value of β ($\beta = 5$). (a) Clustered samples in the feature space. (b) Cumulative histogram of the confidences assigned by CELF. (c) fusion results when the threshold is fixed to 0.5.

Figure 5.10 shows the results when a larger value of β ($\beta = 40$) is used. As it can be seen in Figure 5.10(a), some of the clusters become scattered. For instance, cluster 1 (black) becomes spatially split into 3 different regions. Even if the cumulative histogram and the fusion result shown in Figures 5.10(b) and 5.10(c) are correct, these results may not be reasonable. This is because the clusters do not share many common features (not similar in feature space) and our concept of context becomes not well defined. Moreover, during testing, the context identification step becomes almost random.

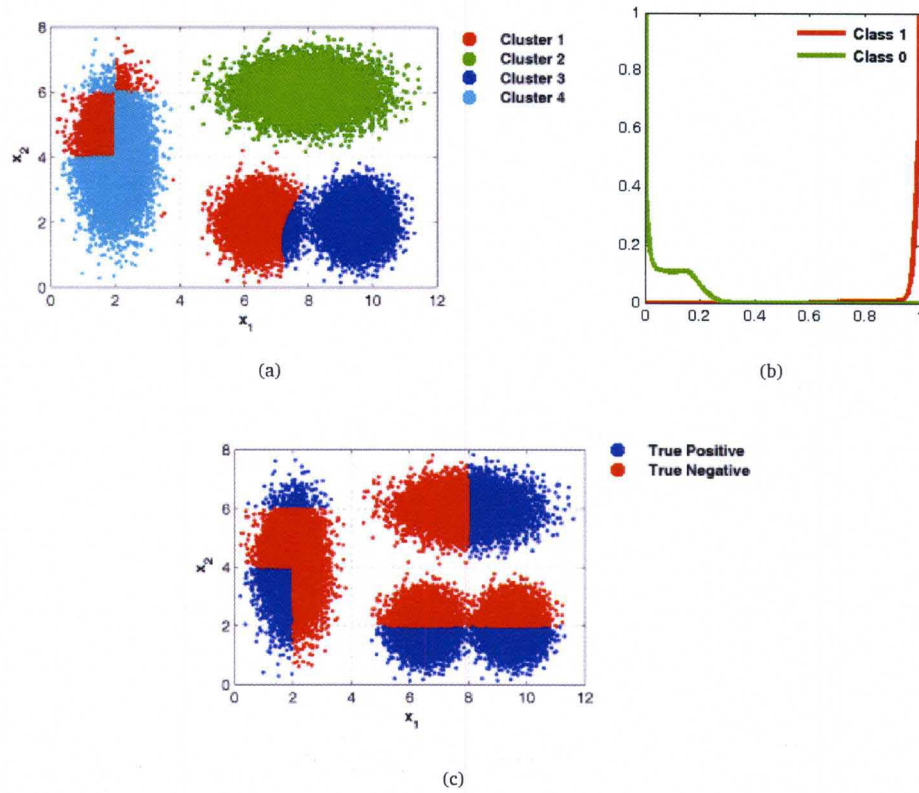


Figure 5.10: Local fusion result using a big value of β ($\beta = 40$). (a) Clustered samples in the feature space. (b) Cumulative histogram of the confidences assigned by the fusion algorithm. (c) fusion results when the threshold is fixed to 0.5.

5.3 CELF with Competitive Agglomeration

CELF and CELF-FD require the specification of the number of clusters. However in most applications, this information may not be known a priori. This problem has been addressed in unsupervised learning and several approaches have been developed [30, 31, 3, 115]. This problem is more acute in the proposed context extraction application. This is because the optimal number of clusters depends on the distribution of the data in the feature space as well as the behavior of the algorithms in the different regions. Thus, even if the data can be visualized in a lower dimensional space, the specification of the number of clusters is still a nontrivial task.

To address this issue, we propose extending the objective function in (5.20) to integrate a regularization term. The resulting algorithm, called Context Extraction for Local Fusion with Competitive Agglomeration (CELFC-CA), starts by partitioning the data into a large number of small clusters. As the algorithm progresses, adjacent clusters compete for data points, and clusters that lose the competition gradually become depleted and vanish. Thus, as the iterations proceed, we obtain a sequence of partitions with a progressively diminishing number of clusters. The final partition is taken to have the "optimal" number of clusters.

The CELFC-CA algorithm minimizes

$$J_3 = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^2 \sum_{k=1}^K v_{ik}^q d_{ijk}^2 + \sum_{i=1}^C \sum_{j=1}^N \alpha_i u_{ij}^2 \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2 - \gamma \sum_{i=1}^C \left(\sum_{j=1}^N u_{ij} \right)^2, \quad (5.26)$$

subject to the constraints in (5.3) and (5.21). As in (5.20), we let $\alpha_i = \beta \sum_{l=1}^K v_{il}^q$ where β is a constant. We should note here that in (5.26), the number of clusters C is not fixed.

The objective function in (5.26) has three components. The first two components, which are similar to those in the CELFC-FD objective function (with $m = 2$), combine clustering, feature discrimination, and multi-algorithm fusion. The global minimum of this component is achieved when the number of clusters C is equal to the number of samples N , i.e. each cluster contains a single data point. The last component in (5.26) is the sum of squares of the cardinalities of the clusters which allows us to control the number of clusters. The global minimum of this term (including the negative sign) is achieved when all points are lumped in one cluster, and all other clusters are empty. When both components are combined and γ is chosen properly, the final partition will minimize the sum of intra-cluster distances, while partitioning the data set into the smallest number of clusters possible. The clusters which are depleted as the algorithm proceeds will be discarded, as explained later.

To optimize J_3 with respect to $\mathbf{U} = [u_{ij}]$, we incorporate the constraints with the aid of Lagrange multipliers. We obtain

$$L = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^2 \sum_{k=1}^K v_{ik}^q d_{ijl}^2 + \beta \sum_{i=1}^C \sum_{j=1}^N u_{ij}^2 \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2 \left(\sum_{k=1}^K v_{ik}^q \right) - \gamma \sum_{i=1}^C \left(\sum_{j=1}^N u_{ij} \right)^2 - \sum_{j=1}^N \lambda_j \left(\sum_{i=1}^C u_{ij} - 1 \right), \quad (5.27)$$

where $\Lambda = [\lambda_1, \dots, \lambda_N]^t$ is a vector of Lagrange multipliers corresponding to the N constraints on \mathbf{U} in (5.21). Computing the derivative of L with respect to u_{ij} and setting it to 0, we obtain

$$u_{ij} = u_{ij}^{CELF} + u_{ij}^{Bias}, \quad (5.28)$$

where

$$u_{ij}^{CELF} = \left[\sum_{l=1}^C (D_{ij} / D_{lj}) \right]^{-1}, \quad (5.29)$$

and

$$u_{ij}^{Bias} = \frac{\gamma}{D_{ij}} (N_i - \tilde{N}_j). \quad (5.30)$$

In (5.29) and (5.30),

$$D_{ij} = \sum_{k=1}^K v_{ik}^q d_{ijk}^2 + \beta \left(\sum_{k=1}^K \omega_{ik} y_{kj} - t_j \right)^2 \left(\sum_{k=1}^K v_{ik}^q \right). \quad (5.31)$$

In (5.30),

$$N_i = \sum_{j=1}^N u_{ij} \quad (5.32)$$

is the fuzzy cardinality of cluster i , and

$$\tilde{N}_j = \left[\sum_{l=1}^C \frac{N_l}{D_{lj}} \right] / \left[\sum_{l=1}^C \frac{1}{D_{lj}} \right], \quad (5.33)$$

is simply a weighted average of the cluster cardinalities, where the weight of each cluster reflects its proximity to the feature point \mathbf{x}_j in question.

The first component of u_{ij} in (5.28), u_{ij}^{CELF} , is the membership term in the CELF-FD algorithm (refer to (5.22)). The second component, u_{ij}^{Bias} , is a signed bias term which depends on the difference between the cardinality of the cluster of interest, and the weighted average of cardinalities with respect to feature point \mathbf{x}_j . For clusters with cardinality higher than average, the bias term is positive, thus appreciating the membership value. On the other hand, for low cardinality clusters, the bias term is negative, thus depreciating the membership value. Moreover, this bias term is also inversely proportional to the distance of feature point \mathbf{x}_j to the cluster of interest \mathbf{c}_i , which serves as an amplification factor. This leads to a gradual reduction of the cardinality of spurious clusters. When the cardinality of a cluster drops below a threshold, we discard the cluster, and update the number of clusters. Since the initial partition has an over-specified number of clusters, each cluster is initially approximated by many small clusters in the beginning. As the algorithm proceeds, adjacent clusters compete. As a result, only few clusters will survive, while others will shrink and eventually become extinct.

Optimization of J_3 with respect to \mathbf{V} , \mathbf{W} , and \mathcal{C} yields the same equations (5.24), (5.7), and (5.10) as those derived for CELF-FD. The CELF-CA algorithm is summarized in Algorithm 5.3.

Illustrative example

To illustrate the behavior of CELF-CA, we use it to partition the same synthetic data shown in Figure 5.5(a). We fix the max number of clusters C_{max} to 10, and let these clusters compete. After 30 iterations, CELF-CA converged and the number of clusters reduced to 5. The final partition is shown in Figure 5.11. As it can be seen, CELF-CA succeeds in partitioning the data into compact and homogeneous clusters where the different algorithms behave consistently within each cluster. Compared to CELF-FD, CELF-CA split cluster 1 (in Figure 5.8(a)) into 2 different clusters (the

Algorithm 5.3 CELF with Competitive Agglomeration (CELF-CA)

Inputs: \mathcal{X} : the features of the data samples.

\mathcal{Y} : the confidences given to the data samples by the different classifiers.

\mathcal{T} : the labels of the data samples.

C_{max} : the maximum number of clusters.

β : the weight of the second term in the objective function.

q : the exponent of the feature weights, $q \in (1, +\infty)$.

ϵ : a given threshold.

Outputs: U : the fuzzy membership matrix of the data samples.

c : the cluster centers.

W : the confidence weights in each cluster.

V : the feature weights in each cluster.

- 1: Fix the maximum number of clusters $C = C_{max}$;
 - 2: Initialize U , W and V .
 - 3: Compute the initial cardinalities N_i for $1 \leq i \leq C$ using (5.32);
 - 4: **repeat**
 - 5: Update the partition matrix U using (5.28);
 - 6: Compute the cardinalities N_i for $1 \leq i \leq C$ using (5.32);
 - 7: **if** $N_i < \epsilon$ **then**
 - 8: discard cluster i ;
 - 9: **end if**
 - 10: Update the number of clusters C ;
 - 11: Update the centers using (5.10);
 - 12: Update V using (5.24).
 - 13: Update W using (5.7).
 - 14: **until** centers stabilize
 - 15: **return** c , U , V , W
-

black and the cyan); these two spherical clusters, even if they are close, should be separated.

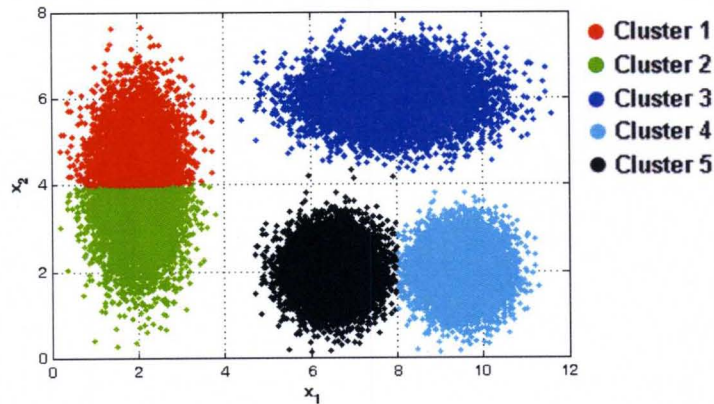


Figure 5.11: Clustered samples in the feature space using CELF-CA. The initial number of clusters was set to 10 and the algorithm converged to 5 distinct clusters.

Table 5.4 shows the accuracy and the aggregation weights assigned to the two classifiers in each cluster. As it can be seen, algorithm 1 is more reliable for clusters 3 and 4 and is selected as the dominant algorithm for these regions. Similarly, algorithm 2 is more reliable for the other two clusters, and is selected as the dominant algorithm.

Table 5.4: Accuracy of each classifier in each cluster and assigned weights by CELF-CA

Clusters		1	2	3	4	5
Accuracy	Algorithm 1	59.81%	85.90%	99.78%	50.92%	50.63%
	Algorithm 2	90.37%	64.75%	54.17%	99.02%	99.02%
Weights	Algorithm 1	0	1	1	0	0
	Algorithm 2	1	0	0	1	1

For this example, the fusion results of CELF-CA are similar to those obtained in Section 5.2. Figure 5.12(a) displays the histograms of the confidences generated by CELF-CA. As it can be seen, the two distributions are almost separable and any threshold in the $[0.3, 0.7]$ range would result in an accuracy of 99.7%. The classification results, using a 0.5 threshold, is shown in Figure 5.12(b).

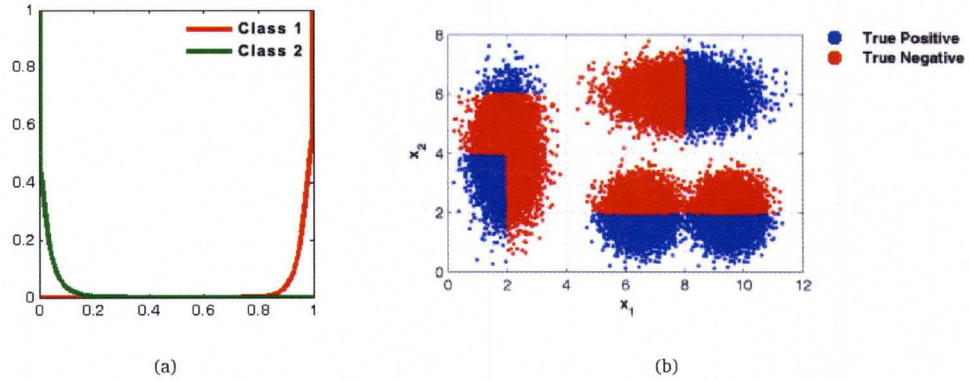


Figure 5.12: Local fusion results using CELF-CA. (a) Cumulative histogram of the confidences assigned by the fusion algorithm. (b) fusion results (using 0.5 as threshold).

5.4 CELF for Multi-Class Data

CELF was designed and developed to support two-class data. In the following, we propose generalizing the algorithm to cover data with multiple classes (CELF-M).

Given L classes, we assume that we have N training observations with desired outputs $\mathcal{T} = \{\mathbf{t}_j = [t_{j1}, \dots, t_{jL}] | j = 1, \dots, N\}$. t_{jl} is equal to 1 if the sample j is from the l^{th} class and 0 otherwise. These samples are processed by K algorithms. Each algorithm k extracts its own feature set, $\mathcal{X}_k = \{\mathbf{x}_{kj} | j = 1, \dots, N\}$, and generates confidence values, $\mathcal{Y}_k = \{\mathbf{y}_{kj} = [y_{k1j}, \dots, y_{kLj}] | j = 1, \dots, N\}$ where y_{klj} is the confidence assigned by classifier k to input j to be in the l^{th} class. The K feature sets are then concatenated to generate one global descriptor as in (5.1).

For simplicity, we formulate CELF-M objective function and optimize it for the case where no feature discrimination is used and the number of clusters is known. Extensions to find the optimal number of clusters and learn feature relevance weights are straightforward following steps similar to those used for CELF-CA and CELF-FD.

CELF-M partitions the feature space and learns the aggregation weights simultaneously by optimizing the following objective function.

$$J_M = \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \sum_{l=1}^L \left(\sum_{k=1}^K \omega_{ik} y_{klj} - t_{jl} \right)^2. \quad (5.34)$$

subject to the constraints in (5.3).

To optimize J_M with respect to $\mathbf{U} = [u_{ij}]$. We incorporate the constraints with the aid of Lagrange multipliers. We obtain

$$L^u = \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \sum_{l=1}^L \left(\sum_{k=1}^K \omega_{ik} y_{klj} - t_{jl} \right)^2 + \sum_{j=1}^N \lambda_j \left(\sum_{i=1}^C u_{ij} - 1 \right), \quad (5.35)$$

where $\Lambda = [\lambda_1, \dots, \lambda_N]^t$ is a vector of Lagrange multipliers corresponding to the N constraints on \mathbf{U} in (5.3). Since the memberships of the different observations

are independent of each other, the above optimization problem can be reduced to N simpler independent problems. For each pattern $j = 1, \dots, N$, we formulate the augmented functional

$$L_j^u = \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{i=1}^C u_{ij}^m \sum_{l=1}^L \left(\sum_{k=1}^K \omega_{ik} y_{klj} - t_{jl} \right)^2 + \lambda_j \left(\sum_{i=1}^C u_{ij} - 1 \right), \quad (5.36)$$

Computing the derivative of L_j^u with respect to u_{ij} and making it equal to 0, we obtain

$$\frac{\partial L_j^u}{\partial u_{ij}} = m u_{ij}^{m-1} D_{ij} + \lambda_j = 0, \quad (5.37)$$

where

$$D_{ij} = \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{l=1}^L \left(\sum_{k=1}^K \omega_{ik} y_{klj} - t_{jl} \right)^2. \quad (5.38)$$

Solving (5.37) for u_{ij} , we obtain

$$u_{ij} = \left(\frac{\lambda_j}{m} \right)^{1/(m-1)} \frac{1}{(D_{ij})^{1/(m-1)}}. \quad (5.39)$$

Taking into account the constraint $\sum_{p=1}^C u_{pj} = 1$, we obtain

$$\left(\frac{\lambda_j}{m} \right)^{1/(m-1)} \sum_{p=1}^C \frac{1}{(D_{pj})^{1/(m-1)}} = 1. \quad (5.40)$$

Solving (5.40) for $\left(\frac{\lambda_j}{m} \right)^{1/(m-1)}$ and substituting this in (5.39), we obtain

$$u_{ij} = \frac{1}{\sum_{l=1}^C (D_{lj}/D_{ij})^{\frac{1}{m-1}}}, \quad (5.41)$$

where D_{ij} is as defined in (5.38).

Derivation of the update equations for the cluster centers are straightforward, as no constraints are imposed on them. As for CELF, we fix $\mathbf{U} = [u_{ij}]$, and $\mathbf{W} = [\omega_{ik}]$, and

set the gradient to zero:

$$\frac{\partial J}{\partial \mathbf{c}_i} = 2 \sum_{j=1}^N u_{ij}^m (\mathbf{x}_j - \mathbf{c}_i) = \mathbf{0}. \quad (5.42)$$

We obtain

$$\mathbf{c}_i = \frac{\sum_{j=1}^N u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^m}. \quad (5.43)$$

To optimize J_M with respect to $\mathbf{W} = [\omega_{ik}]$, we incorporate the constraints using Lagrange multipliers and obtain

$$L = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \sum_{l=1}^L \left(\sum_{k=1}^K \omega_{ik} y_{klj} - t_{jl} \right)^2 + \sum_{i=1}^C \lambda_i \left(\sum_{k=1}^K \omega_{ik} - 1 \right), \quad (5.44)$$

where $\Lambda = [\lambda_1, \dots, \lambda_C]^t$ is a vector of Lagrange multipliers corresponding to the C constraints on \mathbf{W} in (5.3). Since the set of weights within each cluster are independent of each other, the above optimization problem could be reduced to C simpler independent problems. In particular, For $i = 1, \dots, C$, we formulate the augmented functional

$$L_i^\omega = \sum_{j=1}^N u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N u_{ij}^m \sum_{l=1}^L \left(\sum_{k=1}^K \omega_{ik} y_{klj} - t_{jl} \right)^2 + \lambda_i \left(\sum_{k=1}^K \omega_{ik} - 1 \right). \quad (5.45)$$

To obtain the optimal \mathbf{W} , we compute the derivative of L_i^ω with respect to ω_{ik} and set it to 0, i.e,

$$\frac{\partial L_i^\omega}{\partial \omega_{ik}} = 2\alpha \sum_{j=1}^N u_{ij}^m \sum_{l=1}^L y_{klj} \left(\sum_{k=1}^K \omega_{ik} y_{klj} - t_{jl} \right) + \lambda_i = 0. \quad (5.46)$$

Solving (5.46), we obtain

$$\omega_{ik} = \frac{\sum_{j=1}^N u_{ij}^m \sum_{l=1}^L y_{klj} \left(t_{jl} - \sum_{\substack{p=1 \\ p \neq k}}^K \omega_{ip} y_{plj} \right) - \frac{\lambda_i}{2\alpha}}{\sum_{j=1}^N u_{ij}^m \sum_{l=1}^L y_{klj}^2}. \quad (5.47)$$

The Lagrange constant λ_i could be solved using the constraint that $\sum_{l=1}^K \omega_{il} = 1$. Doing so, we obtain

$$\frac{\lambda_i}{2\alpha} = \frac{\sum_{p=1}^K \frac{\sum_{j=1}^N u_{ij}^m \sum_{l=1}^L y_{plj} \left(t_{jl} - \sum_{k=1}^K \omega_{ik} y_{klj} \right)}{\sum_{j=1}^N u_{ij}^m \sum_{l=1}^L y_{plj}^2}}{\sum_{p=1}^K \frac{1}{\sum_{j=1}^N u_{ij}^m \sum_{l=1}^L y_{plj}^2}}. \quad (5.48)$$

From equation (5.47), we can see that algorithm k will be assigned the highest weight, ω_{ik} , in cluster i if it is the most relevant classifier within this cluster. That is, its exclusion (in the K summation in the numerator) will result in the largest deviation from the desired output for samples with high memberships in this cluster.

The resulting algorithm is summarized in Algorithm 5.4.

Algorithm 5.4 CELF for Multi-class data (CELF-M)

Inputs: \mathcal{X} : the features of the data samples.

\mathcal{Y} : the confidences given to the data samples by the different classifiers.

\mathcal{T} : the labels of the data samples.

C : the number of clusters.

m : the fuzzifier, $m \in (1, +\infty)$.

α : the weight of the second term in the objective function.

Outputs: \mathbf{U} : the fuzzy membership matrix of the data samples.

\mathcal{C} : the cluster centers.

\mathbf{W} : the confidence weights in each cluster.

