PERFORMANCE ANALYSIS OF STACKED GENERALIZATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

METE ÖZAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

SEPTEMBER 2008

Approval of the Graduate School of Informatics

Prof. Dr. Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Yasemin Yardımcı
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fatoş Tünay.Y. Vural
Supervisor

Examining Committee Members

Prof. Dr.Yasemin Yardımcı (METU, IS)____________________

Prof. Dr. Fatoş Tünay Yarman Vural (METU, CENG)____________________

Assist. Prof. Dr.İlkay Ulusoy (METU, EENG)____________________

Assist. Prof. Dr. Erhan Eren (METU, IS)____________________

Assist. Prof. Dr. Tuğba Taşkaya Temizel (METU, IS)____________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name : Mete Özay

Signature : __________________

ABSTRACT

PERFORMANCE ANALYSIS
OF
STACKED GENERALIZATION

ÖZAY, Mete
M.S., Department of Information Systems
Supervisor: Prof. Dr. Fatoş Y. Vural

September 2008, 144 pages

Stacked Generalization (SG) is an ensemble learning technique, which aims to increase the performance of individual classifiers by combining them under a hierarchical architecture. This study consists of two major parts. In the first part, the performance of Stacked Generalization technique is analyzed with respect to the performance of the individual classifiers and the content of the training data. In the second part, based on the findings for a new class of algorithms, called Meta-Fuzzified Yield Value (Meta-FYV) is introduced.

The first part introduces and verifies two hypotheses by a set of controlled experiments to assure the performance gain for SG. The learning mechanisms of SG to achieve high performance are explored and the relationship between the performance of the individual classifiers and that of SG is investigated. It is shown that if the samples in the training set are correctly classified by at least one base layer classifier, then, the generalization performance of the SG is increased, compared to the performance of the individual classifiers. In the second hypothesis, the effect of the spurious samples, which are not correctly labeled by any of the base layer classifiers, is investigated.

In the second part of the thesis, six theorems are constructed based on the analysis of the feature spaces and the stacked generalization architecture. Based on the theorems and hypothesis, a new class of SG algorithms is proposed.

The experiments are performed on both Corel data and synthetically generated data, using parallel programming techniques, on a high performance cluster.

Keywords: Ensemble learning, stacked generalization, pattern recognition, parallel computing

ÖZ

YIĞILMIŞ GENELLEME ALGORİTMASININ PERFORMANS ANALİZİ

ÖZAY, Mete
Yüksek Lisans, Bilişim Sistemleri
Tez Yöneticisi: Prof. Dr. Fatoş Y. Vural

Eylül 2008, 144 sayfa

Yığılmış Genelleme Algoritması (YG), bağımsız sınıflandırıcıları sıradüzensel bir mimari altında birleştirerek performanslarını arttırmayı amaçlayan bir topluluk öğrenme tekniğidir. Bu çalışma, iki ana bölümden oluşmaktadır. İlk bölümde, Yığılmış Genelleme tekniğinin performansı, bağımsız sınıflandırıcıların performansına ve eğitim kümesinin içeriğine göre analiz edilmiştir. İkinci Bölümde, Meta-Bulanık Verim Değerleri (Meta-FYV) olarak adlandırılan, yığılmış genelleme için yeni bir algoritma geliştirilmiştir.

İlk bölümde, YG'nin performans kazancını garanti edecek iki hipotezi sunulmuş ve doğruluğu bir dizi kontrollü deney ile sınanmıştır. Deneysel analizlerde, bireysel sınıflandırıcıların performansından daha yüksek performansa ulaşmak için YG'nin öğrenme tekniği incelenmiş ve bağımsız sınıflandırıcılar ile YG'nin performansı arasındaki ilişki araştırılmıştır. Eğer, eğitim kümesindeki örnekler en az bir alt-katman sınıflandırıcı tarafından doğru sınıflandırılırsa, YG'nin genelleştirme performansının bağımsız sınıflandırıcı performanslarına göre arttığı gösterilmiştir. İkinci olarak, herhangi bir alt katman sınıflandırıcı tarafından doğru sınıflandırılamayan parazit örneklerin etkisi incelenmiştir. Herhangi bir alt katman sınıflandırıcı tarafından doğru sınıflandırılamayan örnekleri elemenin YG'nin genel performansını geliştirdiği gösterilmiştir

İkinci bölümde, YG'deki ard arda bağlama işlemi matris cebri ve geometrik veri analizi ile incelenmiştir. Öznitelik uzaylarının ve mimarinin analizine dayalı altı teorem oluşturulmuş ve ispatlanmıştır. Son olarak, deneyler, hem Corel verikümesi üzerinde hem de sentetik olarak üretilen verikümesi üzerinde, yüksek başarımlı bilgisayar kümesinde, parallel programlama teknikleri kullanılarak gerçekleştirilmiştir.

Anahtar Kelimeler: Toplu öğrenme, yığılmış genelleme, örüntü tanıma, paralel hesaplama

To Fatoş Tünay Yarman Vural and Immortal Beloved

ACKNOWLEDGMENTS

I express my sincere appreciation to Prof. Dr. Fatoş Tünay Yarman Vural , who is the architect of the present work, for her guidance and insight not only throughout the research, but the creation and the growth of the present work, via the worker through his life and existence, with her ontology. I want to state my gratefulness to Immortal Beloved, who inspired the being of the thesis and the engineer, who was the "*deaf and blind artist*" of the *state of the art*.

I want to thank to the members of Image Processing and Pattern Recognition Group of Department of Computer Engineering, Ahmet Sayar, Çağlar Şenaras, and especially to Cüneyt Mertayak, Emre Akbaş, and Dr. Mutlu Uysal, for their very precious comments and supports on the implementations.

Thanks go to our faculty secretary Sibel Gülnar and my colleagues Selim Nar and İrfan Emrah Kanat for their spiritual support, at each step, from the beginning through the development of the thesis and to Necla Işıklar, for her endurance for the formation of the thesis.

The suggestions and comments of Asst. Prof. Dr. Tuğba Taşkaya Temizel, especially for her patience, and Asst. Prof. Dr. Alptekin Temizel are gratefully acknowledged, for their advices on parallel computing and research methodologies. I am very grateful to Asst. Prof. Dr. İlkay Ulusoy and Asst. Prof. Dr. Erhan Eren for their support of the motivation on the research.

I want to also thank to Prof. Dr. David Wolpert for his construction of *the state of the art* and precious work on Stacked Generalization, who inspired me about the *art*, at the beginning of the work.

As Newton mentioned, "I was able to see further; because I was standing on the shoulder of giants".

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$S = \{s_i, y_i\}_{i=1}^{N}$	Sample set which consists of samples s_i and their class labels y_i for $i = 1, 2, \ldots, N$.
τ_k	k^{th} feature extractor, for $k = 1, 2, \ldots, K$.
$\underline{x}_{i,k}$	Feature vector of sample s_i extracted by τ_k.
$S_k = \{\underline{x}_{ik}, y_i\}_{i=1}^{N}$	Feature set consisting of feature vectors $\underline{x}_{i,k}$ and class labels y_i, $\forall\ i$
$f_k :\ \underline{x}_{i,k} \to y_i$	The essential classification function that maps each feature vector $\underline{x}_{i,k}$ from feature space to its label y_i.
$h_k :\ \underline{x}_{i,k} \to \hat{y}_{i,k}$	The hypothesis function that maps each feature vector $\underline{x}_{i,k}$ from feature space to its prediction $\hat{y}_{i,k}$.
$S_k{}' = \{x'_{i,k}, y'_i\}_{i=1}^{N'}$	Test dataset, such that, $S_k = S_k^{tr} \cup S_k{}'$.
$h(S_k^{tr}, S_k{}')$	Hypothesis function of the k^{th} classifier trained by S_k^{tr} and tested by the feature set S'_k.
h_k	Shorthand notation for the hypothesis function of the k^{th} classifier.
f^{split}	Dataset splitting algorithm.
$Poly_{\dot{n}}$	The set of trigonometric polynomials of degree $\dot{n}$.
Ξ	Universal constant.
$dist(f, Poly_{\dot{n}})$	Distance metric between the function f and the functions from the polynomial function set $Poly_{\dot{n}}$.
V_c	Vapnik-Chervonenkis (VC) dimension.
$P_e^N(h)$	The error probability of the hypothesis function.
$P_e^N(f)$	The error probability of the essential classification function.
Γ_k	k^{th} descriptor, $k = 1, 2, \ldots, K$.
Υ_k	k^{th} classifier, $k = 1, 2, \ldots, K$.
ω_c	c^{th} class.
$P(\omega_c \mid \underline{x})$	The essential posteriori probability.

$P_k(\omega_c \mid \underline{x})$	The estimated posteriori probability.
$P(error \mid \underline{x})$	Conditional probability error for given $\underline{x}$.
$Loss(\omega^*, \hat{\omega}^*)$	Loss function of the optimal class label of $\underline{x}$, ω^* and the guessed label $\hat{\omega}^*$ with highest $P_k(\omega_c \mid \underline{x})$.
$E_k(.)$	Expectation of Υ_k.
$\Omega = \{\Omega_k\}_{k=1}^{K}$	The output data set.
$f_{p,r}^{combination}$	The mapping that combines the output feature set at p^{th} layer to construct the input feature set at r^{th} layer, $p < r$, $\forall p \geq 1$, $\forall r \geq 2$.
S_{meta}	Meta layer input feature set.
S_{base}	Base layer input feature set.
$\eta_j(\underline{x}_{i,k})$	j^{th}-nearest neighbor of $\underline{x}_{\mathrm{i,k}}$, $\forall j = 1,2,..,\kappa$.
$L(\eta_j(\underline{x}_{i,k}))$	The label of the j^{th}-nearest neighbor of $\underline{x}_{\mathrm{i,k}}$.
$\rho_j(\underline{x}_{i,k})$	The Euclidean distance between $\underline{x}_{\mathrm{i,k}}$ and its j^{th} nearest neighbor $\eta_j(\underline{x}_{i,k})$.
$\mu_c(\underline{x}_{i,k})$	Class membership value of $\underline{x}_{i,k}$.
$\underline{\mu}(\underline{x}_{i,k})$	Class membership vector of $\underline{x}_{i,k}$ obtained by the concatenation of $\mu_c(\underline{x}_{i,k})$.
$\mu^{meta}(s_i)$	Meta layer membership vector of sample s_i.
$\hat{S}_k$	Set of feature vectors that are correctly classified by Υ_k.
$\underline{m}_c$	Mean vector of ω_c.
Σ_c	Covariance matrix of ω_c.
$\sigma_{BC}^{c,\xi}$	Between class variance of ω_c and ω_ξ.
$r_{c,\xi}$	Convergence metric of ω_c and ω_ξ.
M_k	The matrix of the mean values of the classes in Γ_k.
CS$_{\mathrm{SG}}$	The number of correctly classified samples by SG.
$\Re$	Set of real numbers.
$Y \in R(\Psi)$	The matrix Y spans the column space of the matrix Ψ.
$Con(U,V)$	Concatenation of two matrixes *U* and *V*.
$\mathrm{M} \in \Re^{NxCK}$	*N by CK* dimensional membership matrix.
$Y \in \Re^{NxC}$	*N by C* dimensional class label matrix.
$\mathrm{M}(s_i)$	Shorthand matrix notation for $\mu^{meta}(s_i)$.
M^t	Pseudo-inverse of M.
$\hat{\mathrm{M}}$	$Con(\mathrm{M}(s_i))$.
$\hat{Y}$	$Con(Y(s_i))$.

I	Identity matrix.
T_k	The covariance matrix of the classes distributed in Γ_k.
$G=\{G_l,Y_l\}_{l=1}^{K'}$	The set of feature vectors which are classified by at least one classifier, $l=1,2,...,K'$, $K'\leq K$.
MC	Set of samples which can not be classified by at least one classifier, such that, $MC=S-G_l$
$\{\gamma_l\}_{l=1}^{K'}\in\Lambda$	The set of classifiers that can correctly classify $\{G_l,Y_l\}_{l=1}^{K'}$, $K'\leq K$.
$S^{tr}{}_k=\{\underline{x}^{tr}{}_{i,k},y^{tr}{}_i\}_{i=1}^N$	Training dataset of features.
$e(t)$	Classification error at epoch t when a new membership vector is augmented to the meta-layer membership matrix.
$\upsilon(t)$	The ratio of performance increase in SG.
M^{tr}	Meta-layer training membership matrix consisting of membership vectors of training dataset, such that, $\mathrm{M}^{tr}=[\underline{\mu}(\underline{x}^{tr}{}_{i,k})\ \ldots\ \underline{\mu}(\underline{x}^{tr}{}_{i,K})]$.
M^{te}	Meta-layer test membership matrix consisting of membership vectors of test dataset, such that, $\mathrm{M}^{te}=[\underline{\mu}(\underline{x}'_{i,1})\ \ldots\ \underline{\mu}(\underline{x}'_{i,K})]$.
X^{tr}	Solution matrix for $\mathrm{M}^{tr}X^{tr}=Y^{tr}$.
$Perf(SG)$	Classification performance of SG over *S*.
$\mid G\mid$	Cardinality of the set *G*.

The Birds on the high-voltage wires, maximizing the light from the sunset as in SG and Meta-FYV Algorithm

CHAPTER 1

PROLOGUE

"… and the Big Blue Cloud screamed "What is the purpose of the life"
Athena whispered through the Wise Wildflower: "Not the purpose but the process is meaningful"

'Learning is Love'

The epic of grand holy Athena, Book 1

Starting from the Babylonians, people employ computing methodologies for simulating their intelligence to satisfy their natural instincts. In the struggle between the human beings and the nature, they recognized that they could succeed by just having the power over the nature. For this purpose, they, firstly, stole "the fire" from the "gods", in order to take the nature under control using the power of the gods.

They believed that the nature could only be controlled by understanding and implementing the laws of the gods, which creates it. At the same time, they are attracted by the beauty of the nature. In the dilemma of the attraction and the repulsion, the passion has become the love and the adoration.

Under that love and its many folded projections, such as hate, wrangling, glorification and admiration, people have worked and studied rigorously in order to gain the information that will provide them the power. For this purpose, they started to model and simulate the nature, and use these models for their goals. As much as they gain information, they noticed that the dilemma has become more and more complicated and their instinct has been evolved to control the ones that provide the fire, which are the gods.

After a while, the human beings recognized that the main warriors of the nature whom they should fight are themselves. Therefore, they started to employ the intelligence to combat on a new frontier which is the intelligence of the humans. For this purpose, the men have tried to model the intelligence and create the intelligent creatures, in other words, intelligent machines. From the Abacus to digital era, they keep inventing machines which provide monotonically increasing power for humans for the simulations and the controlling.

Throughout the centuries, the intelligent machines have evolved to digital computers. However, the fundamental models and structure of human intelligence is still in its infancy. In order to analyze the human brain, the main conjuncture of the intelligence have been divided into several sub-problems, e.g., language processing, machine learning, pattern recognition, and artificial intelligence by one of the human instincts of the human beings in order to analyze and control the nature, which is divide and conquer.

In pattern recognition, which is the focus of this thesis, one of the main problems is to project the objects of the nature through the computational representations, to the artificial spaces that are created by the humans. In other words, we transform the natural objects to mathematical spaces, which is more comfortable to control the nature under those spaces via the object representations. One of the approaches for the solutions of this transform is to class the objects and describes them under the same labels, which is called clustering and the classification.

In the classification paradigm, since the main goal of the analyses is determining the class labels of the objects, the methodology of the analyses is mostly focused on the investigation of the feature spaces, classification rules, classifier types, etc. The feature spaces are used to represent objects in the abstract spaces. Therefore, the selection of the "best" representation method plays an important role in the class label determination. However, a successful methodology for the selection of the "best" has not been constructed and left as an intuitive task, which is the state of the art, and abandoned to the engineers, which perform art rather than science.

Another task in the determination of the object classes which requires the selection of the “best” model that will perform the “best” clustering or the “best” classification has been left as the state of the art.

Scientists are considered to solve such a huge amount of the state of the art problems as “The Black Artists” so the problems are called “*The black art*”.

In order to attack the *black art* problems, several strategies have been developed. One of them is employing the brute force, which is a mostly try-and-error method by computational power. On the other hand, that methodology has advanced in chaos because of the limitations of the human beings and the machines, compared to almost “unlimited” complexity of *The Black Art* problems, in other words, huge amount of parameters. Another problem of such a methodology is that it deals only with the cause and the reason arguments, without the process in between the arguments and the interaction, mathematically speaking, the transformation.

Another strategy to solve the *black art* problems is extending the present models and the strategies through the multimodal architectures, such as model combination methods, classifier ensemble methods, multilayer models, hierarchical models, etc. Similar to the human instincts and behavior, as discussed above, the instincts of *The Black Artist* have been evolved, and the *black art* problems have been mutated in multimodal forms. In that case, the *black art* problems have become the “best” selection of the “best” representational model combiners, the “best” classification model combiners, etc.

Humorously, we followed our mutated instincts by the guidance of *Athena* and studied on controlling the nature by not focusing on the results, but the processes of the models. In other words, we analyzed the *art* without the concern of seeing it, like the others, who looked and saw the *black,* since they could not see the light of *Athena* that brighten the *black* and transform into light.

In the present work, the *black art* problems of the two layer ensemble learning algorithm, namely, Stacked Generalization (SG), has been examined through the analyses of the algorithm. Theoretical analyses of SG involve the investigation of the feature spaces at each layer and the transformations between the layers. Accordingly, two hypothesis statements are proposed for the generalization performance at the SG. The complementary arguments stated the relations between the feature spaces and the algorithm by focusing on the conditions, where the classification performance increases.

In the experimental analyses, the proposed hypotheses are examined, firstly, by the synthetic datasets which provides an environment for controlling the parameters. In the first set of experiments, the relation of the feature space, which is the projections of the objects, with the outcomes of the algorithm, is investigated. In other words, the behavior of the objects of the nature is considered to be controlled under the artificial problem spaces. During the experiments, the validations of the hypotheses have been confirmed experimentally. In addition, a conceptual equation that defines the relation between the feature space and the performance space of the algorithm is constructed.

In the second part of the experiments, the hypotheses have been tested on the real datasets, which include the projections of the real objects from the real world on the feature space, through features extracted by MPEG 7 descriptors from the Corel Draw Dataset.

During these investigations, the *black art* problems of the SG architecture, which examines the relation between the feature space and the performance and between the classifiers and the performance, are observed. In consequence, three algorithms have been developed that will enlighten the darkness of the *art*, based on the hypotheses.

In the next chapter, the available theoretical and experimental work is surveyed. In Chapter 3, The Stacked Generalization architecture is described. In Chapter 4, the suggested theoretical investigations and the hypotheses are formalized. Chapter 5 presents the experiments in order to validate the hypothesis proposed in this thesis. In Chapter 6, the concatenation operation in SG is analyzed both experimentally and theoretically. Six complementary theorems are introduced and proved based on the hypotheses proposed in Chapter 4. Additionally, the investigations are applied to the classification problem via constructing a meta-layer classification algorithm, which is called Meta-Fuzzy Yield Values (Meta-FYV). Finally, in Chapter 7, the results are discussed.

CHAPTER 2

THE SURVEY ON ENSEMBLE LEARNING

"…and Gaia whispered to Big Blue Cloud through Wise Wildflower"

'Love yields'

The epic of grand holy Athena, Book 1

In this chapter, the challenging problems of the computational learning theory, focusing on the ensemble learning paradigm is surveyed for the purpose of providing the background to the reader, which is necessary to grasp the problems in the available systems.

2.1 The Conjuncture of Computational Learning Theory

Computational Learning Theory is one of the most challenging research areas of the 21st century, since both the definition and the phenomenology of the common sense learning concept has not been well-understood and formalized, yet. The traditional models of computational learning theory are based on constructing the adaptive dynamic algorithms that can acquire information, explore the concepts of the information and generalize the conceptualization in polynomial time complexity by induction or deduction [1].

The fundamental conjunctures of computational learning theory can be investigated in two major streams, namely; epistemology and computer science. Epistemologically, the problem of learning is based on the data inquisition and inference inspired from human learning or other natural learners. In this paradigm, the tasks that perform the learning without analytical models are assumed to be the skill development for humans learning through inbuilt preprogrammed algorithms. However, it can be noticed that this approach restricts the perception of the learning phenomenon. Learning in human brain is known to be the generation of new information through joining the available information, coded in neurons. From this point of view, the learning process takes place in various natural media, other than human brain.

One may consider many natural processes as “learning”. For example, it is not awkward to say that the available information of oxygen and that of hydrogen atoms produces the information to form water molecule. The emission of some of the electrons from the surface by the interaction with photons, the crossover and mutation of the genes in order to cause the evolution and the impregnation of the genes to form a new cell can be considered to produce new information during the processes. Therefore, the learning paradigm can be interpreted as the process of production.

In this interpretation, the learning paradigm is not different from describing the mapping or the transformation functions, in other words, the relations between individual phenomenons on different spaces. Newton states that if the information on the spaces and the states of the beings is available, the transformation function can be approximately achieved. By using the available information on the three laws of the classical dynamics of the particles, it is acknowledged for a long time that more information on the particles could be gained. Following that idea, scientists have been studying to find the “Theory Of Everything” [2] and “Unified Field Theory” [3] starting from the second half of the 20th century. For that purpose, different representations of the objects, such as the studying with the objects in higher or lower dimensions, or recognizing them as the strings as in String Theory [4], [5] or different function descriptions such as uncertainty based interpretation like quantum mechanics have been introduced.

The interpretation contains two conjunctures, namely, information acquisition and its interpretation in order to produce new information. Meanwhile, Feynman [6] stated that the nature and, also, the computation, is not "*classical*". By the investigations on the nature, new approaches to the solutions of the conjunctures have been introduced, such as quantum mechanics, quantum field theory, in other words relativistic quantum mechanics, quantum statistics, etc. It should be noted that all of the effort on the learning theory, such as feature extraction, representation and classifier development, the theories of the physics, such as string representation, wave-particle dualities, and hidden space theories and the mathematics, such as algebraic topology spaces, probabilistic graph theories, and non-deterministic nonlinear equations, are related and focused on the same problem, namely, information acquisition, interpretation and the generalization, for the sake of the production.

From a different perspective, it can be observed that all the effort mentioned above is, ironically, the methodology used by most of the distributed fields of the science. Indeed, this conclusion is not unexpected, since the purpose of all the work is the one inspired by the instincts discussed in the previous chapter, which is the control of the nature.

From the computer science perspective, the problem of learning is defined as developing a mathematical model for the learning phenomenon. The state of the art is focused on the problems of complexity of the learning algorithms, their stability and performance. Complexity requirement of the learning algorithms states two constraints for the algorithms. Firstly, the algorithms should be implemented in polynomial time. Secondly, the tradeoff between the algorithm complexity and the algorithm performance should be acceptable and optimized [7]. By the development of parallel computing methodologies and high performance computing facilities, the tradeoff can be optimized [8]. However, the optimization of the NP complete problems is still an open research area. The generalization ability and the performance criteria of the algorithms are only investigated on very restricted and problem specific domains. Therefore, these problems are still wide open in the machine learning community.

2.2 The Conjuncture of Pattern Recognition

In the framework of pattern recognition, where the learning machines are considered as recognizers or classifiers of the predefined labels, the conjunctures of computational learning theory are attacked by the statistical or structural analysis of the data to extract “useful” information in the algorithmic spaces. Let us now investigate the Pattern Recognition systems, the related problems and their generalization performances.

2.2.1 Pattern Recognition System

Statistical pattern recognition involves three main learning paradigms, namely; supervised learning, unsupervised learning, and the reinforcement learning. In supervised learning, the samples are trained by predefined class labels, however, in unsupervised learning, the predefined class labels are not provided to the classifiers, and in reinforcement learning, the learners are supported by feedbacks [9]. In pattern recognition, the algorithms are formed in two phases; training and testing. The training phase starts by extracting informative features from the data which is, then, used for the modeling of the classifier. Finally, in the testing phase, or the classification phase, the hypothesis spaces, which are constructed in the training phase, are evaluated.

In most of the practical applications, the raw data obtained by the sensor measurements, is embroidered in order to extract the patterns of interest from the measurement environment. This very initial step, called preprocessing, removes or decreases the ambiguity of the methodological and ontological interpretations of the measurement and provides a representation. In the classical models, the assumptions and the lack of knowledge on the states of the observers and the observables (the quantitative and the qualitative significance of the operators and the peers) and the interactions between them (the collapse of the states), compels the interpretations to be constrained.

For the sake of simplicity, we indicate all the samples in both training and test set by the set *S*. If our pattern recognition system involves total of *K* feature extraction algorithms, in the feature extraction step, the information that will be fed to the recognition machine is extracted from the set of *N* patterns $S = \{s_i, y_i\}_{i=1}^{N}$, by an extraction model which maps the raw data to a set of informative features by a linear or non-linear feature extraction mapping τ_k ;

$$S = \{s_i, y_i\}_{i=1}^{N} \xrightarrow{\tau_k} S_k = \{\underline{x}_{ik}, y_i\}_{i=1}^{N} \quad , \qquad \text{(Equation 2.1)}$$

where y_i is the class label of each *d*-dimensional feature vector $\underline{x}_{i,k}$ extracted by k^{th} feature extraction algorithm τ_k , $\forall$ k=1,2,...,K . The process of feature extraction depends on the problem domains by the *no free lunch theorem* [9], [10]. Unfortunately, there is no well-defined methodology to extract a set of features, for a given data. This task is mostly achieved by heuristic techniques, which may yield many redundant and/or irrelevant feature elements. Therefore, a post processing step, such as principle component analysis, and independent component analysis may be employed to reduce the dimension of the feature space [11]. However, the quality and the quantity of the information that is sacrificed causing the information distortion in the mapping, can not be conjectured explicitly, even by using error correcting codes or compression techniques.

In the training step, the correlation and the dependency between the feature vectors of the samples and their class labels is modeled by the classifier, for the estimation of the labels corresponding to the test samples. The *precise* relationship is modeled by a classification function f_k that maps each feature vector $\underline{x}_{i,k}$ from feature space to its label y_i ;

$$f_k : \underline{x}_{i,k} \rightarrow y_i \qquad \text{(Equation 2.2)}$$

The *inference* of the classification function $\{f_k\}_{k=1}^{K} \in F$, where F is the *essential* function space, is the fundamental problem of statistical pattern recognition [12]. The "true" classification function f_k is approximated by a hypothesis function $\{h_k\}_{k=1}^{K} \in H$, where H is the hypothesis space, such that,

$$h_k : \underline{x}_{i,k} \rightarrow \hat{y}_{i,k} \quad , \qquad \text{(Equation 2.3)}$$

where $\hat{y}_{i,k}$ is the prediction of $\underline{x}_{i,k}$ by h_k , by minimizing an error function (risk functional) , such as,

$$error = \sum_{i=1}^{N} \left\| \hat{y}_{i,k} - y_i \right\|^2 \quad , \qquad \text{(Equation 2.4)}$$

where $\|.\|$ indicates the norm space.

Other types of error functions are, also, available [13]. The set of classification functions include linear, non-linear, syntactic or stochastic models [14], [15], [11]. Since the main goal of the classification function is to map the feature vector to a class label of the pattern and their corresponding concepts, the structure of the problem can be characterized as the estimation of the probability density functions of the classes. Especially, for statistical pattern recognition problems, the density functions are estimated by stochastic inference techniques, such as Bayesian [13], Boltzmann [16], Gibbs [16] Learning, Kernel Machines and Statistical Discriminant Analysis.

In order to reduce the error, a feedback to the feature extraction algorithms can be formed, that will enable either the selection of the appropriate feature extractors (descriptor extraction algorithms) or the modification of the feature set [17].

Finally, in the testing phase, the test features S_k' are extracted from the dataset $S_k = \{\underline{x}_{ik}, y_i\}_{i=1}^{N}$ using a splitting f^{split} algorithm, such as cross-validation;

$$S_k = \{\underline{x}_{ik}, y_i\}_{i=1}^{N} \xrightarrow{f^{split}} S_k' = \{x'_{i,k}, y'_i\}_{i=1}^{N'} \quad , \qquad \text{(Equation 2.5)}$$

where $N' < N$, such that, $S_k = S^{tr}{}_k \cup S_k'$, $S^{tr}{}_k$ is the training feature set. The feature space of the test samples is mapped to the label set, with the estimated $h(\underline{x}'_{i,k})$ function;

$$h(\underline{x}'_{i,k}) = \hat{y}'_{i,k} \quad , \qquad \text{(Equation 2.6)}$$

and the performance of the $h(S^{tr}{}_k, S_k')$ function trained by $S^{tr}{}_k$ and tested by the feature set S_k' is inspected by a function such as,

$$Performance(h(S^{tr}{}_k, S_k')) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{y}'_{i,k}}(S_k') \qquad \text{(Equation 2.7)}$$

where $\delta_{\hat{y}'_{i,k}}(S_k')$ is the Dirac measure defined as;

$$\delta_{\hat{y}'_{i,k}}(S_k') = \begin{cases} 1, \hat{y}'_{i,k} \in S_k' \\ 0, \hat{y}'_{i,k} \notin S_k' \end{cases} \qquad \text{(Equation 2.8)}$$

2.2.2 The Curse of Dimensionality Problem

There are many problems of the available pattern recognition systems. One of the major problems of the algorithms is called "the curse of dimensionality", which states the exponential dependence of the number of the samples on the dimension of the feature vector space. In [18], Bishop states that dependence as a partitioning problem of the sample space by the representation in the feature space, for the mapping to class label space.