- 1: Initialize \mathbf{U} and \mathbf{W} .
 - 2: **repeat**
 - 3: Update \mathcal{C} using (5.43).
 - 4: Update \mathbf{W} using (5.47).
 - 5: Update \mathbf{U} using (5.41).
 - 6: **until** parameters do not change significantly
 - 7: **return** \mathcal{C} , \mathbf{U} , \mathbf{W}
-

Illustrative Example

To illustrate the behavior of CELF-M, we use it to partition and fuse a simple synthetic data. This data set is designed to illustrate the need for local fusion, and consists of 2,000 samples that belong to three classes: 500 samples from class 1, 1,000 samples from class 2, and 1,000 samples from class 3. Suppose that each sample has been processed by two different algorithms. Each algorithm, k , extracts one feature (x_k) and assigns one output value (y_k). Figure 5.13 displays this data in the 2-D feature space (x_1, x_2) where samples from class 1 are represented by red dots, and samples from class 2 are represented by green dots, and samples from class 3 are represented by black dots. As it can be seen, the data form 2 distinct clusters in the feature space.

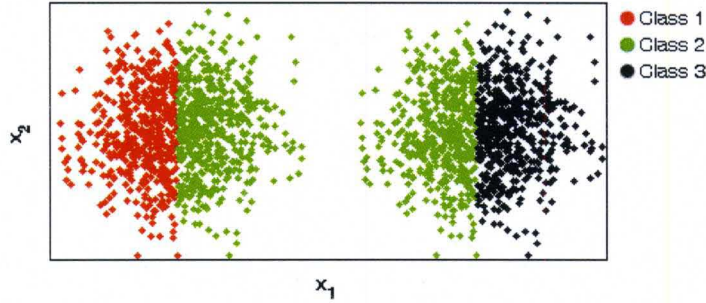


Figure 5.13: Synthetic data of a 3-class problem in the 2-D feature space.

In Figure 5.14, we display the classification results of the two classifiers. As it can be seen, none of the two classifiers classify this data perfectly as both figures include many misclassified samples. In fact, the accuracy of both classifiers is 75%. More importantly, the performance of each classifier varies in different regions of the feature space. For instance, in the left cluster, classifier 1 has an accuracy of 100%, and classifier 2 has an accuracy of 50%. On the other hand, for the right cluster, the accuracy of classifier 1 is 50%, and of classifier 2 is 100%.

Figure 5.15(a) illustrates the clustering result using CELF-M when the number of clusters C is set to 2. As it can be seen, our approach identifies the two clusters and

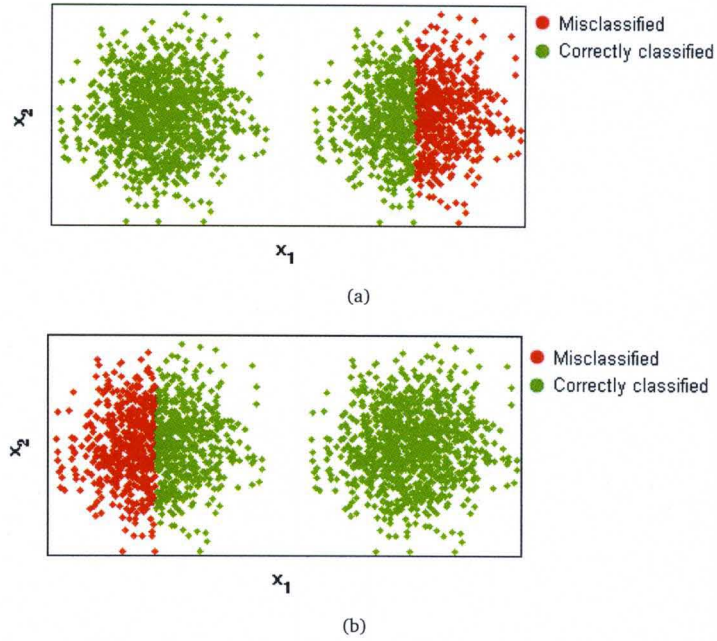


Figure 5.14: Classification results of (a) the first classifier and (b) the second classifier.

assigns the highest weight to the most reliable classifier in each context. Within the left cluster, CELF-M assigns a weight of 1 to the first classifier and a weight of 0 to the second one. Referring to Figure 5.14, we can see that, within this cluster, classifier 1 has an accuracy of 100% and classifier 2 has an accuracy of 50%. On the other hand, within the right cluster, CELF-M assigns a weight of 0 to the first classifier and 1 to the second one. In fact, within this cluster, classifier 1 has an accuracy of 50% and classifier 2 has an accuracy of 100%. The fusion result is shown in Figure 5.15(b) where CELF-M has an accuracy of 100%.

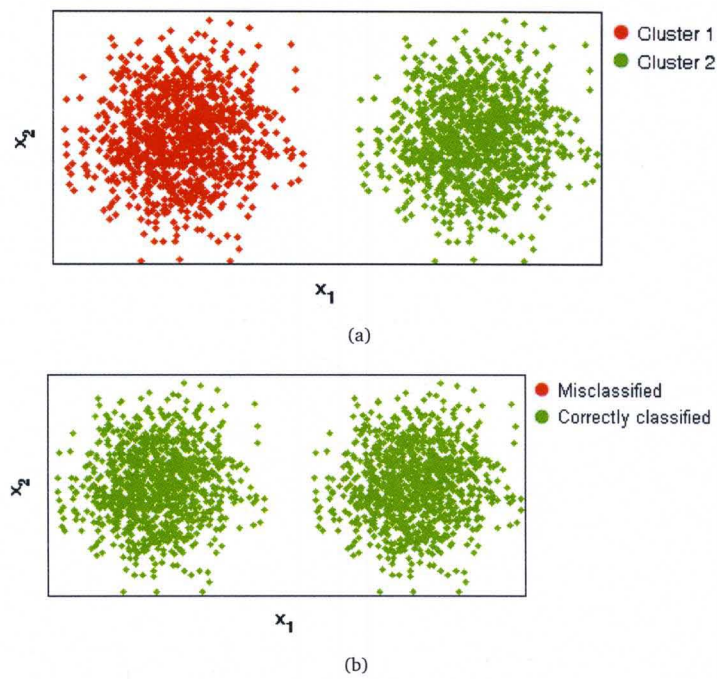


Figure 5.15: Local fusion results using CELF-M. (a) Clustered samples in the feature space. A different color is used for each cluster. (b) fusion results (using 0.5 as threshold).

NON-LINEAR LOCAL FUSION

In the previous chapter, we introduced our local fusion approach using a simple linear aggregation to assign weights to the individual classifiers. This may not be the optimal way to combine the algorithms within each context. To make the fusion of the algorithms' decisions for each context more effective, in this chapter, we propose extensions to CELF that use non-linear fusion approaches. In particular, we introduce two variants of CELF. The first one uses Neural Networks fusion, and the second one is based on Fuzzy Integrals fusion.

6.1 Local Fusion with Neural Networks

The proposed approach, called CELF with Neural Networks (CELF-NN), aims to partition the feature space into different contexts and, simultaneously, adapt a two-layers Neural Network to each context to fuse the individual confidence values. Each network has K inputs (the K classifiers' decision), H hidden neurons in the hidden layer, and L outputs (the L classes). Let f be the activation function, ρ_{khi} be the weight that connects the k^{th} input to the h^{th} neuron (of the hidden layer) of the i^{th} Neural Network, ψ_{hli} be the weight that connects the h^{th} neuron to the l^{th} output of

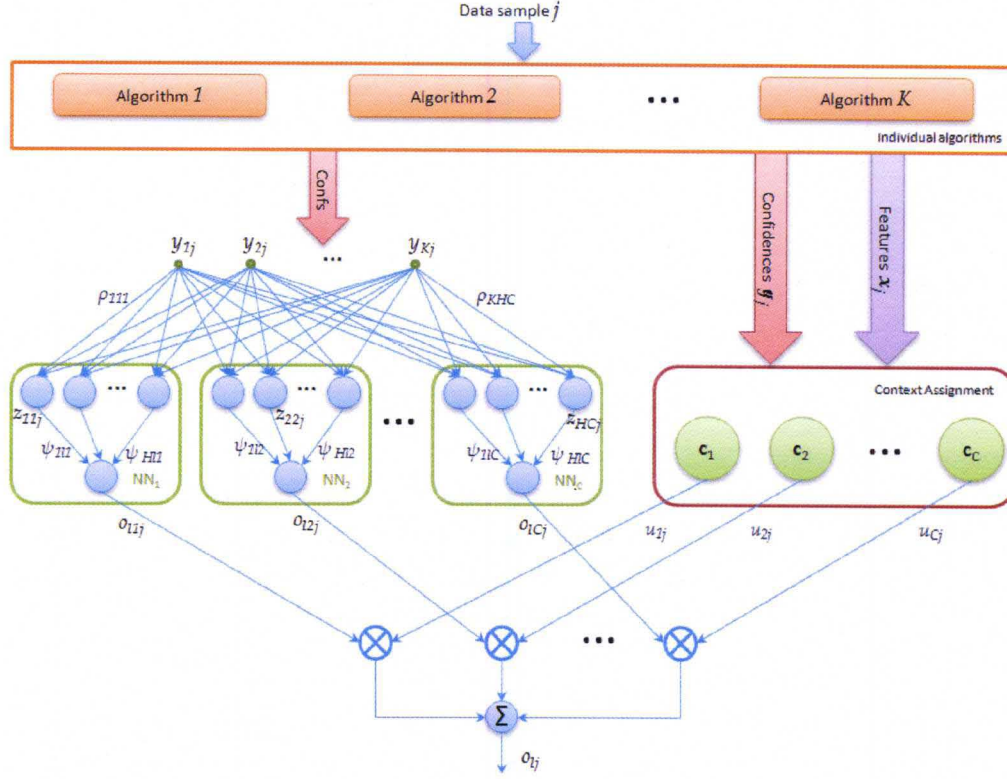


Figure 6.1: Architecture of the proposed CELF-NN

the i^{th} Neural Network, z_{hij} be the output of the h^{th} neuron (of the hidden layer) in the i^{th} Neural network for the sample j , and o_{lij} be the l^{th} output of the i^{th} Neural network for the sample j . Figure 6.1 displays the architecture of the proposed approach. For clarity, we display only the l^{th} output of each Neural Network. This figure highlights the two main components of the training phase, namely, context extraction and decision fusion. As in CELF, the context extraction step uses both the features extracted by various algorithms to partition the training input samples into C different contexts, i.e, each training sample j is assigned to each context j with a fuzzy membership u_{ij} . The decision fusion step uses the confidence values assigned by the individual algorithms to adapt a two-layers Neural Network to each context. The final output o_{lj} , for the sample j , is the weighted aggregation of the C Neural Networks' output, i.e.,

$$o_{lj} = \sum_{i=1}^C u_{ij} o_{lij} . \quad (6.1)$$

CELF-NN partitions the feature space and learns the weights of the different Neural Networks simultaneously by optimizing the following objective function.

$$J_{NN} = \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \sum_{l=1}^L (o_{lij} - t_{jl})^2. \quad (6.2)$$

subject to

$$\sum_{i=1}^C u_{ij} = 1 \quad \forall j, \quad \text{and} \quad u_{ij} \in [0, 1] \quad \forall i, j. \quad (6.3)$$

To optimize J_{NN} with respect to $\mathbf{U} = [u_{ij}]$. We incorporate the constraints with the aid of Lagrange multipliers. We obtain

$$L^u = \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m (o_j - t_j)^2 + \sum_{j=1}^N \lambda_j \left(\sum_{i=1}^C u_{ij} - 1 \right), \quad (6.4)$$

where $\Lambda = [\lambda_1, \dots, \lambda_N]^t$ is a vector of Lagrange multipliers corresponding to the N constraints on \mathbf{U} in (5.3). Since the memberships of the different observations are independent of each other, the above optimization problem can be reduced to N simpler independent problems. For each pattern $j = 1, 2, \dots, N$, we formulate the augmented functional

$$L_j^u = \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{i=1}^C u_{ij}^m (o_j - t_j)^2 + \lambda_j \left(\sum_{i=1}^C u_{ij} - 1 \right), \quad (6.5)$$

Computing the derivative of L_j^u with respect to u_{ij} and making it equal to 0, we obtain

$$\frac{\partial L_j^u}{\partial u_{ij}} = m u_{ij}^{m-1} D_{ij} + \lambda_j = 0, \quad (6.6)$$

where

$$D_{ij} = \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha (o_j - t_j)^2. \quad (6.7)$$

Solving (6.24) for u_{ij} , we obtain

$$u_{ij} = \left(\frac{\lambda_j}{m} \right)^{1/(m-1)} \frac{1}{(D_{ij})^{1/(m-1)}}. \quad (6.8)$$

Taking into account the constraint $\sum_{l=1}^C u_{lj} = 1$, we obtain

$$\left(\frac{\lambda_j}{m}\right)^{1/(m-1)} \sum_{l=1}^C \frac{1}{(D_{lj})^{1/(m-1)}} = 1. \quad (6.9)$$

Solving (6.27) for $\left(\frac{\lambda_j}{m}\right)^{1/(m-1)}$ and substituting this in (6.8), we obtain

$$u_{ij} = \frac{1}{\sum_{l=1}^C (D_{lj}/D_{ij})^{\frac{1}{m-1}}}, \quad (6.10)$$

where D_{ij} is as defined in (6.7).

The computations of the cluster centers are straightforward, as no constraints are imposed on them. To minimize J_{NN} with respect to the centers \mathbf{c}_{ik} , we fix $\mathbf{U} = [u_{ij}]$, $\mathbf{\Upsilon} = [\rho_{khi}]$, and $\mathbf{\Psi} = [\psi_{hli}]$, and set the gradient to zero:

$$\frac{\partial J}{\partial \mathbf{c}_i} = 2 \sum_{j=1}^N u_{ij}^m (\mathbf{x}_j - \mathbf{c}_i) = \mathbf{0}. \quad (6.11)$$

We obtain

$$\mathbf{c}_i = \frac{\sum_{j=1}^N u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^m}. \quad (6.12)$$

To adjust a weights of different layers of the neural network, we fix $\mathbf{C} = [\mathbf{c}_{ik}]$, and $\mathbf{U} = [u_{ij}]$, and optimize J_{NN} with respect to o_{ij} using gradient descent methods. Given a constant η , It can be shown that the weights need to be adjusted using:

$$\Delta \psi_{hli} = \eta \frac{\partial J_{NN}}{\partial o_{lij}} \times \frac{\partial o_{lij}}{\partial \psi_{hli}} \quad (6.13)$$

$$= \eta \delta_{o_{lij}} z_{hij}, \quad (6.14)$$

and

$$\Delta \rho_{khi} = \eta \frac{\partial J_{NN}}{\partial o_{lij}} \times \frac{\partial o_{lij}}{\partial z_{hij}} \times \frac{\partial z_{hij}}{\partial \rho_{khi}} \quad (6.15)$$

$$= \eta \delta_{z_{hij}} y_{kj}, \quad (6.16)$$

where

$$\delta_{o_{lij}} = 2\alpha u_{ij}^m (t_{lij} - o_{lij}) f'_{o_{lij}}, \quad (6.17)$$

and

$$\delta_{z_{hij}} = f'_{z_{hij}} \sum_{l=1}^L \delta_{o_{lij}} \psi_{hli}. \quad (6.18)$$

In (6.17) and (6.18), f' is the derivative of the activation function f . In this thesis, we use the bipolar sigmoidal and $f'_o = (1 - o^2)/2$.

Notice that, in (6.17), given a sample j and cluster i , the weight adjustment depends on the membership of the sample to the correspondent cluster. In fact, if sample j is typical of cluster i , its membership u_{ij} is close to 1. In this case, the weights of the neural network i are adjusted to minimize the error J_{NN} . On the other hand, if sample j is not likely to belong to cluster i , its membership u_{ij} would be close to 0. In this case, the weights of network i are not adjusted even if network i misclassifies the sample.

Inspired by the diagram presented in [132], Figure 6.2 illustrates the update process of the neural network designed for cluster i for a given a sample j and explains both the flow of the signal, and the flow of the error within the network. Using the gradient descent technique, The back propagation of the error $2\alpha u_{ij}^m (o_{lij} - t_{jl})$ is divided into functional steps such as calculation of the error signal vector $\delta_{o_{lij}}$ and calculation of the weight matrix adjustment $\Delta \psi_{hli}$ of the output layer. The diagram also illustrates the calculation of the internal error signal $\delta_{z_{hij}}$ and of the resulting weight adjustment $\Delta \rho_{khi}$ of the input layer.

The resulting algorithm is summarized in Algorithm 6.1.

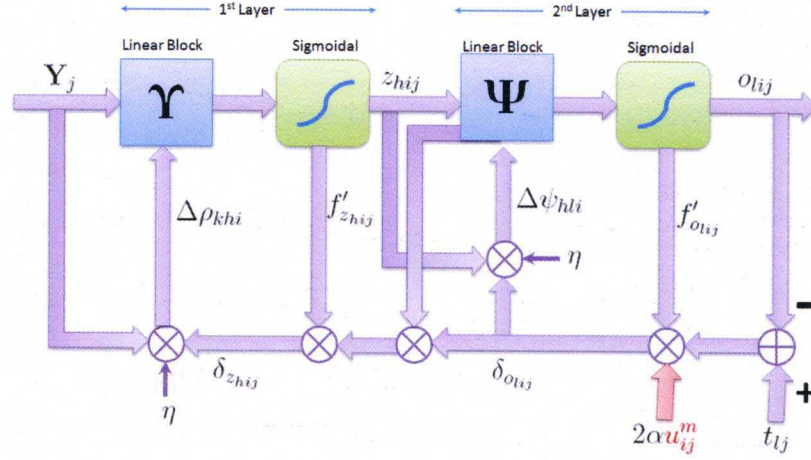


Figure 6.2: Block diagram illustrating signal flow for the error back-propagation algorithm.

Algorithm 6.1 CELF with Neural Networks (CELF-NN)

Inputs: \mathcal{X} : the features of the training data samples.

\mathcal{Y} : the confidences given to the data samples by the different classifiers.

\mathcal{T} : the labels of the data samples.

C : the number of clusters.

m : the fuzzifier, $m \in (1, +\infty)$.

α , the weight of the second term in the objective function.

η , the update coefficient of the NN.

L , the size of the hidden Layer.

Outputs: \mathbf{U} : the fuzzy membership matrix of the data samples.

\mathcal{C} : the cluster centers.

Ψ, Υ : The weights of C Neural Networks.

- 1: Initialize \mathbf{U} , Ψ , and Υ .
 - 2: **repeat**
 - 3: Update \mathcal{C} using (6.12).
 - 4: Update \mathbf{U} using (6.10).
 - 5: Update Ψ and Υ using (6.14) and (6.16).
 - 6: **until** parameters do not change significantly
 - 7: **return** \mathcal{C} , \mathbf{U} , Ψ , and Υ
-

Illustrative Example To illustrate the behavior of CELF-NN, we use it to partition and fuse a simple synthetic data. This data set is designed to illustrate the need for local fusion, and consists of 2,000 samples that belong to two classes: 1,266 samples from class 0 (*negative*) and 734 samples from class 1 (*positive*). Suppose that each sample has been processed by two different algorithms. Each algorithm, k , extracts one feature (x_k) and assigns one output value (y_k). Figure 6.3 displays

this data in the 2-D feature space (x_1, x_2) where samples from class 0 are represented by red dots and samples from class 1 are represented by green dots. As it can be seen, the data form 2 distinct clusters in the feature space, and each cluster has samples from both classes.

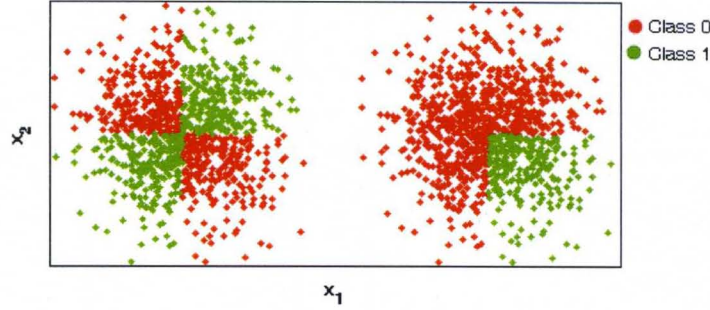


Figure 6.3: Synthetic data in the 2-D feature space. Class 1 samples are shown as red dots, class 2 samples are shown as green dots, and class 3 samples are shown as black dots.

In Figure 6.4, we display the classification results of the two classifiers. As it can be seen, none of the two classifiers classify this data perfectly as both figures include many misclassified samples. In fact, the accuracy of classifier 1 is 63.3%, and of classifier 2 is 61.5%.

To illustrate the performance of the local fusion approach, we compare the results of CELF-NN with the global Neural Network fusion and the baseline CELF approach (with linear aggregation). Figure 6.5(a) displays the cumulative histogram of the confidences assigned by the global fusion algorithm. As it can be seen, the fusion cannot achieve perfect classification as the distribution of the two classes overlap. In fact, for a threshold of 0.5, the accuracy of the fusion is 86.7%. The fusion result is shown in Figure 6.5(b). The cumulative histogram of the confidences assigned by the baseline CELF is displayed in Figure 6.6(a). For a threshold of 0.6, the accuracy of the fusion is 87.4%, which is almost similar to the accuracy obtained by the global Neural Networks fusion.

Figure 6.7(a) illustrates the clustering result using CELF-NN with the number of clusters C set to 2 and with the same parameters used in the global Neural Network

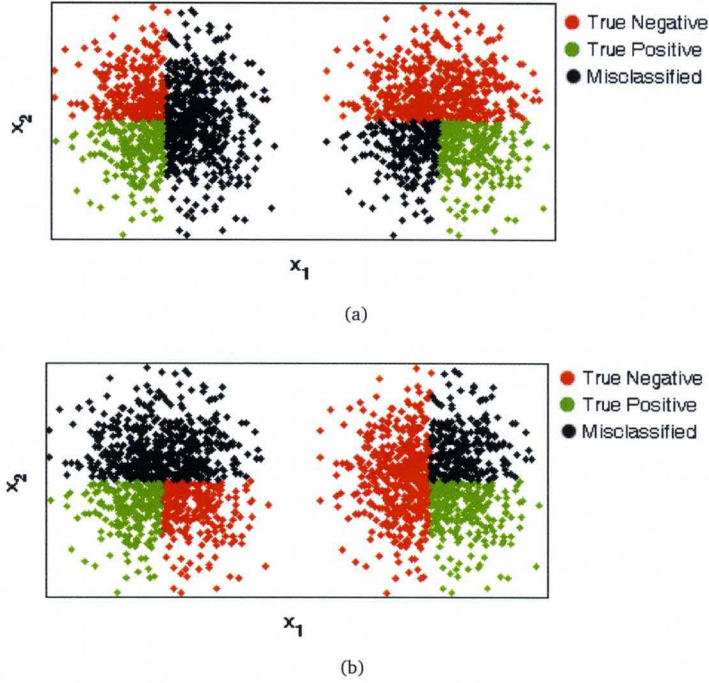


Figure 6.4: Classification result of (a) the first classifier and (b) the second classifier.