Formally, Vapnik [14] states this exponential dependence in the problem of approximating functions, such as the approximation to the function f by its closest function from the set of trigonometric polynomials of degree $\dot{n}$, $Poly_{\dot{n}}$ with the parameter space of dimension $N_n = d^n$ (N_n is the number of parameters) where $f \in \Phi^*$, Φ^* is a set of functions defined on the d-dimensional cube, $[0,1]^d$ and f is s times differentiable by the boundary

$$| f^{(s)}(\underline{x}) - f^{(s)}(\underline{x}') | \leq \Xi \, | \underline{x} - \underline{x}' |^{\varepsilon} \qquad \text{(Equation 2.9)}$$

for some integer s , $0<\varepsilon<1$, and $\Xi < \pi\sqrt{3}/2$ is a universal constant, and the exponential dependency is;

$$dist(f, Poly_n) \leq Cons(f) N_n^{-(s+\varepsilon)/d} \quad , \qquad \text{(Equation 2.10)}$$

where $dist(f, Poly_{\dot{n}}) = \inf_{f^* \in M_n} \sup_{\underline{x}} | f(\underline{x}) - f^*(\underline{x}) |$, that is the distance between f and the closest function f^* from the set $Poly_{\dot{n}}$, and $Cons(f)$ is a constant of the function f.

Since it is a crucial task to control the number of parameters of a classifier, there is no explicit solution to the dimensional curse problem. However, there are some experimental work and approaches to the solution. Empirically, the dimensionality relation ratio, which is the ratio of the sample size, N_s, and the dimension of the feature space, d, is considered as a dimensionality metric, and chosen to be greater than 10 [9].

2.2.3 The Generalization Problem

Another problem in pattern recognition is the generalization ability of the classifiers, which is defined as the performance criteria stated by equation (2.7). Training the classifiers with the optimization constraint of equation (2.4), does not guarantee the generalization performance and may also cause a distortion of the hypothesis space. There may be three reasons of this distortion. First of them is the overtraining of the training set, which happens because of an intensive optimization on the training set. The second one is the underfitting of the classifiers because of trying to fit a more complex function *f* with less complex hypothesis *h*. Finally, one is the curse of dimensionality, or the increasing degree of freedoms of the parameters affect the classifier performance, generating an ill-conditioned partition in the feature space.

One of the theoretical explanations to the overtraining of the classifier and the classifier complexity is stated by Vapnik[14], constructing a probabilistic bound for the classifier error and capacity. Vapnik defines the capacity of the hypothesis space *H* in terms of the number of samples that minimize the risk functional, in equation (2.4), which is called Vapnik-Chervonenkis (VC) dimension of *H* [15]. Equivalently, Vapnik [14] defines the VC-dimension of a set of indicator functions $Q(x,\alpha)$, $\alpha \in \Lambda$, is the maximum number *b* of vectors $\underline{x}_1, \ldots, \underline{x}_b$, which can be separated (*shattered*) in all 2^b possible ways using functions of this set. Formally, Theodoridis et. al [11] defines VC dimension($\mathrm{V_c}$) of *F* , which is the set of binary classifiers, as the largest integer $b \geq 1$, for which $S(F,b)= 2^b$ and if for each sample $\{\underline{x}_i, y_i\}_{i=1}^N$, $S(F,b)= 2^N$, then the $\mathrm{V_c}$ approaches to infinity.

$\mathrm{V_c}$ constructs a bound for the error probability in the classifier. For the training error probability, $P_e^N(h)$, and the essential classification error probability $P_e^N(f)$, that depends on the nature of the data independent of the training set, VC theory states that with a probability at least (1-*p*);

$$P_e^N(f) \leq P_e^N(h) + \Phi(\frac{V_c}{N}) \quad , \qquad \text{(Equation 2.11)}$$

where

$$\Phi(\frac{V_c}{N}) = \sqrt{\frac{V_c(\ln(\frac{2N}{V_c}+1)) - \ln(\frac{p}{4})}{N}} \quad . \qquad \text{(Equation 2.12)}$$

One of the most important results of the VC theory is that it guarantees maximum generalization capability in case of finite VC dimension and increasing number samples, by minimizing the classification error difference between the *F* and *H* spaces. However, VC theory limits the learning capability with VC dimension, by stating that the learning capability is limited by the VC dimension, independent of the probability distribution of the samples. However, in some of the real life applications that can be approximated successfully by the stochastic approaches, especially for the one that can provide a-priori information, *h* with small VC(H) are preferred over *h* with higher VC(H) [15]. This is one of the pitfalls of the theory.

In addition to the VC theory for the analysis of the generalization problem, another approach is training the classifiers within the combinations of the training set, by cross validation, or leave-one-out approaches. In that case, the classifier is avoided from the overtraining of the training feature space. One of the architectures that form a solution for the problem is Stacked Generalization [19], which will be discussed in the next chapter.

2.3 The Conjuncture of Classifier Combination

"An oligarchy is said to be that in which the few and the wealthy and a democracy,
that in which the many and the poor are the rulers."
Aristotle, *Politics.*

"…the desires of the less reputable majority are controlled by
the desires and wisdom of the superior minority."
Plato, *The Republic, Book II.*

The problems of pattern recognition, discussed in the previous sections, which constrains the state-of-the-art solutions, have been attacked by the researchers with different classical paradigms. The general approaches have been selection of the most appropriate feature extraction or classification functions, dimensionality reduction with linear kernels, using nature inspired phenomenon, such as Gibbs and Boltzmann learning, simulated annealing, Artificial Neural Networks and Genetic Algorithms [18], [13], [16]. However, after a while, they all have been entered into the same state-of-the-art paradox and the main concern has become the selection of the best models that correspond to the solution for the specified problem domain [20].

By the development of hybrid systems, it is noticed that complementary classifiers could be more successful than the individuals under restricted problem domains and the classifier combination paradigm, has entered the scene of pattern recognition [20]. In the classifier ensemble, similar to the classical paradigm, several ad hoc methods, such as Hierarchical ARTMAP [21] and multilayer Neural Networks [13], inspired from human learning and vision, have been constructed.

There have been several theoretical studies for the ensemble learning [22], [23]. Most of such theoretical approaches are based on linear combination methods such as averaging, mixture of experts, linear perception, bagging and regressions [24], [25], [26]. However, none of them can provide methods for the analysis of non-linear methods, such as Neural Networks, fuzzy combiners, and concatenation methods [27], [28] [29]. Since the theoretical analyses on the nonlinear combiners are threatened by the lack of analytical methodologies to extract information through the projections of the spaces, the state of interest is mostly focused on the experimental studies. However, because of the increasing number of parameters causing high degree of freedom and the stochastic nature of the architectures, there exists no generalized explanation of the relatively higher performance of the SG, observed during the experimentation.

Dietterich [30] explains the success of the classifier ensembles, or ensemble learning algorithms in the framework of three perspectives; statistical, computational and representational point of views.

From the statistical perspective, when the size of the space H is smaller than the size of the sample space S, the single classifier selection approach may be threatened by the selection of less generalized h, since there may be degenerate solutions for the selection space. Dietterich states that using all of the h and averaging the outputs of h would reduce the risk.

From the computational perspective, when the selection of the best h among H could not be guaranteed, such as in the gradient descent case of Neural Networks, the actual classification function f could never be approximated. Similar to the solution to the problems in pattern recognition, a weighted combination of the classifiers is offered.

In the representational problem, where the hypothesis space *H* may not contain a relatively "good" approximation *h* to the actual classification function *f*, searching the optimal hypothesis in *H* may fail. A solution can be provided by aggregating the functions for the expansion of the function space, which would increase the possibility of finding optimal *h* with respect to *f*.

2.3.1 The Architecture of the Classifier Ensemble System

A classifier ensemble architecture consists of four basic levels; data level, feature level, classifier level and the combination level [31]. Before discussing the levels of the architecture, the terminological ambiguity in the literature on the classifier combination methods should be clarified.

There are two basic methodologies for ensemble learning approaches; fusion and selection. In the fusion approach, individual data or function spaces interact with each other and carry the information about the entire feature space and supposed to work together. On the other hand, in the selection approach, each space is considered by its own, as a subspace of the whole data or function space. The optimal space is, then, chosen by the methods such as voting. Throughout this work, the fusion paradigm will be considered.

Fusion of sensors in the ensemble classifiers is achieved by three types of models; complementary, competitive, and cooperative [32].

In the complementary models, sensors are fused in order to provide a complete description of data space. The complementary models are usually employed in surveillance application for object recognition, detection and tracking [33].

In the competitive models, sensors provide independent measurements of the same information, which is similar to the human vision system, such as capturing images of an object from different perspectives.

In the cooperative models, the data obtained from the individual sensors are combined by producing a data space which can not be obtained from the individual sensors. Meanwhile, these three models can be applied to the other levels of the ensemble learning architecture.

In the feature level fusion, two common methodologies are the direct fusion and the feature space combination. The common type of feature extractors used in visual media processing are descriptor extractors, which provide a mapping from the objects to a set of features which include semantic and syntax information, which are called descriptors, in feature space [46] . Henceforth, in the text, we will assume that the feature extractors are descriptor extractors and the features are descriptors.

The direct fusion of the feature spaces spans the features $\underline{x}_{i,k}$ extracted by different descriptor extractors $\tau_k \; \forall \; k = 1, 2, ...K$, that construct descriptors Γ_k which are directly fed into each individual classifier Υ_k with the corresponding hypothesis functions $h_k, \forall \; k = 1, 2, ...K$, for K classifiers. In that schema, the number of descriptors and the classifiers are taken the same [34], [35]. In the combination of the features, the feature spaces are fused using a combination rule, such as aggregation. In this study, both of the approaches are implemented in the algorithms, in classifier and combination levels.

In the classification level, a subspace of *H* is either used with homogenous (same type of classifiers) or heterogeneous (different type of classifiers) set of classifiers from different spaces H_k. In that case, three types of classifiers, complementary, competitive, and cooperative, can be operated. Moreover, base classifiers may, also, be ensemble classifiers in an ensemble learning architecture [21]. In the meta classification level, the predictions of each individual classifier are combined either in parallel, such as stacked generalization or Neural Networks or sequential, such as boosting and bagging, hierarchically [36], [37], [39]. Since the main goal of the thesis is the investigation of the relationships between the base-layer and meta-layer classifiers, the detailed analysis of the Stacked Generalization architecture will be discussed in the next chapter.

In the design of the ensemble learning architecture, there are two popular construction methods [20]. The first method optimizes the meta classifier by fixing the parameters of the base classifiers and the spaces in the lower levels, which is called decision optimization. The second one is the coverage optimization, whose goal is the optimization of the base classifier parameters and the data dependent spaces relative to the meta classifier, in terms of performance.

From the training point of view, there are three types of meta classifiers according to Kuncheva [31]. The first type is independent of the feature space and do not need retraining after the training of the base classifiers. The second one is data dependent, either implicitly or explicitly. Explicitly data dependent classifiers use the fusion functions of feature vectors. On the other hand, implicitly dependence allows the parameter optimization independent of the feature vectors, just before the prediction of the labels. The third type of the meta classifiers are the dynamic combiners, which can be evaluated during the training of the individual classifiers, such as AdaBoost and dynamic weighting [40], [41], [42], [43].

The main conjunctures of the pattern recognition have been switched into the multimodal forms for the ensemble learning paradigm. Similar problems have appeared for representation of the features, reducing the redundancy in the feature space, finding the best classifier and the problems of the selection of the best combination methods, at all the levels of the ensemble learning.

In the data fusion level, data representation has been evolved into the problem of the selection of the best models for the data space construction partitioning and the distribution among the descriptors. Data acquisition is accomplished by either the distributed sensors or by the same type of sensors measuring different states of the same object.

One of the major problems of data fusion is the selection of the sensor types or structures for data acquisition, for a specific domain. In other words, the measurement or metrology problem is considered as the state of the art [44]. However, the solutions depend on the skill of the artist. The second problem rises because of the gap between the meaning of the data and the corresponding low-level descriptor. For example, representing high level information received from LIDAR, camera, and the photovoltaic detector in the same data space is an extremely difficult task and my yield a lot of discrepancy and redundancy. This problem, also, frustrates the success of the data normalization techniques.

Data space transformations and projections keep their challenges within the addition of the feature level fusion conjectures, in the next level of representation. Since the feature fusion algorithms are at the heart of the ensemble learning architectures, the general approach for the analysis of the architecture is the investigation of the feature space in terms of the space parameters, such as mean variance, covariance, and correlation coefficients, which are related to the performance of the ensemble.

The problem of designing the feature spaces, can be analyzed in two phases. In the first phase, the problem is to define an appropriate descriptor that will map the raw data space to the feature space by the minimum topological deformations. The relationship between the raw data space and the descriptors can be analyzed either in terms of topological mappings [45], statistical analysis [46], [47], [48] or the semantic analysis of the samples [49].

In the second phase of the feature space design, the selection of the subspace from the whole feature space is the point of interest. In all of the subspace selection methods, the main goal is the selection, based on the diversity that is, selecting the most "diverse" subspace. The idea behind the diversity construction is to achieve a group of classifiers that are expert on a distinct rule. In other words, the main goal is to obtain distinct spaces spanned by different representation of the objects, which are competitive, complete, and cooperative. The idea is studied and widely accepted by the pattern recognition community and applied to each level of the ensemble architecture [31], [22], [24].

The subspaces produced by descriptors are either selected by random or nonrandom methods. In the random selection methods, the randomly selected parameters satisfy a dissimilarity metric between the samples. For this purpose, methods like bagging or boosting, or stochastic selection methods, such as genetic algorithms or simulated annealing are used [31].

In the nonrandom selection methods, class label correlated relations and the diversity based on these relations are considered. Moreover, inversely, ensemble learning algorithms such as AdaBoost, Genetic Ensemble, Majority Voting and Stacked Generalization can be used for the feature selection [50], [51], [52].

At the input of the classifiers, either the same set of features are fed into different type of classifiers, or the selected subsets are fed into the same or different type of classifiers, each of which span distinct hypothesis or classifier spaces. As mentioned above, the feature subspace selection is also important at the present level. Error analysis can also be used, to ensure that each classifier has relatively distinct errors on the predictions of the hypothesis, constructing the individual expertise [53], [54].

In various studies, the relationship between the feature spaces, the hypothesis spaces of intra and inter classifiers, that is the agreement of the classifiers, and the hypothesis spaces of the combiners are determined in terms of both generalization performances and topological structures.

In the topological analysis perspective, the methods such as, correlation coefficients of the classifier predictions, predicate analysis, class conditional probability distribution analysis, statistical stability analysis, bias-variance decomposition analysis, and entropy related transformation can be employed [56], [57], [58]. The bias-variance decompositions are mostly used in the error and the performance estimations of the architecture. This approach gives the ability to control the classifier parameters more efficiently, and to understand the relationship between the classifiers and the architecture.

2.3.2 Bias-Variance and Noise in Ensemble System

The bias is considered as the divergence from the essential prediction by the classifier, whereas, the variance as the class prediction divergence of the classifier, independent of the essential state or the label, of the sample [31].

To simplify the notation let us resume the subscript of the descriptor $\underline{x}_{i,k}$ and use $\underline{x}$ for this section. Kohavi and Wolpert [56], [59], define the bias, variance and the noise respectively, as follows;

$$Bias = \frac{1}{2}\sum_{c=1}^{C}(P(\omega_c \mid \underline{x}) - P_k(\omega_c \mid \underline{x}))^2 \quad , \qquad \text{(Equation 2.13)}$$

$$Variance = \frac{1}{2}\sum_{c=1}^{C}(1 - P_k(\omega_c \mid \underline{x})^2) \quad , \qquad \text{(Equation 2.14)}$$

$$Noise = \frac{1}{2}\sum_{c=1}^{C}(1 - P(\omega_c \mid \underline{x})^2) \quad , \qquad \text{(Equation 2.15)}$$

for the fixed label ω_c, given $\underline{x}$ of the distribution $P_k(\omega_c \mid \underline{x})$ predicted by the k^{th} classifier , for C classes, $\forall c= 1,2,.....,C$ and K classifiers, $\forall\ k= 1,2,...,K$.

As mentioned before, the above interpretation can be used for the error analysis of the ensemble. By defining the probability of the error for a given $\underline{x}$ as the divergence between the prediction of the classifier and the true class label of $\underline{x}$;

$$P(error \mid \underline{x}) = 1 - \sum_{c=1}^{C} P(\omega_c \mid \underline{x}) P_k(\omega_c \mid \underline{x}) \quad . \qquad \text{(Equation 2.16)}$$

the total error can be calculated over the entire sample space *S* by;

$$P(error) = \sum_{S} P(error \mid \underline{x}) p(\underline{x}) dx \quad .$$

Using equations (Equation 2.11- Equation 2.14), the conditional error is;

$$P(error \mid \underline{x}) = 1 - \sum_{\omega_c} P(\omega_c \mid \underline{x}) P_k(\omega_c \mid \underline{x}) + \frac{1}{2}\sum_{\omega_c} P(\omega_c \mid \underline{x})^2 + \frac{1}{2}\sum_{\omega_c} P_k(\omega_c \mid \underline{x})^2$$

$$- \frac{1}{2}\sum_{\omega_c} P(\omega_c \mid \underline{x})^2 - \frac{1}{2}\sum_{\omega_c} P_k(\omega_c \mid \underline{x})^2 \quad , \qquad \text{(Equation 2.17)}$$

$$P(error \mid \underline{x}) = \frac{1}{2}(\sum_{\omega_c} (P(\omega_c \mid \underline{x}) - P(\omega_c \mid \underline{x}))^2) + \frac{1}{2}(1 - \sum_{\omega_c} P_k(\omega_c \mid \underline{x})^2)$$

$$+ \frac{1}{2}(1 - \sum_{\omega_c} P(\omega_c \mid \underline{x})^2) \quad , \qquad \text{(Equation 2.18)}$$

Therefore,

$$P(error \mid \underline{x}) = Bias + Variance + Noise \quad . \qquad \text{(Equation 2.19)}$$

In addition to the Wolpert's interpretation, Domingos [60] defines the bias, variance and noise, respectively, as follows

$$Bias = Loss(\omega^*, \hat{\omega}^*) \quad , \qquad \text{(Equation 2.20)}$$

where $Loss(\omega^*, \hat{\omega}^*)$ is the loss function of the optimal class label of $\underline{x}$, ω^* and the guessed label $\hat{\omega}^*$ with largest $P_k(\omega_c \mid \underline{x})$. For zero-one decomposition;

$$Loss(\omega, \hat{\omega}) = \begin{cases} 0, & \omega^* = \hat{\omega}^* \\ 1, & \omega^* \neq \hat{\omega}^* \end{cases} \quad . \qquad \text{(Equation 2.21)}$$

Similarly, variance is defined as,

$$Variance = E_k(Loss(\hat{\omega}^*, h_k(\underline{x}))) \quad , \qquad \text{(Equation 2.22)}$$

where E_k is the expectation on all possible instances of the classifier *k*, and the noise is;

$$Noise = E_k(Loss(f(\underline{x}), \omega^*)) \quad . \qquad \text{(Equation 2.23)}$$

For the zero-one loss decomposition,

$$Variance \equiv \sum_{\omega \neq \hat{\omega}^*} P_k(\omega \mid \underline{x}) = 1 - P_k(\hat{\omega}^* \mid \underline{x}) \quad , \quad \text{(Equation 2.24)}$$

and

$$Noise = 1 - P(\omega^* \mid \underline{x}) \quad . \quad \text{(Equation 2.25)}$$

In that case, the error is defined by;

$$P(error \mid \underline{x}) = c_1 \times Noise + Bias + c_2 \times \mathrm{var}\,iance \quad , \quad \text{(Equation 2.26)}$$

where c_1 and c_2 are constants. In zero-one error case, and for two class classification, and *Bias =0*;

$$P(error \mid \underline{x}) = (1 - 2 \times Noise) \times Variance + Bias + Noise \quad . \quad \text{(Equation 2.27)}$$

For *Bias =1*;

$$P(error \mid \underline{x}) = (2 \times Noise - 1) \times Variance + Bias - Noise \quad . \quad \text{(Equation 2.28)}$$

It can be seen that, for biased samples, as the *Variance* decreases, $P(error \mid \underline{x})$ decreases, and vice-versa, since $Noise <$ ½ . However, for the biased samples, as the *Variance* increases, $P\left(error \mid \underline{x}\right)$ decreases. It is observed that, for some classifiers, as the variance increases, the bias decreases and vice-versa. Therefore, as the precision and the flexibility of the classifiers increases, like in Neural Networks, their bias decreases. However, changing the parameters in order to increase the performance may damage the sensitivity, and will increase the bias by decreasing the variance. So, an important conjuncture, which is called the bias-variance dilemma, rises.

For some cases, the decomposition theory fails. Domingos recognized that, changing the *k* value of k-nearest neighbor classifier either increases or decreases the error, for different datasets [60]. Another problem of bias-variance decomposition is that it provides an efficient analysis for linear ensemble structures [26], [38]. However, it fails for the nonlinear structures. Therefore, this kind of analysis can not be generalized for all the classifiers.

In this study, we have studied the relative performance variations and the relationships between base-layer classifiers and the ensemble classifier. The detailed discussion is provided in Chapter 3.

At the top level of the ensemble learning architecture, which is called the Meta level, the studies are focused on the selection of the best combiner that will improve the classification performance. There are various types of meta classifier, as much as classical single classifiers, and most of the combiners are the hybrid combinations of them. However, there have been constructed combiners inspired from human vision, like Hierarchical ARTMAP [21], and social committees, like voting [31], [61], [62], etc.

Input to the meta layer can be studied in three major classes; data space of labels, probability density functions, and possibility density functions.

The meta classifiers, which process on the label space, is threatened by the lack of confidence of the classifier predictions. This phenomenon causes problems for defining the certainty of the data space and the weighting of the classifiers. One of the approaches to this problem is to assign random weights to the classifiers, initially, and update the weights by training the meta classifier. However, this is still an open research problem.

Probability density functions may provide more information about the classifiers compared to the pure predictions. There are several meta classifiers that receive the probability density functions (pdf) as feature vectors and augments the data space for the classifier fusion[22], [63], [36], [37]. Even if probability density functions give detailed information on the classifier predictions and the generalization ability, there are challenges on their interpretation. Firstly, reliability of the probability density functions predicted by the classifiers may not be credible. Secondly, there should be a formalism that will consider the probability density functions in a different perspective. However, the available methods are far from explaining the reason of such an evaluation and the success is achieved only on domain specific datasets. Therefore, the approaches still remain as the state of the art, depending on the experience of the researchers [37].

Considering the possibility information of the samples, processed by the classifiers, seems to be a more flexible model. Compared to the probability density function models, the fuzzy membership values are obtained from the base classifiers, and the fuzzy combination rules are applied by the meta classifier [29]. The problem with this approach is the difficulty of constructing the fuzzy feature space significantly. It is not clear whether to use the fuzzified features as vectors in the crisp feature space, or to construct an inherently fuzzy feature space.

2.4 The Peroration

Despite of variety of different attempts to solve the pattern recognition conjunctures using the multimodal approximations, the conjunctures reappeared in a more powerful mutated form for the ensemble learning and classifier fusion paradigms. This was because of the lack of the stable interpretations of the computational learning and pattern recognition theory.

Most of the studies on the ensemble learning are directly focused on achieving “better” generalization performance by repeating the classifier design methodologies on the ensembles. However, in addition to the classical conjunctures of the learning and recognition theories, the new paradigm comes with fundamental problems. Skipping such problems and building new architectures based on incomplete and defecting conjectures, just increase the gap between interpretation of the theories and the architecture design, resulting in a serious paradox.

Despite the powerful conjectures and the postulates of the ensemble learning, or data fusion, some of which are discussed above, it is long far away from constructing stable theories.

In the present work, the main goal is to develop a framework which analyses the architecture of an ensemble learning system. To assure the improved generalization performance, we also attempt to provide explanations of its transformation and classification mechanisms. Based on the experimental and theoretical analysis of the architecture, three classification algorithms are developed.

In the next chapter, the theoretical structure, and the algorithmic variations of stacked generalization is discussed. In Chapter 4, the analytical and in Chapter 5 the experimental investigations of the fuzzy stacked generalization algorithm (fuzzy SG) are introduced, with a new algorithm. In Chapter 6, the concatenation operation at the fuzzy stacked generalization algorithm is analyzed and the analyses are summarized in six theorems, leading to a meta layer classification algorithm, namely, Meta-FYV.

PART 1

PERFORMANCE ANALYSIS OF STACKED GENERALIZATION

In the first part of the thesis, we investigate the relationship between the performances of base layer classifiers and the general performance of Stacked Generalization in terms of function and data space transformations.

In Chapter 3, Stacked Generalization architecture is introduced and in Chapter 4, a fuzzified variation of SG, which is called Fuzzy SG, is theoretically investigated and 2 Hypotheses that define the performance criteria of SG are proposed.

In Chapter 5, the hypotheses are validated with synthetic and real datasets. In addition, a variation of SG for classification which modifies the spurious samples from the training data is proposed.

CHAPTER 3

STACKED GENERALIZATION ARCHITECTURE

3.1 Overview

We have discussed some of the challenging problems of pattern recognition, specifically, ensemble learning algorithms in Chapter 2, focusing on the generalization problem. Wolpert [64] developed a stacking architecture which provides solutions for the overtraining, overfitting, and generalization problems, which is called Stacked Generalization (SG). However, SG brings new problems, which are called "*black art*" and partial solutions to the *black art* problems are discussed [36].

In this chapter, firstly, SG is introduced with its challenging problems. In the second part, the various SG architectures are investigated, and finally the pitfalls of the available architectures are discussed.

3.2 The Stacked Generalization Architecture

Stacked Generalization is an ensemble learning technique, which aims to increase the performance of individual classifiers by combining them under a hierarchical architecture. Individual classifiers at the base-layer, which are called generalizers, are trained by cross-validation and meta-layer classifier is trained on the data which is not used by the individual classifiers. The meta-data may be the combination of different kind of attributes such as crisp predictions, membership values, cross-entropy of the classes and probability density functions. In addition, meta-classifier is tested by the samples which are not introduced at base-layer. Therefore, the training process of the meta-layer classifier is achieved by learning the errors of individual classifiers from their predictions.

Stacked Generalization architecture implements the training process by partitioning the learning set into two subsets which are used for training and testing. SG uses cross validation techniques in order to split the learning set. On the other hand, cross validation techniques are usually used via the winner takes all strategy, which maps the function space to a function that has the highest generalization accuracy. Unlikely, SG uses cross validation in order to gain information on the bias of the individual classifier or classifiers, and use this information for reducing the biases on the higher layers of the architecture by meta layer generalization.

SG may consist of one or more than one classifier. One classifier architectures are used in order to improve the generalization ability of the classifier [64]. Multi-classifier architectures may consist of two or more layers. As discussed on the previous chapter, at the base layer, the base classifiers are trained using in-sample/out-of-sample techniques, such as cross validation or bootstrapping. The trained classifiers are, then, combined at the meta classifier. Meanwhile, more than two layers can be used in the architecture by assigning each layer to each one of the nodes [64].

As a special case of multi-classifier architectures, single classifier architectures are used for density estimation or increasing the generalization accuracy for the classifier [65]. In such kind of architectures, the single classifier is trained on both the base and meta layer.

The motivation of the present work is the theoretical and the experimental analysis of the generalization performance of two layer architecture. We investigate the relationship of the performance of the individual classifiers at the base layer and that of the classifiers at meta-layer. The detailed discussion of the architecture will be provided in the next section.

3.2.1 The Multiple Classifier Architecture

In SG, the *generalizer* is defined as a mapping from the feature set of *N* samples, $S_k = \{\underline{x}_{i,k} \in \Re^d, y_i \in \Re\}_{i=1}^{N}$ together with a query $\underline{x}_{q,k} \in \Re^d$, $\forall q \neq i$, where d is the dimension of the descriptor space, into a guess, $\hat{y}_{q,k} \in \Re$,$\forall q \neq i$. The feature set S_k is split into two sets, such that, $S_k = S^{tr}{}_k \cup \underline{x}_{q,k}$ for one-leave out cross validation.

In the training process performed on the base layer, the classifiers are taught with $S^{tr}{}_k$ and tested with $\underline{x}_{q,k}$. Wolpert [64] defines the cross validation error estimate of the classifiers, Υ_k, with the hypothesis function set, $h_k \in H$ consisting of K classifiers, with respect to S_k by,

$$Error(k) = \frac{1}{N}\sum_{i=1}^{N}(h(S^{tr}{}_k;\underline{x}_{q,k}) - \hat{y}_{q,k})^2 \qquad \text{(Equation 3.1)}$$

where $h(S^{tr}{}_k;\underline{x}_{q,k})$ is the guesses or the predictions of the classifiers, in other words, the set of predictions when the classifier is trained on the training set $S^{tr}{}_k$ and questioned by $\underline{x}_{q,k}$.