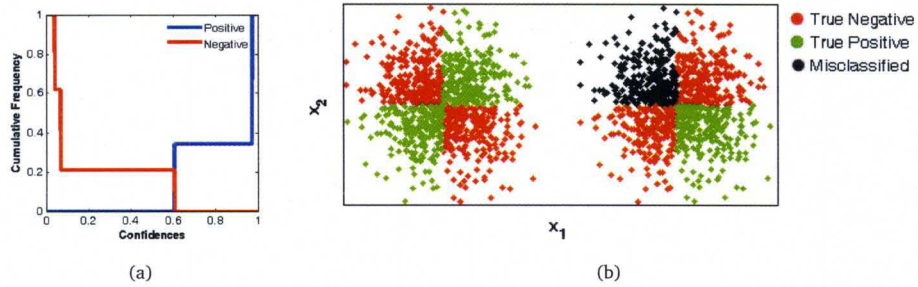


Figure 6.5: Fusion results using a global Neural Network approach. (a) Cumulative histograms of the confidences assigned by the global Neural Network fusion. (b) Assigned label when the threshold is fixed to 0.5.

fusion. As it can be seen, our approach identifies the two clusters. Figure 6.7(b) displays the histograms of the confidences generated by CELF-NN. As it can be seen, the two distributions are separable and any threshold in the $[0.3, 0.7]$ range would result in an accuracy of 100%. The classification results, using a 0.5 threshold, are shown in Figure 6.7(c).

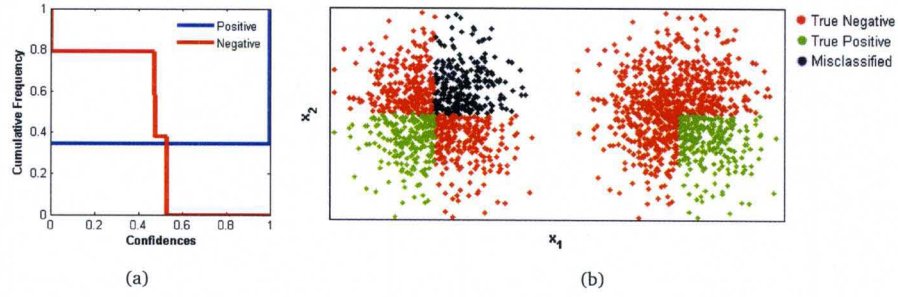


Figure 6.6: Fusion results using the baseline CELF approach. (a) Cumulative histograms of the confidences assigned by CELF. (b) Assigned label when the threshold is fixed to 0.6.

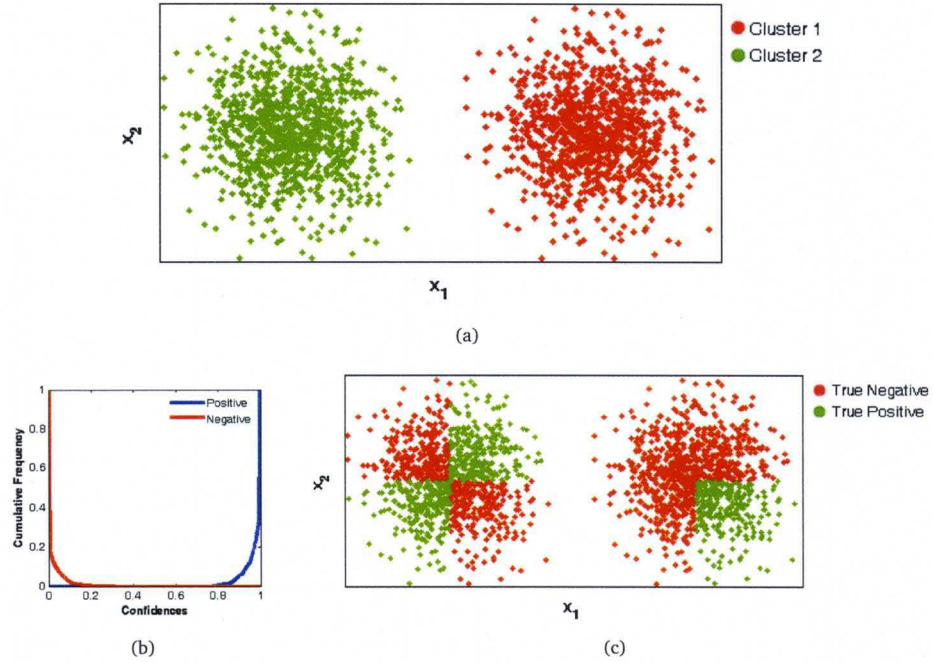


Figure 6.7: Local fusion results using CELF-NN. (a) Clustered samples in the feature space. A different color is used for each cluster. (b) Cumulative histogram of the confidences assigned by the fusion algorithm. (c) fusion results (using 0.5 as threshold).

6.2 Local Fusion with Fuzzy Integrals

Fusion methods based on the fuzzy integral [114] have the desirable property of assigning weights to subsets of classifiers to take into account the interaction between them. In the following, we propose generalizing CELF by replacing the linear fusion component with the fuzzy integral.

The proposed approach, called CELF with Fuzzy Integrals (CELF-FI), partitions the feature space and learns the fuzzy measures simultaneously by optimizing the following objective function.

$$J_{FI} = \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \left(C_{g_i}(\hat{\mathbf{y}}_j) - t_j \right)^2, \quad (6.19)$$

subject to

$$\sum_{i=1}^C u_{ij} = 1 \quad \forall j, \quad \text{and} \quad u_{ij} \in [0, 1] \quad \forall i, j. \quad (6.20)$$

In (6.19), $\hat{\mathbf{y}}_j = [\hat{y}_{1j}, \dots, \hat{y}_{Kj}]$ is the set of confidence values assigned by the K algorithms to sample j sorted in ascending order, g_i is the Sugeno measure associated with cluster i , and, C_{g_i} is the Choquet integral with respect to g_i . For each g_i , we associate a coefficient λ_i that satisfies (2.21) and (2.22). The objective function in (6.19) can be rewritten as

$$J_{FI} = \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \left(\sum_{k=1}^K [\hat{y}_{kj} - \hat{y}_{(k-1)j}] \cdot g_i(A_k) - t_j \right)^2, \quad (6.21)$$

where $A_k = \{k, \dots, K\}$ and $\hat{y}_{0j} = 0$.

To optimize J_{FI} with respect to $\mathbf{U} = [u_{ij}]$. We incorporate the constraints with the aid of Lagrange multipliers. We obtain

$$L^u = \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{j=1}^N \sum_{i=1}^C u_{ij}^m \left(C_{g_i}(\hat{\mathbf{y}}_j) - t_j \right)^2 + \sum_{j=1}^N \lambda_j \left(\sum_{i=1}^C u_{ij} - 1 \right), \quad (6.22)$$

where $\Lambda = [\lambda_1, \dots, \lambda_N]^t$ is a vector of Lagrange multipliers corresponding to the N constraints on \mathbf{U} in (5.3). Since the memberships of the different observations are independent of each other, the above optimization problem can be reduced to N simpler independent problems. For each pattern $j = 1, 2, \dots, N$, we formulate the augmented functional

$$L_j^u = \sum_{i=1}^C u_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha \sum_{i=1}^C u_{ij}^m (C_{g_i}(\hat{\mathbf{y}}_j) - t_j)^2 + \lambda_j \left(\sum_{i=1}^C u_{ij} - 1 \right), \quad (6.23)$$

Computing the derivative of L_j^u with respect to u_{ij} and making it equal to 0, we obtain

$$\frac{\partial L_j^u}{\partial u_{ij}} = m u_{ij}^{m-1} D_{ij} + \lambda_j = 0, \quad (6.24)$$

where

$$D_{ij} = \|\mathbf{x}_j - \mathbf{c}_i\|^2 + \alpha (C_{g_i}(\hat{\mathbf{y}}_j) - t_j)^2. \quad (6.25)$$

Solving (6.24) for u_{ij} , we obtain

$$u_{ij} = \left(\frac{\lambda_j}{m} \right)^{1/(m-1)} \frac{1}{(D_{ij})^{1/(m-1)}}. \quad (6.26)$$

Taking into account the constraint $\sum_{l=1}^C u_{lj} = 1$, we obtain

$$\left(\frac{\lambda_j}{m} \right)^{1/(m-1)} \sum_{l=1}^C \frac{1}{(D_{lj})^{1/(m-1)}} = 1. \quad (6.27)$$

Solving (6.27) for $\left(\frac{\lambda_j}{m} \right)^{1/(m-1)}$ and substituting this in (6.26), we obtain

$$u_{ij} = \frac{1}{\sum_{l=1}^C (D_{lj}/D_{ij})^{\frac{1}{m-1}}}, \quad (6.28)$$

where D_{ij} is as defined in (6.25).

To minimize J_{FI} with respect to the centers \mathbf{c}_{ik} , we fix $\mathbf{U} = [u_{ij}]$, and $\mathbf{G} = [g_{ik}]$, and set the gradient to zero:

$$\frac{\partial J}{\partial \mathbf{c}_i} = 2 \sum_{j=1}^N u_{ij}^m (\mathbf{x}_j - \mathbf{c}_i) = \mathbf{0}. \quad (6.29)$$

We obtain

$$\mathbf{c}_i = \frac{\sum_{j=1}^N u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^m}. \quad (6.30)$$

Differentiation of (6.19) with respect to the Sugeno measure, g_i , does not have a closed-form solution. Thus, we use a gradient descent approach and update it in every iteration. As a convention, the measure of a singleton set $\{l\}$ is called a density and is denoted by $g_{il} = g_i(\{l\})$. Given a learning rate η , the density g_{il} is updated using

$$g_{il} = g_{il} - \eta \frac{\partial J_{FI}}{\partial g_{il}}, \quad (6.31)$$

where

$$\frac{\partial J_{FI}}{\partial g_{il}} = 2\alpha \sum_{j=1}^N u_{ij}^m \left(\sum_{k=1}^K [\hat{y}_{kj} - \hat{y}_{(k-1)j}] \cdot g_i(A_k) - t_j \right) \left(\sum_{k=1}^K [\hat{y}_{kj} - \hat{y}_{(k-1)j}] \cdot \frac{\partial g_i(A_k)}{\partial g_{il}} \right). \quad (6.32)$$

Notice that, in (6.32), given a sample j and cluster i , the weight adjustment depends on the membership of the sample to the correspondent cluster. In fact, if the sample j is typical of cluster i , its membership u_{ij} is close to 1. In this case, g_i is adjusted to minimize the error J_{FI} . On the other hand, if the sample j is not typical of cluster i , its membership u_{ij} is close to 0. Therefore, g_i is not adjusted even if the fuser misclassifies this sample.

To calculate the partial derivative $\partial g_i(A_k)/\partial g_{il}$, we use an approach similar to the one in [88], and obtain the following cases:

Case 1: $k \neq K$ and $k = l$

$$\frac{\partial g_i(A_k)}{\partial g_{il}} = 1 + \lambda_i g_i(A_{k+1}) + g_{ik} g_i(A_{k+1}) \frac{\partial \lambda_i}{\partial g_{il}} + (1 + \lambda_i g_{ik}) \frac{\partial g_i(A_{k+1})}{\partial g_{il}}. \quad (6.33)$$

Case 2: $k \neq K$ and $k \neq l$

$$\frac{\partial g_i(A_k)}{\partial g_{il}} = g_{ik} g_i(A_{k+1}) \frac{\partial \lambda_i}{\partial g_{il}} + (1 + \lambda_i g_{ik}) \frac{\partial g_i(A_{k+1})}{\partial g_{il}}. \quad (6.34)$$

Case 3: $k = K$ and $k = l$

$$\frac{\partial g_i(A_k)}{\partial g_{il}} = 1. \quad (6.35)$$

Case 4: $k = K$ and $k \neq l$

$$\frac{\partial g_i(A_k)}{\partial g_{il}} = 0. \quad (6.36)$$

In (6.33) and (6.34),

$$\frac{\partial \lambda_i}{\partial g_{il}} = \begin{cases} \frac{\lambda_i^2 + \lambda_i}{(1 + \lambda_i g_{il}) \left(1 - (\lambda_i + 1) \sum_{k=1}^K \frac{g_{ik}}{1 + \lambda_i g_{ik}} \right)} & \text{if } \lambda \neq 0. \\ K & \text{if } \lambda = 0. \end{cases} \quad (6.37)$$

The resulting algorithm is summarized below in Algorithm 6.2.

Illustrative Example To illustrate the behavior of the proposed fusion approach, we first use it to partition and fuse a simple synthetic data. This data set is designed to illustrate the need for local fusion, and consists of 2,000 samples that belong to two classes: 1,000 samples from class 0 (*negative*) and 1,000 samples from class 1 (*positive*). Suppose that each sample has been processed by three different algorithms. Each algorithm, k , extracts one feature (x_k) and assigns one output value (y_k). Figure 6.8 displays this data in the 2-D feature space (x_1, x_2) where

Algorithm 6.2 CELF with Fuzzy Integrals (CELF-FI)

Inputs: \mathcal{X} : the features of the training data samples.

\mathcal{Y} : the confidences given by the different classifiers.

\mathcal{T} : the labels of the data samples.

C : the number of clusters.

m : the fuzzifier, $m \in (1, +\infty)$.

α , the weight of the second term in the objective function.

η , the learning rate.

Outputs: \mathbf{U} : the fuzzy membership matrix of the data samples.

\mathcal{C} : the cluster centers.

$\mathbf{G} = [g_{ik}]$: Sugeno measures.

1: Initialize \mathbf{U} and \mathbf{G} .

2: **repeat**

3: Update \mathcal{C} using (6.30).

4: Update \mathbf{U} using (6.28).

5: Update \mathbf{G} using (6.31) for few iterations.

6: **until** parameters do not change significantly

7: **return** \mathcal{C} , \mathbf{U} , and \mathbf{G}

samples from class 0 are represented by red dots and samples from class 1 are represented by green dots. As it can be seen, the data form 2 distinct clusters in the feature space, and each cluster has samples from both classes.

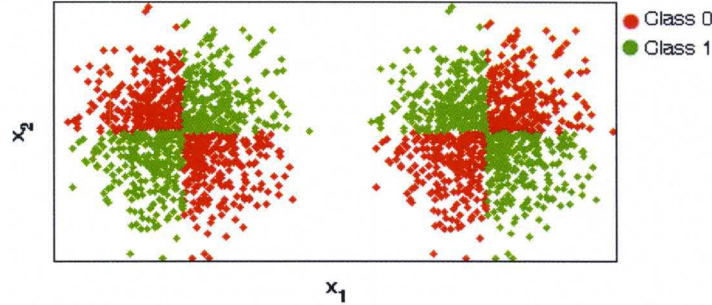


Figure 6.8: Synthetic data in the 2-D feature space. Class 0 samples are shown as red dots and class 1 samples are shown as green dots.

In Figure 6.9, we display the classification results of the three classifiers. As it can be seen, none of the three classifiers classify this data perfectly as all three figures include many misclassified samples. In fact, the accuracy of classifier 1 is 63.2%, of classifier 2 is 74.9%, and of classifier 3 is 61.7%. More importantly, the performance of each classifier varies in different regions of the feature space. For instance, in the left cluster, classifier 1 has an accuracy of 75.4%, and classifier 3 has an accuracy

of 49%. On the other hand, for the right cluster, the accuracy of classifier 1 is 51%, and of classifier 3 is 74.4%.

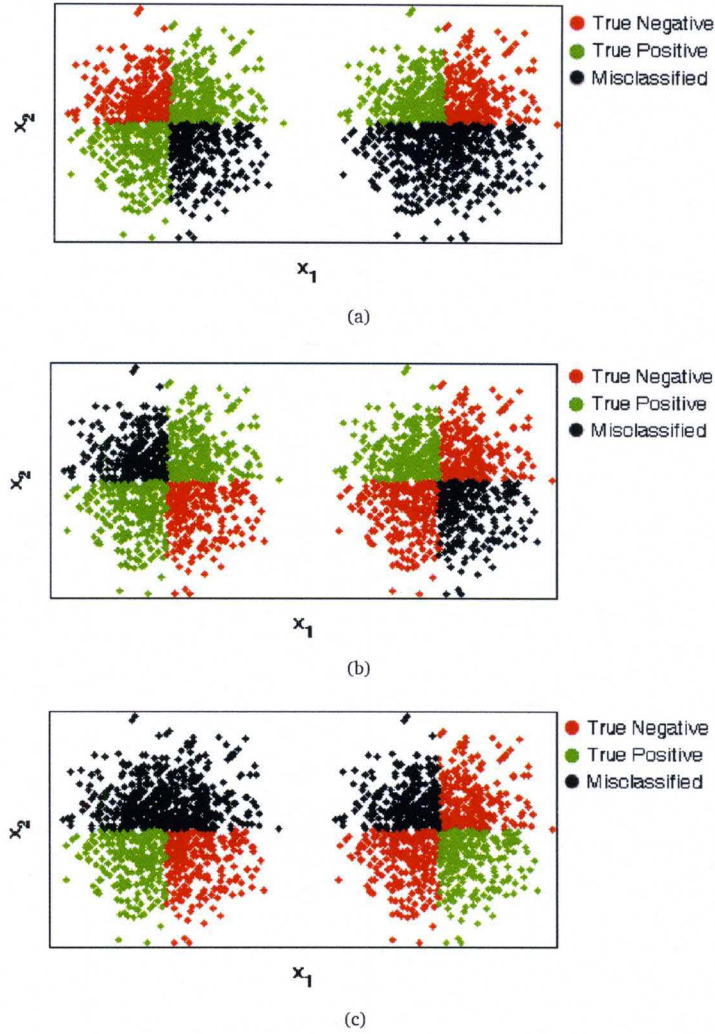


Figure 6.9: Classification result of (a) the first classifier, (b) the second classifier, and (c) the third classifier.

To illustrate the performance of CELF-FI, we compare the results of CELF-FI with the fusion using global fuzzy integral and the baseline CELF approach. Figure 6.10(a) displays the cumulative histogram of the confidences assigned by the global fusion. As it can be seen, the fusion cannot achieve perfect classification as the distribution of the two classes overlap. In fact, for a threshold of 0.5, the accuracy of the fusion is 74.9% which is not any better than the best individual classifier. These results, shown in Figure 6.10(b), are similar to those obtained by classifier 2

only. The cumulative histogram of the confidences assigned by the baseline CELF is displayed in Figure 6.11(a). For a threshold of 0.5, the accuracy of the fusion is 77.7%, which is almost similar to the accuracy obtained by the global fuzzy integral fusion.

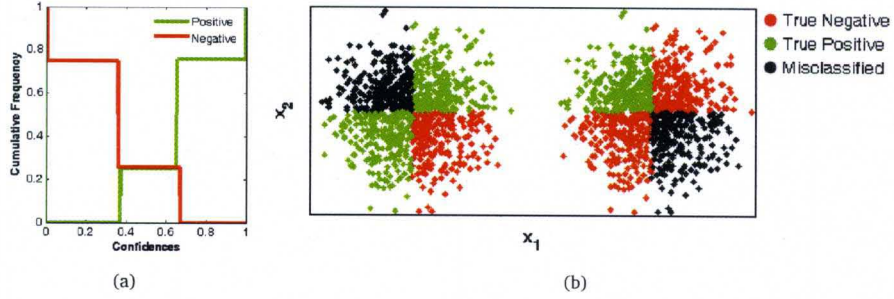


Figure 6.10: Fusion results using a global fuzzy integral approach. (a) Cumulative histograms of the confidences assigned by the global fuzzy integral fusion. (b) Assigned label when the threshold is fixed to 0.5.

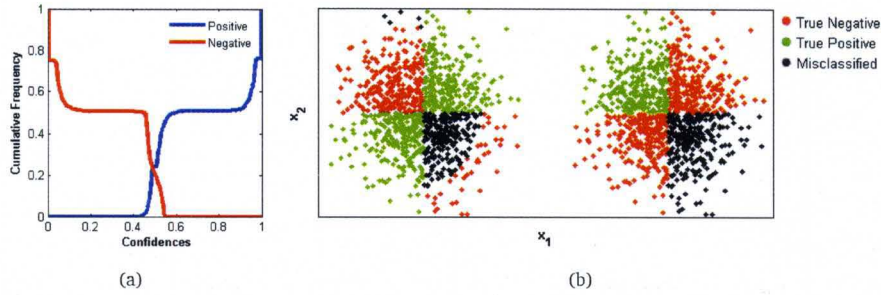


Figure 6.11: Fusion results using CELF. (a) Cumulative histograms of the confidences assigned by CELF. (b) Assigned label when the threshold is fixed to 0.5.

Figure 6.12(a) illustrates the clustering result using CELF-FI with the number of clusters C set to 2, the fuzzifier m set to 2, the parameter α set to 10, and the learning rate η set to 0.1 (same as the one used in the global fuzzy integral). Figure 6.12(b) displays the histograms of the confidences generated by CELF-FI. As it can be seen, the two distributions are separable and any threshold in the $[0.3, 0.7]$ range would result in an accuracy of 100%. The classification results, using a 0.5 threshold, are shown in Figure 6.12(c).

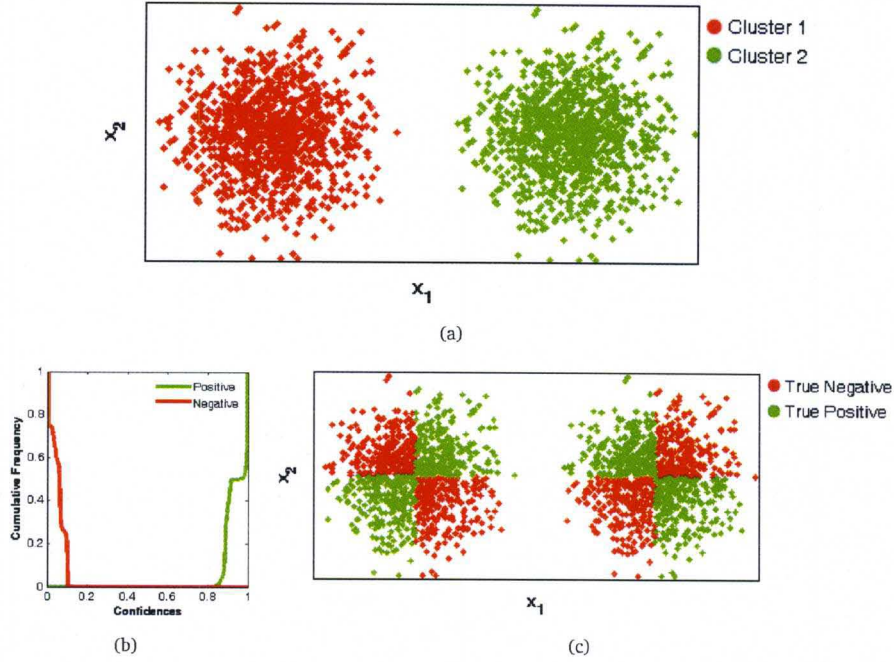


Figure 6.12: Local fusion results using CELF-FI. (a) Clustered samples in the feature space. A different color is used for each cluster. (b) Cumulative histogram of the confidences assigned by the fusion algorithm. (c) fusion results (using 0.5 as threshold).

In order to gain further insight into the behavior of the local and the global approach, in Figure 6.13, we display the Shapley index of each classifier, and in Figure 6.14, we display the interaction indices between each pair of classifiers assigned by the global fuzzy integral and CELF-FI (within each cluster).

In Figure 6.13, we can see that the global fusion assigned roughly the same Shapley score to each algorithm; which means that all the classifiers contribute in the fusion result with approximately the same part. This is expected since the 3 classifiers have comparable overall performance. The proposed CELF-FI, on the other hand, assigns cluster dependent values to the Shapley indices. In particular, for cluster 1 a high Shapley index is assigned to the first two classifiers, and a low Shapley index is assigned to the third classifier. However for cluster 2, CELF-FI assigns a high Shapley index to the last two classifiers, and assigns a low Shapley index to the first one. In fact, in order to obtain better fusion results, CELF-FI learned to discard the third classifier in cluster 1, and to discard the first classifier in cluster 2. Referring

to Figure 6.9, we can see that the third classifier has the worst accuracy (49%) in cluster 1, and the first classifier has the worst accuracy (51%) in cluster 2.

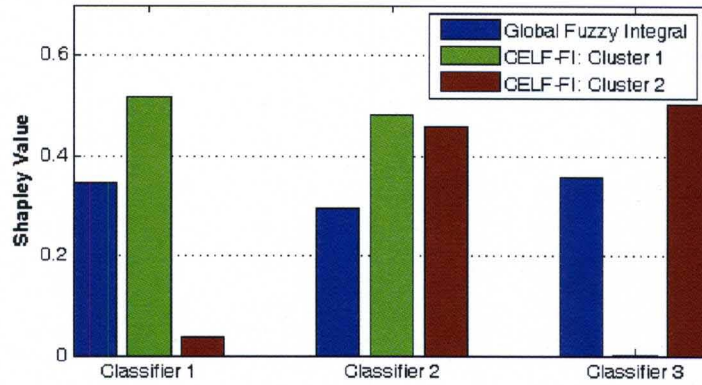


Figure 6.13: Shapley values of the different classifiers assigned by the global fusion and the local fusion (within each cluster).

In Figure 6.14, we can see that, the global fusion with fuzzy integral assigns null interaction indexes to all classifiers' pairs. Thus, a linear aggregation was used to fuse the three classifiers. On the other hand, the local approach assigns a positive interaction index to (classifier 1, classifier 2) within cluster 1, and a negative interaction index to (classifier 2, classifier 3) within cluster 2. In fact, when we refer to Figure 6.9, we can see that within cluster 1, classifiers 1 and 2 have to be both satisfied in order to detect samples from Class 1. However, within cluster 2, it is sufficient to satisfy classifier 2 or 3.

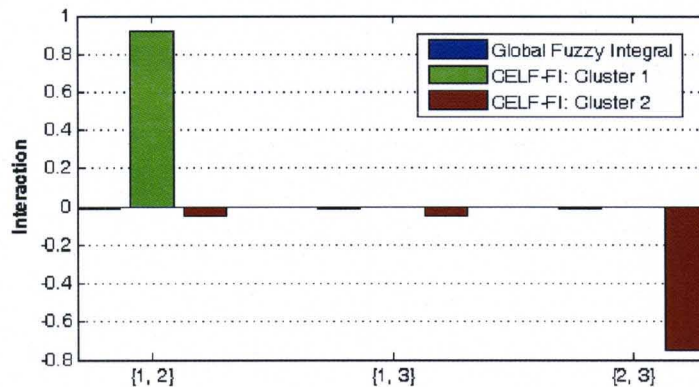
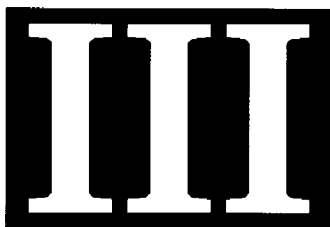


Figure 6.14: Interactions indices of the different pair of classifiers assigned by the global fusion and the local fusion (within each cluster).



EXPERIMENTAL RESULTS

APPLICATION TO LANDMINE DETECTION

In this chapter, we apply the proposed fusion method to the problem of land mine detection. We fuse the output of several landmine detection algorithms that have different preprocessing, different features, and different classification approaches in order to improve the detection performance. In particular, CELF was applied to two different data collections. The first dataset was collected using the NIITEK vehicle mounted Ground Penetration Radar(GPR) system, and CELF was used to fuse the results of four detection algorithms. The second one contains data collected using the *Autonomous Mine Detection System* (AMDS). The latter system has two sensors: GPR and Wideband ElectroMagnetic Induction (WEMI). Different algorithms were used for each sensor. In this case, CELF was used for multi-sensor multi-algorithm fusion to combine the outputs of four different algorithms.

For this application, since it involves high dimensional feature spaces, where the samples are highly sparse, we use CELF with the feature relevance weighting option, and a low fuzzifier value ($m = 1.2$) in order to obtain consistent clusters. As a result, we could not find the optimal number of cluster using CELF-CA. This is because CELF-CA requires that $m = 2$. Consequently, we fix the number of clusters, C , and

we assume that this number is sufficient to cover the different regions of the feature space.

7.1 Landmine Detection Using a Vehicle Mounted GPR System

7.1.1 Data Collection

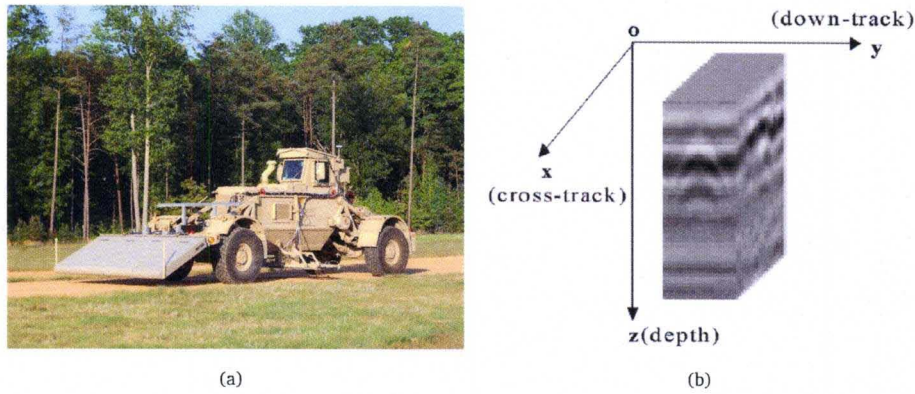


Figure 7.1: GPR data collection. (a) Vehicle-mounted GPR system. (b) An example of GPR scans.

The data used in this experiment consist of a sequence of raw GPR measurements collected by a vehicle-mounted GPR array [54] (see Figure 7.1(a)). The GPR collects 24 channels of data. Adjacent channels are spaced approximately 5 centimeters apart in the cross-track direction, and sequences (or scans) are taken at approximately 5 centimeter down-track intervals. The system uses an antenna that generates a wide-band pulse from 200 MHz to 7GHz. Each A-scan, that is, the measured waveform collected in one channel at one down-track position, contains 416 time samples, each corresponding to roughly 8 picoseconds. We often refer to the time index as depth although, since the radar wave travels through different media, this index does not represent a uniform sampling of depth. Thus, we model GPR input data as a three-dimensional matrix of sample values, $S(z, x, y)$, $z = 1, \dots, 416$,

$x = 1, \dots, 24$, $y = 1, \dots, N_s$, where N_s is the total number of collected scans, and the indices z , x , and y represent depth, cross-track position, and down-track positions respectively. A sample volume of GPR input data is illustrated in Figure 7.1(b). Figure 7.2 displays down-track B-scans (sequences of A-scans from a single channel) and cross-track B-scans (sequences of A-scans from a single scan). The surveyed object position is highlighted in each figure. The objects scanned are a) a high-metal content antitank mine, b) a low-metal antitank mine, and d) a wood block.

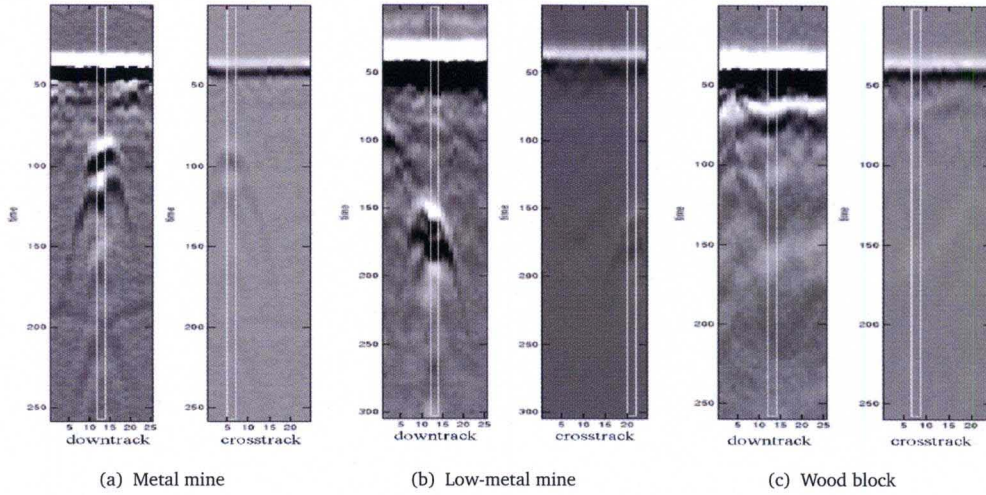


Figure 7.2: NIITEK Radar down-track and cross-track B-scans pairs for (a) a high-metal mine, (b) a low-metal mine, and (c) a wood block.

In our experiments, we use data collected between November 2002 and July 2006 from 4 geographically distinct test sites (A, B, C and D). Sites A, B, and D are temperate climate test facilities with prepared soil and gravel lanes. Site C is an arid climate test facility with prepared soil lanes. The statistics of the data are shown in Table 7.1. Site B has the largest number of collections and the largest number of alarms. The data collected from Sites B and D have emplaced buried clutter. Although the lanes at Sites A and C are prepared, they still contain non-emplaced clutter objects. Both metal and non-metal non-emplaced clutter objects such as ploughshares, shell casings, and large rocks have been excavated from these sites. The emplaced clutter objects include steel scraps, bolts, sort-drink cans, concrete blocks, plastic bottles, wood blocks, and rocks. In all, there are 12 collections

having 19 distinct mine types that can be classified into 3 categories: *anti-tank metal*(ATM), *anti-tank with low metal content* (ATLM), and *simulated mines* (SIM). The targets were buried up to 6 inches deep. Many of these mine types are present at several sites. The data include 1,560 mine encounters in a sample ground area of 41,807.57/ m^2 .

Table 7.1: Statistics of dataset 1

Site	Site A	Site B	Site C	Site D	Total
# Lanes	3	6	2	1	12
# Mine Types	9	15	9	5	19
# Mine Alarms	183	821	62	494	1560
# Clutter encounters	0	15	0	196	211
# Clutter Alarms post prescreen	0	4	0	46	50
Area (m^2)	14,812.83	15,630.62	4,054.39	7,309.73	41,807.57

The distribution of mine targets at different depths is shown in Table 7.2. As it can be seen, mines buried at 1" through 6" occupy 87.5% of the total targets encountered vs. 12.5% surface-laid or flush-buried mines.

Table 7.2: Burial depth of mines in dataset 1

Depth	Surface	0"	1"	2"	3"	4"	5"	6"	Total
ATLM	12	92	90	204	122	134	47	76	777
ATM	6	37	124	68	151	34	119	77	616
SIM	48	0	20	47	23	29	0	0	167
Total	66	129	234	319	296	197	166	153	1560

7.1.2 Evaluation Method

The individual algorithms and the fusion approach were implemented for use within the Testing/training Unified Framework (TUF) System with lane-based cross-validation (in which each mine lane is in turn treated as a test set with the rest of the lanes

used for training). The results of this process are scored using the Mine Detection Assessment and Scoring (Midas) system developed by the institute for Defense Analysis [6].

Since the set of potential false alarm locations is infinite (limited only by the precision of the marking system), we cannot consider typical *receiver operating characteristic* (ROC) curves comparing probability of detection (PD) vs. probability of false alarm (PFA) because the denominator in the PFA calculation is not well defined. For this reason, The scoring is performed in terms of Probability of Detection (PD) vs. False Alarm Rate (FAR). Confidence values are threshold at different levels to produce *Free-Response Receiver Operating Characteristic* (FROC) curve.

During data collection, a global positioning system (GPS) was used with known locations of buried landmines to generate ground truth files that indicate the approximate locations of the landmine signatures in the GPR data files. For scoring purposes, alarms within a certain radial distance of 25 *cm* from the edge of a mine are considered detections and alarms more than 25 *cm* from landmine edges are considered false alarms. Then, given a threshold, the PD is defined to be the number of mines detected divided by the number of mines. The FAR is defined as the number of false alarms per square meter.

It is often the case that a single dominating classifier (one producing statistically lower FAR at every PD value), does not exist. Furthermore, in many practical cases such as humanitarian demining, the best algorithm may be the one at which 100% detection is achieved with the lowest false alarm rate, no matter what other properties the FROC may display. For other time-critical demining applications where some level of missed mines may be tolerated, the best FROC may be the one at which the probability of detection is highest at a given constant false alarm rate. Our application falls in the second category.

7.1.3 Motivation for Multi-Algorithm Fusion

In the following, we consider four landmine discrimination algorithms of distinct character; namely, the prescreener [117], the EHD [51], the HMM [38, 29] and the SCF [55]. These algorithms were described in details in Section 4.2.1.

In this section, we compare the performance of the individual detectors and justify the need to fuse their results to improve the overall performance of the system. Figure 7.3 displays the FROCs obtained by applying three detection algorithms (EHD, SCF and HMM) and the prescreener to the entire data collection. As it can be seen, the EHD and the SCF detectors have the best overall performance. However, this does not necessarily mean that they are *consistently* the best algorithms. For instance, Figure 7.4(a) displays the results averaged over site A of the collection only. For this subset, the SCF is the best algorithm and the EHD is the second best one. However, in Figure 7.4(b), which displays the results averaged over site D only, the EHD is the best algorithm and the HMM is the second best one.

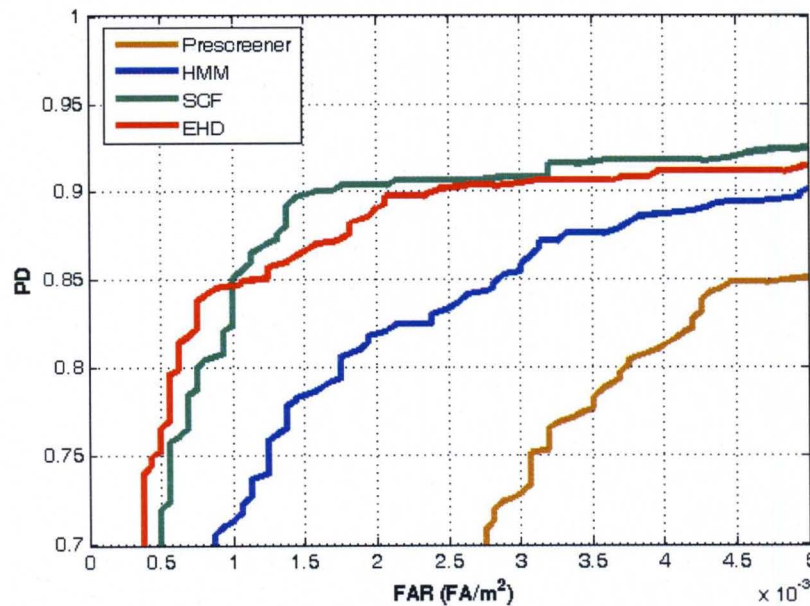


Figure 7.3: Performance of the different detectors on the entire data collection

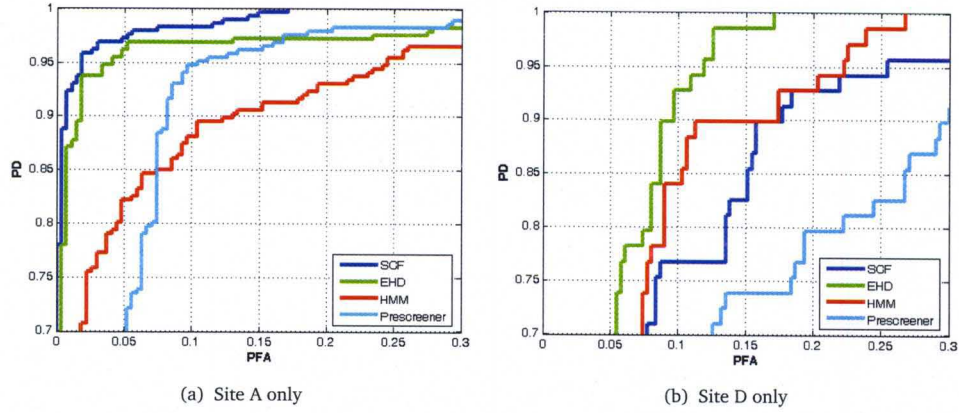


Figure 7.4: Performance of the detectors on two different sites

From the results displayed in Figure 7.4, one can reach the conclusion that there is no single algorithm that can consistently outperform all others detectors. In fact, the relative performance of different detectors can vary depending on the geographical site and soil and weather conditions. Moreover, even within the same site, the relative performance of the different algorithms can vary significantly depending on the mine type, burial depth, and other unknown factors. Figure 7.5 displays the performance of the detectors on mines buried at different depths. As it can be seen in Figure 7.5(a), the EHD is the best algorithm on shallow mines. However, as it appears in Figure 7.5(b), the same algorithm has the worse performance for deep mines.

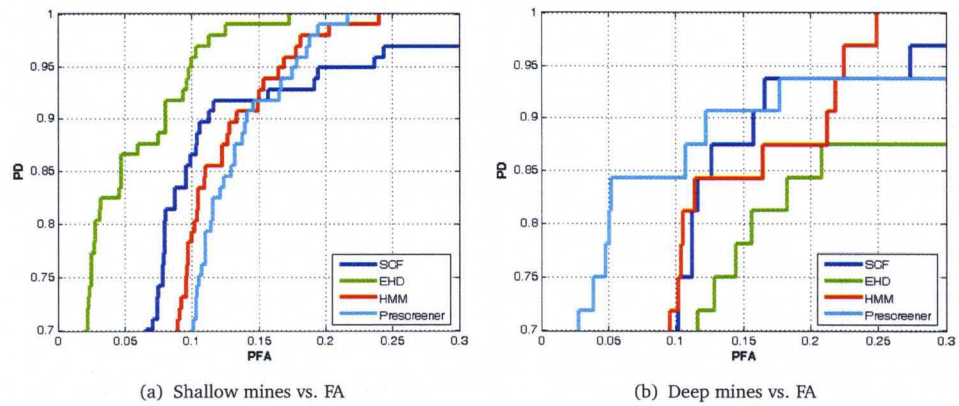


Figure 7.5: Performance of the detectors on mines buried at different depths

7.1.4 Multi-Algorithm Fusion

In this section, we apply the proposed CELF, CELF-FI, and CELF-NN methods to the landmine data set described in Section 7.2.1. Then, we compare their performance with standard fusion methods; namely, fuzzy integrals [114], bayesian fusion [106], and global linear fusion. The global fusion approach uses the same aggregation method as the CELF approach. We do this by simply setting the number of clusters to 1. The fusion algorithms were trained and tested using 12 lane-based cross-validation.

For each cross validation, the training data consists of a set of GPR alarms. Each alarm is processed by three discrimination algorithms (EHD, HMM, and SCF) and the prescreener outlined in Section 4.2. The features extracted from these alarms are then fed to CELF to partition the aggregate feature space into $C = 10$ clusters.

Table 7.3 displays the content of the 10 identified clusters. As it can be seen, most clusters include alarms of similar types, and thus may be considered as homogeneous contexts. For instance, some clusters are dominated by low metal mines (e.g. cluster 1). Also, some clusters include mainly mine (e.g. cluster 10), others include mainly false alarms (e.g. cluster 2), and others include a mixture of both.

Table 7.4 shows the aggregation weights assigned by CELF to each classifier in each cluster. As it can be seen, the performance of the different algorithms can vary significantly from one context to another. For instance, in context 2, the EHD has the highest performance and gets the highest weight. This context includes only FA (refer to Table 7.3). Figure 7.6 shows the histograms of the confidences assigned to the alarms in context 2 by the different detectors. Notice that this cluster does not include mines, and thus we cannot generate ROC for this context. As it can be seen, the distribution of the EHD confidence values is shifted to the left more than all others indicating low confidence values for all false alarms. The highest confidence assigned by the EHD algorithm in this context is 0.3134, which is lower

Table 7.3: Distribution of the alarms among the 10 clusters for one cross validation set

Cluster	AT Mines		False Alarms	
	Metal	Low Metal	Blank	Clutter
1	30	148	44	17
2	0	0	193	6
3	0	12	180	6
4	0	3	188	6
5	1	25	89	14
6	55	61	15	21
7	0	2	247	3
8	37	85	5	10
9	13	35	97	17
10	47	100	0	0

Table 7.4: Weights assigned to each classifier in each cluster

Cluster	1	2	3	4	5	6	7	8	9	10
SCF	0.39	0.00	0.05	0.31	0.50	0.21	0.00	0.26	0.56	0.00
EHD	0.61	1.00	0.95	0.34	0.30	0.66	1.00	0.57	0.39	0.28
HMM	0.00	0.00	0.00	0.35	0.07	0.13	0.00	0.17	0.00	0.72
Prescreener	0.00	0.00	0.00	0.01	0.13	0.00	0.00	0.01	0.04	0.09

than the highest confidences assigned by the other detectors: 0.6512, 0.3908, and 0.8824 assigned by the SCF, HMM, and the prescreener algorithms respectively. For this context, CELF identified the EHD algorithm as best detector. Table 7.5 shows some representatives alarms from this context and the confidence values assigned to them by the different detectors. As it can be seen, most of the FA may be caused by sub-layers in the soil. These sub-layers appear to be affecting the SCF and HMM detectors more than the EHD as this algorithm assigns the lowest confidences for most of these alarms.