The output data space of the base layer may consist of the predictions of each classifier, Υ_k with the hypothesis function h_k, on the feature set S_k ; $\hat{y}_{i,k} = \{h(S^{tr}{}_k;\underline{x}_{i,k})\}_{k=1}^{K}$, and the actual labels of the samples, y_i $\forall i=1,2,..,N.$.

Define a mapping $f_{p,r}{}^{combination}$, which combines all the output Ω_k to generate the input feature vector for the next layer. The output data space of the p^{th} layer, which consists of the set of outputs, $\Omega = \{\Omega_k\}_{k=1}^{K}$, $\forall k$, is then, transformed into the input data space of the r^{th} layer, $p < r$, $\forall p \geq 1$, $\forall r \geq 2$,using a combination mapping $f_{p,r}^{combination}$;

$$\Omega \xrightarrow{f_{p,r}^{combination}} S_r \qquad \text{(Equation 3.2)}$$

The number of layers may be greater than 2. The mapping of $f_{p,r}^{combination}$ continues up to the top layer, which is called meta-layer. For two layer architecture, the input feature set at the layer with $r = 2$ is the meta-layer with S_{meta} defined by S_2. Base layer input feature set S_{base} is defined by S_1 at the layer with $r = 1$, which consists of descriptors. Through the chapter and the thesis, we will consider two layer architecture with the meta-layer at the second layer.

In the mapping, the predictions of the base layer classifiers are transformed as being the meta layer feature vectors of the meta classifier in order to form the meta layer training dataset with the actual labels, $S_{meta} = \{\underline{x}_i^{meta}, y_i\}_{i=1}^{N}$. At the meta layer, the meta classifier Υ_{meta} with hypothesis function h_{meta} ,is trained on S_{meta}, using the cross validation technique.

The purpose of the meta layer training process is to teach Υ_{meta} the biases of the base layer classifiers. Therefore, the meta layer input space should consist of the predictions that provide information on the base layer classifiers. These predictions can be the label estimates, probability density functions or possibility density functions, which are discussed on the previous chapter.

3.2.2 The Black Art Problems in SG

In this subsection, we will investigate the various approaches to form the SG architecture.

In the bottom-up approach, the input data space, S_{base} ,of the base layer can be generated either by one descriptor or multiple individual descriptors, Γ_{k}, $\forall\ k = 1, 2, \ldots, K$ (Figure 3.1). The created feature set can be fed into the same type of classifiers or different type of classifiers, Υ_k having the hypothesis functions h_k, $\forall\ k = 1, 2, \ldots, K$. There are various works examining the classifier combinations [65]-[68].

Among these combinatorial options, feeding each classifier with a different feature set provides an important property. In that case, each classifier gains the ability to be an expert on different aspects of the object represented by the feature space. This approach not only avoids the normalization problem created by the combinations of the features of different nature or decision of each classifier, but also reduces the curse of dimensionality.

On the other hand, there is no reason for choosing one of them instead of the others. This is one of the challenging problems of SG and left as the state of the art. Therefore, it is called as *black art.*

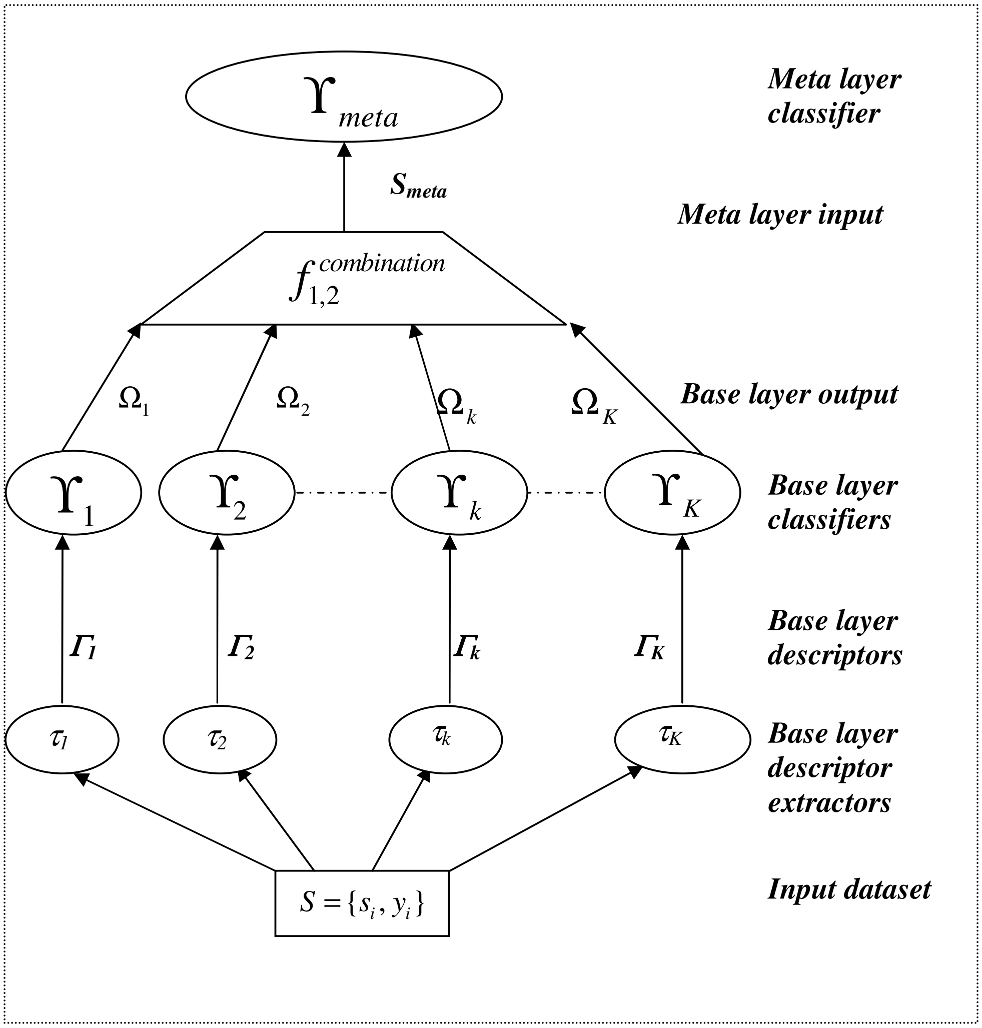


Figure 3.1: 2 layer Stacked Generalization architecture

The selection of the individual classifiers is another *black art* problem. Wolpert [64] offers to use different types of classifiers in order to gain wealthy information. Meanwhile, the algorithm complexity criteria should also be considered in the classifier selection. Another criteria on the classifier selection is selecting the classifiers as they could provide complementary and cooperative information on both the structure of the samples and their generalization accuracy. Therefore, they should neither compress and damage their individual predicted information by redundant information nor they present the same information that has already been introduced by the other classifiers.

The information generated at the output of the classifier Υ_k, which is Ω_k, $\forall$ $k=1,2,\ldots,K$, form the base layer output space. The selection of the type of the information among Ω_k's is another *black art.* This type of information may be selected from the meta feature classes discussed above or they may be either dataset related attributes such as the number of samples, correlation coefficients, statistical skew and kurtosis values or entropy related features, such as class entropy, attribute entropy, or cross entropy [66].

One of the risks that should be considered is the increasing dimensionality while gathering such information. While constructing the input space for the meta layer, the dilemma of the dimensionality reduction and the loss of information is a challenging problem for the determination of the feature combination model, which is another *black art* at the base layer output space.

Majority voting, weighted voting, cross validation, concatenation, linear combination and boosting are some of the combination techniques for the meta layer [68]. The fuzzy SG model proposed by Uysal, Akbas, and Yarman-Vural [21**Error! Reference source not found.**] reduces the base layer output space dimension by using fuzzy k-nearest neighbor (fuzzy k-nn) classifiers at the base layer. Moreover, fuzzy k-nn classifiers provide information on the membership values of each sample, for each classifier. The detailed theoretical analysis of the fuzzy SG will be provided in the next chapter.

The combined features at the input space, S_{meta}, are then, fed into the meta classifier, Υ_{meta} with the hypothesis function h_{meta}. Wolpert [64] recommends training the meta classifier on S_{meta}, in order to teach the meta classifier the information on the classifier biases. However, some meta classifiers such as majority voting, does not need training at the meta layer. A detailed discussion of training and non-training strategies for the meta classifiers is provided by Kuncheva [70]. Surprisingly, for some cases, relatively less expensive non-training strategies may result in the higher classification performance by just changing the type of the meta classifier. This shows the sensitivity of the SG architecture to the application domain. Therefore, the selection of the training strategy and the meta classifier type is the state of the art problem at the meta layer.

3.3 Discussion

The suggested solutions to the black art problems in the literature are domain specific. In most of the work on ensemble learning and especially on SG, various classifiers are examined on several datasets, using several training strategies. However, most of the analyses are far from explaining the reasons of the success and failure of the SG architectures.

One of the reasons of such a lack of analysis is that the common perspective on the community is constructing a “better” classifier, generalizer or the learner on a specific problem domain. Therefore, the challenging problems of the ensemble learning and SG seem to be as the mutated problems of the single classification architectures adapted to the multiple layer and multiple classifiers. Ho [71] suggests that in order to avoid such an infinite recurrence and not considered to lose the sight of the problems, the fundamental challenging problems should be reviewed.

In the present work, we focused on the theoretical and experimental analysis of the SG architecture. This analysis revealed an SG algorithm which yielded a better performance. As a consequence of the analyses, three algorithmic variations of SG are developed. The analyses and the proposed algorithms are introduced in the Chapter 4, 5 and 6.

CHAPTER 4

FUZZY STACKED GENERALIZATION:
ANALYSIS AND VARIATIONS

> "Of course it is very interesting to know how humans can learn. However, this is not necessarily the best way for creating an artificial learning machine. It has been noted that the study of the birds was not very useful for constructing the airplane."
>
> *The Nature of Statistical Learning*
> *Vladimir N. Vapnik*

4.1 Overview

In the previous chapter, Stacked Generalization (SG) architecture, is introduced with its state of the art problems, called *black art.*

In this chapter, we employ a two-level fuzzy SG [21], [72] and investigate the *black art* problems in terms of the parameters that affect the performance of SG. The experimentations revealed us to propose two hypotheses, which assure an improvement on the overall performance of SG. The first hypothesis, involves a sufficient condition on the performance of the individual base-layer classifiers, to improve the overall SG performance. The hypothesis claims that the overall performance of SG reaches an upper bound and gets higher than the performance of the individual classifiers, provided that each training data is correctly labeled by at least one base-layer classifier.

The second hypothesis deals with the effect of the training samples, which cannot be correctly classified by any of the base-level classifiers. It is shown that the elimination of the samples which are misclassified by all of the base layer classifiers, from the training data improves the overall performance of SG. The hypotheses are tested and verified on both real and artificially generated data.

In the next section, fuzzy SG architecture and performance description problem is discussed. In the third section, the proposed hypotheses will be elaborated. The first hypothesis constructs a relationship between the performance of SG and the performance of individual classifiers, whereas the second one introduces an experimental approach to the black art problem in order to increase the classification performance of SG relative to the attributes of the base layer output feature space and the classification behavior of the base layer classifiers. The experiments, which examine the validity of the hypotheses will be provided in the next chapter.

4.2 Fuzzy Stacked Generalization Architecture

Stacked Generalization is an ensemble learning technique that combines more than one classifier in a hierarchical architecture. There is no restriction on the number of classifiers and the number of layers that will be used in the architecture. However, the 2-layer architectures with one classifier at the meta-layer and multiple classifiers at the base layer is very popular and are used for increasing the classification performance of the individual classifiers, at the base layer. The other types of architectures are used for object recognition, image retrieval, density estimation, etc. [72], [73] ,[74].

A 2-layer fuzzy SG classifier receives feature vectors at the input of the base layer fuzzy k-nn classifiers. Then, the outputs of the individual classifiers are combined by concatenating all the outputs of the base-layer classifiers to feed the fuzzy k-nn classifier at the meta-layer.

Base Layer Input Space

In the bottom-up approach, the image dataset, $S = \{s_i\}$, $\forall\ i = 1, 2, .., N$, consisting of N images, is mapped to the feature dataset S_k, using k^{th} descriptor,

$$S = \{s_i\} \xrightarrow{\tau_k} S_k = \{\underline{x}_{i,k}, y_i\}_{k=1}^{K}, \ \forall i \qquad , \qquad \text{(Equation 4.1)}$$

where $\underline{x}_{i,k}$ is the i^{th} feature vector of the *N*-sample training data extracted by the k^{th} descriptor and y_{i} is the label of the corresponding sample. The extracted features at each descriptor Γ_k are fed into each classifier Υ_k, directly. Therefore, the number of classifiers is equal to the number of descriptors.

In the present work, MPEG 7 descriptors are implemented on Corel Draw dataset. The detailed information on the structures of the descriptors is introduced in Chapter 5.

Base Layer Classifiers: Fuzzy k-Nearest Neighbor Classifiers

In the base layer fuzzy k-nn classifiers, the feature data set S_k, firstly, is divided into *N* parts, such that, $S_k = S^{tr}{}_k \cup \underline{x}_{i,k}$ $\forall$ *i=1,2,..,N,* by one-leave out cross validation. In the training process performed on the base layer, the classifiers are taught with $S^{tr}{}_k$ and tested with $\underline{x}_{i,k}$.

In the fuzzy k-nn, the membership value of the sample $\underline{x}_{i,k}$ corresponding to the c^{th} class, ω_c, $\forall\ c = 1,2,..,C$, is calculated by each classifier Υ_k with the hypothesis function h_k as,

$$\mu_c(\underline{x}_{i,k}) = \frac{\sum_{j=1}^{\kappa} L(\eta_j(\underline{x}_{i,k}))(\rho_j(\underline{x}_{i,k}))^{-\frac{2}{m-1}}}{\sum_{j=1}^{\kappa}(\rho_j(\underline{x}_{i,k}))^{-\frac{2}{m-1}}} \quad , \qquad \text{(Equation 4.2)}$$

where $L(\eta_j(\underline{x}_{i,k}))$ is the label of the j^{th}-nearest neighbor of $\underline{x}_{\text{i,k}}$, which is $\eta_j(\underline{x}_{i,k})$ and $\rho_j(\underline{x}_{i,k}) = \left\| \underline{x}_{i,k} - \eta_j(\underline{x}_{i,k}) \right\|$ is the Euclidean distance between $\underline{x}_{\text{i,k}}$ and $\eta_j(\underline{x}_{i,k})$, $\forall\ j = 1,2,..,\kappa$. *m* is the fuzzification parameter and taken as *m=2,* as in [75].

The performance of each base layer classifier Υ_k is calculated from the membership vector of each $\underline{x}_{\text{i,k}}$, $\underline{\mu}(\underline{x}_{i,k}) = \left[\mu_1(\underline{x}_{i,k})\ \mu_2(\underline{x}_{i,k}) \ldots \mu_c(\underline{x}_{i,k}) \ldots \mu_C(\underline{x}_{i,k})\right]$

$$\hat{y}_{i,k} = \max(\underline{\mu}(\underline{x}_{i,k})) \qquad \text{(Equation 4.3)}$$

where $\hat{y}_{i,k}$ is the estimated label of $\underline{x}_{\text{i,k}}$ by h_k of Υ_k.

By equation (2.7), the classification performance of h_k is defined as,

$$Perf(h_k) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\hat{y}_{i,k}}(S_k) \qquad . \qquad \text{(Equation 4.4)}$$

Base Layer Output Space, Concatenation and Meta Layer Input Space

Base layer output space is spanned by the membership vectors for each $\underline{x}_{\mathrm{i,k}}$ that are calculated by each classifier Υ_k .

It should be noted that the membership vectors in equation (4.2) satisfies $\sum_{c=1}^{N}\mu_c(\underline{x}_{i,k}) = 1$. Incidentally, this equation aligns each sample on a line at the output space of the base layer classifiers satisfying the line equation. Therefore, the base layer classifiers can be considered as transformations which map the input feature space of any dimension into a point on a line in *C* (number of classes) dimensional space. This property will allow us to fix the dimension of the input space to the number of classes at the output space which is the input to the meta layer input space.

Concatenation of all the outputs of base layer classifier will yield an input space to meta layer with a fixed dimension *CK* through the mapping from this space to the meta layer input space. The class membership values $\mu_c(\underline{x}_{i,k})$ obtained from each classifier Υ_k is concatenated by forming the feature vector of the meta layer classifier Υ_{meta} with the hypothesis function h_{meta};

$$\underline{\mu}^{meta}(s_i) = \left[\mu_1(\underline{x}_{i,1}) \;...\; \mu_{\mathrm{C}}(\underline{x}_{i,1}) ... \mu_1(\underline{x}_{i,K}) ... \mu_C(\underline{x}_{i,K})\right] \qquad , \qquad \text{(Equation 4.5)}$$

$$\underline{\mu}^{meta}(s_i) = \left[\mu_C(\underline{x}_{i,K})\right] \qquad \text{(Equation 4.6)}$$

by satisfying $\sum_{k=1}^{K}\sum_{c=1}^{C}\mu_c(\underline{x}_{i,k}) = K$. Remarking that each classifier Υ_k is fed by each descriptor Γ_k. Therefore, the proposed feature vector $S_{meta} = \{\underline{\mu}^{meta}(s_i)\}_{i=1}^{N}$ is a *CK* dimensional feature vector.

Note that the dimension of the feature space at meta-layer is independent from the dimensions of the feature spaces at the base layer classifiers. Therefore, no matter how high is the dimension of the feature vectors at the base layer, this architecture fixes the dimensions at meta-layer to *CK.* This may be considered as a partial solution to dimensional curse problem provided that CK is bounded to a value to assure statistical stability to avoid curse of dimensionality.

The membership vector of the training dataset, $\underline{\mu}(\underline{x}^{tr}{}_{i,k})$, is constructed by the leave-one-out cross-validation at Υ_k and is fed into the meta classifier the training dataset, S^{tr}_{meta} , by concatenation. Similarly, the membership vector of the test dataset, $\underline{\mu}(\underline{x}'_{i,k})$, is constructed by teaching Υ_k with the training dataset and examining with the test dataset, and then, is fed as the testing dataset for the meta classifier, S^{te}_{meta}, by concatenation.

Meta Layer Classifier

The meta fuzzy k-nn classifier Υ_{meta} proposes the classification on the feature vector S_{meta} either by training [72] or without training. Similar to the base layer fuzzy k-nn, the meta-layer fuzzy k-nn calculates the memberships by the equation (4.2) and the performance is calculated by the equation (4.4).

4.2.1 Some Comments on the Performance of Fuzzy Stacked Generalization

The performance of the SG varies greatly depending on several parameters. Specifically, the type of the individual classifiers, the dimensions of feature vectors, the size and distribution of training set, number of classes, and the relationship between all of these parameters affect the performance.

It is well known that, for real life applications, where there are too many linearly non-separable classes, even if we increase the dimension of the feature space to achieve seperability, the classification performance does not increase due to the curse of dimensionality problem. On the other hand, reducing the dimensionality by the methods such as principal component analysis, normalization, and feature selection algorithms causes the lack of information. Therefore, one needs to find a balance between the dimensionality curse and the information deficiency with the seperability problems, in designing the classifiers.

Employing distinct descriptors for each classifier enables us to split various coherent properties of feature space, such as color, shape, and texture, which yields a set of relatively low dimensional feature spaces, for the individual classifiers at the base layer. It, also, allows us to control and improve the performance of the individual classifier, independent of the other.

Moreover, implementing the classifiers on their individual feature spaces allows us to deal with the features with consistent information, since each sample is considered in its individual space with its specific attributes without any deformation of the space that could happen because of the interactions with the other spaces. In addition, the dimension can be restricted in the individual spaces without loss of information. Moreover, individual classifiers produce complete information about the space and their structure, which is the information on the quantity of the generalization ability of the classifier on each sample. Therefore, the dilemma of curse of dimensionality and the information gain could be controlled, up to a point. As a result, this approach yields a better performance than feeding the same and most of the time high dimensional feature vector to all of the classifiers [75].

At the meta-layer, concatenating the membership values of the base layer classifiers as input vectors, also, controls the degree of the curse of dimensionality by allowing the meta-classifier to do the mappings from the base layer output space to the meta-layer feature space with two degrees of freedom parameters such as the number of classes and the classifiers of the base layer space, in some manner. Even if, the dimension of the vector space that are input to the base layer classifiers varies a great deal, in an architecture of C classes and K classifiers each of which is fed by K descriptors, the dimension of the meta-layer input vector is fixed by;

$$\dim(S_{meta}) = CK \qquad \text{(Equation 4.7)}$$

This formula indicates that for a relatively low number of classes and classifiers, the size of the input vector for the meta-classifiers is reasonably small, which assures to avoid the curse of dimensionality problem. Concatenation of the vectors at the output of base layer classifiers helps us to learn different properties of the samples, which may result in substantial improvement in the performance. However, the concatenation technique increases the dimension of the feature vector by *CK*. If one deals with a problem of a high number of classes, which may also require high number of base layer classifiers for high performance, the dimension of the feature space at the meta-layer becomes large, causing again curse of dimensionality. This phenomenon is observed clearly for the architectures with higher number of classifiers[75]. However, one may realize that addition of each class will bring the additional samples which represent that class in the training set. Therefore, increasing the number of classes will automatically increase the number of samples in the training data. This fact gives directions for the success of SG even for high number of classes. We will discuss these issues in the next chapter.

An analysis to show the decrease in performance as the number of classes and the classifiers increase is provided in [75]. Since there are several parameters such as the types of classifiers, the number of classes, the number of descriptors, the distribution of the feature vectors, and the mean and variances of these distributions, there is no generalized model that defines the behavior of the performance for high class numbers.

The available research studies on the performance analysis on SG usually investigate the linear architectures [24], [25]. However, the architecture used in this work is doubly non-linear, because of the nonlinearity of the classifiers and the construction of the meta-layer feature vector.

4.3 A Discussion on the Performance of Fuzzy Stacked Generalization

As mentioned in the previous sections, the performance of the SG can not be predicted by analyzing the parameters, such as the number of the individual classifiers, classes, and the training samples, because of the large number of parameters that affect the overall system in a highly nonlinear structure. Especially, the choice of the techniques for constructing the meta-layer input space and the choice of the meta-classifier are extremely complicated problems. Although the fuzzy classification technique, used in two-layer architecture and concatenation of the output vectors of the individual classifier fixes the dimension of the input vector in the meta-layer, the increase in the class number brings many problems, which results in a decrease in the performance when the number of classifiers are limited. However, it is highly desirable to define a framework, which ensures an increase in the performance of the SG, compared to the performance of the individual classifiers. Otherwise, using the expensive SG algorithms is nothing, but, waste of time and effort, with unsatisfactory results.

Wolpert [64], Ting and Witten [63] states that the SG architecture performs better when the individual classifiers can identify the different parts of the feature space. In such a case, the feature space of the input data of base layer classifiers are mapped to a well separated feature space which include the clusters of the samples.

During our experimentations, we noticed that the generalization performance of the overall SG highly depends on the design of the training data. The performance of the individual classifiers by the cross validation, at base layer, provides important information about the contribution of each sample in the training data to the generalization performance at meta-level. In other words, if a sample or a group of samples in the training data is correctly classified by at least one classifier at base layer, then, this group of samples contributes to improve the overall performance of SG. Otherwise, these samples become spurious and distort the feature space at the input of the meta-layer. This observation is consistent with Wolpert [64], Ting and Witten[63] which assures that the individual classifiers can identify the different parts of the feature space. Therefore, during the construction of the meta-layer input space which concatenates the feature spaces obtained from the individual classifiers, it is wise to eliminate the spurious samples, which spoils the seperability of the feature space. In order to assure the seperability at the input of meta-layer, we modified the base layer feature space to include only the samples that are correctly classified by at least one classifier, in cross validation.

Mathematically, in a 2-layer fuzzy SG architecture consisting of *C* classes and *K* classifiers, let's assume that the dataset $\hat{S}_k$ consists total of *N* samples belonging to *C* classes, each of which is correctly classified by at least one classifier. Then, the number of samples which are correctly classified by the k^{th} classifiers, n_k, is defined by the set;

$$\hat{S}_k = \{\underline{x}_{i,k}, y_i\}_{i=1}^{n_k} \qquad \text{(Equation 4.8)}$$

Using the equations (4.3) and (4.4), we can define a lower bound for the performance of the k^{th} classifier Υ_k with the hypothesis function h_k as,

$$Perf(h_k) \geq \frac{n_k}{N} \quad , \qquad \text{(Equation 4.9)}$$

for the special case that each sample is correctly classified by at least one classifier. In the extended case, where each sample can be classified correctly by individual classifiers, there would be overlapping between the correctly classified datasets defined by the equation (4.8), and there may be the correctly classified sample sets which are defined by each classifier, that is, $\hat{S}_{corr} = \hat{S}_1 \cap \hat{S}_2 \cap \hat{S}_3$.

For a special case that C=3, and K=3, and with the assumption that each classifier is correctly classified by just one classifier, the samples belonging to ω_c is correctly classified just by the k^{th} classifier and $c=k$. Therefore, $\{\underline{x}_{i,1}\}_{i=1}^{n_1} \in \omega_1$, $\{\underline{x}_{i,2}\}_{i=1}^{n_2} \in \omega_2$ and $\{\underline{x}_{i,3}\}_{i=1}^{n_3} \in \omega_3$. The Venn diagram representation of the sets is indicated in (Figure 4.1) :

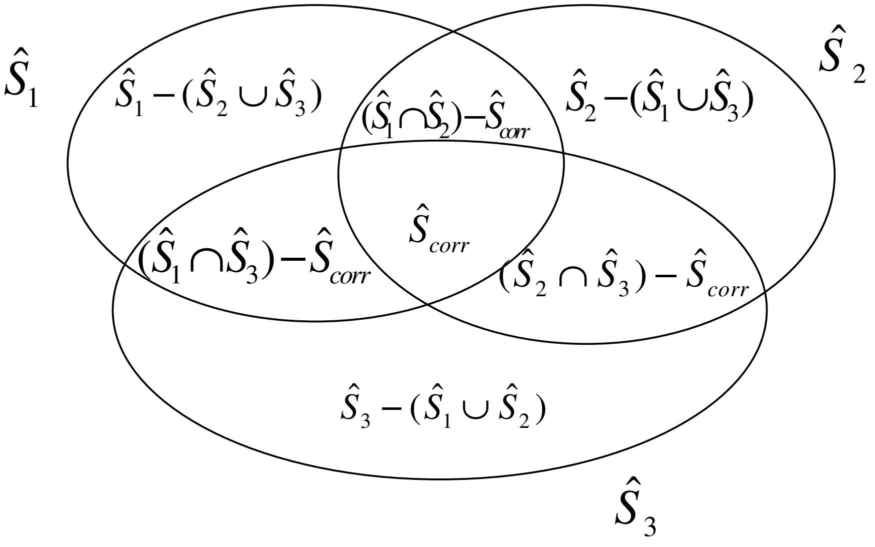

$\hat{S}_1$
$\hat{S}_1 - (\hat{S}_2 \cup \hat{S}_3)$
$(\hat{S}_1 \cap \hat{S}_2) - \hat{S}_{corr}$
$\hat{S}_2 - (\hat{S}_1 \cup \hat{S}_3)$
$\hat{S}_2$
$\hat{S}_{corr}$
$(\hat{S}_1 \cap \hat{S}_3) - \hat{S}_{corr}$
$(\hat{S}_2 \cap \hat{S}_3) - \hat{S}_{corr}$
$\hat{S}_3 - (\hat{S}_1 \cup \hat{S}_2)$
$\hat{S}_3$

Figure 4.1: Venn diagram representation of the correctly classified samples in the classifiers

In that case, ignoring the order of the features and assuming that the dataset is divided into 3 parts, each of which consists of n_k samples with $N = n_1 + n_2 + n_3$;

$$\hat{S}_1 - (\hat{S}_2 \cup \hat{S}_3) = \{\underline{x}_{i,1}, y_{i,1}\}_{i=1}^{n_1}, \qquad Perf(h_1) = \frac{n_1}{N} \qquad \text{(Equation 4.10)}$$

$$\hat{S}_2 - (\hat{S}_1 \cup \hat{S}_3) = \{\underline{x}_{i,2}, y_{i,2}\}_{i=1}^{n_2}, \qquad Perf(h_2) = \frac{n_2}{N} \qquad \text{(Equation 4.11)}$$

$$\hat{S}_3 - (\hat{S}_1 \cup \hat{S}_2) = \{\underline{x}_{i,3}, y_{i,3}\}_{i=1}^{n_3}, \qquad Perf(h_3) = \frac{n_3}{N} \qquad \text{(Equation 4.12)}$$

and the other subsets would be empty sets.