Unlike context 2, context 9 includes a mixture of different alarms. For this context,

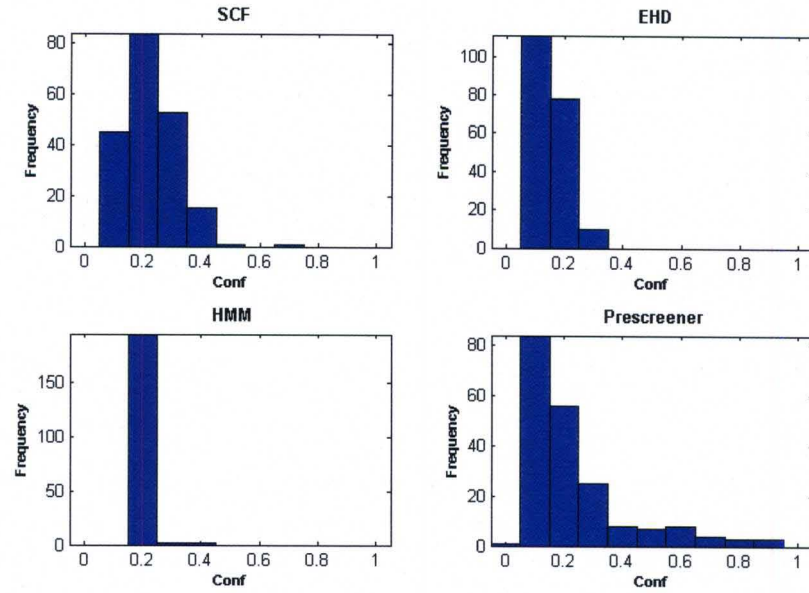
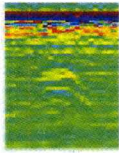
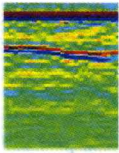

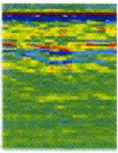
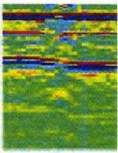
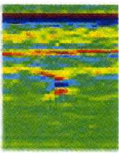


Figure 7.6: Distribution of the confidence values assigned by the different detectors for alarms assigned to context 2

Table 7.5: Representative alarms from context 2

Alarm						
EHD	0.15	0.15	0.09	0.12	0.17	0.13
SCF	0.24	0.22	0.22	0.13	0.12	0.11
HMM	0.20	0.20	0.20	0.20	0.20	0.20
Prescreener	0.23	0.87	0.17	0.16	0.45	0.21

as it can be seen in Figure 7.7, the SCF and the EHD have the best performances. Table 7.6 shows some representatives mines and false alarms from this context and the confidence values assigned to them by the different detectors. The EHD algorithm has the best performance for mine alarms, and the SCF has the second best performance. However for FA, the EHD assigns high confidence values; and the confidences assigned by the SCF algorithm have generally lower values. These FA appear to be different from those shown in Table 7.5 and may be caused by clutter objects or simply by disturbed soil. For this context, CELF combines the confidences

assigned by the EHD and the SCF by assigning comparable weights to these algorithms to get better performance.

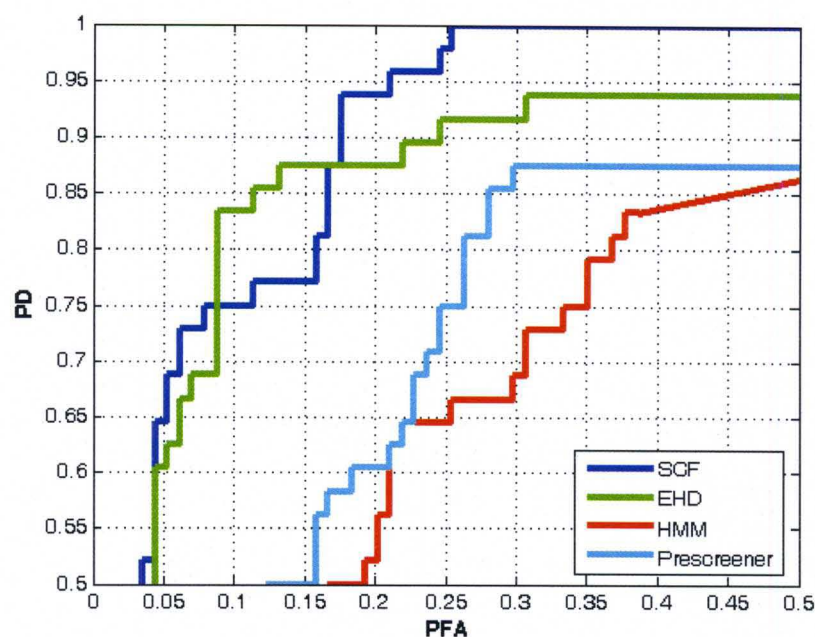
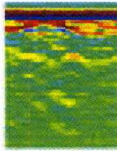
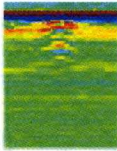
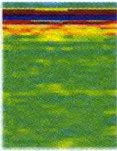
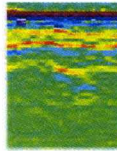
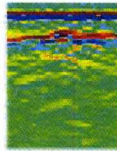
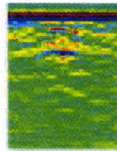


Figure 7.7: Performance of the different detectors for alarms assigned to context 9

Table 7.6: Representative mines and false alarms from context 9

	Mines			FA		
Alarm						
EHD	0.81	0.74	0.80	0.25	0.47	0.62
SCF	0.46	0.50	0.57	0.40	0.25	0.24
HMM	0.38	0.20	0.20	0.20	0.52	0.200
Prescreener	0.08	0.37	0.12	0.30	0.27	0.09

Another interesting context is number 10. This context includes only Mines (refer to Table 7.3). Figure 7.8 shows the histograms of the confidences assigned to the alarms assigned to context 10 by the different detectors. Table 7.7 shows some representatives mines from this context and the confidence values assigned to them by the different detectors. As it can be seen, the HMM algorithm assigns generally

higher confidences than those assigned by the other detectors. That explains why, for this context, CELF assigns the highest weight to the HMM.

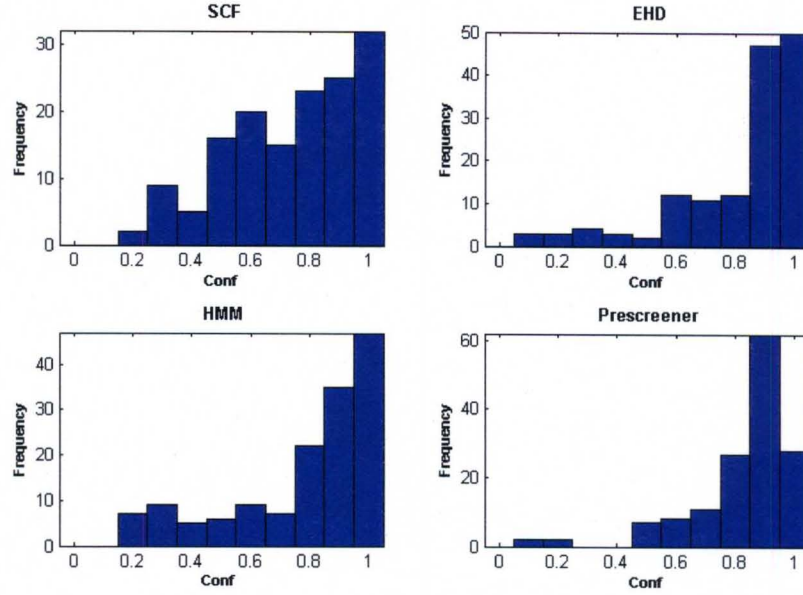
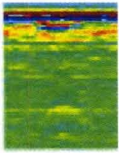
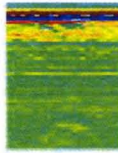
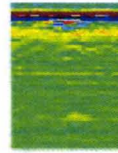
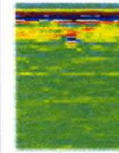
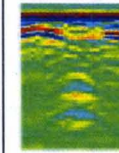
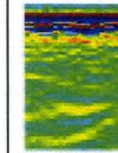


Figure 7.8: Distribution of the confidence values assigned by the different detectors for alarms assigned to context 10

Table 7.7: Representative alarms from context 10

Alarm						
EHD	0.97	0.99	0.57	0.97	0.96	0.98
SCF	0.65	0.67	0.96	0.65	0.80	0.99
HMM	0.83	0.99	0.97	0.99	0.80	0.99
Prescreener	0.92	0.85	0.89	0.95	0.69	0.94

To illustrate the advantage of CELF over global fusion, and the benefits of partitioning the feature space into clusters, we fuse the 4 algorithms using the same approach but without clustering, i.e., we let $C = 1$. Table 7.8 shows the aggregation weights assigned by the global fusion approach to each classifier. The highest weight was assigned to the EHD detector since it has the best overall performance

(refer to Figure 7.3), and the lowest weight was assigned to the prescreener since it has the worse overall performance (refer to Figure 7.3).

Table 7.8: Weights assigned by the global fusion approach

	EHD	SCF	HMM	Prescreener
Weights	0.67	0.21	0.12	0.00

The ROCs resulting from the individual detectors, the global fusion, the bayesian fusion [106], the fuzzy integrals [114] (refer to Chapter 2) and the proposed CELF approach on sites A, B, C, and D are shown in Figures 7.9(a), 7.9(b), 7.9(c), and 7.9(d) respectively. We note that the proposed context-extraction for local fusion approach outperforms all individual detectors and the other fusion approaches on site A, site B and site C significantly. However, the performance of CELF on site D is comparable to the other fusion approach. In fact, on this site, the EHD has the best performance. And since EHD has the highest overall performance on all sites, The global fusion approach assigns a high weight to this detector(refer to Table 7.8).

Figure 7.10 displays the ROCs resulting from the individual detectors, the global fusion approaches, and the proposed CELF, CELF-FI, and CELF-NN approaches using all data with lane-based cross-validation. We note that our proposed approaches outperforms all individual detectors and the other fusion approaches significantly. Also, we note that CELF-FI and CELF-NN slightly outperform the linear CELF approach. This is due to the non-linear aspect of CELF-FI and CELF-NN.

7.2 Landmine Detection Using AMDS System

7.2.1 Data Collection

In this section, we report results using data collected with the NIITEK Inc. Autonomous Mine Detection System (AMDS). This system includes a Ground Penetrating Radar (GPR) and a Wideband Electro-Magnetic Induction (WEMI) sensor

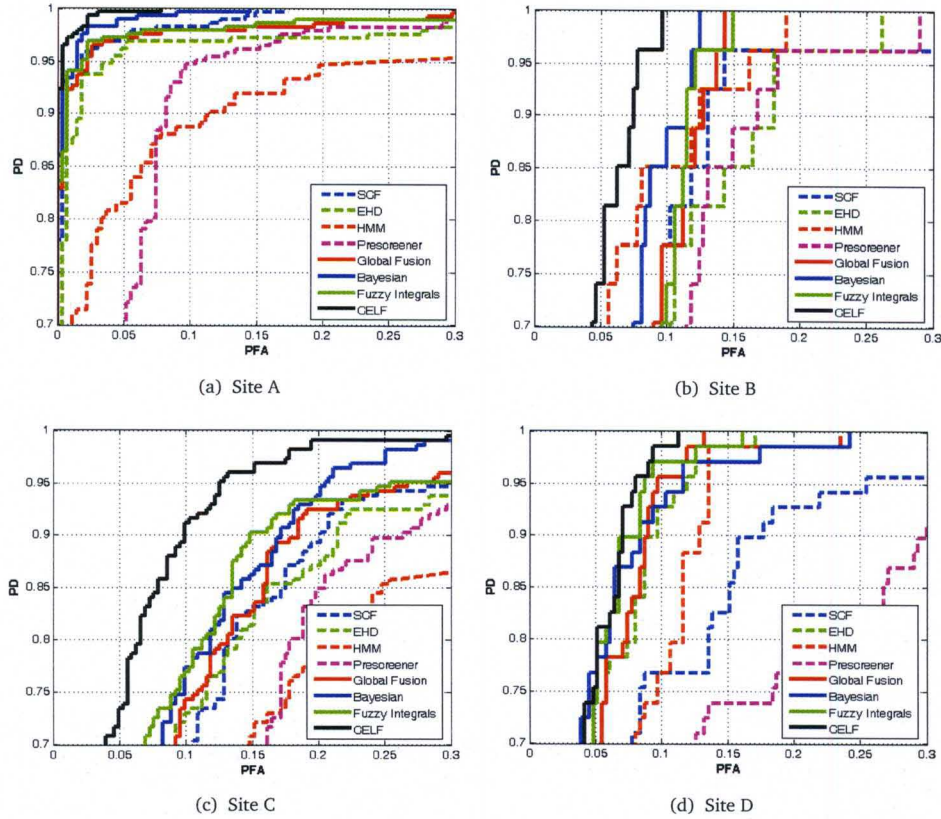


Figure 7.9: Performance of the individual detectors and the different fusion methods on (a) site A, (b) site B, (c) site C, (d) site D.

and is shown in Figure 7.11. It was used to acquire large sets of co-located GPR and WEMI data from 2 geographically distinct test sites (site A and site B). The two sites are partitioned into grids with known mine locations. Over all, there are 28 distinct mine types that can be classified into 4 categories: *anti-tank metal* (ATM), *anti-tank with low metal content* (ATLM), *anti-personnel metal* (APM), and *anti-personnel with low metal content* (APLM). The targets were buried up to 5 inches deep. Multiple data sets at different times were collected at each site resulting in a large and diverse collection of mine and clutter signatures.

In this data collection, clutter arises from two different processes. One type of clutter is emplaced and surveyed. Objects used for this clutter can be classifier into 2 categories: *High Metal Clutter* (HMC) and *Non-Metal Clutter* (NMC). High metal clutter such as steel scraps, bolts, soft-drink cans, is emplaced and surveyed

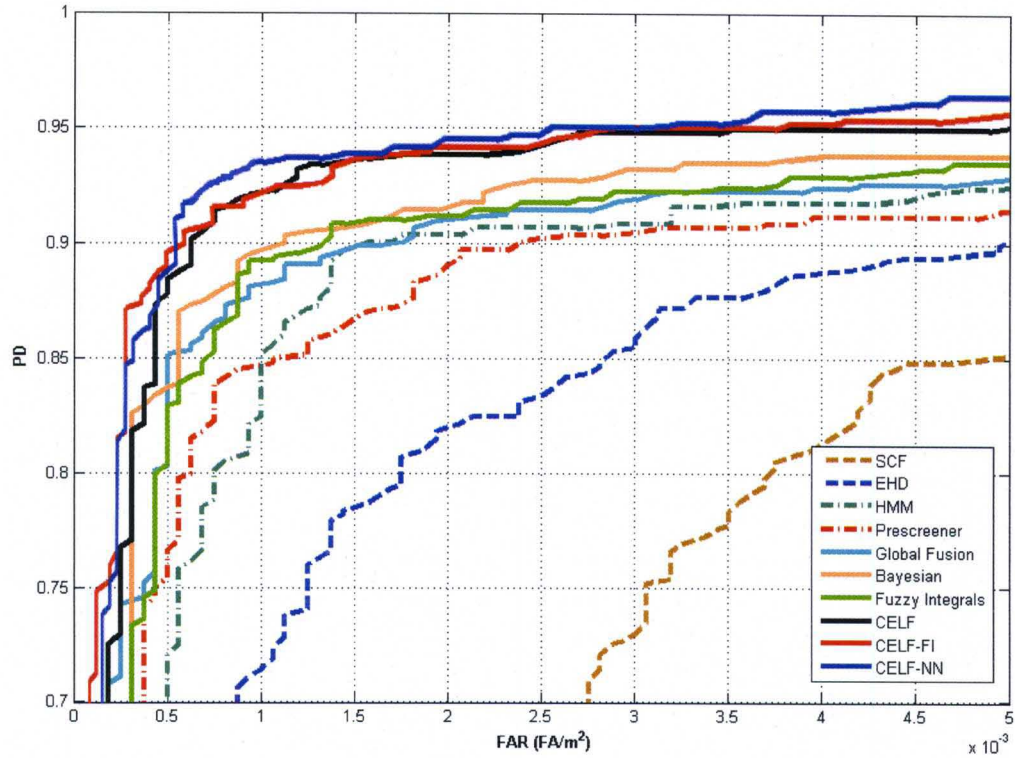


Figure 7.10: Performance of the individual detectors and the different fusion methods on the entire collection using lane-based cross-validation

in an effort to test the robustness of the detection algorithms, and in particular those using the WEMI sensor. Non-metal clutter such as concrete blocks and wood blocks is emplaced and surveyed in an effort to test the robustness of the GPR based detection algorithms. The other type of clutter, referred to as blank, is caused by disturbing the soil.

The AMDS data collection includes a total of 311 mine signatures and 564 clutter signatures. The statistics of this collection is shown in Table 7.9, and the depth distribution for all objects is shown in Table 7.10.

7.2.2 Motivation for Multi-Sensor Multi-Algorithm Fusion

Our extensive experiments have lead us to the conclusion that the performance of most detection systems can be significantly effected by various factors, and there is



Figure 7.11: NIITEK Autonomous Mine Detection System.

Table 7.9: Statistics of the data collection used in our experiment

Type		Site A	Site B	Total / Category	Total / Type
AP	HM	16	40	56	187
	LM	38	93	131	
AT	HM	6	20	26	124
	LM	28	70	98	
EA	HMC	224	68	292	564
	NMC	72	68	140	
	Blank	52	80	132	
Total		436	439	875	875

no single sensor or algorithm that can consistently outperform all others. In fact, the relative performance of different sensors and detectors can vary significantly depending on the mine type, geographical site, soil and weather conditions, and burial depth.

Table 7.10: Burial depth of all objects in the data collection

Depth	Mine			Clutter		
	Site A	Site B	Total	Site A	Site B	Total
Surface	0	27	27	52	80	132
1"	12	104	116	70	46	116
2"	36	48	84	78	44	122
3"	28	34	62	88	18	106
4"	12	0	12	60	20	80
5"	0	10	10	0	8	8
Total	88	223	311	348	216	564

To illustrate the above point, in Figure 7.12, we show the Receiver Operating Characteristic (ROC) curves of the four discrimination algorithms on subsets of the data collection outlined in section 7.2.1. The different ROCs display the performance of the algorithms when different types of mines are scored. For instance, in Figure 7.12(a), only anti-tank (AT) mines are considered. In this case, the HMM and EHD detectors have the best performance. This is because AT mines are large enough to have good GPR signatures and many of them have low metal content. This explains the relatively lower performance of the MFIT algorithm. However, for anti-personal (AP) mines, the MFIT detector has the best performance at high probability of detection as shown in Figure 7.12(b). In this case, several AP mines have weak GPR signatures and cannot be easily detected by any of the GPR algorithms. Figures 7.12(c) and 7.12(d) display the ROC curves when only high metal mines or low metal mines are considered. Besides the mine type, the relative performance of the different algorithms depend on other factors, such as burial depth, soil properties, and weather conditions.

The above examples suggest using different sensors, algorithms and/or features to accommodate for the different conditions. However, this task may not be as simple as it sounds since it is not possible to characterize the performance of each algorithm on all possible variations. Moreover, it may not be possible to know the

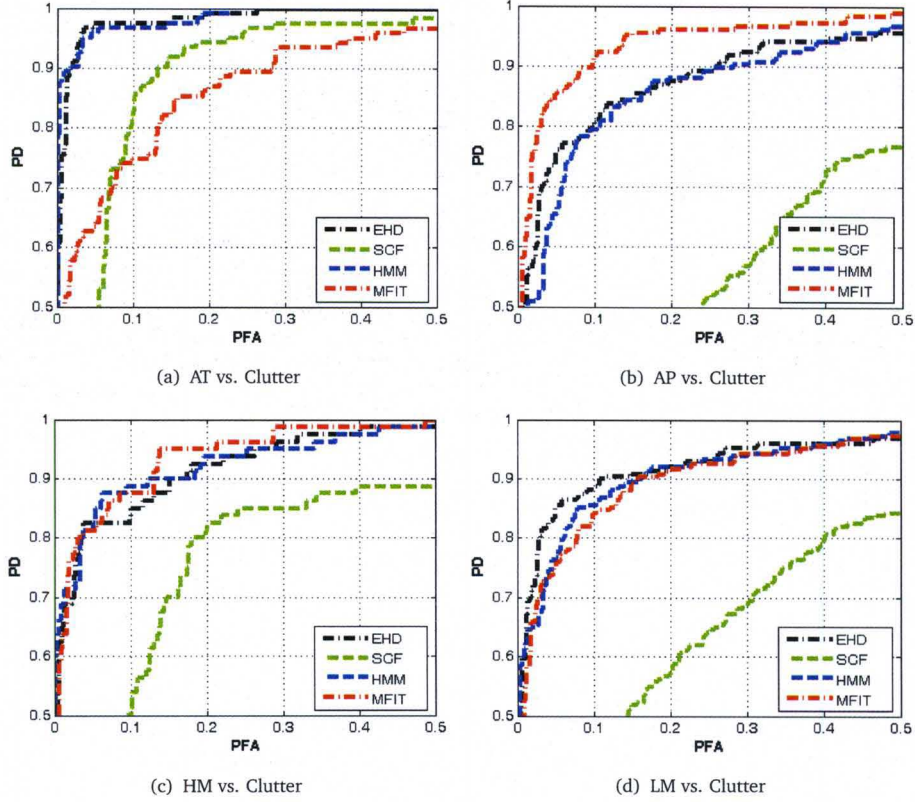


Figure 7.12: Performance of the individual detectors for different types of mines when: (a) only anti-tank (AT) mines are considered, (b) only anti-personal (AP) mines are considered, (c) only high-metal (HM) mines are considered, (d) only low-metal (LM) mines are considered.

characteristics of the test site. Thus, the selection of the optimal subset of algorithms is not a trivial task and needs to be learned in an unsupervised way.

7.2.3 Multi-sensor Multi-algorithm Fusion

We apply the proposed CELF, CELF-FI, and CELF-NN methods to the landmine data set described in Section 7.2.1. We compare its performance with other commonly used fusion methods, and global linear fusion. The global fusion approach uses the same aggregation method as CELF. We do this by simply setting the number of clusters to 1. The fusion algorithms were trained and tested using 6 fold cross-validation. For each cross validation, the training data consists of a set of co-located

GPR and WEMI alarms. Each alarm is processed by the four discrimination algorithms (EHD, HMM, SCF, and MFIT). The features extracted from these alarms and their confidence values are then fed to CELF to partition the aggregated feature space into contexts.

Table 7.11 displays the content of the 13 identified contexts. As it can be seen, most clusters include alarms of similar types, and thus, each one may be considered as a homogeneous context. For instance, some contexts (e.g. 2 and 11) are dominated by high metal mines. Others are dominated by AT mines (e.g. 3 and 9) or AP mines (e.g. 4 and 12). Also, some contexts include mainly mine or clutter alarms. Others include a mixture of both. Alarms that are grouped into the same context share common GPR and/or WEMI features. However, some contexts do not always correspond to alarms of the same type. In this case, other factors such as burial depth and soil properties can affect the grouping of the signatures. For instance, for the GPR sensor, some shallowly buried AP mines can have signatures as strong as the deeply buried AT mines.