Therefore, following equation (4.2), the membership vector obtained from the k^{th} classifier, Υ_k, fed by k^{th} descriptor, Γ_k, for the i^{th} sample would be;

$$\underline{\mu}(\underline{x}_{i,k}) = \left[\mu_1(\underline{x}_{i,k}) \ \mu_2(\underline{x}_{i,k}) \ . \ . \ . \ \mu_c(\underline{x}_{i,k}) \ . \ . \ . \ \mu_C(\underline{x}_{i,k}) \right] \qquad \text{(Equation 4.13)}$$

If $\{y_i\}_{i=1}^{n_k} = \omega_c$, then $\mu_c(\underline{x}_{i,k}) = \max(\underline{\mu}(\underline{x}_{i,k})), \forall i = 1, 2..., n_k$, with the normalization property that $\sum_{c=1}^{C} \mu_c(\underline{x}_{i,k}) = 1$.

and the membership vectors obtained from each classifier is, respectively;

$$\underline{\mu}(\underline{x}_{i,1}) = \left[\mu_1(\underline{x}_{i,1}) \mu_2(\underline{x}_{i,1}) \mu_3(\underline{x}_{i,1}) \right] \quad ,$$

$$\mu_1(\underline{x}_{i,1}) = \max(\underline{\mu}(\underline{x}_{i,1})) \quad , \qquad \text{(Equation 4.14)}$$

$$\underline{\mu}(\underline{x}_{i,2}) = \left[\mu_1(\underline{x}_{i,2}) \mu_2(\underline{x}_{i,2}) \mu_3(\underline{x}_{i,2}) \right] \quad ,$$

$$\mu_2(\underline{x}_{i,2}) = \max(\underline{\mu}(\underline{x}_{i,2})) \quad , \qquad \text{(Equation 4.15)}$$

$$\underline{\mu}(\underline{x}_{i,3}) = \left[\mu_1(\underline{x}_{i,3}) \mu_2(\underline{x}_{i,3}) \mu_3(\underline{x}_{i,3}) \right] \quad ,$$

$$\mu_3(\underline{x}_{i,3}) = \max(\underline{\mu}(\underline{x}_{i,3})) \quad , \qquad \text{(Equation 4.16)}$$

In the concatenation process, the concatenated membership matrix is obtained by equation (4.6);

$$\mu^{meta}(s_i) = \left[\underline{\mu}(\underline{x}_{i,1}) \ \underline{\mu}(\underline{x}_{i,2}) \ \underline{\mu}(\underline{x}_{i,3}) \right] \quad . \qquad \text{(Equation 4.17)}$$

In the meta-classifier, the performance is calculated based on the meta-layer feature matrix obtained from equation (4.17), in other words, the membership values. Since the performance of the meta-classifier fuzzy κ-nn Υ_{meta} with the hypothesis function h_{meta}, $Perf\left(h_{meta}\right)$, is inversely proportional to the Euclid distance metric, which is used for the membership vector calculation in equation (4.2);

$$Perf\,(h_{meta}) \sim \frac{1}{\rho_j(\underline{x}_{i,k})} \qquad \text{(Equation 4.18)}$$

where $\rho_j(\underline{x}_{i,k})$ is the distance between $\underline{x}_{i,k}$ and its j^{th} nearest neighbor. Therefore, in order to maximize the performance of the meta-classifier, we should minimize the distance metric for the correctly classified samples, and maximize the metric for the misclassified samples;

$$\max(Perf(h_{meta})) \sim \begin{cases} \min(\rho_j(\underline{x}_{i,k})), y_i \in \{y_j\}_{j=1}^{\kappa} \\ \max(\rho_j(\underline{x}_{i,k})), otherwise \end{cases} , \qquad \text{(Equation 4.19)}$$

For the general case, *C* classes and *K* classifiers which are fed by *K* descriptors, and each k^{th} classifier, Υ_k, is fed by k^{th} descriptor, Γ_k,

$$\rho_j(\underline{x}_{i,k}) = \left\| \underline{\mu}(\underline{x}_{i,k}) - \eta_j(\underline{\mu}(\underline{x}_{i,k})) \right\| , \qquad \text{(Equation 4.20)}$$

$$\rho_j(\underline{x}_{i,k}) = [(\underline{\mu}(\underline{x}_{i,k}) - \eta_j(\underline{\mu}(\underline{x}_{i,k})))^T (\underline{\mu}(\underline{x}_{i,k}) - \eta_j(\underline{\mu}(\underline{x}_{i,k})))]^{\frac{1}{2}} \qquad \text{(Equation 4.21)}$$

for $k = 1, 2, ..., K$, $i = 1, 2, ..., N$ and *j=1,2….. κ*.

The above theoretical analysis shows that, if the i^{th} sample and the κ samples are from the same class, the performance would be preserved, since the class membership values of the samples would be closer and the metric would minimize.

On the other hand, if the samples do not belong to the same class, the class membership values of the samples obtained from the classifiers should be exclusively different in order to maximize the metric. This fact indicates that the individual classifiers should be able to make diverse predictions on the samples for improved performance.

Additionally, the number of the classifiers that can classify the samples accurately in a diverse range directly affects the distance metric. Therefore, as the number of the classifiers that can perform the accurate classification increase, even in a diverse and partial range, the performance of the general architecture increases, proportionally.

The performance increase is due to the equations (4.14-4.16), which show that the membership vectors obtained from each individual classifier form a line equation in the feature space, where the correctly classified and the misclassified samples are assembled on the line. In that case, the distance metric is applicable to the distances between the lines. Moreover, even if the membership vectors obtained from each classifier is linearly dependent, the concatenation process destructs the linear dependency, successfully.

Following the above theoretical analysis, the first hypothesis for an increased performance of the overall SG could be stated as follows;

Hypothesis 1 **(Performance of SG)***:* In a 2-layer Homogenous Stacked Generalization architecture consisting of C-classes and K-classifiers, fed by K distinct descriptors; if a group of samples belonging to a specific class in a data set can be classified correctly by at least one classifier, then, the performance of the SG gets higher than that of the performances of the individual classifiers. The overall performance of SG increases as the number of samples which are correctly classified by at least one classifier, is increased.

While the theoretical analysis of the performance description of the SG architecture is based on the fuzzy k-nn, the hypothesis could be tested by the other variations of the stacked generalization. Independent of the classifier and the feature combination structures, the hypothesis mentions a diversity criterion for the feature and the classifier spaces at the base layer that would increase the overall performance.

In the real world problems, the diversity and completeness of the feature sets can not be controlled easily, and the conformity can not be obtained clearly. In that case, the diversity of the classification results can be controlled by force. One of the methodologies in order to achieve the diversity, is the spurious sample elimination. In complementary with Hypothesis 1, the sample elimination method is advanced in Hypothesis 2:

Hypothesis 2 **(Elimination of Spurious Samples)***:* In a 2-layer Homogenous Stacked Generalization architecture consisting of C-classes and K-classifiers, fed by K distinct descriptors; in the leave-one-out cross validation if the training samples that can not be correctly validated by any of the base layer classifiers are eliminated from the meta-layer input dataset for the meta-layer classification process, then the overall performance of the SG in the test stage gets higher than that of the performance of the classification of the whole set including the samples misclassified by the base layer classifiers.

One of the drawbacks of the second hypothesis is that the sample elimination method causes the loss of data on the samples and may cause the curse of dimensionality. On the other hand, sample elimination controls the decisions of the individual classifiers at the meta-layer and enables the diversity at the meta feature set. In addition, as the number of classifiers that can classify the samples correctly increase the performance of the architecture increase proportionally.

In the next chapter, the experiments that examine the validity of the hypotheses will be introduced.

CHAPTER 5

TESTING AND VALIDATING THE PERFORMANCE INCREASE IN STACKED GENERALIZATION

"Give a man a hammer, and every problem looks like a nail."

Paraphrase of comment by J. Friedman

5.1 Overview

For the experiments, synthetic data sets and the Corel Dataset are employed. Firstly, a variety of synthetic data are produced systematically, such that each classifier labels at least one group of data correctly. Then, the effects of classification performances of the individual classifiers on the performance of SG are examined over different data sets. In order to reduce the number of variables that change the parameters discussed in the previous chapter, the data sets are produced by Gaussian distribution via changing the mean and covariance of the distributions. This enables us to overlap the classes as much as we like, so that, we can control the performance of each classifier, to label the samples correctly or incorrectly.

In the second part of the experiments, the datasets with the features extracted from the Corel Dataset using 8 of MPEG-7 [48] descriptors, which are color layout(CL), color structure(CS), edge histogram (EH), region shape (RS), Haar (H), dominant color (DC), scalable color (SC), and homogenous texture (HT), are constructed. Then, the relations of the datasets that are constructed by the classification of the individual descriptor features and the SG classification performance are studied.

All the experiments on the Corel Dataset are discussed in the third part. Experiments are implemented by Matlab, using C/C++ and MPI library, on METU High Performance Cluster.

5.2 Preparation of Synthetic Datasets

In order to study the performance of SG in a controlled experiment apparatus, d-dimensional Gaussian data sets are generated as the representation of each class. While constructing the data sets, with the mean vector, $\underline{m}_c$ and the covariance matrix Σ_c of class *c* with the class conditional density,

$$f(\underline{x} \mid \underline{m}_c, \Sigma_c) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma \mid}} \exp\left[-\frac{1}{2}\left(\underline{x} - \underline{m}_c\right)^T \Sigma_c^{-1}\left(\underline{x} - \underline{m}_c\right) \right]$$

that affect the bias and variance, are systematically varied. Experiments are performed by changing both $\underline{m}_c$ and Σ_c to get an intuitive feeling about the behavior of the SG. One can easily realize that there are explosive alternatives for changing the parameters of the class conditional densities, in the *d*-dimensional vector space. However, it is quite intuitive that rather then the changes in the class scatter matrix, the amount of overlaps among the classes affect the performance of the individual classifiers. Therefore, during the experiments we suffice to control only the amount of overlaps. This task is achieved by fixing the covariance matrix Σ_c, in other words, within-class variance and changing the mean values of the individual classes, which varies the between-class variances, σ_{BC} .

The data sets are displayed in 2-dimensional Euclidean space for the visualization of the feature vectors and for the simplicity of the calculations of controlled experiments. Defining v_i as the eigen-vector of Σ, and, λ_i are the eigen-values of the data set, we have,

$$\sum v_i = \lambda_i v_i \quad . \qquad \text{(Equation 5.1)}$$

Therefore, the central position of the sample distribution, constructed by data sets in 2-dimensional space is defined by ν_1 and $\nu_{2,}$ and the propagation is defined by $\lambda_1^{1/2}$ and $\lambda_2^{1/2}$. In the data sets, the covariance matrices are held fix and equal, the eigenvalues on both axes are equal. As a result, the data sets are generated by the circular Gaussian function with fixed radius.

5.3 Validation of Hypothesis-1 on Synthetic Dataset Experiments

In the experiments, 2 different apparatus are prepared. In the first apparatus, the relationship between the performance of the classifiers at the base layer and the performance of SG is explored. In the second apparatus, the performance of SG is investigated in terms of between-class-variances.

5.3.1 Comparison of the Performances of Individual Classifiers and Fuzzy SG

In this set of experiments, a variety of data sets is generated in such a way that each sample is correctly labeled by at least one classifier, in the base layer. The number of samples in each class is taken as 250, and for 12 classes, 3000 samples with 2-dimensional feature sets are fed to each classifier as input. The performances of the classifiers are observed by fixing the covariance matrices and changing the mean values of Gaussian class conditional density which generalizes the feature vectors. Therefore, the classes that are to be classified by high accuracy are distributed separately, from the other classes which are overlapped in some ratios.

In order to avoid the misleading information in this gradual overlapping process, the classes are first generated apart from each other to assure the linear separablity, in the initialization step. Then, the distance among the mean values of the classes are gradually decreased. The ratio of decrease is selected as one tenth of between-class variance of each 2 classes ω_c and ω_ξ. $\forall$ c $\neq \xi$, $c = 1, 2, .., C$, $\xi = 1, 2, .., C$, which is, $r_{c,\xi} = \frac{1}{10}\sigma_{BC}^{c,\xi}$, where $\sigma_{BC}^{c,\xi} = \left\| m_c - m_\xi \right\|$. The termination condition for the algorithms is $\sum_{c,\xi} \sigma_{BC}^{c,\xi} = 0$,$\forall$ c $\neq \xi$. At each epoch, only one of the mean values approaches to the mean value of another class, while keeping the rest of the mean values fixed. Defining *K* as the number of classifiers fed by *K* descriptors and *C* as the number of classes, the data generation algorithm is given below:

Algorithm 5. 1: The synthetic dataset generation algorithm

```
Initialize: Generate linearly separable data sets of classes separately
for each $\xi$ =1,2,.....,C-1
    for each k=1,2,........,K
        for each c=1,2,........,C
            in the k^{th} classifier, $\Upsilon_k$, group C classes by moving the dataset of
            $\omega_c$ over the dataset of $\omega_\xi$ via $r_{c,\xi}$, such that the class will be
            overlapped, $\sigma_{BC}^{c,\xi} = 0$.
        end for(c)
    end for(k)
    split the data sets into two randomly selected parts, and construct test
    and training sets.
    perform classification in SG using test and training sets.
end for( $\xi$ )
```

In the first set of the experiments, 12 classes, each consisting of 250 samples, are classified using 7 base layer classifiers. The feature sets are prepared with fixed and equal $\mathrm{T}_k = [\Sigma_1 \ . \ . \ . \ \Sigma_c]^T$, which is the covariance matrix of the classes distributed in Γ_k, $\mathrm{T}_k = \begin{pmatrix} 5 & 5 \\ 5 & 5 \end{pmatrix}$, $\forall\ c, c = 1, 2, .., 12$, *k=1,2,…, 7*. In other words, $\lambda_1^{1/2} = 5$ and $\lambda_2^{1/2} = 5$ (equation 5.1).

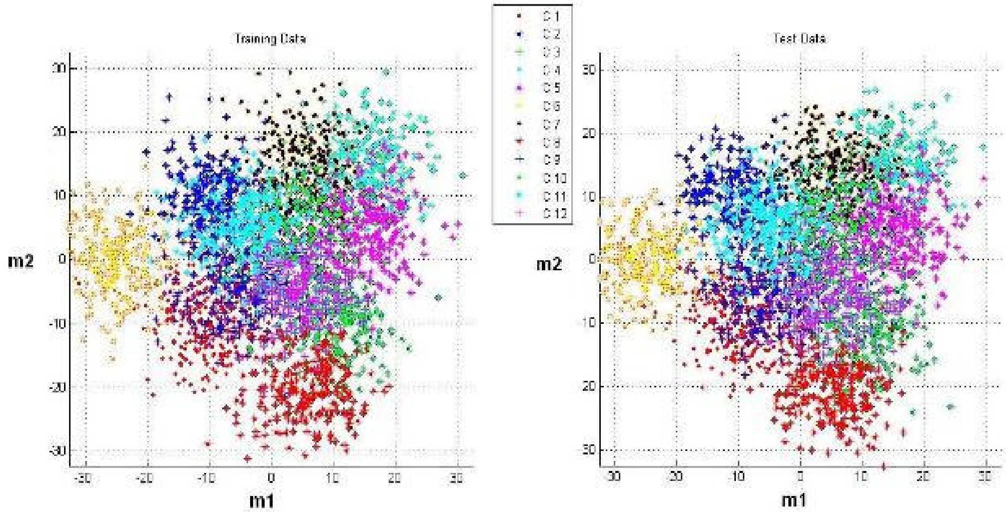
Meanwhile, the classes are distributed with different σ_{BC} and converged towards each other using Algorithm 5.1. The convergence metric, $r_{c,\xi}$, is selected as 5. The matrix $M_k = [m_{c,k}]_{c=1}^{12}$, with the row vectors that contain the mean values of each class c at each descriptor $k=1,2,\ldots, 7$, $m_{c,k}$, are as follows,

$$M = [M_1 \,|\, M_2 \,|\, M_3 \,|\, M_4 \,|\, M_5 \,|\, M_6 \,|\, M_7] \qquad ,$$

$$M = \begin{bmatrix}
-10 & -10 & -10 & -10 & -10 & -10 & -10 & -10 & -10 & -10 & 10 & -15 & -25 & -25 \\
-10 & 10 & -10 & 10 & -10 & 10 & -10 & 10 & -25 & -25 & 0 & 0 & -15 & 10 \\
10 & -10 & 10 & -10 & 10 & -10 & 20 & -10 & 15 & -15 & -10 & -10 & -25 & -25 \\
15 & 15 & 15 & 15 & 25 & 25 & 15 & 15 & 15 & 15 & 10 & 10 & -15 & 10 \\
15 & 5 & -25 & 0 & -15 & 5 & -15 & 5 & -15 & 5 & 15 & 15 & 5 & -10 \\
-25 & 0 & 15 & 5 & 15 & 5 & 15 & 5 & 15 & 5 & 15 & 5 & 0 & 0 \\
5 & 15 & 5 & 15 & 5 & 15 & 5 & 15 & 5 & 15 & 10 & 15 & -25 & 25 \\
5 & -20 & 5 & -20 & 5 & -20 & 5 & -15 & 5 & -15 & -15 & -10 & 25 & -25 \\
-5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & 15 & 10 & 25 & 25 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 0 & 0 & 25 & 0 \\
-5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -15 & 10 & -10 & 10 \\
5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 25 & -25 & 10 & -10
\end{bmatrix}$$

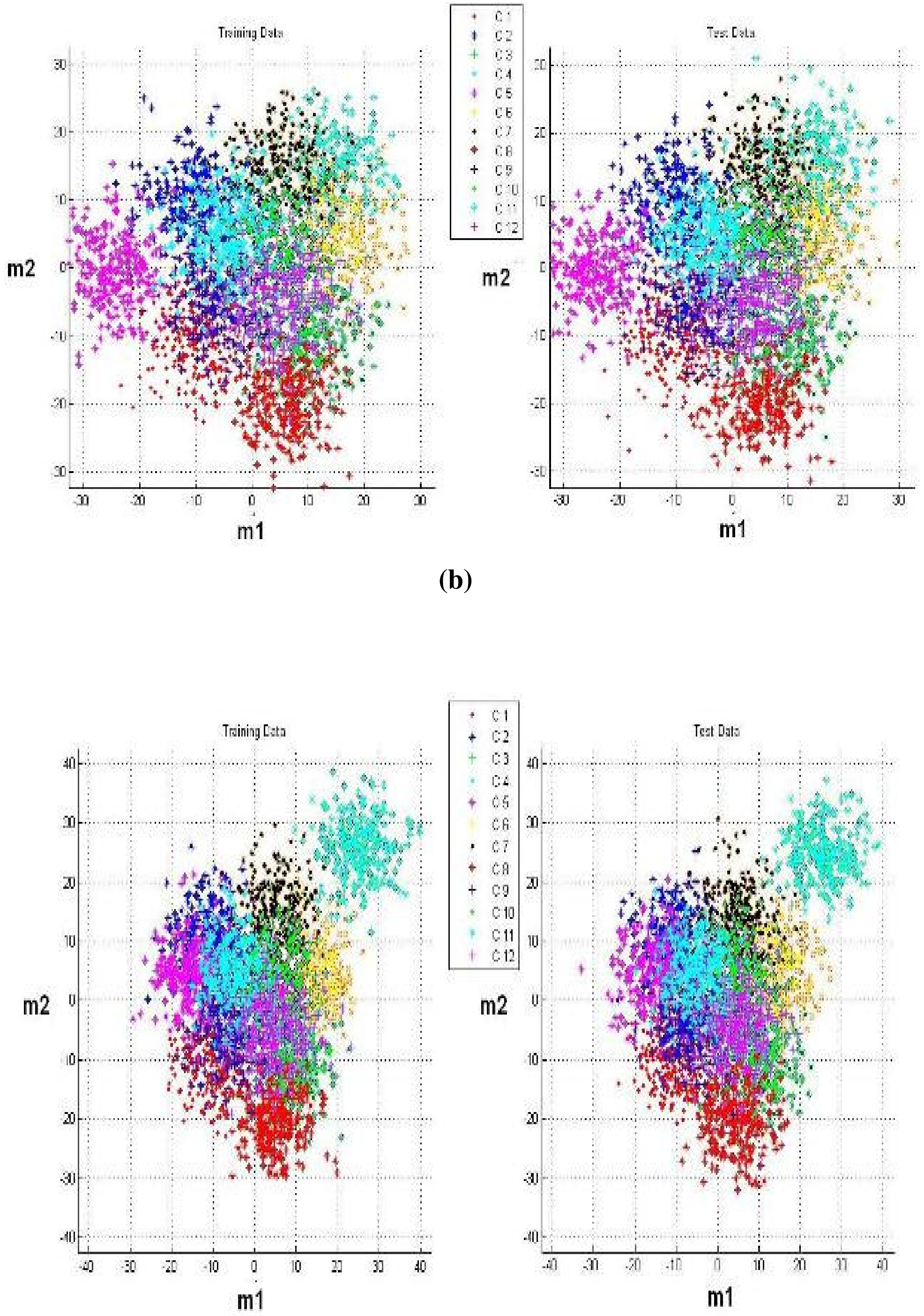
With the matrix introduced above, the distribution of the classes in each descriptor is controlled such that the classification performances of the individual classifiers for a specific class are limited to around 90%.

In Figure 5.1, the distributions of the classes at each descriptor are visualized from *(a)* to *(g)*, from *Descriptor 1* to *Descriptor 7*, respectively.



(a)

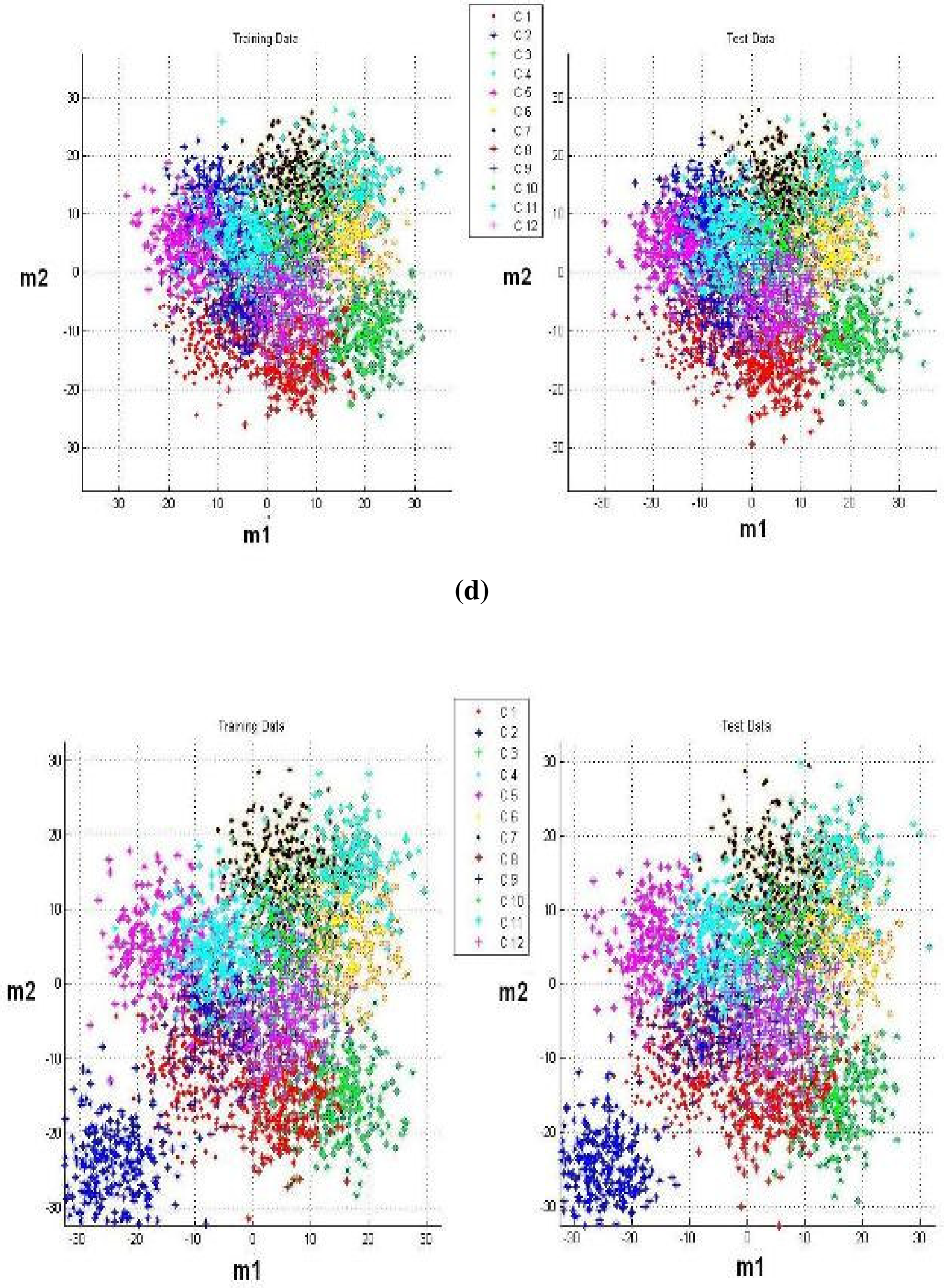
Figure 5.1: Distribution of the classes at, **(a)** Descriptor 1, **(b)** Descriptor 2, **(c)** Descriptor 3, **(d)** Descriptor 4, **(e)** Descriptor 5, **(f)** Descriptor 6, **(g)** Descriptor 7



(b)

(c)

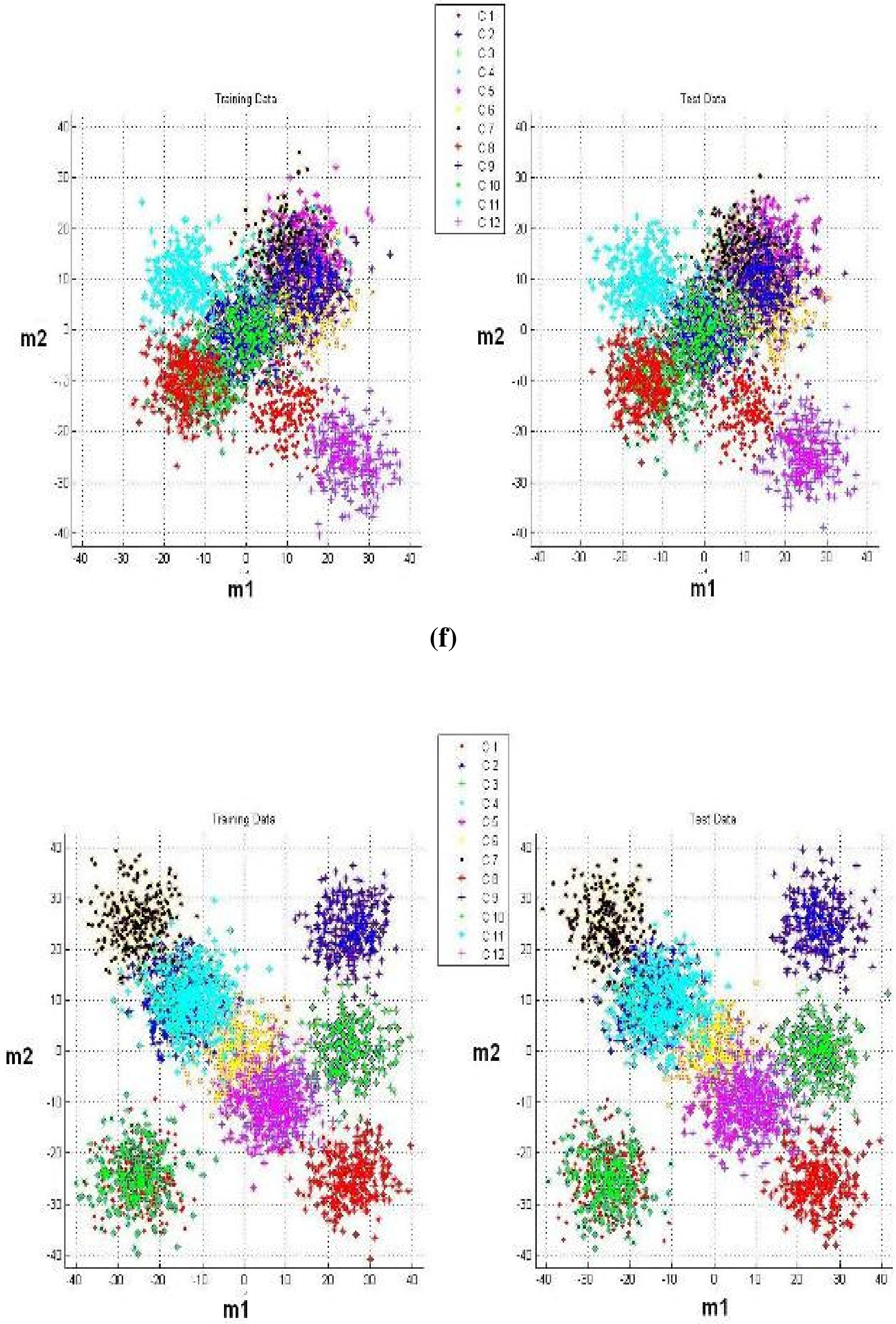
Figure 5.1 (Continued)



(d)

(e)

Figure 5.1 (Continued)



(f)

(g)

Figure 5.1 (Continued)

In this epoch, it is visualized that different classes are distributed with different topologies in each classifier by different overlapping and discrimination attributes. For examples, the ninth class is located with a higher distance relative to the other classes in Descriptor 7, while it is overlapped with the other classes in other descriptors. The classification performance of the ninth class for each distribution can be seen from Table 5.1. In this way, the classification behaviors of the classes are controlled through the topological distributions and the performance criteria in equation (4.19) is verified.