Table 7.12 shows the aggregation weights assigned by CELF to each classifier within each context. As it can be seen, the performance of the different algorithms can vary significantly from one context to another. For instance, in contexts 1, 6, 7 and 13, the highest weight is assigned to the MFIT detector. These contexts contain mainly blank and Non-Metal Clutter (NMC) (refer to Table 7.11), and MFIT assigns low confidence values to these alarms since they have low (or no) metal content. Consequently, since the desired output is 0, CELF assigns the highest aggregation weight to MFIT. Figure 7.13 shows the cumulative histograms of the confidences assigned by the different detectors in context 1. As it can be seen, the maximum confidence assigned by MFIT is 0.3 which is lower than those assigned by EHD, SCF and HMM (0.9, 1, and 0.8 respectively).

Another interesting context is number 3. This context includes mainly AT mines with low-metal content and clutter with high metal content. Figure 7.14 displays

Table 7.11: Distribution of the alarms among the 13 contexts identified by CELF

Context	Mines				Clutters		
	ATM	ATLM	APM	APLM	Blank	HMC	NMC
1	0	0	0	0	25	3	44
2	12	0	14	0	0	4	0
3	0	25	8	1	0	25	0
4	0	5	5	52	0	56	0
5	0	33	0	26	1	12	0
6	0	1	0	0	31	7	26
7	0	0	0	0	22	3	24
8	5	3	5	4	0	63	0
9	1	26	0	2	2	6	5
10	0	4	0	2	14	15	14
11	8	0	21	9	0	33	0
12	0	0	1	32	0	49	0
13	0	0	0	3	37	8	27

Table 7.12: Weights assigned to each classifier in each context for the entire training data

Context	1	2	3	4	5	6	7	8	9	10	11	12	13
EHD	0.10	0.53	0.63	0.36	0.42	0.14	0.00	0.55	0.84	0.43	0.42	0.47	0.22
SCF	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.11	0.13	0.00	0.00	0.00	0.00
HMM	0.00	0.37	0.32	0.19	0.36	0.00	0.00	0.00	0.00	0.00	0.15	0.12	0.00
MFIT	0.90	0.10	0.02	0.46	0.22	0.86	1.00	0.33	0.03	0.57	0.42	0.41	0.78

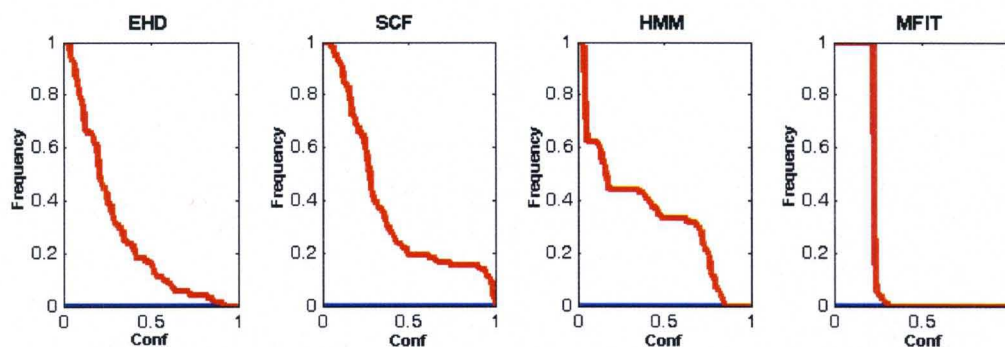


Figure 7.13: Cumulative histograms of the confidence values assigned by the different detectors to samples within context 1.

the ROC curves of the different detectors in this context. As it can be seen, HMM and EHD have the best performance in this context. This explains the high weights assigned to these detectors (refer to Table 7.12). This is because the AT mines have a relatively large size, and thus a strong GPR response. The MFIT detector, on the other hand, is not a good discriminator for this type of alarms and gets assigned a weight close to zero. In particular, MFIT tends to assign low confidence values to ATLM mines and higher confidence values to most HM clutters.

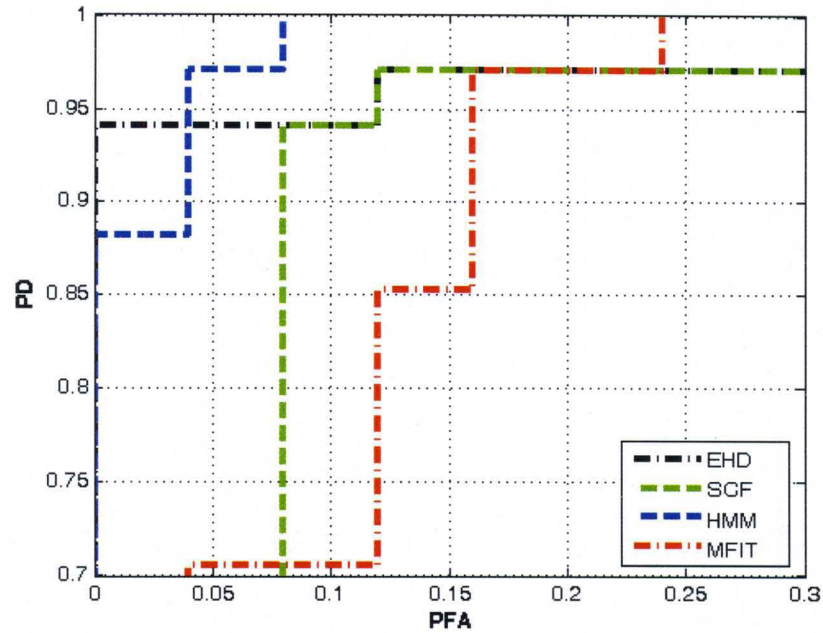
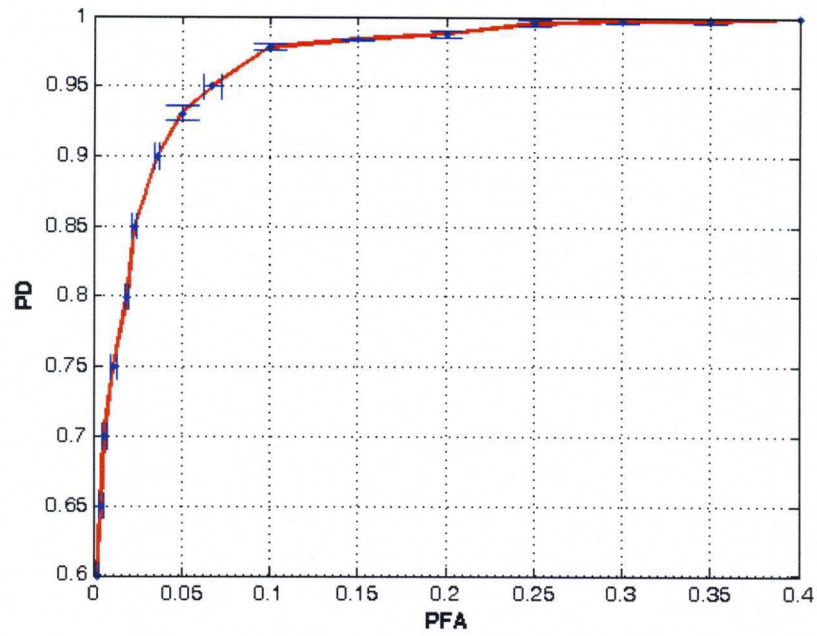


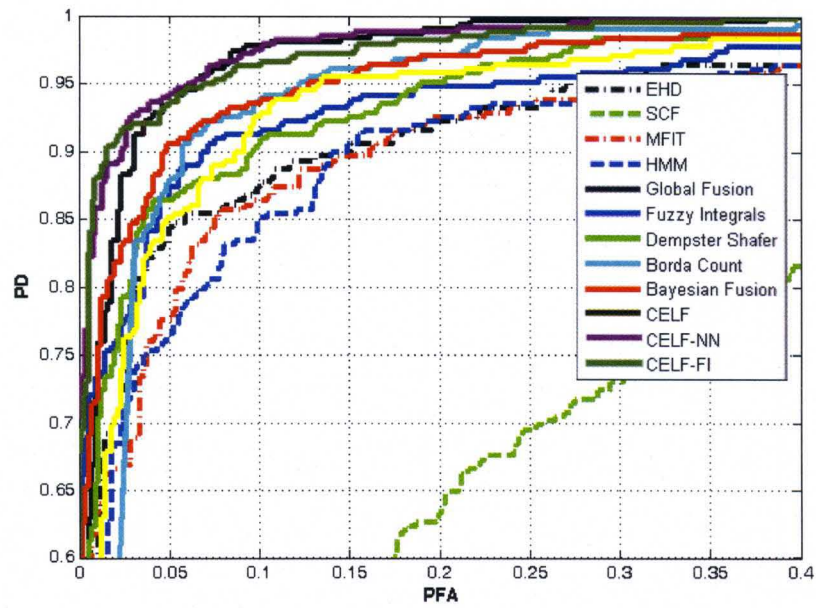
Figure 7.14: ROC curves generated by the different detectors for samples assigned to context 3.

To take into account the effect of initialization and local minima, we ran CELF 100 times using different random initializations. Figure 7.15(a) shows the results where we display the mean of all ROC curves along with the standard deviation. As it can be seen, the deviation are small indicating that CELF is not very sensitive to initialization. This can be attributed in part to the fuzzy nature of the algorithm where each alarm gets assigned to each context with a different membership degree and that the overall fusion is averaged over all clusters (refer to equation (5.19)).

For comparison purposes, we fuse the 4 algorithms (HMM, EHD, SCF, and MFIT) using other commonly used fusion methods including the fuzzy integrals [114, 18,



(a)



(b)

Figure 7.15: Probability of detection (PD) vs. probability of false alarms (PFA) of (a) average and standard deviation of CELF over 100 runs, (b) the individual detectors and the different fusion methods on the entire collection using 6 fold cross-validation.

45, 65, 5], bayesian fusion [106], Borda count [19], and Dempster Shafer [108]. To emphasize the benefits of local fusion and partitioning the feature space into contexts, we also fuse the 4 algorithms using the same linear aggregation used in CELF but without partitioning the feature space, i.e., we use $C = 1$. We will refer to this method simply global fusion. The ROC curves resulting from all fusion methods (including the proposed CELF, CELF-FI, and CELF-NN approaches) are shown in Figure 7.15(b). First, we note that even with a simple global fusion, we obtain results that outperform all individual detectors. This is because these detectors operate on different sensor data, use different preprocessing, feature extraction, and classification methods. This diversity allows the fusion to take advantages of the strengths of the individual detectors, overcome their weakness, and achieve a higher accuracy. Second, the proposed approaches outperforms all individual detectors and the other fusion methods. For instance, for a 90% probability of detection (PD), CELF reduces the probability of false alarm (PFA) by 100% when compared to the global fusion and by 40% when compared to the next best fusion method (Borda count). Also, we note that the results of CELF-FI and CELF-NN are slightly better than the CELF result. This can be explained by the non-linear aspect of CELF-FI and CELF-NN.

OTHER APPLICATIONS OF CELF

CELF was designed and implemented mainly for landmine detection. However, our approach can be applied to many other problems. In this chapter, we apply the proposed fusion method to 3 different problems; namely, semantic video indexing, image database categorization, and phoneme recognition.

8.1 Semantic Video Indexing

In this section, we use CELF to label MPEG-1 movies from the TRECVID-2002 data collection¹ [111].

8.1.1 Data Collection

This collection consists mainly of Internet Archive of advertising, educational, industrial, amateur films produced between 1930 and 1970 by corporations, non-profit organizations, and trade groups. This collection included a total of 73.3 hours of video data partitioned into a search test set (40.12 hours); a feature development

¹This is the latest TRECVID data collection that is publicly available with no copyright issues.

set (training and validation; 23.26 hours); a feature test set (5.07 hours); and a shot boundary test set (4.85 hours). For our experiment, we used the feature development set for training and the feature test set for testing and evaluation. Each shot in this collection can belong to one (or more) of the 10 semantic concepts: 'Outdoors', 'Indoors', 'Face', 'People', 'Cityscape', 'Landscape', 'Text Overlay', 'Speech', 'Instrumental Sound', and 'Monologue'. For all data, we used the shot boundaries provided by NIST [111]. Table 8.1 summarizes the data used in our experiment. The number of shots used to train and test each of the 10 semantic concepts is shown in Table 8.2.

Table 8.1: Feature development and test sets of the TRECVID-2002 collection used in our experiment

	# of movies	# of shots	Running time (h)
Training	80	6,330	23.26
Testing	23	1,848	5.07

Table 8.2: Number of shots per semantic concept used in our experiment

	Training	Testing
Outdoors	3,290	1,277
Indoors	1,425	494
Face	626	589
People	2,158	681
Cityscape	1,895	699
Landscape	654	184
Text Overlay	811	144
Speech	4,387	1,815
Instrumental Sound	3,498	1,568
Monologue	54	57

8.1.2 Low-Level Descriptors and Classifiers

The goal of our experiment is to illustrate that the proposed context dependent fusion is a framework that can improve the performance by partitioning the feature space into disjoint regions and identifying local expert algorithms for each region. Thus, we did not attempt to optimize the feature extraction nor the classifier design components. We simply use a set of generic MPEG-7 descriptors [84, 85] and a simple k -NN classifier [23]. Other descriptors and classifiers can be easily integrated into our approach. In particular, the following set of low-level descriptors are extracted and used to construct the low-level feature space for context extraction.

1. **Color Structure Descriptor (CSD):** This descriptor expresses local color structure in an image using an 8x8-structuring element. It counts the number of times a particular color is contained within the structuring element as the structuring element scans the image. The HMMD color space is used in this descriptor.
2. **Scalable Color Descriptor (SCD):** This descriptor addresses the inter-operability issue by fixing the color space to HSV, with a uniform quantization of the HSV space to 256 bins. The SCD is a color histogram in the HSV color space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bin numbers and bit representation accuracy over a broad range of data rates. Among 256 bins, only the highest 32 frequency components are used in our experiment.
3. **Edge Histogram Descriptor (EHD):** This descriptor is designed to represent the spatial distribution, frequency, and directionality of the edges. First, simple edge detector operators are used to identify edges and group them into five categories: vertical, horizontal, 45° diagonal, 135° diagonal, and isotropic (non-edge). Then, local, global, and semi-local edge histograms are generated. [27].

4. **Homogeneous Texture Descriptor (HTD):** This descriptor is based on the Gabor descriptors proposed by Manjunath et al. [85] to represent the texture. Each image is filtered by 30 Gabor filters that are generated with 5 different scales and 6 different orientations. The texture feature is represented by the average and standard deviation of each filtered image.

8.1.3 Results

CELF, CELF-FI, and CELF-NN were used to partition the feature space into 50 clusters, and fuse the result of the different classifiers presented in the previous section. Since we are using only visual features, we only use the first 7 semantic concepts as the last 3 concepts require textual and audio features.

Figure 8.1 displays some representative keyframe images from 6 typical clusters obtained by one run of CELF. As expected, these images appear similar, and thus may be considered as a homogeneous context. In fact, each cluster is dominated by images from few concepts. Context 3 includes mainly images from the 'Text Overlay' concept. Context 11 is dominated by images from the 'Outdoors' and 'People' concepts. Context 12 is dominated by images from the 'People' and 'Face' concepts. Context 27 is dominated by images from the 'Outdoors' and 'Cityscape' concepts. Context 33 is dominated by images from the 'Indoors' and 'People' concepts. And, context 39 is dominated by images from the 'Outdoors' and 'Landscape' concepts.

For comparison purposes, we fuse the 4 classifiers (k -NN based on CSD, SCD, EHD, and HTD) using global Neural Network. The performance of the different classifiers and fusion algorithms is measured in terms of the 'Mean Averaged Precision (MAP)' [2]. Basically, MAP is a non-interpolated average precision. It corresponds to the area under an ideal (non-interpolated) recall/precision curve. To compute MAP, the average precision for each semantic concept is first calculated. Average

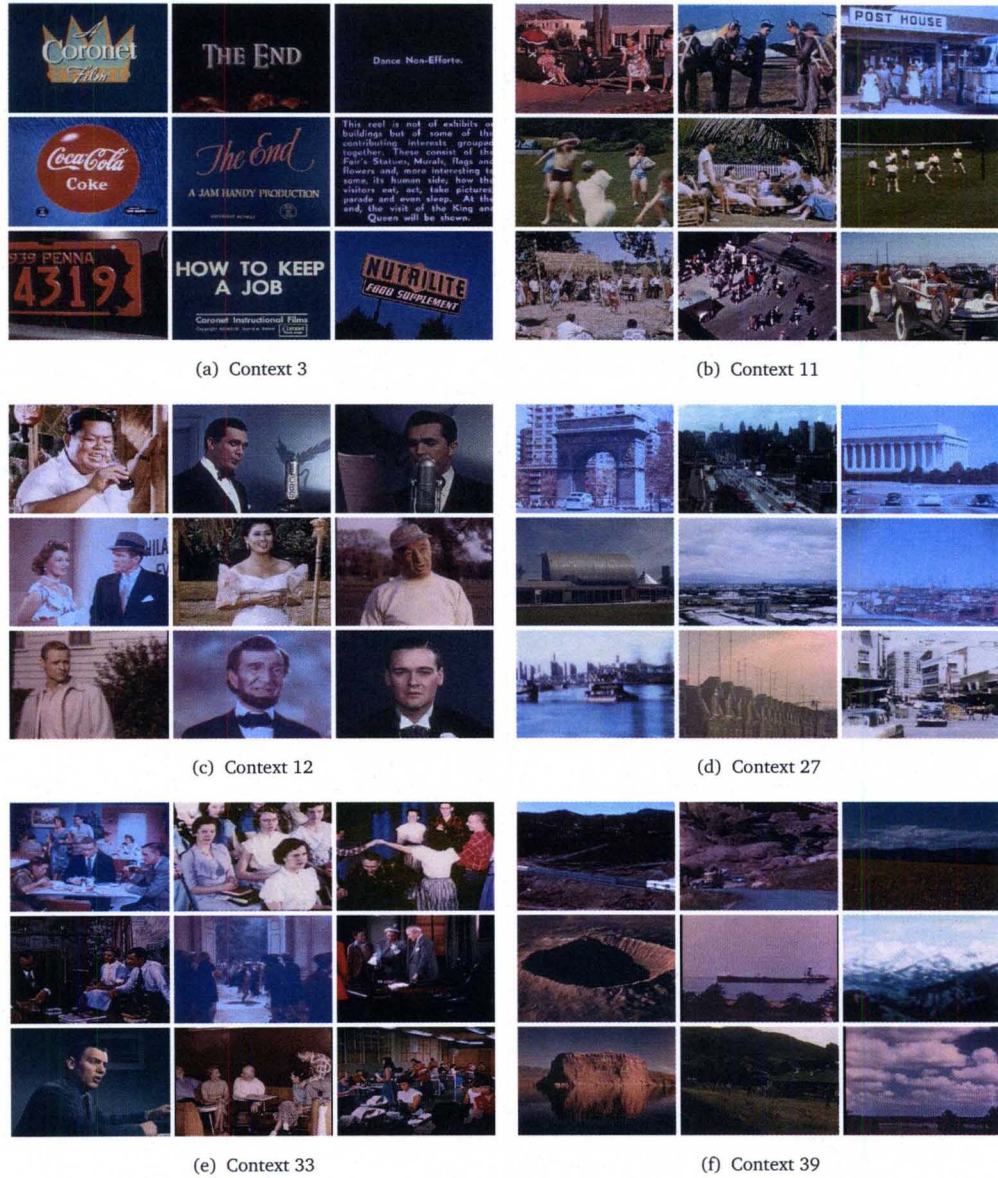


Figure 8.1: Sample keyframes from 6 of the contexts identified by CELF

precision favors highly ranked relevant documents and allows comparison of results with different sizes. The averages are then combined (averaged) across all semantic concepts in the appropriate set to create the non-interpolated MAP for that set. Table 8.3 displays the MAP of the individual algorithms, the global Neural Network fusion, and the proposed approaches (CELF, CELF-FI, and CELF-NN).

Table 8.3: MAP values for the individual classifier and the fusion algorithms averaged over the test data

	CSD	SCD	EHD	HTD	Global NN	CELF	CELF-FI	CELF-NN
Outdoors	0.58	0.64	0.72	0.58	0.74	0.75	0.72	0.81
Indoors	0.22	0.23	0.31	0.29	0.25	0.36	0.36	0.37
Face	0.21	0.23	0.33	0.32	0.33	0.33	0.39	0.41
People	0.28	0.35	0.40	0.37	0.38	0.41	0.46	0.46
Cityscape	0.40	0.48	0.49	0.39	0.52	0.50	0.51	0.54
Landscape	0.08	0.08	0.16	0.12	0.15	0.18	0.22	0.21
Text Overlay	0.57	0.58	0.62	0.59	0.70	0.72	0.78	0.80

We note that, for all concepts, the CELF-NN approach outperform all individual classifiers and the global fuser. CELF, and CELF-FI improve the result of the individual classifiers for most of the concept (6 concepts over 7). But, for some concepts, they are not able to do so (e.g. 'outdoors' concept for CELF-FI, and 'Face' concept for CELF). In the other side, the global Neural Network is not able to improve the results of the individual classifiers for 3 of the 7 concepts. However, we can conclude that, in average, the proposed adaptive approaches proves better performance than the global fusion approach (Neural Networks).

8.2 Image Database Categorization

In this section, CELF is used to label a subset of 3000 color images from the COREL image collection. This subset includes 30 categories. Each category contains 100 images. To generate the test dataset, 25 images from each category were randomly selected. The remaining images were used for training.

8.2.1 Features Descriptors

The goal of our experiment is to illustrate that the proposed context dependent fusion is a framework that can improve the performance by partitioning the feature space into disjoint regions and identifying local expert algorithms for each region. Thus, we did not attempt to optimize the feature extraction nor the classifier design components. We simply use a set of generic MPEG-7 descriptors [85] and few other commonly used features, and a simple k -NN classifier [23]. The features are selected to balance the color, texture, structure, and textual properties of an image. Other descriptors and classifiers can be easily integrated into our approach. In particular, we use the CSD, SCD, EHD, and HTD features (described in Section 8.1.2). In addition, we extract:

- **Wavelet Texture Descriptor (WTD):** Each image is analyzed at different frequencies with different resolutions. We use the Haar filter bank to decompose the image into three scales. This would result in a total of 10 components (approximation at scale three, and horizontal, vertical, and diagonal components at the three scales). Then, for each image, the mean and standard deviation of the components are computed.
- **Thesaurus Text Descriptor (TTD):** The TTD is used to represent the semantic knowledge contained within an image. We use the approach proposed in [26, 25] to automatically annotate images using a multi-modal thesaurus developed through unsupervised clustering. The annotating words are represented in a vector form where each component indicates the presence or absence of particular word.

8.2.2 Results

To illustrate the performance of the proposed adaptive fusion approaches, we use them to fuse the result of the different classifiers presented in the previous section. Here, CELF-NN and CELF-M were used to partition the feature space into C clusters. We let $C = 20$ as we assume that this is sufficient to cover the variations in the given data. For comparison purpose, we fuse the 6 classifiers (k -NN based on WTD, CSD, EHD, HTD, SCD, and TTD with $k = 20$) using global Neural Network.

Table 8.4: Distribution of the images among the 20 contexts

Cluster	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Sailing	3	1	0	3	2	5	4	13	1	1	3	0	3	1	4	16	2	6	6	1
Antelope	0	0	0	2	2	0	3	1	0	0	1	2	0	29	8	1	19	4	2	1
Gardens	2	1	0	0	0	1	1	0	8	13	0	2	0	7	26	0	4	2	8	0
Horses	0	0	2	1	0	0	10	3	3	0	0	0	0	38	5	4	8	0	1	0
Auto Racing	1	0	2	25	2	2	0	7	0	0	1	0	20	1	0	6	5	1	0	2
Aviation	0	1	0	2	1	30	0	9	0	0	1	4	1	0	0	0	1	0	0	25
Eagles	0	0	0	1	6	34	0	5	5	0	2	2	1	2	1	3	8	1	2	2
Beaches	4	0	0	0	0	0	0	0	28	1	1	18	0	0	5	3	0	1	14	0
Butterflies	1	0	0	2	52	3	0	2	0	0	3	2	4	0	0	3	2	1	0	0
Cougars	0	3	0	3	0	0	33	1	0	0	0	0	0	11	0	3	0	4	0	17
Deserts	3	0	0	0	0	1	2	1	14	4	0	1	0	7	6	6	14	1	13	2
Diving	14	1	0	1	0	0	1	0	12	5	0	1	5	3	11	1	1	5	14	0
Whales	0	0	1	13	1	3	0	27	0	0	9	1	2	0	1	10	0	2	1	4
Elephants	0	0	0	0	0	0	12	0	0	0	0	0	0	16	7	1	33	4	0	2
Flowers	2	1	0	4	4	6	3	4	0	0	2	0	0	5	1	2	9	0	4	28
Fungi	19	6	5	2	0	0	2	0	2	3	0	1	3	2	2	1	2	14	11	0
Mountains	3	1	8	8	0	1	14	3	0	4	0	0	2	1	1	2	0	14	1	12
Lions	1	0	0	3	1	0	1	6	5	0	42	9	0	0	0	1	1	0	1	4
Bears	5	0	0	2	0	0	4	1	2	10	0	0	1	18	12	1	1	13	4	1
Rome	11	0	1	1	0	0	4	2	11	5	0	0	3	4	8	8	0	6	11	0
Skiing	1	0	53	1	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0
Sunrises	7	0	3	1	0	0	0	0	0	0	0	0	59	0	2	3	0	0	0	0
Tigers	7	2	0	4	1	0	6	3	0	4	0	1	6	5	10	9	0	10	7	0
Waterfalls	1	2	0	5	0	0	0	6	10	0	24	10	3	1	0	4	2	1	4	2
Wolves	15	1	0	3	2	0	0	1	0	1	0	0	10	1	15	9	0	11	6	0
Building	8	0	0	1	0	0	0	0	11	23	0	2	0	5	10	2	0	5	8	0
Buses	2	67	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	3	0	0
Cars	2	0	3	25	2	0	0	7	1	0	0	0	12	1	2	18	1	1	0	0
Castles	0	0	1	5	5	0	0	13	9	1	2	18	2	2	6	7	0	0	3	1
Colorado	0	0	0	6	14	1	0	27	1	0	8	0	2	0	0	8	2	1	5	0

Table 8.4 displays the content of the 20 contexts identified by CELF. As it can be seen, most contexts include images of "similar" categories, and thus, each one may be considered as a homogeneous context. For instance, some contexts (e.g. 2, 3 and 5) are dominated by images from only one category. Others are dominated by

images from 2 categories (e.g. 6 and 11). Few others include a mixture of different categories (e.g. 16, 18 and 19).

Figure 8.2 shows the aggregation weights assigned by CELF-M to each classifier within each context. As it can be seen, the performance of the different algorithms can vary significantly from one context to another. For instance, in context 2, the highest weight was assigned to the WTD classifier. This context contains mainly images of buses (see Figure 8.3). Since these images have similar structures, the highest weight was assigned to a texture descriptor (WTD). On the other hand, in context 6, the highest weight was assigned to the SCD classifier. This context contains mainly images from aviation and eagles categories. Figure 8.4 shows some representative images from this context. These images have blue sky as a common background. Thus, a color descriptor, such SCD, is more efficient to distinguish them from the other images.

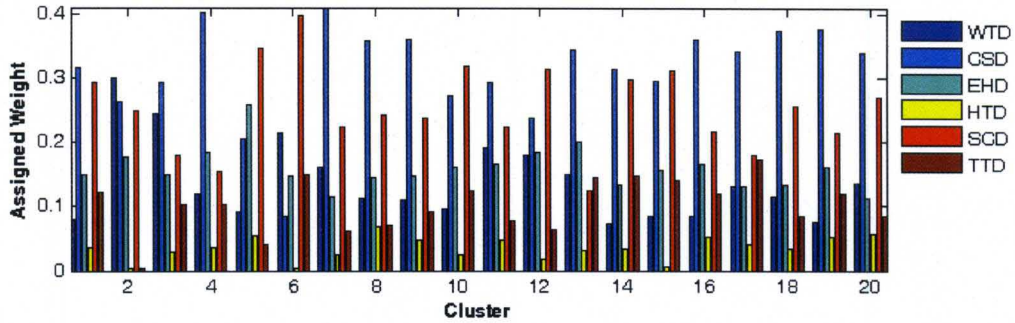


Figure 8.2: Weights assigned to each classifier in each context

The performance of the different classifiers and fusion algorithms is reported in Table 8.5. First, we note that even though the CSD based classifier has the best overall accuracy (61%), it doesn't have the best performance for some categories. For instance, it has the worst performance on the aviation category. For this category, the best performance is obtained by the TTD based classifier, which has the second worst overall accuracy. Second, we note that all fusion algorithms (Neural Networks, CELF-M, and CELF-NN) outperform all individual classifiers. In fact,



Figure 8.3: Representative images from context 2



Figure 8.4: Representative images from context 6

Table 8.5: Accuracy of the individual classifiers and the fusion algorithms

Category	WTD	CSD	EHD	HTD	SCD	TTD	Global NN	CELF-M	CELF-NN
Sailing	0.36	0.68	0.44	0.04	0.48	0.28	0.80	0.80	0.84
Antelope	0.08	0.84	0.04	0.04	0.68	0.24	0.84	0.88	0.88
Gardens	0.64	0.68	0.68	0.32	0.40	0.36	0.80	0.80	0.80
Horses	0.68	0.96	0.72	0.52	0.84	0.92	0.96	0.96	0.96
Auto Racing	0.52	0.56	0.64	0.40	0.48	0.60	0.80	0.88	0.92
Aviation	0.48	0.32	0.44	0.44	0.44	0.72	0.52	0.60	0.72
Eagles	0.52	0.80	0.48	0.28	0.72	0.40	0.80	0.80	0.84
Beaches	0.76	0.84	0.52	0.12	0.68	0.52	0.84	0.88	0.88
Butterflies	0.44	0.68	0.64	0.52	0.64	0.36	0.80	0.80	0.80
Cougars	0.60	0.68	0.36	0.24	0.52	0.12	0.76	0.84	0.84
Deserts	0.04	0.64	0.56	0.20	0.60	0.52	0.68	0.76	0.80
Diving	0.16	0.40	0.04	0.08	0.40	0.16	0.64	0.68	0.80
Whales	0.24	0.80	0.48	0.20	0.68	0.44	0.84	0.84	0.84
Elephants	0.28	0.60	0.24	0.20	0.48	0.68	0.64	0.72	0.80
Flowers	0.12	0.56	0.08	0	0.48	0.16	0.56	0.60	0.68
Fungi	0.24	0.76	0.16	0.04	0.80	0.04	0.76	0.80	0.84
Mountains	0.12	0.28	0.04	0	0.32	0.04	0.32	0.36	0.44
Lions	0.20	0.68	0.24	0.28	0.48	0.40	0.88	0.88	0.88
Bears	0.32	0.60	0.28	0.20	0.80	0.20	0.84	0.88	0.92
Rome	0.16	0.36	0.04	0.20	0.44	0.04	0.56	0.60	0.68
Skiing	0.76	0.72	0.68	0.76	0.72	0.60	0.84	0.96	0.96
Sunrises	0.76	0.72	0.80	0.48	0.48	0.68	0.92	0.92	0.96
Tigers	0.08	0.52	0.04	0.16	0.12	0.04	0.28	0.48	0.56
Waterfalls	0.24	0.36	0.08	0.12	0.36	0.04	0.32	0.36	0.48
Wolves	0.08	0.36	0.12	0.12	0.24	0.28	0.56	0.60	0.68
Building	0.32	0.56	0.16	0.12	0.60	0.12	0.60	0.64	0.72
Buses	0.92	0.88	0.80	0.56	0.80	0.44	1.00	1.00	1.00
Cars	0.24	0.56	0.44	0.08	0.48	0.48	0.72	0.76	0.80
Castles	0	0.44	0.08	0.36	0.36	0.12	0.44	0.56	0.64
Colorado	0.44	0.52	0.32	0.20	0.32	0.40	0.64	0.68	0.72
Overall Accuracy	0.36	0.61	0.35	0.24	0.53	0.35	0.70	0.74	0.79

these classifiers operate on different properties of the images; namely color, texture, structure, and textual properties. This diversity allows the fusion to take advantages of the strengths of the individual classifiers, overcome their weakness, and achieve a higher accuracy. Third, the proposed CELF-NN approach outperforms CELF-M and the global Neural Network fusion. In fact, the non-linear fusion used in CELF-NN has a better performance than the linear aggregation used in CELF-M. Besides, the global Neural Network fusion tends to neglect the classifiers with low overall accuracy, even though these classifiers may have good performance for some categories. For instance, the HTD based classifier, which has the worst overall accuracy, has the best performance on the skiing category; and, as we mention before, the TTD based classifier, which has the second worst overall accuracy, has the best performance on the aviation category (72%). For this category, the global Neural Network fusion has a bad performance (52% which is lower than the accuracy obtained by TTD). On the other side, even if CELF-NN couldn't have better performance than TTD, it didn't degrade the accuracy of the best classifier (72%).

8.3 Phoneme Recognition

In this experiment, we illustrate the performance of the proposed local fusion using the phoneme data which was used in the European ROARS ESPRIT project[4]. The aim of this project was the development and implementation of a real time analytical system for French and Spanish speech recognition. The data include 5404 samples and are composed of two classes: class 1 corresponds to nasal vowels, and has 3818 samples, and class 2 corresponds to oral vowels, and includes 1586 samples. Each data sample is represented by 5 features.

The phoneme data was not generated from multiple sources of information and is not intended to test fusion algorithms. To adapt this data to our application, we assume that we have 3 sets of features and a different classifier is trained for each set. These sets are extracted as subsets from the original 5 features. The first set

includes features 1, 2, and 3, the second one includes features 1, 2, and 4, and the third one includes features 2, 3, and 5. For each set, we use a simple k -NN classifier to generate a confidence value.

To validate the results, we use a five fold cross-validation. For each fold, we classify the training data using the 3 k -NN classifiers with their appropriate feature subsets. Then, we use CELF, CELF-NN, and CELF-FI, with the Feature Discrimination (FD) and the Competitive Agglomeration (CA) aspects, to partition the training data into different clusters. Starting with 20 clusters, our approaches reduced the number of clusters to 4. For each cluster, our approaches learned the optimal fusion model.

Figure 8.5 displays Receiver Operating Characteristic (ROC) curves that compare the performance of the individual classifiers, the global fusion (that uses all features in one classifier), and the proposed local fusion approaches. As it can be seen, the proposed approaches have the best overall performance. We notice too that, for this problem, CELF-NN and CELF-FI have bad performance compared to the baseline CELF. In fact, this is due to overfitting problems.

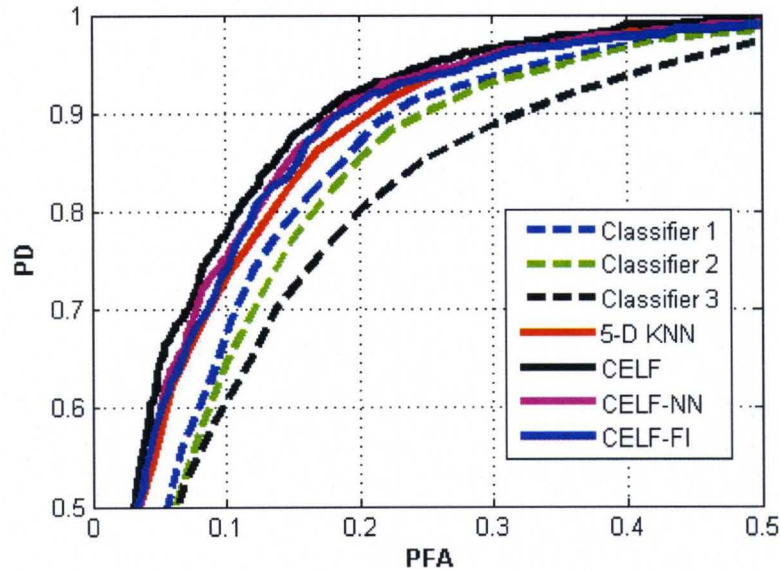


Figure 8.5: Comparison of the three individual classifiers that use subsets of the features, with the global and local fusion for the phoneme data set.

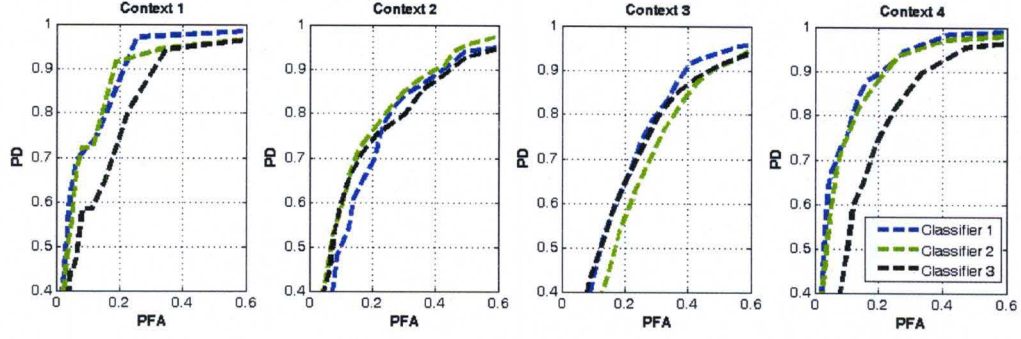


Figure 8.6: Performance of the three individual k -NN classifiers within each of the 4 contexts generated by CELF-CA.

To gain some insight of the behavior of CELF, in Figure 8.6, we display the performance of the three individual k -NN classifiers for the four different contexts. These curves are obtained by hardening the fuzzy partition generated by CELF, and testing the samples within each cluster independently. As it can be seen, the relative performances of the individual classifiers (i.e. the different features here) varies significantly from one context to another. For instance for context 1, classifier 1 (which uses features 1, 2 and 3) has the best overall performance. Consequently, this classifier is considered the most reliable one for this cluster and is assigned the highest aggregation weight as shown in Table 8.6. Similarly, for context 2, classifier 2 is assigned the highest weight.

Table 8.6: Assigned weights to each classifier in each cluster.

Context #	1	2	3	4
Classifier 1	0.52	0.08	0.41	0.58
Classifier 2	0.41	0.62	0.17	0.41
Classifier 3	0.07	0.30	0.42	0.01

CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

We have presented a new fusion approach, called Context Extraction for Local Fusion (CELF). This approach thrives to adapt the fusion method to different regions of the feature space. It is based on a novel objective function that combines context identification and multi-algorithm fusion criteria into a joint objective function. The context identification component was designed to partition the input feature space into different contexts that share similar features and similar response to the different classification algorithms. The fusion component was designed to learn the optimal fusion parameters within each context.

In order to deal with different applications scenario, we proposed several variations of CELF. First, we proposed CELF-FD, an extension that includes a feature discrimination component. This version is advantageous when dealing with high dimensional feature spaces and/or when the number of features extracted by the individual algorithms varies significantly. Then, we proposed CELF-CA, an extension that adds a regularization term to the objective function to introduce competition among the clusters and to find the optimal number of clusters in an unsupervised

way. We have also generalized CELF to support classification with multiple classes (CELF-M). Finally, we proposed two variations that use non-linear fusion. The first one, CELF-NN, is based on Neural Networks, and the second one, CELF-FI, is based on Fuzzy Integrals. The latter variation assigns weights to subsets of classifiers to take into account the interaction between them.

We applied our proposed approaches to fuse multiple landmine detection algorithms that use different sensors, features, and different classification methods. We have shown that the proposed methods can identify meaningful and coherent contexts where different expert algorithms can be identified. Our extensive experiments have also indicated that CELF and its variants outperform the individual classifiers and several standard global fusion methods. We have shown also that our approaches offer good results on other applications as well. In particular, semantic video indexing, image database categorization, and phoneme recognition.

The performance of the different variations of CELF depend mainly on the application and the data set. According to our experiments, CELF-CA showed a degraded performance when dealing with high dimensional feature spaces. This is because it requires the fuzzifier m to be equal to 2, which is not a good choice for high dimensional data. Generally, the non-linear extensions of CELF (CELF-NN and CELF-FI) offer better results than the baseline algorithm. However, in some cases, their performance can degrade due to *overfitting* problems.

9.2 Future Work

Although our approaches have shown promising results, there is still room for improvement. For instance, CELF-CA requires a fixed value of the fuzzifier ($m = 2$) and proved to be not a good choice for high dimensional data. Future research may include investigating other approaches to find the optimal number of clusters.

Potential methods include the AIC (Akaike's Information Criterion) [3, 115] and MDL (Minimum Description Length) [48].

The first term in the objective function of CELF is based on minimizing the sum of squared error and requires the sum of the fuzzy membership to be 1. This may be not be robust in the presence of noisy data. One possible approach to robustify CELF is to convert it to a possibilistic approach [102]. This modification can reduce the effect of the outliers and would generate more meaningful contexts.

Another interesting modification to CELF is to replace the sum of squared error in the second term of the objective function of CELF by a Minimum Classification Error (MCE) term [89]. This modification can improve the results when the outputs of the individual algorithms are distant from the desired ones.

CELF is designed to partition the feature space, learn feature relevance weights and fusion parameters for each context, and possibly learn the optimal number of clusters. This can be a complex optimization problem and is prone to local minima. One possible solution to alleviate this problem is to replace the first term in the objective function (unsupervised learning) with a semi-supervised learning term [78]. In fact for several applications, partial supervision information may be available and may be explored in partitioning the high-dimensional feature space to obtain semantically meaningful contexts.

Finally, the quality of the obtained clusters may need to be assessed and used in the fusion. For instance, a context with good validity measure should be more reliable than a context with worse validity.

REFERENCES

- [1] *Proc. of the 6th International Conf. on Ground Penetrating Radar(GPR'96)*. Tohoku University, Sendai, Japan, 1996.
- [2] Trec-10 proceedings appendix on common evaluation measures. Technical report, 2002. Retrieved from <http://trec.nist.gov/pubs/trec10/appendices/measures.pdf> on Mar. 30, 2007.
- [3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [4] P. Alinat. Periodic progress report 4. Thomson report TS II-Number 5516, ROARS Project ESPRIT, 1993. ASM 93/S/EGS/NC/079.
- [5] S. Auephanwiriyaikul, J. M. Keller, and P. D. Gader. Generalized choquet fuzzy integral fusion. *Information Fusion*, 3(1):69 – 85, 2002.
- [6] L. Ayers and E. Rosen. MIDAS: Mine detection assessment and scoring user's manual v1.1. Technical report, Institute for Defense Analysis, 2004.
- [7] P. Bach. Neutron activation and analysis. In *EUREL International Conf. On The Detection of Abandoned Land Mines*, pages 58–61. Edinburgh, UK, 1996.
- [8] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

- [9] Y. Bi, D. Bell, H. Wang, G. Guo, and J. Guan. Combining multiple classifiers using dempster's rule for text categorization. *Applied Artificial Intelligence*, 21(3):211–239, 2007.
- [10] D. R. Brown. Multisensor vehicular mine detection testbed for humanitarian demining. In *Proc. of the Technology and the Mine Problem Symposium*, pages 73–78. Naval Postgraduate School, Monterey, California, 1996.
- [11] H. Brunzell. Detection of shallowly burried objects using impulse radar. *IEEE Transactions on Geoscience and Remote Sensing*, 37:875–886, 1999.
- [12] C. Bruschini and B. Gros. A survey on sensor technology for landmine detection. *The journal of humanitarian demining*, 2(1), 1998.
- [13] D. Carevic. Clutter reduction and target detection in ground-penetrating radar data using wavelets. volume 3710, pages 973–978. SPIE, 1999.
- [14] D. Carevic. Kalman filter-based approach to target detection and target-background separation in ground-penetrating radar data. In *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets IV*, pages 1284–1288. Orlando, Florida, 1999.
- [15] L. Carin, H. Yu, Y. Dalichaouch, and C. Baum. On the wideband emi response of a rotationally symmetric permeable and conducting target. *IEEE Transactions on Geosci. Remote Sensing*, 39:1113–1206, 2001.
- [16] M. Ceccarelli and A. Petrosino. Multi-feature adaptive classifiers for sar image segmentation. *Neurocomputing*, 14:345–363, 1977.
- [17] S.-B. Cho and J. H. Kim. Combining multiple neural networks by fuzzy integral for robust classification. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 25 of 2, pages 380–384, 1995.
- [18] G. Choquet. Theory of capacities. *Annales de l'Institut Fourier*, 5:131–295, 1955.

- [19] J. C. de Borda. *Mémoire sur les élections au scrutin*. Histoire de l'Académie Royale des Sciences, Paris, 1781.
- [20] D. Denneberg. *Non-additive measure and integral*. Kluwer Academic Publishers Group, Dordrecht, 1994.
- [21] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [22] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. In *Machine Learning*, pages 139–172, 1987.
- [23] E. Fix and J. Hodges. Discriminatory analysis: Nonparametric discrimination: Small sample performance. Technical Report 11, USAF School of Aviation Medicine, Randolph Field, Texas, August 1952.
- [24] G. D. Forney. The viterbi algorithm. *Proc. IEEE*, 61:268–278, 1973.
- [25] H. Frigui and J. Caudill. Building a multi-modal thesaurus from annotated images. In *18th International Conference on Pattern Recognition. ICPR 2006.*, volume 4, pages 198 –201, 2006.
- [26] H. Frigui and J. Caudill. Unsupervised image segmentation and annotation for content-based image retrieval. In *IEEE International Conference on Fuzzy Systems*, pages 72 –77, 2006.
- [27] 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*, 17(1):185–199, 2009.
- [28] H. Frigui, P. Gader, and K. Satyanarayana. Landmine detection with ground penetrating radar using fuzzy k -nearest neighbors. In *proceedings of the IEEE Conference on Fuzzy Systems*, Budapest, Hungary, 2004.