Table 5.1: Comparison of the performances (perf %) of individual Classifiers (Υ_1, Υ_2, Υ_3, Υ_4, Υ_5, Υ_6 and Υ_7) with respect to the classes (C) and the performance of SG

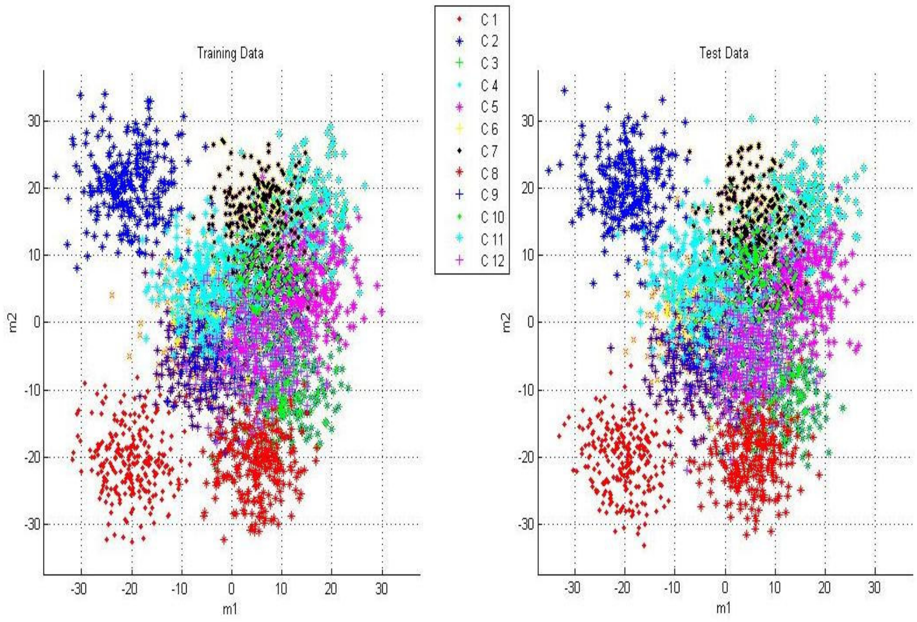
	Υ_1	Υ_2	Υ_3	Υ_4	Υ_5	Υ_6	Υ_7	**SG**
C1	66	63.6	67.6	62.8	61.6	85.6	50.0	**100**
C2	67.2	60.8	49.6	50.8	***98.4***	38.4	36.8	**100**
C3	54.4	58.8	50.8	***85.2***	72.4	53.6	47.6	**99.2**
C4	66.8	64.0	***96.8***	66.4	61.6	22.8	37.6	**100**
C5	60.8	***90.0***	56.0	63.6	75.2	38.8	48.4	**100**
C6	***91.6***	57.2	69.6	54.0	66.0	43.6	73.6	**100**
C7	57.2	55.2	65.2	57.6	60.8	37.2	***94.4***	**100**
C8	78.4	75.6	86.0	69.2	54.4	61.6	***97.6***	**100**
C9	40.8	41.2	36.0	36.0	32.8	26.0	***99.6***	**100**
C10	44.0	32.4	32.0	38.0	37.6	43.2	***95.6***	**100**
C11	32.0	35.2	33.6	40.0	39.6	***92.8***	38.8	**99.6**
C12	37.6	39.6	34.4	52.0	44.4	***97.2***	63.6	**99.6**
Total Performance	58.0	56.1	56.5	56.3	58.7	53.4	65.3	**99.9**

In Table 5.1, performances of individual classifiers and the performance of SG are given for an instance of dataset generated by the above algorithm. For that particular instance, note that, the performance of the individual classifiers are in between 26-75% for 12 classes and the overall performance of SG is 99%. Highest performance for each classifier indicates the class with the largest between class variances, which are indicated by underline.

As mentioned above, the data sets are constructed in such a way that each sample is correctly recognized by at least one classifier. Although the performances of individual classifiers are around 55%, the classification performance of SG is around 100%, verifying Hypothesis 1.

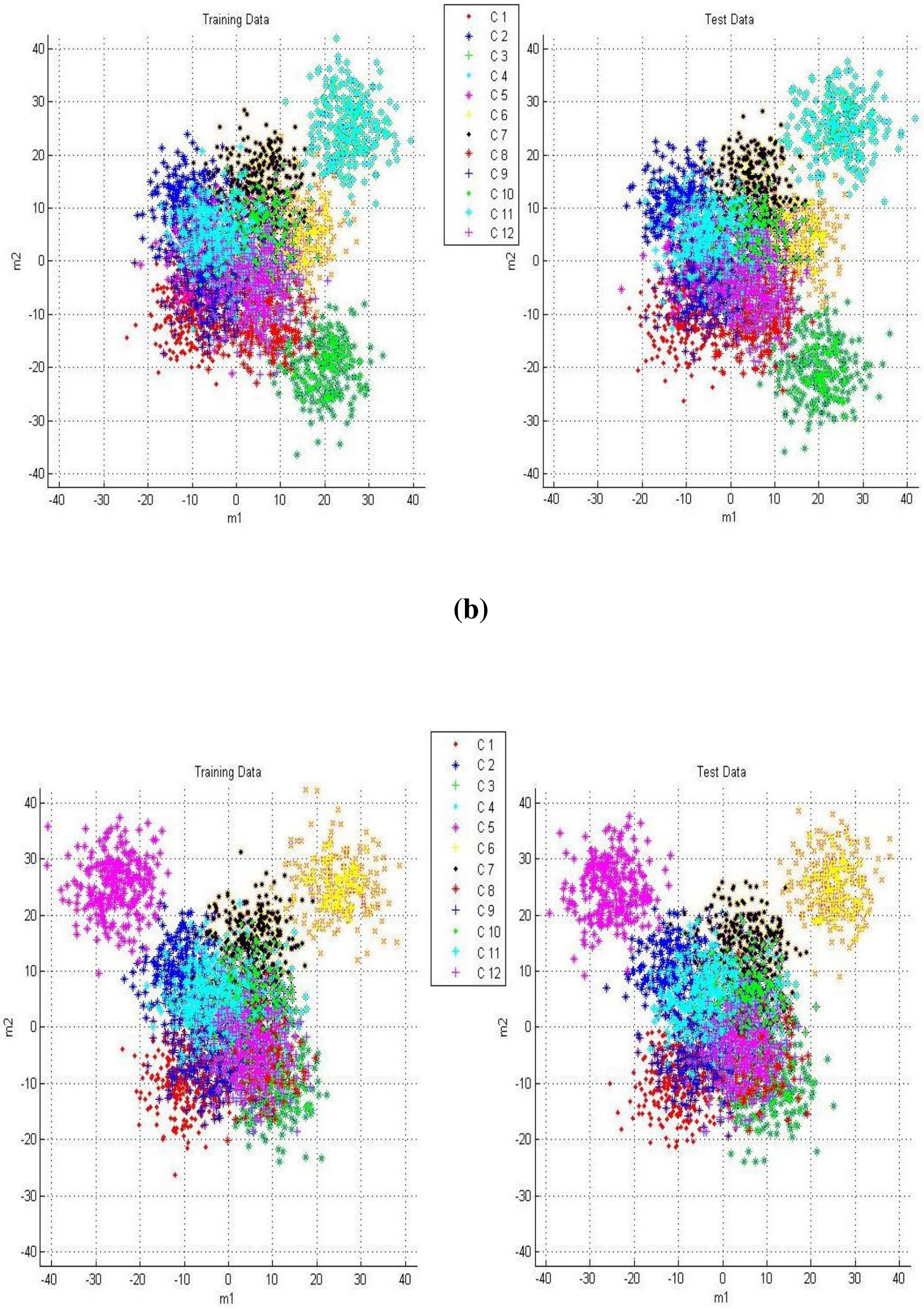
$$M = \begin{bmatrix} -20 & -20 & -10 & -10 & -10 & -10 & -10 & -10 & 10 & -10 & 10 & -15 & 15 & 5 \\ -20 & 20 & -10 & 10 & -10 & 10 & -10 & 10 & -5 & -10 & 0 & 0 & -5 & 10 \\ 10 & -10 & 20 & -20 & 10 & -10 & 10 & -10 & 15 & -15 & -10 & -10 & -10 & -5 \\ 15 & 15 & 25 & 25 & 5 & 5 & -5 & 10 & 15 & 15 & 10 & 10 & -15 & 10 \\ 15 & 5 & -5 & 0 & -25 & 25 & -10 & 5 & -5 & 5 & 15 & 15 & 5 & -10 \\ -5 & 0 & 15 & 5 & 25 & 25 & 15 & 5 & 15 & 5 & 15 & 5 & 0 & 0 \\ 5 & 15 & 5 & 15 & 5 & 15 & 25 & 25 & 5 & 10 & 10 & 15 & -5 & 5 \\ 5 & -20 & 5 & -10 & 5 & -5 & 25 & -25 & 5 & -15 & -15 & -10 & 5 & -5 \\ -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -25 & -25 & 15 & 10 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 25 & 25 & 0 & 0 & 5 & 0 \\ -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -25 & 25 & -10 & 10 \\ 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 15 & -15 & 25 & -25 \end{bmatrix}$$

In Figure 5.2, the distribution of the classes with the given matrix *M* is introduced.



(a)

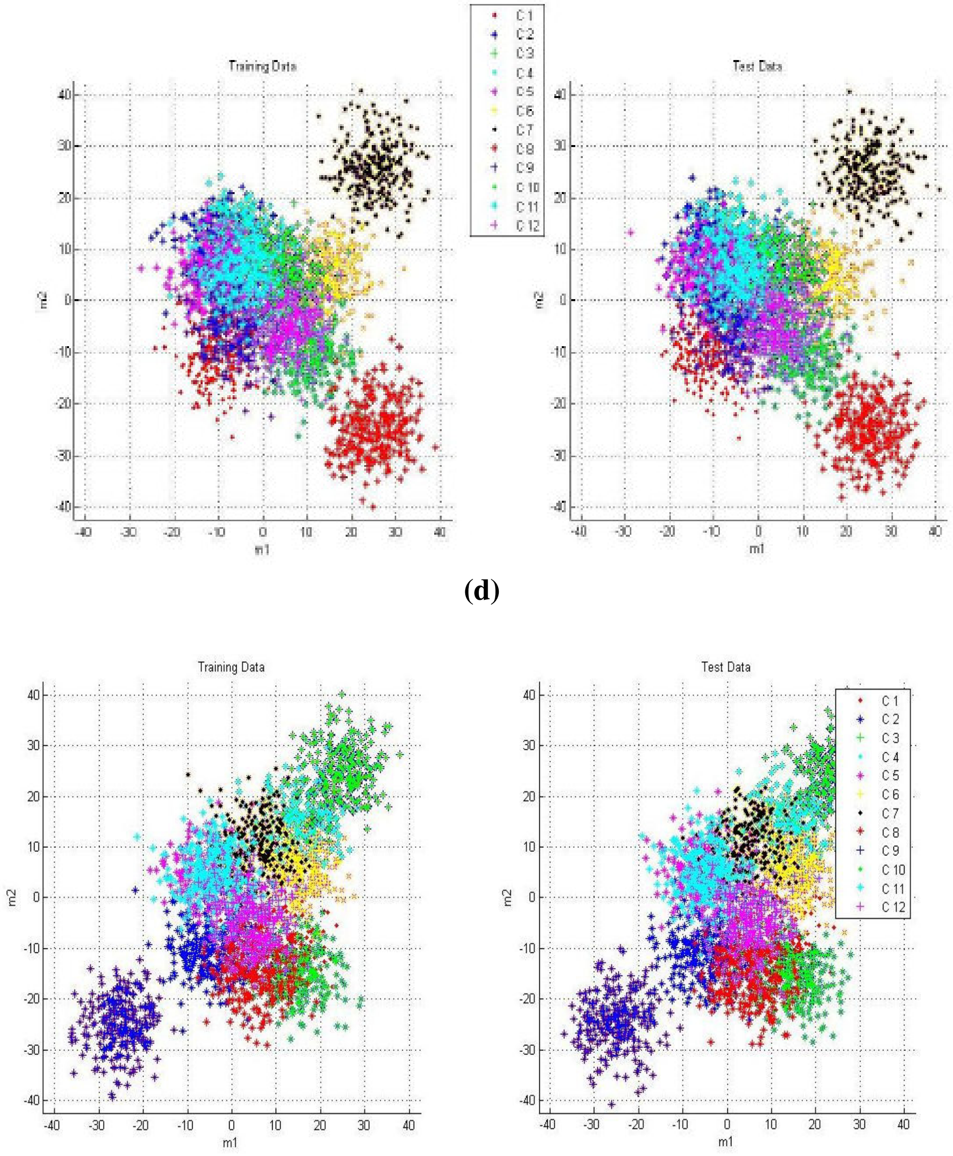
Figure 5.2: Distribution of the classes at, **(a)** Descriptor 1, **(b)** Descriptor 2, **(c)** Descriptor 3, **(d)** Descriptor 4, **(e)** Descriptor 5, **(f)** Descriptor 6, **(g)** Descriptor 7 In that epoch, the classes are distributed in a topology different from Figure 5.1.



(b)

(c)

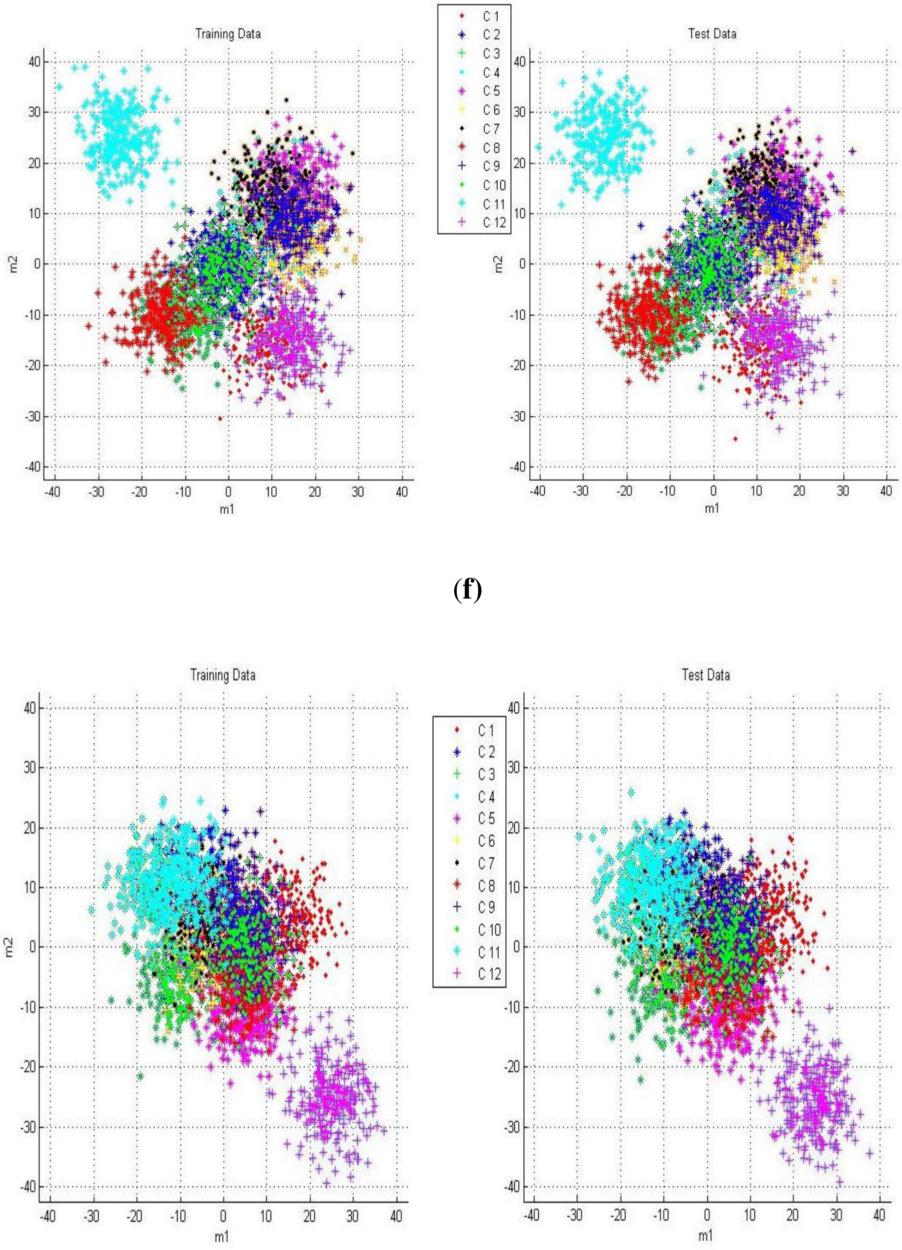
Figure 5.2 (Continued)



(d)

(e)

Figure 5.2 (Continued)



(f)

(g)

Figure 5.2 (Continued)

In that case, different classes are distributed at higher relative distances and with different overlapping ratios.

In Table 5.2, the performance results of another epoch of the experiments are introduced. The corresponding mean value matrix of each class at each descriptor is the following;

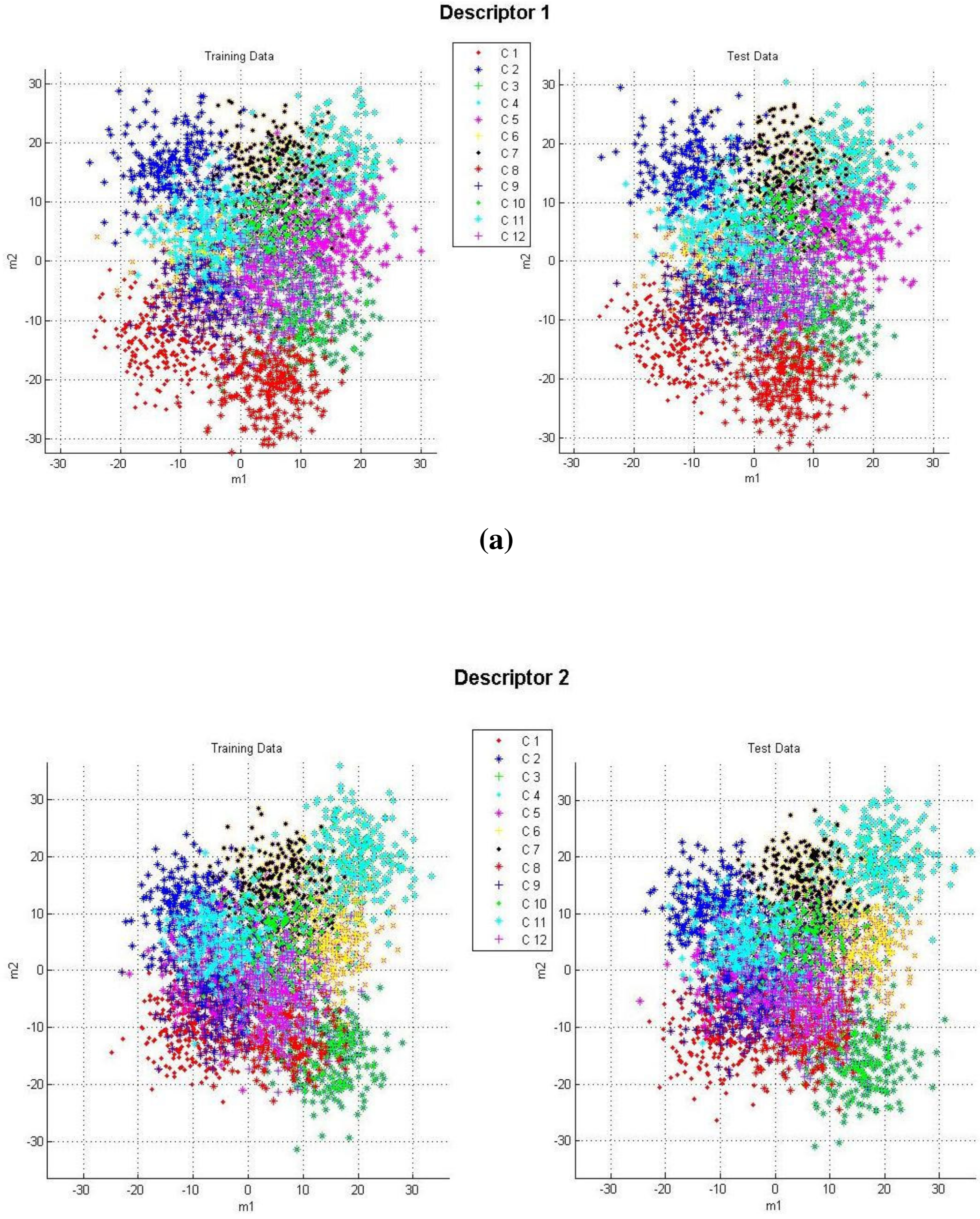
Table 5.2: Comparison of the performances (perf %) of individual Classifiers (Υ_1, Υ_2, Υ_3, Υ_4, Υ_5, Υ_6 and Υ_7) ith respect to the classes (C) and the performance of SG

	Υ_1	Υ_2	Υ_3	Υ_4	Υ_5	Υ_6	Υ_7	**SG**
C1	***97.2***	67.6	68.4	69.6	28.0	53.6	65.6	***100***
C2	***96.8***	63.2	63.6	41.6	67.6	44.4	30.0	***100***
C3	56.4	***95.2***	57.2	66.8	56.8	47.2	66.4	***99.6***
C4	60.8	***98.0***	22.8	30.8	62.0	24.4	46.0	***100***
C5	56.8	24.0	***96.8***	27.2	44.8	38.8	50.4	***100***
C6	32.8	68.4	***97.6***	71.2	57.2	43.6	14.0	***100***
C7	54.0	65.6	74.4	***96.8***	52.4	36.8	24.4	***99.6***
C8	77.2	43.6	29.6	***98.4***	48.0	65.6	27.6	***99.6***
C9	45.2	34.0	35.2	35.2	***98.8***	24.8	29.2	***100***
C10	40.0	33.6	22.4	47.6	***90.4***	33.6	18.0	***100***
C11	49.2	28.4	38.0	28.0	38.4	***100.0***	26.0	***100***
C12	34.8	34.4	22.4	34.4	44.4	65.2	***98.8***	***100***
Total Performance	58.433	54.667	52.367	53.967	57.4	48.167	41.367	99.9

In the third set of the experiments, samples are distributed in the descriptors such that each classifier can correctly classify at least one class with approximately 80% performance limit. The corresponding mean value matrix is,

$$M = \begin{bmatrix} -12.5 & -12.5 & -10 & -10 & -10 & -10 & -10 & -10 & 10 & -10 & 10 & -15 & 15 & 5 \\ -10 & 15 & -10 & 10 & -10 & 10 & -10 & 10 & -5 & -10 & 0 & 0 & -5 & 10 \\ 10 & -10 & 15 & -15 & 10 & -10 & 10 & -10 & 15 & -15 & -10 & -10 & -10 & -5 \\ 15 & 15 & 19 & 19 & 5 & 5 & -5 & 10 & 15 & 15 & 10 & 10 & -15 & 10 \\ 15 & 5 & -5 & 0 & -17.5 & 17.5 & -10 & 5 & -5 & 5 & 15 & 15 & 5 & -10 \\ -5 & 0 & 15 & 5 & 17.5 & 17.5 & 15 & 5 & 15 & 5 & 15 & 5 & 0 & 0 \\ 5 & 15 & 5 & 15 & 5 & 15 & 17.5 & 17.5 & 5 & 10 & 10 & 15 & -5 & 5 \\ 5 & -20 & 5 & -10 & 5 & -5 & 17.5 & -17.5 & 5 & -15 & -15 & -10 & 5 & -5 \\ -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -15 & -15 & 15 & 10 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 22.5 & 22.5 & 0 & 0 & 5 & 0 \\ -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -10 & 10 & -10 & 10 \\ 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 15 & -15 & 15 & -15 \end{bmatrix}$$

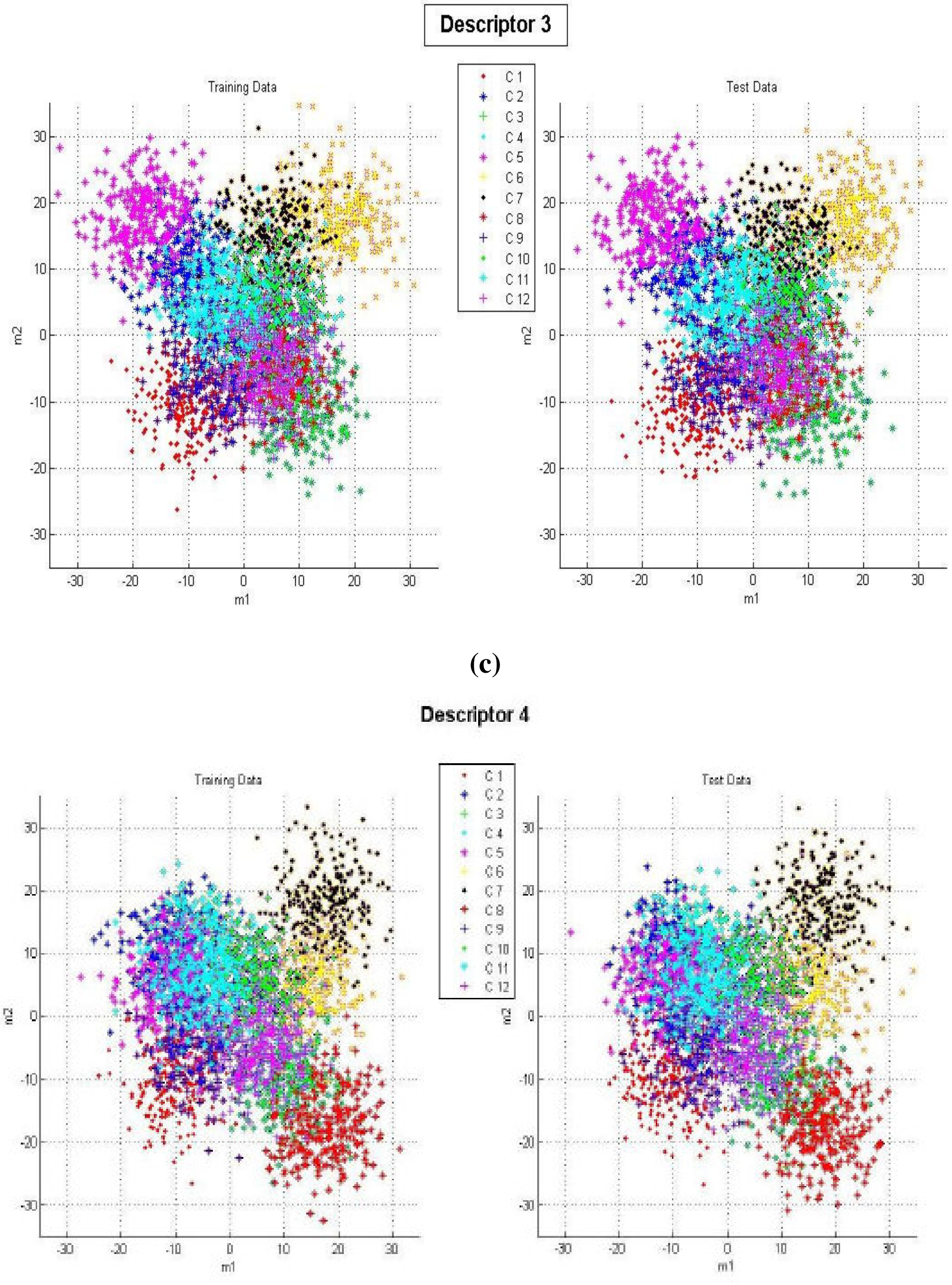
In Figure 5.3, the distribution of the classes with the given matrix *M* is introduced.



(a)

(b)

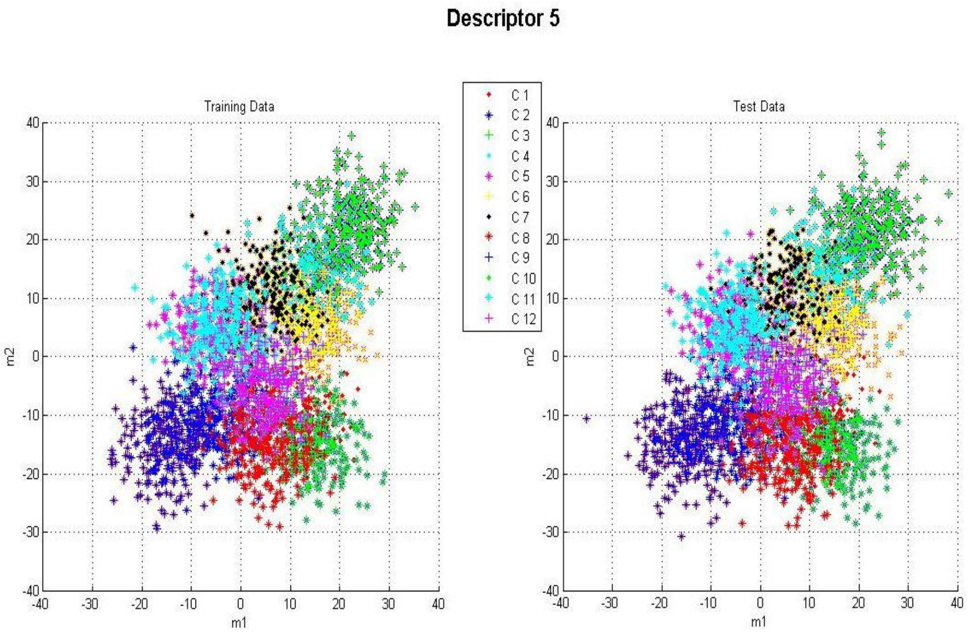
Figure 5. 3: Distribution of the classes at, **(a)** Descriptor 1, **(b)** Descriptor 2, **(c)** Descriptor 3, **(d)** Descriptor 4, **(e)** Descriptor 5, **(f)** Descriptor 6, **(g)** Descriptor 7



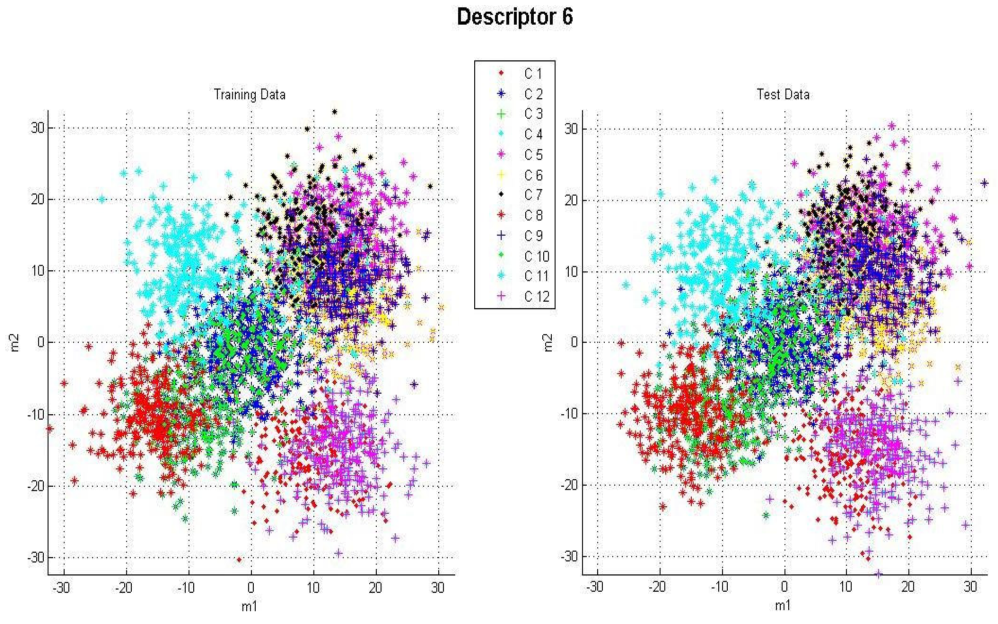
(c)

(d)

Figure 5.3 (Continued)

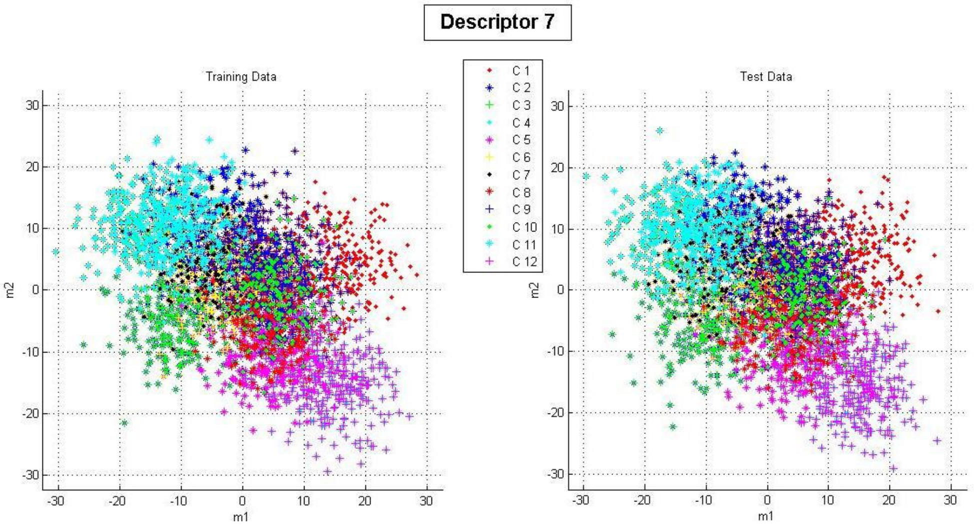


(e)



(f)

Figure 5.3 (Continued)



(g)

Figure 5.3 (Continued)

In Table 5.3, the performance results of the experiment are introduced. The corresponding mean value matrix of each class at each descriptor is the following;

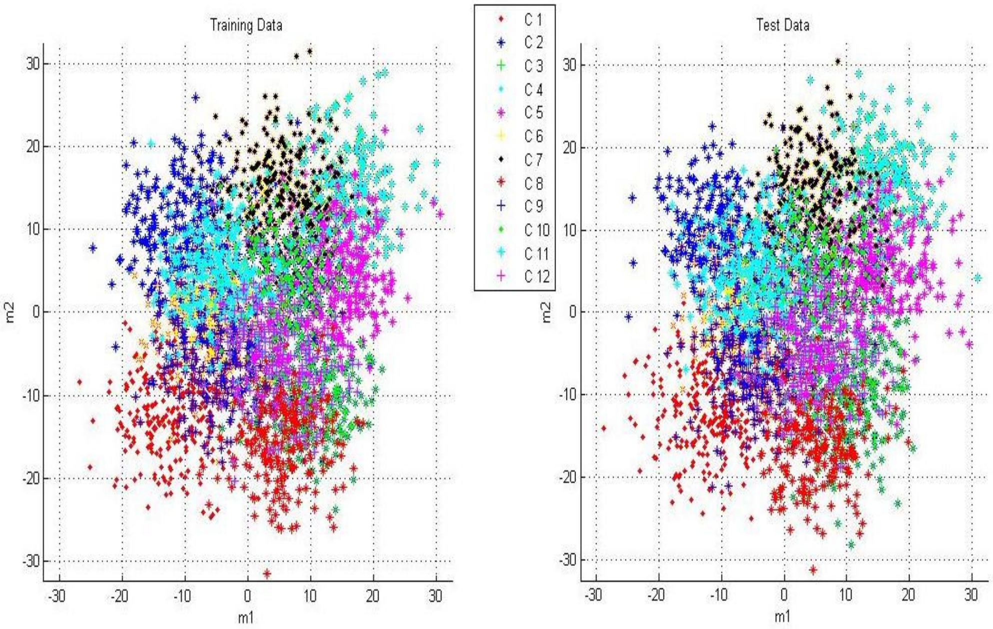
Table 5.3 Comparison of the performances (perf %) of individual Classifiers (Υ_1, Υ_2, Υ_3, Υ_4, Υ_5, Υ_6 and Υ_7) ith respect to the classes (C) and the performance of SG

	Υ_1	Υ_2	Υ_3	Υ_4	Υ_5	Υ_6	Υ_7	**SG**
C1	***82.8***	63.6	66.0	71.2	32.0	54.0	67.2	***99.6***
C2	***73.2***	63.6	48.0	34.4	51.6	37.6	29.6	***97.2***
C3	55.2	***78.0***	59.6	51.2	62.4	46.8	69.6	***98.4***
C4	61.2	***82.0***	26.0	31.2	44.4	17.6	52.8	***98.4***
C5	53.2	23.2	***76.8***	29.6	41.2	39.6	45.2	***100***
C6	24.8	66.4	***87.2***	62.0	56.4	42.4	21.2	***98.8***
C7	54.0	63.2	54.8	***88.4***	55.2	36.8	23.6	***98.4***
C8	***80.8***	39.2	22.8	74.8	45.2	63.2	23.6	***96.4***
C9	39.6	33.2	33.2	29.6	***83.6***	21.6	29.6	***99.2***
C10	38.4	35.6	30.8	47.6	***82.8***	38.0	24.0	***99.2***
C11	33.2	30.0	30.8	30.4	38.8	***84.4***	29.6	***96.4***
C12	40.4	33.2	28.0	40.4	32.4	58.8	***81.2***	***99.2***
Total Performance	53.1	50.9	47	49.2	52.2	45.1	41.4	98.4

In the fourth set of the experiments, samples are distributed in the descriptors such that each classifier can correctly classify at least one class with approximately 70% performance limit. The corresponding mean value matrix is,

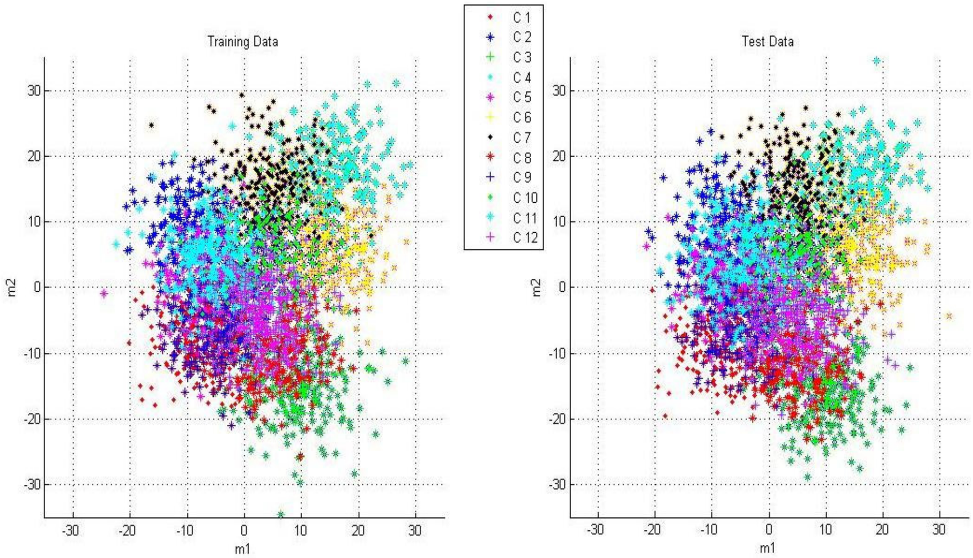
$$M = \begin{bmatrix} -12 & -12 & -7.5 & -7.5 & -10 & -10 & -7.5 & -7.5 & 10 & -10 & 10 & -15 & 10 & 5 \\ -10 & 10 & -8 & 8 & -10 & 10 & -10 & 10 & -5 & -10 & 0 & 0 & -5 & 10 \\ 10 & -10 & 10 & -15 & 10 & -10 & 10 & -10 & 10 & -15 & -10 & -10 & -5 & -5 \\ 15 & 15 & 15 & 17.5 & 5 & 5 & -5 & 10 & 15 & 15 & 10 & 10 & -15 & 10 \\ 15 & 5 & -5 & 0 & -15 & 15 & -10 & 5 & -5 & 5 & 15 & 15 & 5 & -10 \\ -5 & 0 & 15 & 5 & 15 & 15 & 10 & 5 & 10 & 5 & 10 & 5 & 0 & 0 \\ 5 & 15 & 5 & 15 & 5 & 15 & 10 & 15 & 5 & 10 & 10 & 15 & -5 & 5 \\ 5 & -15 & 5 & -10 & 5 & -5 & 15 & -15 & 5 & -15 & -15 & -10 & 5 & -5 \\ -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -10 & -15 & 15 & 10 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 20 & 20 & 0 & 0 & 5 & 0 \\ -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 10 & -10 & 10 \\ 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 5 & -5 & 15 & -15 & 10 & -15 \end{bmatrix}$$

In Figure 5.4, the distribution of the classes with the given matrix M is introduced.

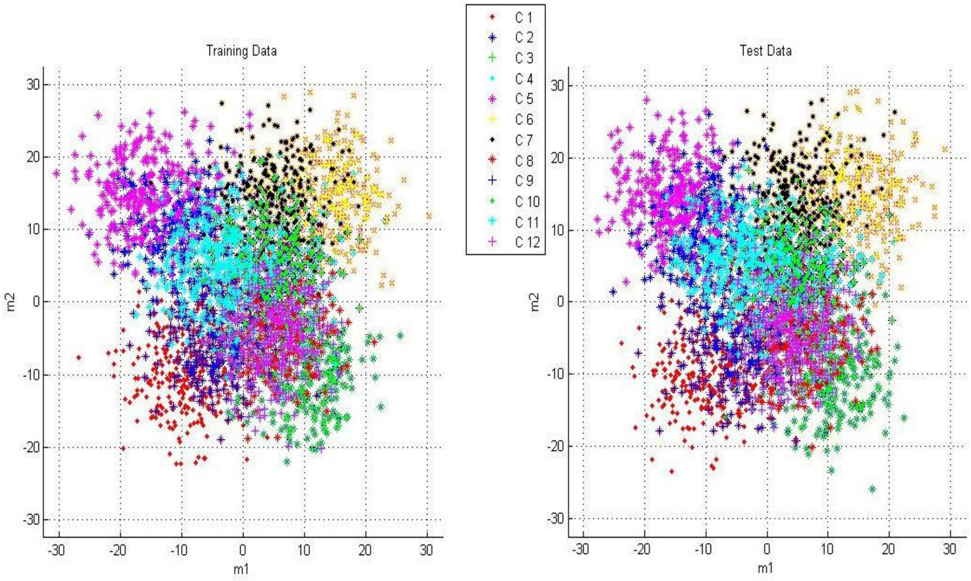


(a)

Figure 5. 4: Distribution of the classes at, **(a)** Descriptor 1, **(b)** Descriptor 2, **(c)** Descriptor 3, **(d)** Descriptor 4, **(e)** Descriptor 5, **(f)** Descriptor 6, **(g)** Descriptor 7

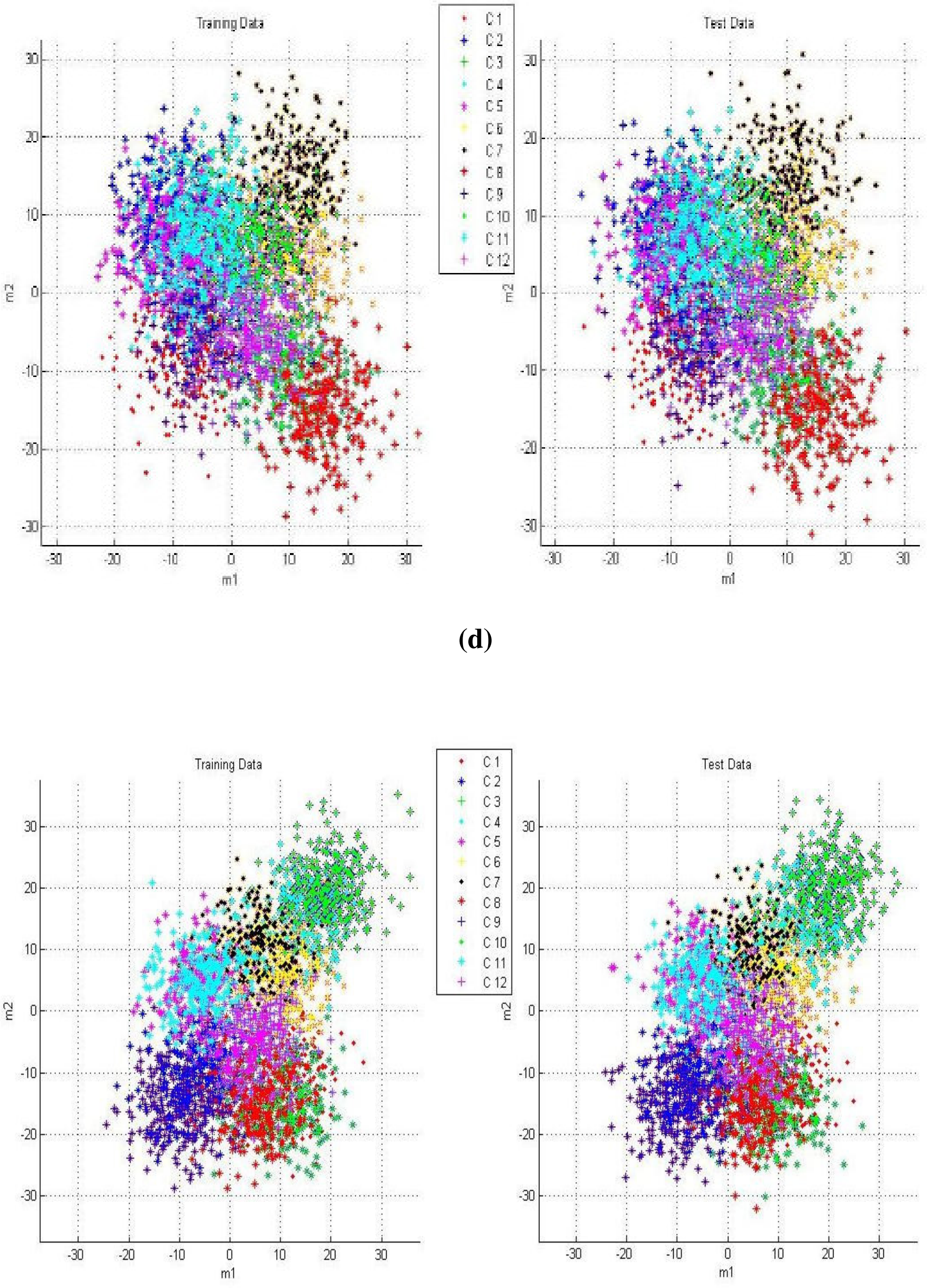


(b)



(c)

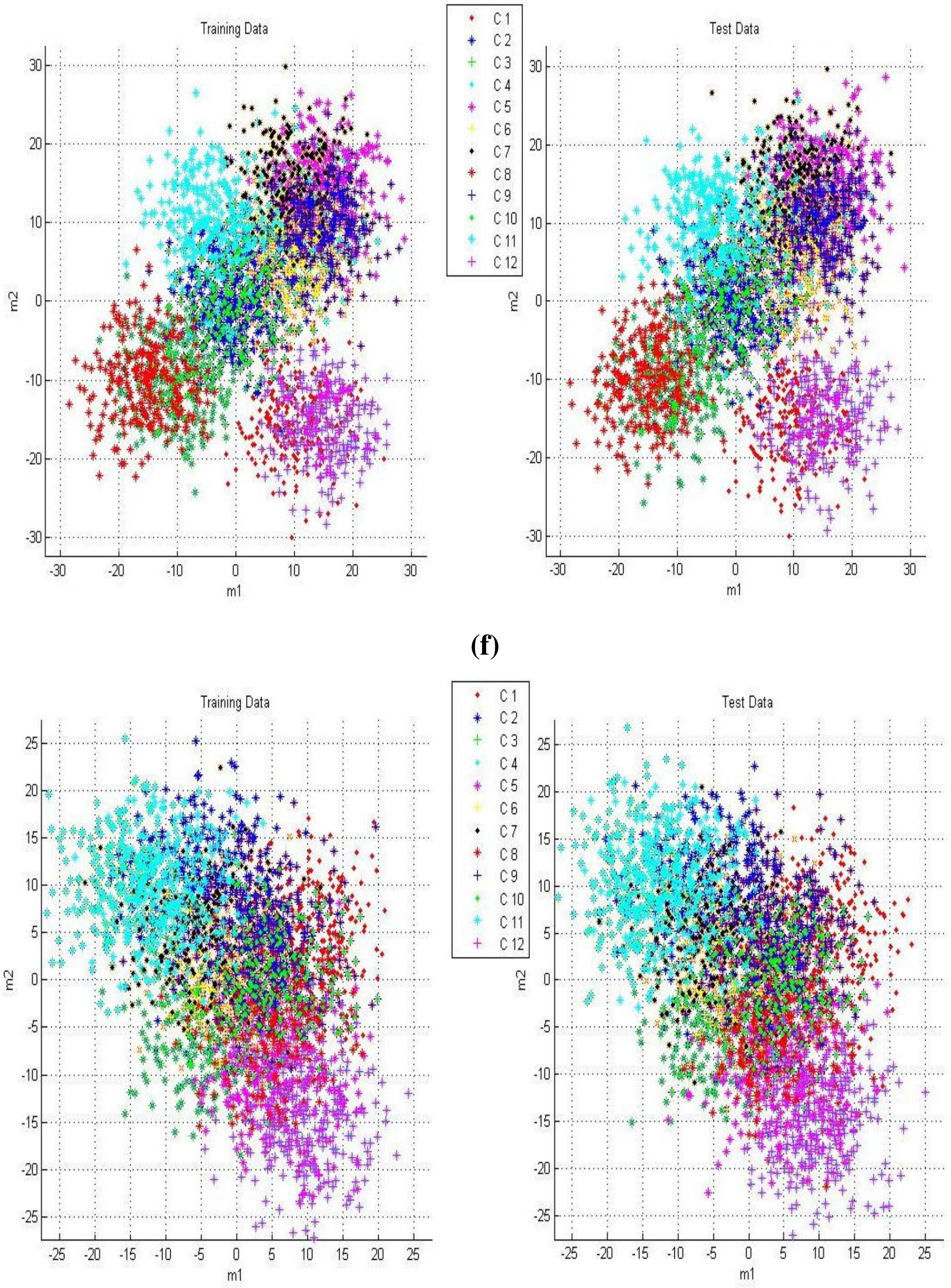
Figure 5.4 (Continued)



(d)

(e)

Figure 5.4 (Continued)



(f)

(g)

Figure 5.4 (Continued)

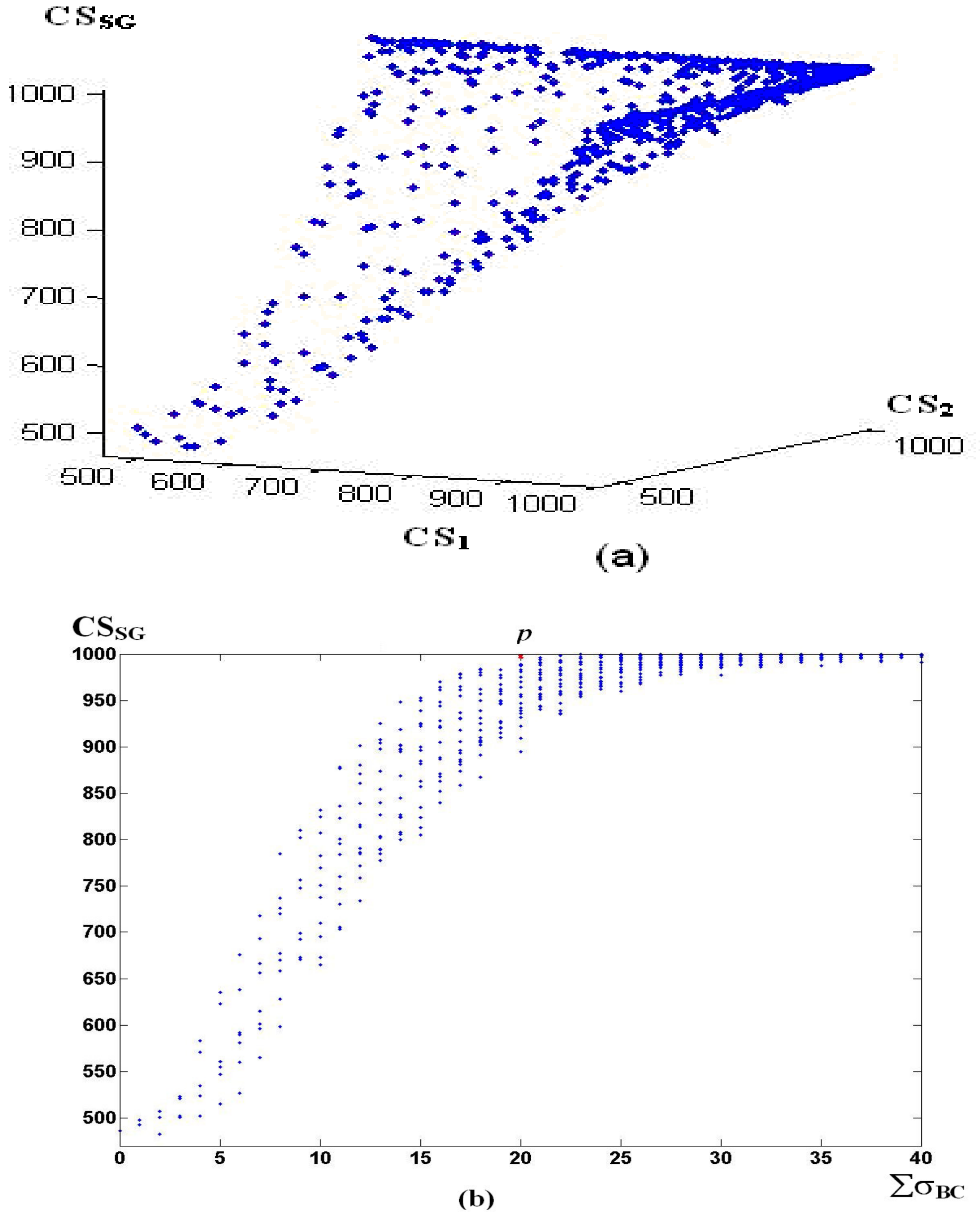
In Table 5.4, the performance results of the experiment are introduced. The corresponding mean value matrix of each class at each descriptor is the following;

Table 5.4: Comparison of the performances (perf %) of individual Classifiers (Υ_1, Υ_2, Υ_3, Υ_4, Υ_5, Υ_6 and Υ_7) ith respect to the classes (C) and the performance of SG

	Υ_1	Υ_2	Υ_3	Υ_4	Υ_5	Υ_6	Υ_7	SG
C1	***75***	*42*	*68*	*52*	*36*	*62*	*46*	***99***
C2	***64***	*45*	*41*	*38*	*43*	*37*	*32*	***98***
C3	*46*	***72***	*60*	*40*	*39*	*52*	*46*	***88***
C4	***68***	***72***	*23*	*33*	*45*	*17*	*59*	***98***
C5	*54*	*22*	***70***	*28*	*40*	*42*	*32*	***100***
C6	*22*	*68*	***74***	*50*	*46*	*28*	*18*	***97***
C7	*65*	*62*	*50*	***72***	*44*	*34*	*20*	***96***
C8	*55*	*30*	*25*	***75***	*44*	*61*	*18*	***89***
C9	*36*	*24*	*36*	*30*	***67***	*32*	*23*	***100***
C10	*42*	*32*	*24*	*27*	***74***	*32*	*21*	***98***
C11	*31*	*17*	*34*	*16*	*38*	***70***	*26*	***95***
C12	*33*	*28*	*27*	*41*	*38*	***67***	***68***	***100***
Total Performance	49.3	42.9	44.3	41.8	46.1	44.4	34.2	96.4

5.3.2 Comparison of the Between Class Variance with the Performance of Stacked Generalization

In this set of experiments, the relationship among the sum of the between class variances of the data sets in each classifier and the performance of SG is explored. In the first experiment, the data set is constructed by forming 2-classes, each having 500 samples, in 2-dimensional feature space. In Figure 5.5 -a, the axes represent the number of samples correctly classified by classifier 1, CS_1 and the number of samples correctly classified by classifier 2, CS_2 , respectively, and the z axis represents the number of samples correctly classified by SG, CS_{SG}. In Figure 5.5-b, the relationship between the total σ_{BC} in the base classifiers, $\sum\sigma_{BC}$ and the number of samples correctly classified by SG, CS_{SG} is shown.