- [29] H. Frigui, K. C. Ho, and P. Gader. Real-time land mine detection with ground penetrating radar using discriminative and adaptive hidden markov models. *EURASIP Journal on Applied Signal Processing*, 12:1867–1885, 2005.
- [30] H. Frigui and R. Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7):1109–1119, 1997.
- [31] H. Frigui and R. Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, 1999.
- [32] H. Frigui and O. Nasraoui. Simultaneous clustering and attribute discrimination. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 158–163, 2000.
- [33] H. Frigui and O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition Journal*, 37:567–581, 2004.
- [34] H. Frigui and S. Salem. Fuzzy clustering and subset feature weighting. In *IEEE International Conference on Fuzzy Systems*, 2003.
- [35] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
- [36] P. Gader, W. H. Lee, and J. N. Wilson. Detecting landmines with ground penetrating radar using feature-based rules, order statistics, and adaptive whitening. *IEEE Transactions on Geoscience and Remote Sensing*, 42(11):2522–2534, 2004.
- [37] P. Gader, W. H. Lee, and J. N. Wilson. Detecting landmines with ground penetrating radar using feature-based rules, order statistics, and adaptive whitening. *IEEE Transactions on Geoscience and Remote Sensing*, 42(11):2522–2534, 2004.

- [38] P. Gader, M. Mystkowski, and Y. Zhao. Landmine detection with ground penetrating radar using hidden markov models. *IEEE Transactions on Geoscience and Remote Sensing*, 39:1231–1244, 2001.
- [39] P. Gader, B. Nelson, H. Frigui, G. Vaillette, and J. Keller. Fuzzy logic detection of landmines with ground penetrating radar. *Signal Processing, special issue on fuzzy logic in signal processing*, 80:1069–1084, 2000.
- [40] P. D. Gader, H. Frigui, B. Nelson, G. Vaillette, and J. M. Keller. New results in fuzzy set based detection of landmines with gpr. In *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets IV*, pages 1075–1084. Orlando, Florida, 1999.
- [41] P. D. Gader, D. Hepp, B. Forester, T. Peurach, and B. T. Mitchell. Pipelined systems for recognition of handwritten digits in usps zip codes. In *Proceedings of U.S. Postal Service Advanced Technology Conference*, Washington, D.C., 1990.
- [42] P. D. Gader, M. A. Mohamed, and J. M. Keller. Fusion of handwritten word classifiers. *Pattern Recogn. Lett.*, 17(6):577–584, 1996.
- [43] P. D. Gader, M. A. Mohamed, and J. M. Keller. Fusion of handwritten word classifiers. *Pattern Recognition Letters*, 17:577–584, 1996.
- [44] T. Gozani. Inspection techniques based on neutron interrogation. In *Proc. of SPIE. Physics-Based Technologies for the Detection of Contraband*, number 2936, pages 9–20. 1996.
- [45] M. Grabisch. A new algorithm for identifying fuzzy measures and its application to pattern recognition. volume 1, pages 145–150, 1995.
- [46] M. Grabisch. A graphical interpretation of the choquet integral. *IEEE Transactions on Fuzzy Systems*, 8:627–631, Oct 2000.

- [47] M. Grabisch, J. Marichal, R. Mesiar, and E. Pap. *Aggregation Functions*. Cambridge University Press, 2009.
- [48] P. D. Grünwald. *The Minimum Description Length Principle*, volume 1 of *MIT Press Books*. The MIT Press, June 2007.
- [49] A. Gunatilaka and B. A. Baertlein. Subspace decomposition technique to improve gpr imaging of anti-personnel mines. In *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets V*, pages 1008–1018. Orlando, Florida, 2000.
- [50] H. Frigui, L. Zhang, P. D. 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*, Orlando, FL, USA, 2007.
- [51] H. Frigui and P. D. Gader. Detection and discrimination of land mines based on edge histogram descriptors and fuzzy k-nearest neighbors. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, 2006.
- [52] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
- [53] S. Hashem. Optimal linear combinations of neural networks. *Neural Networks*, 10(4):599–614, 1997.
- [54] K. J. Hintz. Snr improvements in NIITEK ground penetrating radar. In *Proceedings of the SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets IX*, Orlando, FL, USA, 2004.
- [55] K. Ho, L. Carin, P. Gader, and J. Wilson. An investigation of using the spectral characteristics from ground penetrating radar for landmine/clutter discrimination. *IEEE Transactions on Geoscience and Remote Sensing*, 46(4):1177–1191, 2008.

- [56] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:66–75, 1994.
- [57] Y. S. Huang and S. C. Y. A method of combining multiple classifiers-a neural network approach. *Proc. 12th Int'l Conf. Pattern Recognition and Computer Vision*, pages 473–475, 1994.
- [58] Y. S. Huang and S. C. Y. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):90–94, 1995.
- [59] F. Huenupan, N. B. Yoma, C. Molina, and C. Garreton. Confidence based multiple classifier fusion in speaker verification. *Pattern Recognition Letters*, 29(7):957–966, 2008.
- [60] R. A. Jacobs. Methods for combining experts probability assessments. *Neural Computation*, 7(5):867–888, 1995.
- [61] C. Ji and S. Ma. Combined weak classifiers. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 494–500. MIT Press, Cambridge, 1997.
- [62] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.
- [63] H.-J. Kang, K. Kim, and J. H. Kim. Optimal approximation of discrete probability distribution with kth-order dependency and its application to combining multiple classifiers. *Pattern Recognition Letters*, 18:515–523, 1997.
- [64] L. Kaufman and P. Rousseeuw. *Finding Groups in Data*. John Wiley and Sons, 1989.
- [65] S. Kim, H. Tizhoosh, and M. Kamel. Choquet integral-based aggregation of image template matching algorithms. In *Fuzzy Information Processing Society*,

2003. *NAFIPS 2003. 22nd International Conference of the North American*, pages 143 – 148, 24-26 2003.

- [66] F. Kimura and M. Shridar. Handwritten numerical recognition based on multiple algorithms. *Pattern Recognition*, 24(10):969–983, 1991.
- [67] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- [68] G. Klir and Z. Wang. *Fuzzy Measure Theory*. Plenum, New York, 1992.
- [69] G. J. Klir and M. J. Wierman. *Uncertainty-Based Information: Elements of Generalized Information Theory*. Heidelberg, Physica-Verlag, 1998.
- [70] R. Krishnapuram, H. Frigui, and O. Nasraoui. Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. i. *IEEE Transactions on Fuzzy Systems*, 3(1):29–43, 1995.
- [71] R. Krishnapuram, O. Nasraoui, and H. Frigui. The fuzzy c spherical shells algorithms: A new approach. *IEEE Transactions on Neural Network*, 3(5):663–671, 1992.
- [72] L. Kuncheva. Change-glasses approach in pattern recognition. *Pattern Recogit.Lett.*, 14:619–623, 1993.
- [73] L. Kuncheva. Clustering-and-selection model for classifier combination. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, volume 1, pages 185–188, 2000.
- [74] L. I. Kuncheva. Switching between selection and fusion in combining classifiers: An experiment. *IEEE Transactions on Systems, Man, and CYbernetics-Part B*, 32(2):146–156, 2002.

- [75] L. I. Kuncheva. *Combining Pattern Classifiers*. Wiley-Interscience, New York, 2004.
- [76] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.
- [77] L. Lam and C. Y. Suen. Optimal combination of pattern classifiers. *Pattern Recognition Letters*, 16:945–954, 1995.
- [78] T. Lange, M. Law, A. Jain, and J. Buhmann. Learning with constrained and unlabelled data. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 731 – 738 vol. 1, 2005.
- [79] S. Le Hegarat-Mascle, I. Bloch, and D. Vidal-Madjar. Introduction of neighborhood information in evidence theory and application to data fusion of radar and optical images with partial cloud cover. *Pattern Recognition*, 31(11):1811–1823, 1998.
- [80] K. Leszczynski, P. Penczek, and W. Grochulski. Sugeno’s fuzzy measure and fuzzy clustering. *Fuzzy Sets and Systems*, 15(2):147–158, March 1985.
- [81] R. Liu and B. Yuan. *Information Fusion*, chapter Multiple classifiers combination by clustering and selection, pages 163–168. 2001.
- [82] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [83] E. Mandler and J. Schurmann. Combining the classification results of independent classifiers based on the dempster-shafer theory of evidence. *Pattern Recognition and Artificial Intelligence*, pages 381–393, 1988.

- [84] B. S. Manjunath, J. rainer Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:703–715, 1998.
- [85] B. S. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG 7: Multimedia Content Description Language*. John Wiley, 2002.
- [86] J. E. McFee and Y. Das. Advances in the location and identification of hidden explosive munitions. *Defence Research Establishment Suffield, Report*, (548):83, 1991.
- [87] I. McLean. E.J. nanson, social choice, and electoral reform. *Australian Journal of Political Science*, 31:369–385, 1996.
- [88] A. Mendez-Vazquez, P. Gader, J. Keller, and K. Chamberlin. Minimum classification error training for choquet integrals with applications to landmine detection. *IEEE Transactions on Fuzzy Systems*, 16(1):225–238, Feb. 2008.
- [89] A. Mendez-Vazquez, P. D. Gader, J. M. Keller, and K. Chamberlin. Minimum classification error training for choquet integrals with applications to landmine detection. *IEEE Transactions on Fuzzy Systems*, 16(1):225 – 238, 2008.
- [90] R. S. Michalski and R. E. Stepp. *Machine Learning: an artificial intelligence approach*, volume 1, chapter Learning from observation: Conceptual clustering, pages 331–163. Morgan Kaufmann, 1983.
- [91] J. T. Miller, T. H. Bell, J. Soukup, and D. Keiswetter. Simple phenomenological models for wideband frequency-domain electromagnetic induction. *IEEE Transactions on Geoscience and Remote Sensing*, 39(6):1294–1298, 2001.
- [92] D. Mills. *Improvements to Mine Detectors*. Number PO1408. Australian patent application, 1996.
- [93] M. Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine*, 12(2):34–51, 1991.

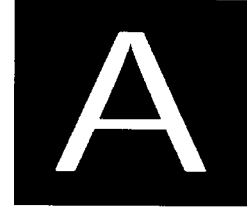
- [94] R. B. Moler. Nuclear techniques for mine detection research. In *Technical Report*. Lake Luzerne, NY, Sponsored by BRDEC, Ft. Belvoir, VA, U.S.A., 1985.
- [95] R. B. Moler. Nuclear and atomic methods of mine detection. In *Workshop Report*. Contract DAAK70-89-C-0002, Dep. of the Army BRDEC, Ft. Belvoir, VA, U.S.A., 1991.
- [96] P.W. Munro and B. Parmanto. Combining neural network regresion estimates with regularized linear weights. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 592–598. MIT Press Cambridge, 1997.
- [97] T. Murofushi. A technique for reading fuzzy measures (i): The shapley value with respect to a fuzzy measure. In *2nd Fuzzy Workshop*, page 39–48, Nagaoka, Japan, Oct 1992. in Japanese.
- [98] T. Murofushi and S. Soneda. Techniques for reading fuzzy measures (iii): Interaction index. In *9th Fuzzy Syst. Symp.*, page 693–696, Sapporo, Japan, May 1993. in Japanese.
- [99] T. Murofushi and M. Sugeno. An interpretation of fuzzy measures and the choquet integral as an integral with respect to a fuzzy measure. *Fuzzy Sets Syst.*, 29(2):201–227, 1989.
- [100] C. Nadal, R. Legault, and C. Y. Suen. Complementary algorithms for the recognition of totally unconstrained handwritten numerals. *Proc. 10th Int'l Conf. Pattern Recognition*, pages 443–449, 1990.
- [101] A. K. Novakoff. Faa bulk technology overview for explosive detection. In *Proc. of SPIE.*, number 1824, pages 2–12. 1992.
- [102] N. Pal, K. Pal, J. Keller, and J. Bezdek. A possibilistic fuzzy c-means clustering algorithm. *Fuzzy Systems, IEEE Transactions on*, 13(4):517 – 530, 2005.
- [103] E. Pap. *Null-additive set functions*. Kluwer Academic Publishers, 1995.

- [104] L. A. Rastrigin and R. H. Erensterin. *Method of Collective Recognition (in Russian)*. Moscow, Russian: Energoizdat, 1981.
- [105] K. L. Russell, J. E. McFee, and W. Sirovyak. Remote performance prediction for infrared imaging of buried mines. In *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets II*, pages 762–769. Orlando, Florida, 1997.
- [106] D. Ruta and B. Gabrys. An overview of classifier fusion methods. *Computing and Information systems*, 7(1):1–10, 2000.
- [107] W. R. Scott. Broadband electromagnetic induction sensor for detecting buried landmines. pages 22–25, 2007.
- [108] G. Shafer. *A Mathematical Theory of Evidence*. Princeton, NJ, Princeton University Press, 1996.
- [109] J.-R. Simard. Improved landmine detection capability (ILDC): Systematic approach to the detection of buried mines using ir imaging. In *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets II*, volume 3079. Orlando, Florida, 1997.
- [110] K. Sirlantzis, S. Hoque, and M. C. Fairhurst. Trainable multiple classifier schemes for handwritten character recognition. In *Proceedings of the 3rd International Workshop on Multiple Classifier Systems*, pages 319–322, Cagliari, Italy, 2002.
- [111] A. F. Smeaton and P. Over. The trec-2002 video track report, 2002.
- [112] G. D. Sower and S. P. Cave. Detection and identification of mines from natural magnetic and electromagnetic resonances. In *Proc. of SPIE.*, volume 2496, pages 1015–1024. Orlando, Florida, 1995.
- [113] M. Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974.

- [114] H. Tahani and J. M. Keller. Information fusion in computer vision using the fuzzy integral. *IEEE Transactions on Systems, Man and Cybernetics*, 20(3):733–741, 1990.
- [115] A. Takasu. On the number of clusters in cluster analysis. In *DS '98: Proceedings of the First International Conference on Discovery Science*, pages 419–420, London, UK, 1998. Springer-Verlag.
- [116] S. L. Tantum, Y. Wei, V. S. Munshi, and L. M. Collins. A comparison of algorithms for landmine detection and discrimination using ground penetrating radar. In *Proceedings of the SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets*, pages 728–735, 2002.
- [117] P. A. Torrione, C. S. Throckmorton, and L. M. Collins. Performance of an adaptive feature-based processor for a wideband ground penetrating radar system. *IEEE Transactions on Aerospace and Electronic Systems*, 42(2):644–658, 2006.
- [118] A. H. Trang, P. V. Czipott, and D. A. Waldron. Characterization of small metallic objects and nonmetallic anti-personnel mines. In *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets I*. Orlando, Florida, 1997.
- [119] K. Tsipis. Report on the landmine brainstorming workshop of aug. 25-30, nov. 96. In *Report No.27, Program in Science and Technology for International Security*. MIT, Cambridge, MA, USA. Web: <http://mcnutt.mit.edu/PSTIS/minereport/minereport.html>, 1996.
- [120] K. Van Erp and L. Schomaker. Variants of the borda count method for combining ranked classifier hypotheses. In *Proc. of the Seventh International Workshop on Frontiers in Handwriting Recognition*, volume 11-13, pages 443–452. 2000.

- [121] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10:988–999, 1999.
- [122] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis. Soft combination of neural classifiers: A comparative study. *Pattern Recognit. Lett.*, 20:429–444, 1999.
- [123] J. N. Wilson and P. D. Gader. Use of the borda count for landmine discriminator fusion. In R. S. Harmon, J. T. Broach, and J. H. Holloway, editors, *Detection and Remediation Technologies for Mines and Minelike Targets XII.*, volume 6553 of *the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, 2007.
- [124] T. R. Witten. Present state of the art in ground-penetrating radars for mine detection. In *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets III*, pages 576–586. Orlando, Florida, 1998.
- [125] I. J. Won, D. A. Keiswetter, and D. R. Hansen. Gem-3: a monostatic broadband electromagnetic induction sensor. *J. Environ. Eng. Geophys.*, 2:53–64, 1997.
- [126] K. Woods, J. Kegelmeyer, W. P., and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):405–410, 1997.
- [127] L. Xu and S. ichi Amari. *Encyclopedia of Artificial Intelligence*, volume 3, chapter Combining Classifiers and Learning Mixture-of-Experts, pages 318–326. IGI Global (IGI) publishing company, 2009.
- [128] L. Xu, A. Krzyzak, and C. Y. Suen. Methods for combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418–435, 1992.

- [129] S. Yu, R. K. Mehra, and T. R. Witten. Automatic mine detection based on ground penetrating radar. In *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets IV*, page 961Ú972. Orlando, Florida, 1999.
- [130] 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. volume 2, pages II-177–II-180, 2008.
- [131] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
- [132] J. Zurada. *Introduction to artificial neural systems*. West Publishing Co., St. Paul, MN, USA, 1992.



ABBREVIATIONS

AMDS	Autonomous Mine Detection System
ANN	Artificial Neural Network
APLM	Anti-Personal with Low Metal content
APM	Anti-Personal Metal
ATLM	Anti-Tank with Low Metal content
ATM	Anti-Tank Metal
CDF	Context-Dependent Fusion
CELF	Context Extraction for Local Fusion
CELF-CA	CELF using Competitive Agglomeration
CELF-FD	CELF with Feature Discrimination
CELF-FI	CELF with Fuzzy Integrals fusion
CELF-M	CELF for Multiple classes

CELLF-NN	CELLF with Neural Networks fusion
DST	Dempster-Shafer Theory
DT	Decision Template
EHD	Edge Histogram Descriptor
EDS	Energy Density Spectrum
EMI	Electro-Magnetic Induction
FA	False Alarm
FAR	False Alarm Rate
FCM	Fuzzy C-Mean
FROC	Free-Response Receiver Operating Characteristic
GPR	Ground Penetration Radar
HM	High Metal
HMC	High Metal Clutter
HMM	Hidden Markov Models
k-NN	k -Nearest Neighbors
LM	Low Metal
MD	Metal Detactor
MFIT	Model FITting detector
MIDAS	MIne Detection Assessment and Scoring

MLP	Multi-Layer Perceptron
NMC	Non-Metal Clutter
PD	Probability of Detection
PFA	Probability of False Alarm
ROC	Receiver Operating Characteristic
SCAD	Simultaneous Clustering and Attribute Discrimination
SCF	Spectral Correlation Feature
SIM	SIMulated mines
TUF	Testing/training Unified Framework
WEMI	Wideband Electro-Magnetic Induction

SYMBOLS

$\ \cdot\ $: Euclidian distance
C	: Number of clusters
N	: Number of samples
K	: Number of classifiers
L	: Number of classes
i	: Index of the clusters
j	: Index of the samples
k	: Index of the classifiers
l	: Index of the classes
H	: Hidden layer size
h	: Index of the neurones in the hidden layer
\mathbf{x}_j	: Feature descriptor of the j^{th} sample

$\mathcal{T} = \{t_j\}$: Desired output
$\mathcal{T} = \{t_{jl}\}$: Desired output (Multi-class data)
$\mathcal{Y}_k = \{y_{kj}\}$: Confidence generated by the k^{th} classifier
$\mathcal{Y}_k = \{y_{kj}\}$: Confidence generated by the k^{th} classifier (Multi-class data)
\mathbf{c}_i	: Center of the i^{th} cluster
d_{ijk}	: Euclidian distance in the k^{th} sub-space
$\mathbf{U} = [u_{ij}]$: Membership degree matrix
$\mathbf{W} = [\omega_{ik}]$: Aggregation weight matrix
$\mathbf{V} = [v_{ik}]$: Feature relevance weight matrix
$\mathbf{\Upsilon} = [\rho_{dki}]$: Connection weights between the inputs and the hidden layer
$\mathbf{\Psi} = [\psi_{ldi}]$: Connection weights between the hidden layer and the outputs
g	: Fuzzy measure
g_i	: Fuzzy measure for context i
C_g	: Choquet integral with respect to g
I_g	: Interaction Index of g
ϕ_g	: Shapley Value of g

CURRICULUM VITAE

Name: Ahmed Chamseddine Ben Abdallah

Address: CECS Department
Speed School of Engineering
University of Louisville
Louisville, KY 40292

Education:

Ph.D., Computer Science & Engineering

Dec. 2010

University of Louisville, *Louisville, Kentucky*

M.S., Mathematical Engineering

Dec. 2005

Tunisia Polytechnic School, *Tunis, Tunisia*

B.Eng., Polytechnic Engineering

June 2004

Tunisia Polytechnic School, *Tunis, Tunisia*

Journal Publications:

1. **A. C. Ben Abdallah**, H. Frigui, and P. Gader, "*Adaptive Local Fusion with Fuzzy Integrals*", IEEE Transactions on Fuzzy Systems (Under review).
2. H. Frigui, **A. C. Ben Abdallah**, and P. Gader, "*Context Extraction for Local Fusion*", Pattern Recognition (Under review).
3. **A. C. Ben Abdallah**, H. Frigui, and P. Gader, "*Ensemble Neural Networks for Adaptive Information Fusion*", (Under preparation).

Conference Publications:

1. **A. C. Ben Abdallah**, H. Frigui, and P. Gader, "*Adaptive Local Fusion with Neural Networks*", International Conference on Artificial Neural Networks (ICANN 2010), Greece, September 2010.
2. **A. C. Ben Abdallah**, H. Frigui, and P. Gader "*Local Fusion with Fuzzy Integrals*", 2010 IEEE International Conference on Fuzzy Systems (under review).
3. **A. C. Ben Abdallah**, H. Frigui, and P. Gader "*Context Extraction for Local Fusion using Fuzzy Clustering and Feature Discrimination*", 2009 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2009), Korea, August 2009.
4. H. Frigui, **A. C. Ben Abdallah**, and P. Gader "*Context-dependent fusion for landmine detection with multisensor systems*", SPIE, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIV, Orlando, April 2009.
5. H. Frigui, J. Caudill, and **A. C. Ben Abdallah**, "*Fusion of Multi-Modal Features for Efficient Content-Based Image Retrieval*", IEEE World Congress on Computational Intelligence (WCCI 2008), Hong Kong, June 2008.

6. H. Frigui, P. Gader, and **A. C. Ben Abdallah**, "*A Generic Framework for Context-Dependent Fusion with Application to Landmine Detection*", SPIE Defense and Security Symposium, Orlando, March 2008.