(a)

(b)

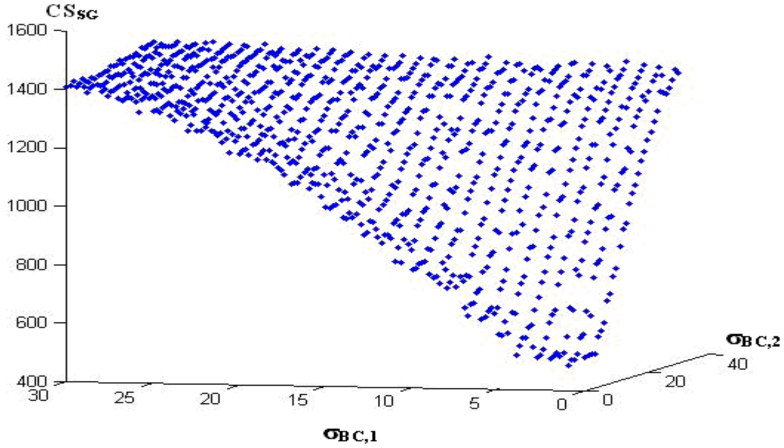
Figure 5.5: **(a)** The relationship between the number of samples correctly classified by base classifiers and the number of samples correctly classified by SG, **(b)** the relationship between the total σ_{BC} in base classifiers and the number of samples correctly classified by SG

In Figure 5.5-a, it is observed that some points are concentrated at two intersecting lines on top of z-axis. This is because of the fact that once a classifier reaches a relatively higher performance compared to the other classifier, this performance dominates the other classifier resulting an accumulation at the top edges.

In Figure 5.5-b, the performance of SG reaches an asymptote at the point *p* where the classes are well separated. Notice that, up to the point *p*, different combinations of σ_{BC} 's that sums up to the same value $\sum\sigma_{BC}$ results in different SG performance, and that results in an interval of performances for a fixed $\sum\sigma_{BC.}$

In Figure 5.6-a, similar experiments of 3-class problem consisting of 1500 samples are displayed. In Figure 5.6-a, the relationship between the total σ_{BC} for the first classifier, $\sigma_{BC,1}$ and the total σ_{BC} for the second classifier, $\sigma_{BC,2}$ and the number of samples correctly classified by SG, CS_{SG} is shown. In Figure 5.6-b, the relationship between the total σ_{BC} for the base classifiers, $\sum\sigma_{BC}$ and the number of samples correctly classified by SG, CS_{SG} is shown.

In the experiments, a sigmoidal relationship between the performances of the base layer classifiers and the performance of SG is observed.



(a)

Figure 5.6: (a) The relationship between the total σ_{BC} in the first and second

classifier, and the number of samples correctly classified by SG, **(b)** total σ_{BC} of 3 classes and the performance of SG

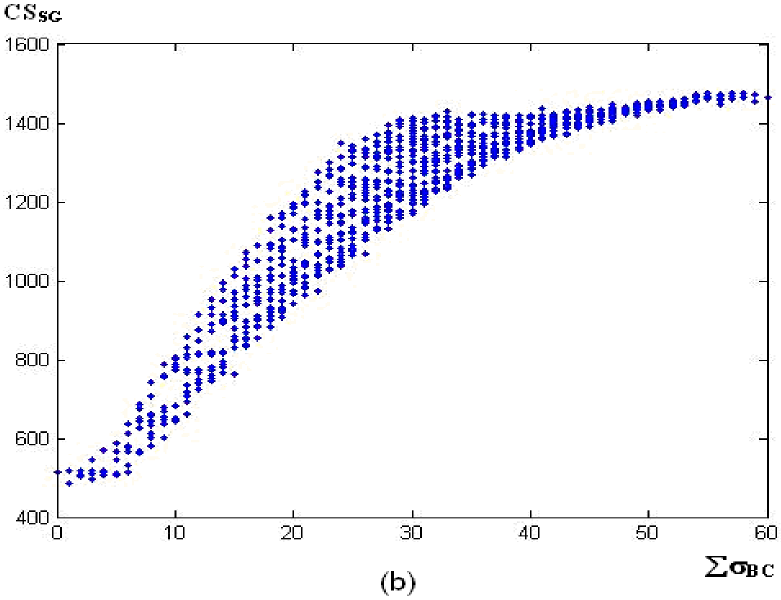


Figure 5.6 (Continued)

In the figures, it is observed that the same value of total σ_{BC} in the base classifiers have several different corresponding correctly classified values of SG, in other words, different topological distribution of the classes corresponding to the same $\sum \sigma_{BC}$ results in different classification performances.

The sigmoid function observed from the experiments is,

$$Performans(SG) = \frac{\mathrm{A}}{(1+\mathrm{B}.\exp(-\sum_{c} \Sigma_{c}.\mathrm{C}))} \qquad \text{(Equation 5.2)}$$

depends on parameters A, B and C. which is to be estimated from the training data.

This function may lead to an open ended discussion about relation between the between-class variance of the base layer input features and the performance of the SG. If there is such a relationship, the mappings between the feature spaces of the hierarchical architecture may be estimated through equation (5.2).

5.4 Validation of Hypothesis-1 on Corel Data

In this section, the validation of the *Hypothesis -1* is examined with the experiments on the Corel Dataset classes using 7 MPEG-7 visual descriptors Color Structure (32 dimensional), Color Layout (12 dimensional), Edge Histogram (80 dimensional), Region-based Shape (35 dimensional), Dominant Color (16 dimensional), Scalable Color (64 dimensional), Homogenous Texture (62 dimensional) [46] and Haar Coefficients (195 dimensional) [76]. In the experiments, 4 to 8 descriptor combinations; 4 (Color Structure , Color Layout, Edge Histogram, Region-based Shape), 5 (Color Structure , Color Layout, Edge Histogram, Region-based Shape, Haar) , 6 (Color Structure , Color Layout, Edge Histogram, Region-based Shape, Haar, Dominant Color) , 7 (Color Structure , Color Layout, Edge Histogram, Region-based Shape, Haar, Dominant Color, Scalable Color), and 8 (Color Structure , Color Layout, Edge Histogram, Region-based Shape, Haar, Dominant Color, Scalable Color, Homogenous Texture) descriptors are used.

MPEG-7 standard is developed by Moving Picture Experts Group in order to describe the audio, video and visual multimedia contents by the acquisition of the maximum information coded in the media for a broad range of applications [77] ,[78],[79].

MPEG-7 descriptors are chosen to be the feature extractors on Corel Dataset since they generate descriptions with high variance and a well-balanced cluster structure [46]. Since the mean and the standard deviation of the features describe the location and distribution of the samples, the high variability property allows us to construct highly distinguished samples for the Corel samples. In addition, the descriptors are independent of each other by providing high between class variance values. Therefore, the structures of the feature spaces are consistent with the synthetic datasets, and provide wealthy information variability.

The 10 classes used for the experiments are New Guinea, Beach, Rome, Bus, Dinosaurs, Elephant, Roses, Horses, Mountain, and Dining, each contain 100 samples from the dataset, and 50 of the samples of each class are used for the training and the remaining 50 samples are used for testing. In the homogenous SG structure, all of the classifiers are fuzzy k-NN, with optimized k-values for each iteration. In the experiments, fuzzy knn is implemented both in Matlab and C++, where C++ implementations classified 2% more samples than Matlab implementations. For C++ implementations, the fuzzified modification of Approximate Nearest Neighbor library is used.

In the first group of the experiments, the samples in the training and test set which cannot be correctly classified by at least one classifier are labeled as misclassified ($MC_{training}$) and $MC_{testing}$,respectively. These samples are eliminated from the data sets, therefore new dataset consists of the samples which are correctly labeled by at least one classifier at the base layer. Defining *K* as the number of classifiers fed by *K* descriptors, $mem_{CV,k}$ as the membership vector obtained from the cross validation on the training set in the kth classifier, $mem_{test,k}$ as the membership vector obtained from test set in the kth classifier, $MC_{training,k}$ as MC set obtained over the training data set in the kth classifier, $MC_{testing,k}$ as MC set obtained over the test data set in the kth classifier, the algorithm is given below:

Algorithm 5. 2 Misclassified training and testing data elimination algorithm

```
for each k=1,2,…..,K
    Calculate $\mathrm{MC}_{\mathrm{training},k}$
    Calculate $\mathrm{MC}_{\mathrm{testing,k}}$
    Calculate $\mathrm{mem}_{\mathrm{test},k}$
    Calculate $\mathrm{mem}_{\mathrm{CV},k}$
end for(k)
Calculate $\mathrm{MC}_{\mathrm{training}} = \bigcup_{k=1}^{K} (\mathrm{MC}_{\mathrm{training,k}})$
Concatenate $\mathrm{mem}_{\mathbf{CV},k}$ ,for the meta-layer input training dataset
Eliminate the samples of x from $\mathrm{mem}_{\mathrm{CV}}$ where $\underline{\mathrm{x}} \in \mathrm{MC}_{\mathrm{training}}$
Calculate $\mathrm{MC}_{\mathrm{testing}} = \bigcup_{k=1}^{K} (\mathrm{MC}_{\mathrm{testing,k}})$
Concatenate $\mathrm{mem}_{\mathrm{test},k}$for the meta-layer input test dataset
Eliminate the samples of z from $\mathrm{mem}_{\mathrm{test}}$ ,where $\underline{\mathrm{z}} \in \mathrm{MC}_{\mathrm{testing}}$
Perform the meta–layer classification
```

The performances of the classifications using 10 classes are introduced on the Table 5.5.

Table 5.5 Performances of 10 Class Experiments for test data

10 Class Experiments	Without MC Sample Elimination	With MC Sample Elimination	Performance Gain
4 Descriptors	85.6%	86.9%	1.3%
5 Descriptors	86.6 %	88.0%	1.4%
6 Descriptors	85.6%	87.2%	1.6%
7 Descriptors	85.4%	86.0%	0.6%
8 Descriptors	85.8%	87.0%	1.2%

In Table 5.6, the number of samples eliminated from each class, that is the number of MC samples for each class, in both training (tr) and testing (te) phases are given.

Table 5.6: Number of MC samples from each class, in 10 class classification with 4 Descriptor, 5 Descriptor, 6 Descriptor, 7 Descriptor and 8 Descriptor experiments, with each class of 50 samples, in both training (tr) and testing (te) phases

Classes	4 D		5D		6D		7D		8D	
	te	tr	te	tr	te	tr	te	tr	te	tr
New Guinea	3	1	2	1	2	0	1	0	1	0
Beach	4	9	3	5	3	5	2	5	2	5
Rome	2	1	2	1	2	1	1	0	1	0
Bus	6	3	6	3	4	1	3	1	2	1
Dinosaurs	5	6	3	4	3	2	1	1	1	1
Elephant	2	2	1	2	1	2	1	2	1	2
Roses	6	2	4	2	3	2	2	2	2	2
Horses	2	0	2	0	1	0	1	0	1	0
Mountain	2	4	2	4	2	4	2	4	2	3
Dining	3	5	2	5	2	4	2	4	2	4
Total Number of MC samples	35	33	27	27	23	21	16	19	15	18

In the experiments, it is observed that, when the meta-layer input data space is constructed by eliminating MC samples from the base layer output space, the classification performance of SG is increased. This observation offers an approach to one of the black art problems of SG which is the construction of meta-layer input space.

As it is expected, as the number of classifiers increase, the number of MC samples in the tests decreased, and the performance gain decreased. However, this is because of the fact that MC samples of test sets are eliminated. In the next section, by only eliminating the samples from the training sets, the performance gain increased as the number of the descriptors increased.

5.5 Validation of Hypothesis-2 on Corel Data

In this section, the validation of the *Hypothesis -2* is examined with the experiments on the Corel Dataset classes. In the experiments, 4 to 8 descriptor combinations of the MPEG 7 descriptors are used over 10 to 20 classes, each of which contain 100 samples from the dataset. 50 of the samples of each class are used for the training and the remaining 50 samples are used for testing. In the homogenous SG structure, all of the classifiers are fuzzy k-NN, with optimized k-values for each iteration.

In the experiments, the samples that are not correctly labeled by any of the base layer classifiers are determined by cross-validation on the training and test dataset. These samples are considered as spurious samples and placed on misclassified (MC) sample sets obtained in training and testing processes, $MC_{training}$. , and $MC_{training}$ samples are extracted from training data sets.

In this section, the elements, $\underline{x} \in MC_{training}$ are extracted from the meta-layer input data set(in other words, base layer output set) in order to construct the data set in a topology that will include only the correctly classified samples. So, in some manner, a dataset to be the input for the meta-layer that contains the samples that are defined by at least one of the base layer classifiers is constructed.

Defining K as the number of classifiers fed by *K* descriptors, $mem_{CV,k}$ as the membership vector obtained from the cross validation on the training set in the k^{th} classifier, $mem_{test,k}$ as the membership vector obtained from test set in the k^{th}

classifier, and $\mathrm{MC}_{\mathrm{training,k}}$ as MC set obtained over the training data set in the k$^{\mathrm{th}}$ classifier, the sample elimination algorithm is given below:

Algorithm 5. 3: Misclassified training data elimination

```
1. for each k=1,2,.....,K
2.     Calculate $\mathrm{MC}_{\mathrm{training},k}$
3.     Calculate $\mathrm{mem}_{\mathrm{test},k}$
4.     Calculate $\mathrm{mem}_{\mathrm{CV},k}$
5. end for(k)
6. Calculate $\mathrm{MC}_{\mathrm{training}} = \bigcup_{k=1}^{K}(\mathrm{MC}_{\mathrm{training,k}})$
7. Concatenate $\mathrm{mem}_{\mathbf{CV},k}$ ,for the meta-layer input training dataset
8. Eliminate the samples of $\underline{x}$ from $\mathrm{mem}_{\mathrm{CV}}$ where $\underline{\mathrm{x}} \in \mathrm{MC}_{\mathrm{training}}$
9. Perform the meta–layer classification
```

After extracting the samples $\underline{x}$ **,** from the dataset, the features of the remaining samples obtained from different classifiers are concatenated and fed into the meta-layer classifier. In Table 5.7, the experiments implemented on four descriptors; color layout, color structure, edge histogram and region shape with 10 classes of 100 samples; New Guinea, Beach, Rome, Bus, Dinosaurs, Elephant, Roses, Horses, Mountain, and Dining are introduced. The performances are introduced for Cross Validation on training set and for the test set. The total number of the elements of the dataset is 1000, with 500 samples for training, and 500 samples for testing.

Table 5.7: 10-classes, 4 Descriptors Experiment

	Training Performance	**Test Performance**
Color Layout	67.2%	67.8%
Color Structure	80.4%	80.6%
Edge Histogram	59.8%	57.4%
Region Shape	38.6%	35.8%
Number of misclassified Samples	33 (6.6 % of the dataset)	35 (7 % of the dataset)

In the experiment, after removed 33 samples from the dataset, 467 samples are fed into the meta-layer classifier. After the classification process, 86.9% performance is obtained with approximately 1.3% performance gain, since SG obtained 85.6% performance without extracting the samples.

Table 5.8: Performances of 10-class classification with 8 Descriptors Experiment

	Training Performance	**Testing Performance**
Color Layout	67.2%	67.8%
Color Structure	80.4%	80.6%
Edge Histogram	59.8%	57.4%
Region Shape	38.6%	35.8%
Haar	61.0%	62.8%
Dominant Color	53.8%	51.0%
Scalable Color	76.6%	77.2%
Homogenous Texture	46.8%	48.6%
Number of MC Samples	18 (3.6 % of the dataset)	15 (3 % of the dataset)

In this experiment introduced by Table 5.8, 8 descriptors are used for the level-0 classifiers. Since the number of the descriptors is increased, the number of MC samples is decreased. In the experiment, 18 are removed from the training data set. The test dataset is classified with a performance of 87.4% by approximately 1.6% performance gain compared to the performance of the original data set 85.8%.

Figure 5.7 shows a sample image that cannot be classified by the 4D, 5D, 6D, and 7D classification experiments of 10 classes. As can be seen clearly, the sample includes different attributes, such as man, sea, and mountain that are meaningful for different descriptors. This causes the distribution of the points of the feature vector extracted from the picture take place in an entangled state with other feature vectors resulting with de-coherence of the classes.

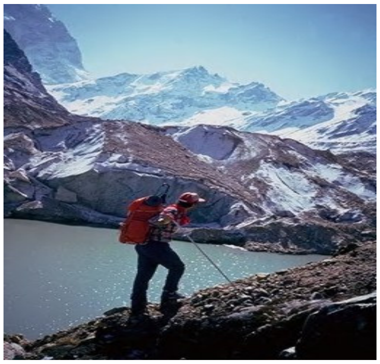


Figure 5.7: A sample image form mountain class that the first 7 descriptors can not define

In the experiment presented by Table 5.9, 5 classes, Autumn, Bhutan, California Sea, Canada Sea, and Canada West classes are added to the data set. Therefore, the total number of the samples became 1500, with 750 for training and 750 for testing. In the experiment, 8 descriptors are used and 96(12% of the dataset) samples are removed from the dataset. After the samples are extracted, 67.7% performance is obtained with 3.2% improvement compared to the case where no sample removal which is 64.5%.

Table 5.9: Number of samples eliminated from the training data in 15 class classification with , 8 descriptors experiment, with each class of 50 samples

Performance	**Training Performance**	**Test Performance**
Color Layout	49.47%	46.67%
Color Structure	61.3%	61.20%
Edge Histogram	42.27%	40.00%
Region Shape	25.20%	22.93%
Haar	46.13%	46.13%
Dominant Color	36.93%	35.07%
Scalable Color	58.00%	61.07%
Homogenous Texture	33.87%	35.60%
Number of MC Samples	96 (12.8 % of the training dataset)	100 (13.3 % of the training dataset)

In Table 5.10, the performances of the 10 class classification experiments by eliminating the MC samples, and without eliminating the MC samples, and the corresponding performance gains are provided.

Table 5.10: Performances of 10-Class Experiments

10-Class Experiments	Without MC Sample Elimination	With MC Sample Elimination	Performance Gain
4 Descriptors	85.6%	86.2%	0.6%
5 Descriptors	86.8 %	87.6%	1.8%
6 Descriptors	85.6%	86.4%	0.8%
7 Descriptors	85.8%	86.2%	0.4%
8 Descriptors	86.4%	87.4.%	1.0%

In the Table 5.11, the number of elements eliminated is given per classes in 5 Descriptors and 8 Descriptors with 15 class classification experiments.

Table 5.11: Number of samples eliminated from each class, in 15-class experiments, with each class of 50 samples

Classes	5 Descriptors Experiment		8 Descriptors Experiment	
	Test	**Training**	**Test**	**Training**
New Guinea	19	17	14	13
Beach	26	15	17	12
Rome	3	1	2	0
Bus	4	8	3	7
Dinosaurs	2	2	1	2
Elephant	9	8	5	3
Roses	15	18	12	16
Horses	22	29	16	19
Mountain	17	13	9	9
Dining	8	6	2	3
Autumn	1	4	1	2
Bhutan	8	4	4	2
California Sea	2	0	1	0
Canada Sea	3	8	2	4
Canada West	12	7	10	4
Total Number of MC Samples	151	139	100	96

In Table 5.12, the performances of the 15 class experiments by eliminating the MC samples, and without eliminating the MC samples, and the corresponding performance gains are provided.

Table 5.12: Performances of 15-Class Experiments

15-Class Experiments	**Without MC Sample Elimination**	**With MC Sample Elimination**	**Performance Gain**
5 Descriptors	65.3%	66.4%	1.1%
6 Descriptors	62.3 %	62.3%	0.0%
7 Descriptors	62.8%	64.0%	1.2%
8 Descriptors	64.5%	67.7%	3.2%

Consequently, since the extra classes reduce the performances of the individual classifiers, the total number of MC samples is increased. As can be seen from the Table 5.11, the added 5 classes intended to disarrange the feature space by damaging the space of the classes New Guinea, Beach, Roses, Horses, and Mountain.

In the next group of experiments, 5 more classes; China, Croatia, Death Valley, dogs and England are added to the present classes. In Table 5.13, the performances of the 20 class experiments by eliminating the MC samples, and without eliminating the MC samples, and the corresponding performance gains are provided.

Table 5.13: Performances of 20-Class Experiments

20-Class Experiments	**Without MC Sample Elimination**	**With MC Sample Elimination**	**Performance Gain**
4 Descriptors	52.4%	54.0%	1.6%
5 Descriptors	50.7%	52.3%	1.6%
6 Descriptors	49.9 %	51.8.%	1.9%
7 Descriptors	50.9%	53.0%	2.1%
8 Descriptors	52.9%	56.2%	3.3%

In the experiments, it is observed that as the number of descriptors increase, the number of MC samples decrease and the data set is less spoiled. In 10 class, 15 class and 20 class experiments, the most successful experiments are implemented by the least and the most number of descriptors, such as, using 4 descriptors and 8 descriptors. However, as the number of descriptors increased, the performance gain increased, since the number of MC samples decreased by the increasing number of descriptors. However, the structure of the classes and the impact of descriptors on the classes affect the results.

PART 2

A NEW THEORETICAL FRAMEWORK FOR ENSEMBLE LEARNING WITH FUSION

In the second part of the thesis, a theoretical analysis of Stacked Generalization architecture is conducted. For this purpose, six theorems are proposed and proved by following the hypotheses of the first part and examining the feature space transformations.

Firstly, the meta-layer input space is theoretically analyzed, where the doubly non-linear architecture of Fuzzy SG is transformed into a linear architecture. Based on the investigations, *Theorem 1,* which promises the existence of the solution to the linear equation obtained at the meta-layer data space is proposed and proved. The rate of increase in the classification performance of SG is formalized by *Theorem 2.*

Secondly, transformations of the feature space from the base-layer descriptors to the output space of the classifiers for the meta-layer input space is geometrically investigated. Then, by investigating the linear separability conditions of meta-layer feature space, *Theorem 3,* which states the conditions for the linear separability is proposed.

Thirdly, from the analyses of the meta-layer feature space and the relationships of the error functions of the base-layer classifiers and the meta-classifier, *Theorem 4*, which states the relationship of the meta-layer feature space and the performance of the classifiers is proposed and proved. Then, the error of the SG is formalized by functional analysis in *Theorem5,* which states the conditions for the performance increase of SG. Through the investigations of the relationship between the base-layer feature space and the hypothesis function spaces of the classifiers, *Theorem 6,* which states the performance bound for SG, in terms of the relationships, is constructed.

CHAPTER 6

A NEW THEORETICAL FRAMEWORK FOR ENSEMBLE LEARNING WITH FUSION AND META-FUZZIFIED YIELD VALUE

6.1 Introduction

In most of the pattern recognition problems, feature space is formed by concatenating more than one descriptor to provide the necessary and sufficient information for representing the object classes. This commonly used technique, employed for feature space fusion, brings extra power to solve the problems in application domains such as face [80], [81], and speech recognition [82], [83] and biometric authentication [84]. In stacked generalization, feature spaces at any layer can be formed by concatenation of descriptors in various forms, as explained in the previous chapter.

The motivation for the concatenation of the feature spaces is to combine non-homogenous feature sets, to form a better feature space [85]. On the other hand, the concatenation process should be considered carefully. Ross and Govindarajan [86] state the major problems of the feature space concatenation as the incompatibility of the feature sets of multiple modalities (e.g., the attributes of fingerprints and eigenfaces), the unknown relationships between features spaces, the curse of dimensionality problem of the concatenated feature vector, and the convenient selection of the classifier for the concatenated feature space. However, feature selection, normalization and dimensionality reduction techniques, such as the kernel machine methods, can be used for the solutions to the feature space fusion problems [87].

Ironically, even though the concatenation methods are successfully applied to the pattern recognition problems, the structure of the concatenation transformation and the argument of its success could not be analyzed and explained clearly. Therefore, the concatenation operation is applied intuitively as a tool from the "black box" of the "*Black* Artist". In the present chapter, the structure of the concatenated feature spaces and the mappings between the spaces are theoretically and experimentally investigated in order to clarify the concatenation process which is the black box of feature level fusion.

In the next section, the concatenation transformations of the generic feature vectors are theoretically discussed. In the third section, the concatenation of the meta-layer features on the fuzzy SG architecture is investigated. In the fourth section, a new stacked generalization algorithm, called meta-fuzzified yield value (Meta-FYV), is introduced.

6.2 The Analysis of Concatenation Operator

Concatenation is a commonly used operator in computer science by different fields, such as logic theory, automata theory [88], pattern recognition[89] and coding theory [90] for different purposes like constructing logical connectives, reducing transmission channel error and feature space construction. In the present work, we focus on the application of the concatenation operator for the fusion of feature spaces.

In pattern recognition, the feature vector is defined as an n-dimensional vector that represents the attributes of the patterns or objects. By definition [91], any feature vector $\underline{x} \in \Re^n$ can be expressed as ;

$$\underline{x} = a_1\underline{e}_1 + a_2\underline{e}_2 + ... + a_n\underline{e}_n \qquad \text{(Equation 6.1)}$$

where, $\{\underline{e}_i\}_{i=1}^{n}$ form the standard basis of $\Re^n$ and $\{a_i\}_{i=1}^{n}$ is the set of coordinates of $\underline{x}$.

In most general case, let us consider the concatenation of two matrixes $U \in \Re^{mxn}$ and $V \in \Re^{mxp}$ consisting of feature vectors, $\{\underline{u}_i\}_{i=1}^{n} \in \Re^{mx1}$, $U = [\,\underline{u}_1 \;\; \underline{u}_2 \;\; \ldots \;\; \underline{u}_n\,]$ and $\{\underline{v}_j\}_{j=1}^{p} \in \Re^{mx1}$, $V = [\,\underline{v}_1 \;\; \underline{v}_2 \;\; \ldots \;\; \underline{v}_p\,]$ with *n* and *p* dimensions, defined by two different basis, $\{\underline{\phi}_i\}_{i=1}^{n}$ and $\{\underline{\varphi}_i\}_{i=1}^{p}$;

$$U = \alpha_1\underline{\phi}_1 + \alpha_2\underline{\phi}_2 + ... + \alpha_n\underline{\phi}_n \quad , \qquad \text{(Equation 6.2)}$$

$$V = \beta_1\underline{\varphi}_1 + \beta_2\underline{\varphi}_2 + ... + \beta_p\underline{\varphi}_p \quad . \qquad \text{(Equation 6.3)}$$

The concatenation operation is another feature matrix obtained by appending the entries of U and V, yielding *n+p* dimensional matrix. The basic motivation for the concatenation is to combine the feature spaces of the patterns for a high level information about the patterns.

Mathematically, the matrix concatenation, $Con(U,V)$ can be defined as,

$$\Psi = Con(U,V) = \begin{bmatrix} U^T \\ V^T \end{bmatrix}^T \quad , \qquad \text{(Equation 6.4)}$$

$$\Psi = [\alpha_1\underline{\phi}_1 + \alpha_2\underline{\phi}_2 + ... + \alpha_n\underline{\phi}_n \;\; \beta_1\underline{\varphi}_1 + \beta_2\underline{\varphi}_2 + ... + \beta_p\underline{\varphi}_p] \qquad \text{(Equation 6.5)}$$

where the concatenation of the basis is,

$$\Phi = [\underline{\phi}_1 \; \underline{\phi}_2 \; \cdots \; \underline{\phi}_n \; \underline{\varphi}_1 \; \underline{\varphi}_2 \; \cdots \; \underline{\varphi}_p] \qquad \text{(Equation 6.6)}$$

The concatenated matrix Ψ is *m* by *n+p* dimensional feature matrix.

One of the challenging problems of the concatenation of feature matrix is the normalization of the features during the concatenation. Since different vectors contain different magnitudes of the attributes, a feature space may dominate the others, which may cause serious problems for the algorithms based on the distance metrics. Another problem with the concatenation is the dilemma of the curse of dimensionality versus dimensionality reduction, which causes the loss of information as discussed in Chapter 3.

Definition 1 (Vectorization): Let $\Psi = (\psi_{ij}) \in \Re^{pxq}$ be a pxq matrix and $\psi_i = (\psi_{i,1}\ \psi_{i,2}\ .\ .\ .\ \psi_{i,p})^T$ is the i^{th} column of Ψ, then $Vec(\Psi) = (\psi^T{}_1\ \psi^T{}_2\ .\ .\ .\ \psi^T{}_q)^T$ is a pq dimensional vector.

In [92], Zhang utilizes the matrix concatenation operation for the solution of a set of matrix equations. According to the theorem proposed by Zhang [92], for a given matrix $\Psi \in \Re^{mxn}$, and $\Psi^t \in \Re^{nxm}$ which is the pseudo-inverse of Ψ ,and $Y \in \Re^{uxl}$, the matrix equation

$$\Psi X \Psi^t = Y \qquad \text{(Equation 6.7)}$$

has a common solution if and only if Y spans $(\Psi^t)^T \otimes \Psi$.

One of the sufficient conditions for the solution is that $Y = \Phi$, in other words, the matrix product is the basis space of the concatenated matrix. One of the most important consequences of the theorem for the pattern recognition is that it states the solution conditions for the regression equations. This theorem can be applied for the regression analysis. In the next section, the theorem will be applied to the meta-layer feature space in the regression analysis for the fuzzy SG architecture.

6.3 The Analysis of Concatenation Operator at Meta Layer in Fuzzy Stacked Generalization

In this section, the effect of concatenation operation at the output of base-layer classifiers in fuzzy SG will be investigated by matrix algebra, geometric data analysis and functional analysis. In Section 6.3.1, the theorem, proposed by Zhang will be applied to the fuzzy SG architecture. In Section 6.3.2, the geometric interpretation of the concatenation in membership vectors, at the output of base layer classifiers will be analyzed. In Section 6.3.3, the concatenation operation will be investigated in terms of the error function of the classification operation at the meta-layer, in complement with Hypothesis 1, which is introduced in Chapter 4.

6.3.1 The Matrix Analysis of Meta-layer Concatenation

The solution of the linear equations of the concatenated matrices, which is introduced in Section 6.2, can be applied for the analysis of the matrices proposed by concatenated membership vectors.

Definition 2: Let, $\mathrm{M}\in\Re^{NxCK}$ be *N by CK* dimensional membership matrix, where $\mathrm{M}(s_i)=\left[\underline{\mu}(\underline{x}_{i,1})\ ...\ \underline{\mu}(\underline{x}_{i,k})\right]$, such that, $\underline{\mu}(\underline{x}_{i,k})=\left[\mu_1(\underline{x}_{i,k})\ .\ .\ .\ \mu_C(\underline{x}_{i,k})\right]$, $\underline{\mu}(\underline{x}_{i,k})\in\Re^{1xC}$ are the class membership vectors of C classes obtained from k^{th} classifier Υ_k, $k=1,2,...,K$ which is fed by k^{th} descriptor, Γ_k, $k=1,2,...,K$, $\forall\ i=1,2,...,N$ samples.

Definition 3: $Y\in\Re^{NxC}$ is *N by C* dimensional class label matrix, where $Y(s_i)=\left[Y_1(s_i)\ .\ .Y_c(s_i)\ .\ \mathrm{Y}_C(s_i)\right]$, such that, for $Y_c(s_i)\in\Re$, $c=1,2,...,C$, and for each sample $\{s_i\}_{i=1}^{N}$, such that ,while s_i belongs to the c^{th} class ω_c, $Y_c(s_i)=\begin{Bmatrix}1, & if\ \ s_i\in\omega_c\\ 0, & otherwise\end{Bmatrix}$ and $Y=[Y_j(s_i)]$, $\forall\ i,j$.

Theorem 1 (Existence Theorem): Given membership matrix M and label matrix *Y*, there exists the solution matrix *X* for the regression equation,

$\mathrm{M}X=Y$

Proof:

Let us define a feature vector $\Psi\in\Re^{mxn}$, its pseudo-inverse $\Psi^t\in\Re^{nxm}$ and the matrix $Y\in\Re^{uxl}$. According to Zhang [92], the linear equation $\Psi X\Psi^t=Y$ would have solution if and only if $Con(Vec(Y_i))\in R(Con(\ (\Psi_i^{\ t})^T\ \otimes\Psi_i))$.

Recall that our aim is to investigate the solution of $\mathrm{M}X=Y$ equation. For this purpose, let $\mathrm{M}=Con(\ (\Psi_i^{\ t})^T\ \otimes\Psi_i)$, $Y=Con(Vec(Y_i))$ and $X=Con(Vec(X_i))$, respectively. Then,

$Vec(\Psi_i X_i \Psi_i^{\ t})=((\Psi_i^{\ t})^T\otimes\Psi_i)Vec(X_i)$, (Equation 6.8)

$((\Psi_i^{\ t})^T\otimes\Psi_i)Vec(X_i)=Vec(Y_i)$, (Equation 6.9)

$\mathrm{M}X=Y$. (Equation 6.10)

Since, the class label matrix $Y\in Span\{\underline{\mu}(\underline{x}_{i,1}),\underline{\mu}(\underline{x}_{i,2}),...,\underline{\mu}(\underline{x}_{i,k})\}$, there exists a solution matrix *X* to the linear matrix equation,

$\mathrm{M}X=Y$ (Equation 6.11)

Lemma 1:

Given membership matrix $\mathrm{M}\in\Re^{NxCK}$, where $\mathrm{M}(s_i)=\left[\underline{\mu}(\underline{x}_{i,1})\ ...\ \underline{\mu}(\underline{x}_{i,k})\right]$ and $h(\underline{x}_{i,k})=\max(\underline{\mu}(\underline{x}_{i,k}))$ with class label matrix $Y\in\Re^{NxC}$. According to the theorem proposed by Zhang [92], the general solution of the matrix equation $\mathrm{M}X=Y$ is,

$$X=Invec_{m,n}\left[\mathrm{M}^tY+(I_{mn}-\mathrm{M}^t\mathrm{M})z\right] \quad , \qquad \text{(Equation 6.12)}$$

where, $z\in\Re^{mn}$ arbitrary vector, and $Invec_{m,n}(x)$ is a matrix $X\in\Re^{mxn}$, such that, $Vec(X)=x$, I_{mn} is mn dimensional identity matrix, $\mathrm{M}=Con(\ (\Psi_i^{\,t})^T\ \otimes\Psi_i)$ and $Y=Con(Vec(Y_i))$ for $\Psi\in\Re^{mxn}$.

Definition 4:

Let's choose a group of feature vectors $G_l=\{\underline{x}_{i,l}\}_{i=1}^{g_l}$ of the samples $s_{i,l}$,with the corresponding label set, $Y_l=\{y_i\}_{i=1}^{g}$, where the samples of the feature set $\{G_l,Y_l\}_{l=1}^{K'}$ covers the whole set of samples *S*, $\bigcup_{l=1}^{K'}s_{i,l}=S$, $\forall\ l\geq 1$ and $\{G_l,Y_l\}_{l=1}^{K'}$ is classified by at least one classifier correctly, $K'\leq K$, for K classifiers. The classifier set which can correctly classify the feature set $\{G_l,Y_l\}_{l=1}^{K'}$ is defined by $\{\gamma_l\}\in\Lambda$ with the hypothesis function $h(G_l)$ and the error function $\varepsilon_l=\left\|h(G_l)-Y_l\right\|$.

Definition 5:

When the membership matrix from the k^{th} classifier which classifies a group of feature vectors, $\bigcup_{k=1}^{K'}s_{i,k}=S$, $\forall\ k\geq 1$, correctly, is added, the meta-layer membership matrix, solution matrix, meta-layer class label matrix and the error function is defined by, $\mathrm{M}(k)$, $X(k)$, $Y(k)$ and $e(k)$, respectively, such that,

$$e(k)=\left\|\mathrm{M}(k)X(k)-Y(k)\right\| \qquad \text{(Equation 6.13)}$$

Theorem 2 (Performance Increase Ratio) : In a 2-layer Homogenous Stacked Generalization architecture consisting of C-classes and K-classifiers, fed by distinct K-descriptors, the performance of SG increases by the ratio $\upsilon(k)=\frac{1}{2}\frac{\left\|e(k)\right\|^2}{e(k)^T\mathrm{M}\mathrm{M}^te(k)-\left\|e(k)\right\|^2}$, where $\mathrm{M}\in\Re^{NxCK}$ is the meta layer membership matrix and $e(k)$ is the error function, when the membership matrix obtained from the k^{th} supplementary and the mutual classifier γ_k is augmented to meta layer feature space.

Proof:

Let's define the cost function of the equation $\mathrm{M}X = Y$ by,

$$J(X,Y) = \|\mathrm{M}X - Y\|^2$$ (Equation 6.14)

In order to minimize the cost function, we may check the minima of the function by,

$$\nabla_X J(X,Y) = 2\mathrm{M}\|\mathrm{M}X - Y\| \quad ,$$ (Equation 6.15)

$$\nabla_Y J(X,Y) = -2\|\mathrm{M}X - Y\| \quad ,$$ (Equation 6.16)

and since $X = \mathrm{M}^t Y$, where M^t is the pseudo inverse of M

$$\nabla_X J(X,Y) = 0 \quad .$$ (Equation 6.17)

The positive gradient descent procedure results in $Y = 0$, however, Y increases as k increases. Therefore, a negative descent procedure proposed by Duda et. al. [13] can be applied, such that,

$$Y(k+1) = Y(k) - \upsilon\frac{1}{2}\left[\nabla_Y J - |\nabla_Y J|\right] \quad ,$$ (Equation 6.18)

$$Y(k+1) = X(k) + 2\upsilon(k)e^+(k) \quad .$$ (Equation 6.19)

where, $\upsilon(k)$ is convergence rate and $e^+(k)$ is the positive part of the error function ,such that,

$$e^+(k) = \frac{1}{2}(e(k) + |e(k)|$$ (Equation 6.20)

Following equation (6.13) $e(k)$ is,

$$e(k) = (\mathrm{MM}^t - I)Y(k)$$ (Equation 6.21)

and

$$e(k+1) = e(k) + 2\upsilon(\mathrm{MM}^t - I)e^+(k)$$ (Equation 6.22)

Then,

$$\|e(k+1)\|^2 = \|e(k)\|^2 + \|2\upsilon(k)(\mathrm{MM}^t - I)e^+(k)\|^2 + 4e(k)^T\upsilon(k)(\mathrm{MM}^t - I)e^+(k)$$

(Equation 6.23)

Since $e(k)^T \mathrm{M} = 0$, where $e(k)^T$ is the transpose of $e(k)$ and,

$$e(k)^T \upsilon(k)(\mathrm{MM}^t - I)e^+(k) = -\upsilon(k)\left\|e^+(k)\right\|^2 \qquad \text{(Equation 6.24)}$$

and since $\mathrm{MM}^t = (\mathrm{MM}^t)^T \mathrm{MM}^t$,

$$\left\|2\upsilon(k)(\mathrm{MM}^t - I)e^+(k)\right\|^2 = \upsilon(k)^2\left\|e^+(k)\right\|^2 - \upsilon(k)^2 e^+(k)\mathrm{MM}^t e^+(k) \qquad \text{(Equation 6.25)}$$

Therefore,

$$\frac{1}{4}(\left\|e(k+1)\right\|^2 - \left\|e(k)\right\|^2) = \upsilon(k)(1-\upsilon(k))\left\|e^+(k)\right\|^2 + \upsilon(k)^2 e^+(k)^T \mathrm{MM}^t e^+(k)$$

(Equation 6.26)

By taking the partial differentiation with respect to $\upsilon(k)$,

$$\frac{\partial}{\partial(\upsilon(k))}(\frac{1}{4}(\left\|e(k)\right\|^2 - \left\|e(k+1)\right\|^2)) = \frac{\partial}{\partial(\upsilon(k))}(\upsilon(k)(1-\upsilon(k))\left\|e^+(k)\right\|^2 + \upsilon(k)^2 e^+(k)^T \mathrm{MM}^t e^+(k))$$

(Equation 6.27)

$$\upsilon(k) = \frac{1}{2}\frac{\left\|e^+(k)\right\|^2}{e^+(k)^T \mathrm{MM}^t e^+(k) - \left\|e^+(k)\right\|^2} \qquad \text{(Equation 6.28)}$$

Since $e(k) > 0$, and $e^+(k) = e(k)$,

$$\upsilon(k) = \frac{1}{2}\frac{\left\|e(k)\right\|^2}{e(k)^T \mathrm{MM}^t e(k) - \left\|e(k)\right\|^2} \qquad \text{(Equation 6.29)}$$

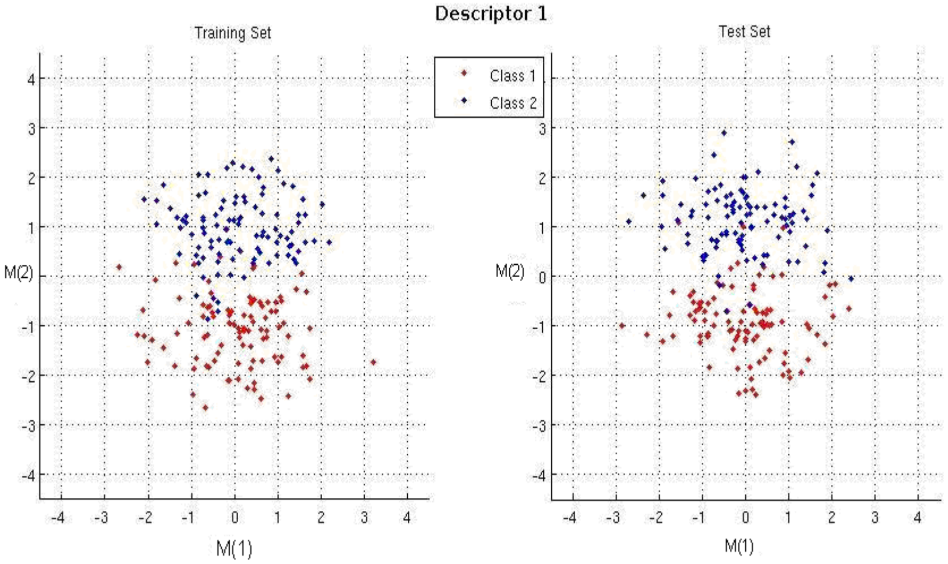
6.3.2 The Geometric Analysis of Meta-layer Concatenation

The column space meta-layer feature matrix $\mathrm{M}(s_i)$ consists of the samples that lie on a set of lines defined by the line equations $\sum_{k=1}^{K} \underline{\mu}(\underline{x}_{i,k}) = K$. This is the consequence of the structure of the membership vectors, which adds to 1 for each classifier.

Therefore, the samples reside on the edges of the hyper-polygon in *CK* dimensional Euclidean Space and have tendencies toward the vertices of the hyper-polygon where the correct estimations of the class labels take place.

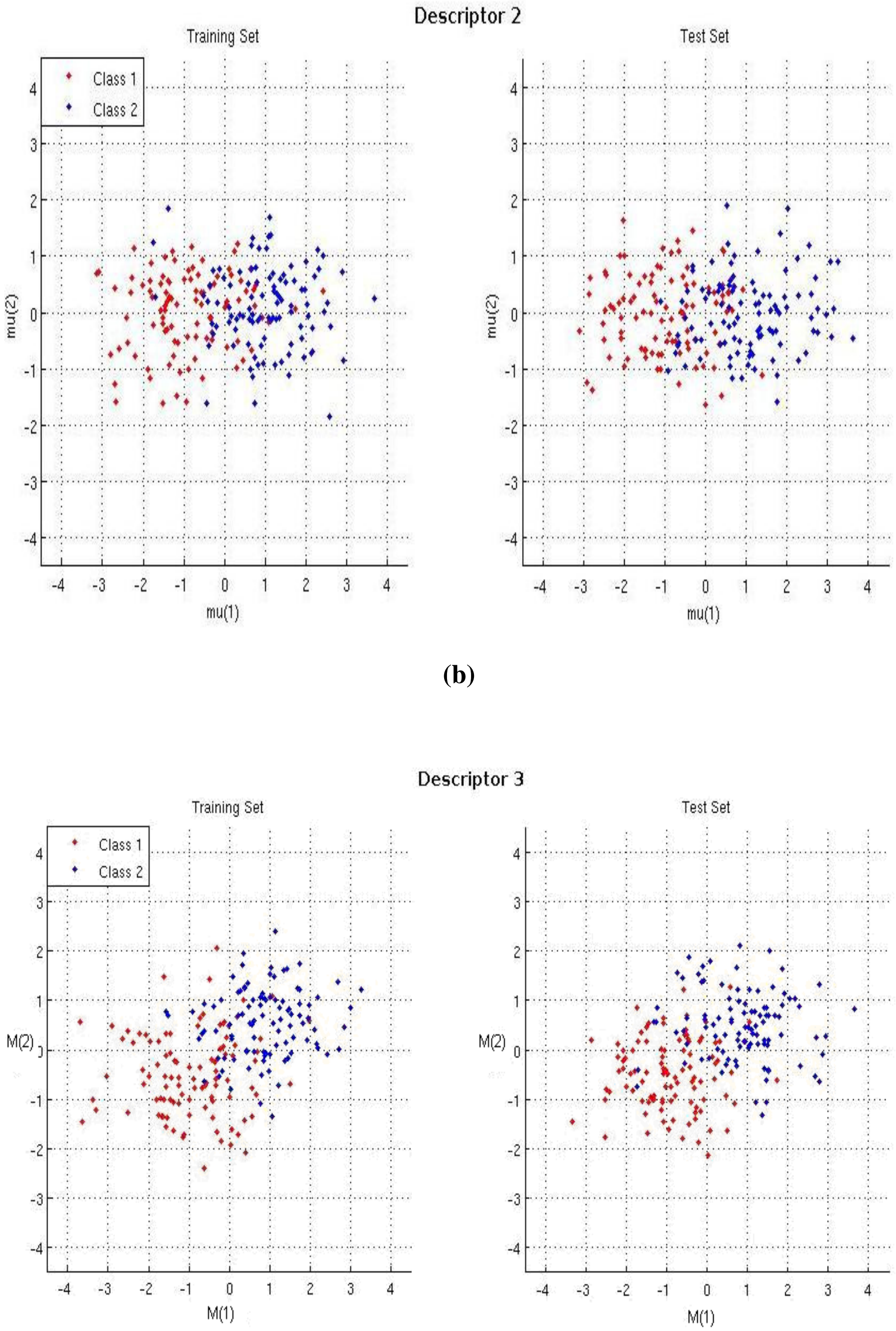
In order to investigate the geometrical properties of the concatenation operator in fuzzy SG, let's consider an experiment based on synthetic datasets, consisting of 2 classes and 3 descriptors. In the base layer feature spaces, the classes will be distributed by Gaussian distribution with the covariance T_k, which is the covariance matrix of the classes distributed in Γ_k and variance matrices M_k, $k=1,2,3$, $M_1 = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}$, $\mathrm{T}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$, $M_2 = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}$, $\mathrm{T}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$ and $M_3 = \begin{pmatrix} -1 & -0.5 \\ 1 & 0.5 \end{pmatrix}, \mathrm{T}_3 = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$, for the first, second and third descriptor, respectively (Figure 6.1).

Each class consists of 100 samples for each test and training datasets, therefore, training and test datasets consist of 200 samples for each descriptor, respectively. The κ values for the fuzzy knn algorithms at the base layer and the meta-layer are chosen as 10, for simplicity.



(a)

Figure 6.1: (a) Feature dataset in descriptor 1, **(b)** Feature dataset in descriptor 2, **(c)** Feature dataset in descriptor 3, consisting of 2 classes, where red samples belong to the first class and blue samples belong to the second class.



(b)

(c)

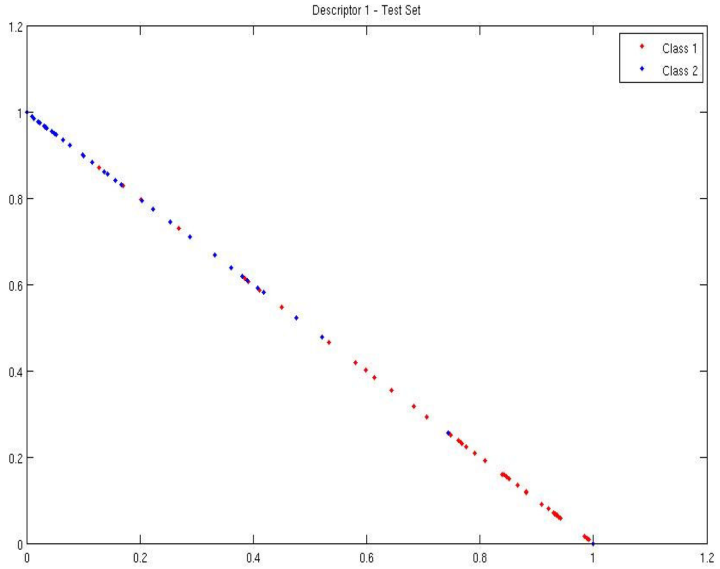
Figure 6.1 (Continued)

The datasets are distributed with different orientations at each descriptor, since they are distributed by different mean values. Therefore, the samples that are classified correctly would be relatively different for each classifier. Meanwhile, σ_{BC} values for two classes are same for the first and second descriptors, and it is relatively different in the third descriptor. Consequently, overlapping of the classes in each descriptor is different.

The output space of the base layer classifiers consist of membership matrices obtained from the classifiers $\underline{\mu}(\underline{x}_{i,k}) \in \Re^{1x2}$, for *k=1,2,3*, *i=1,2,…,200* ,such that,

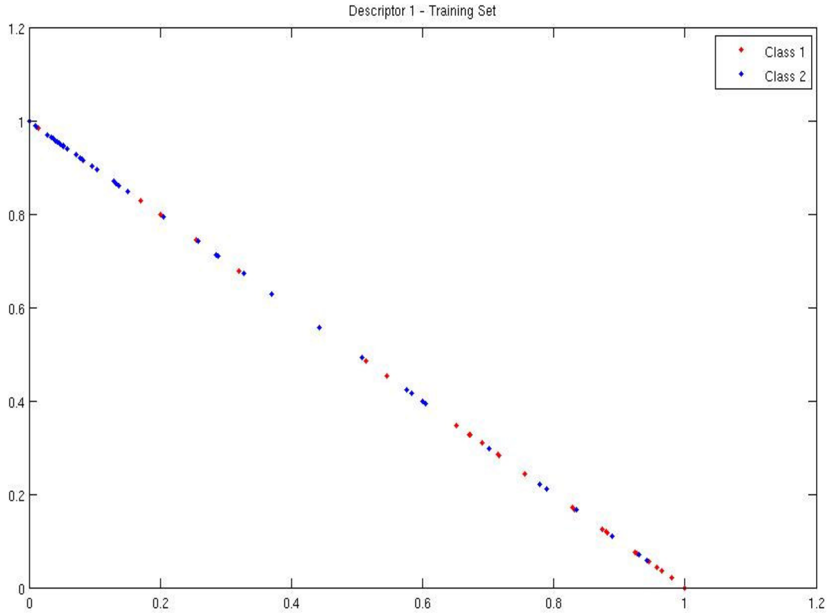
$$\underline{\mu}(\underline{x}_{i,k}) = [\mu_1(\underline{x}_{i,k})\ \mu_2(\underline{x}_{i,k})] \qquad \text{(Equation 6.30)}$$

for 2 classes, by the constraint $\sum_{c=1}^{2} \mu_c(\underline{x}_{i,k}) = 1$, which constructs the line equation. The output space of each base layer classifier can be visualized by plotting the membership vectors of each classifier $\underline{\mu}(\underline{x}_{i,k})$ (Figure 6.2)

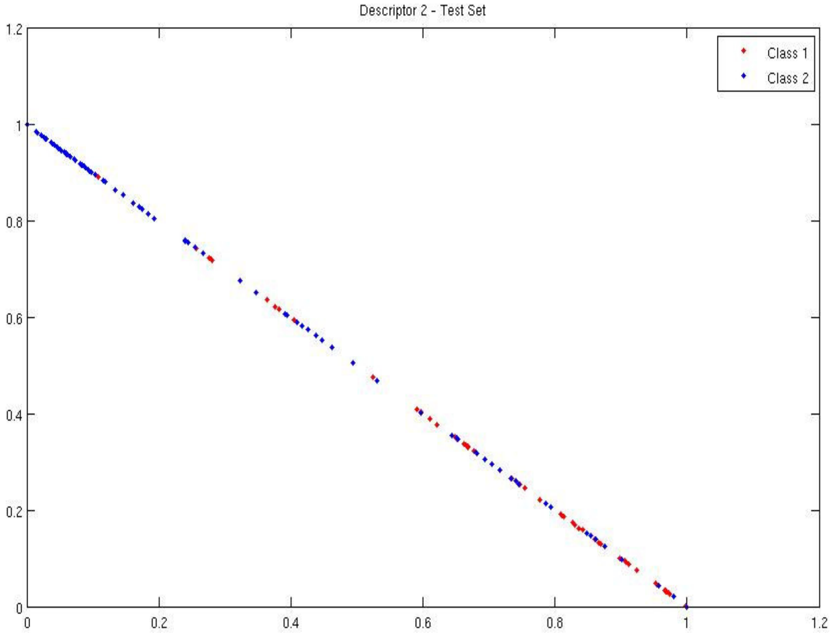


(a-1) Test Set

Figure 6.2: **(a)** Membership space for Descriptor 1, **(b)** Membership space for Descriptor 2, **(c)** Membership space for Descriptor 3, with test and training datasets.

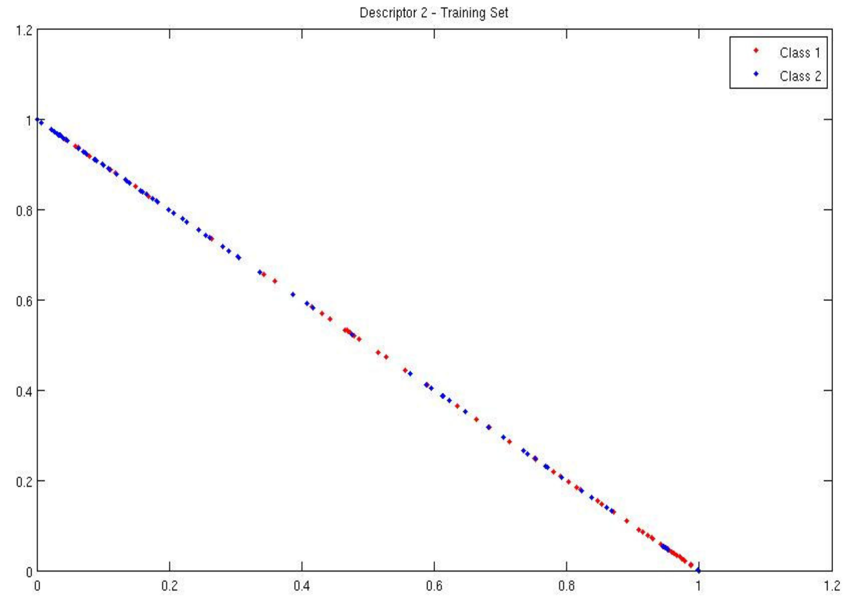


(a-1) Training Set

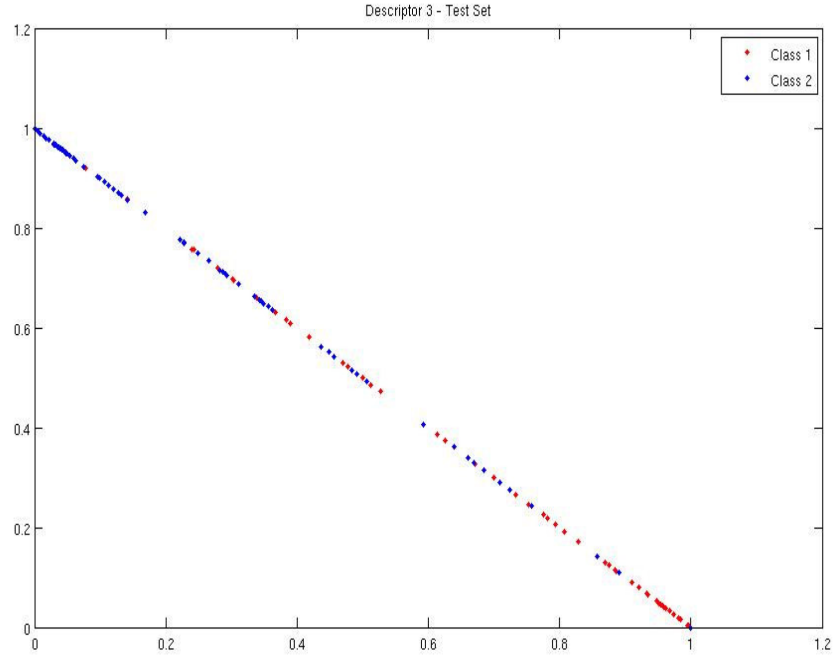


(a-2) Test Set

Figure 6.2 **(Continued)**

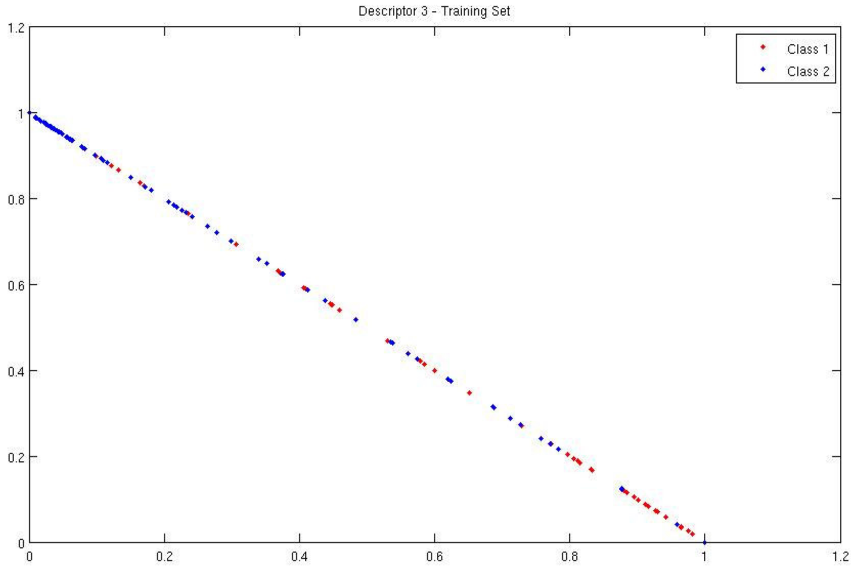


(a-2) Training Set



(a-3) Test Set

Figure 6.2 (**Continued**)



(a-3) Training Set

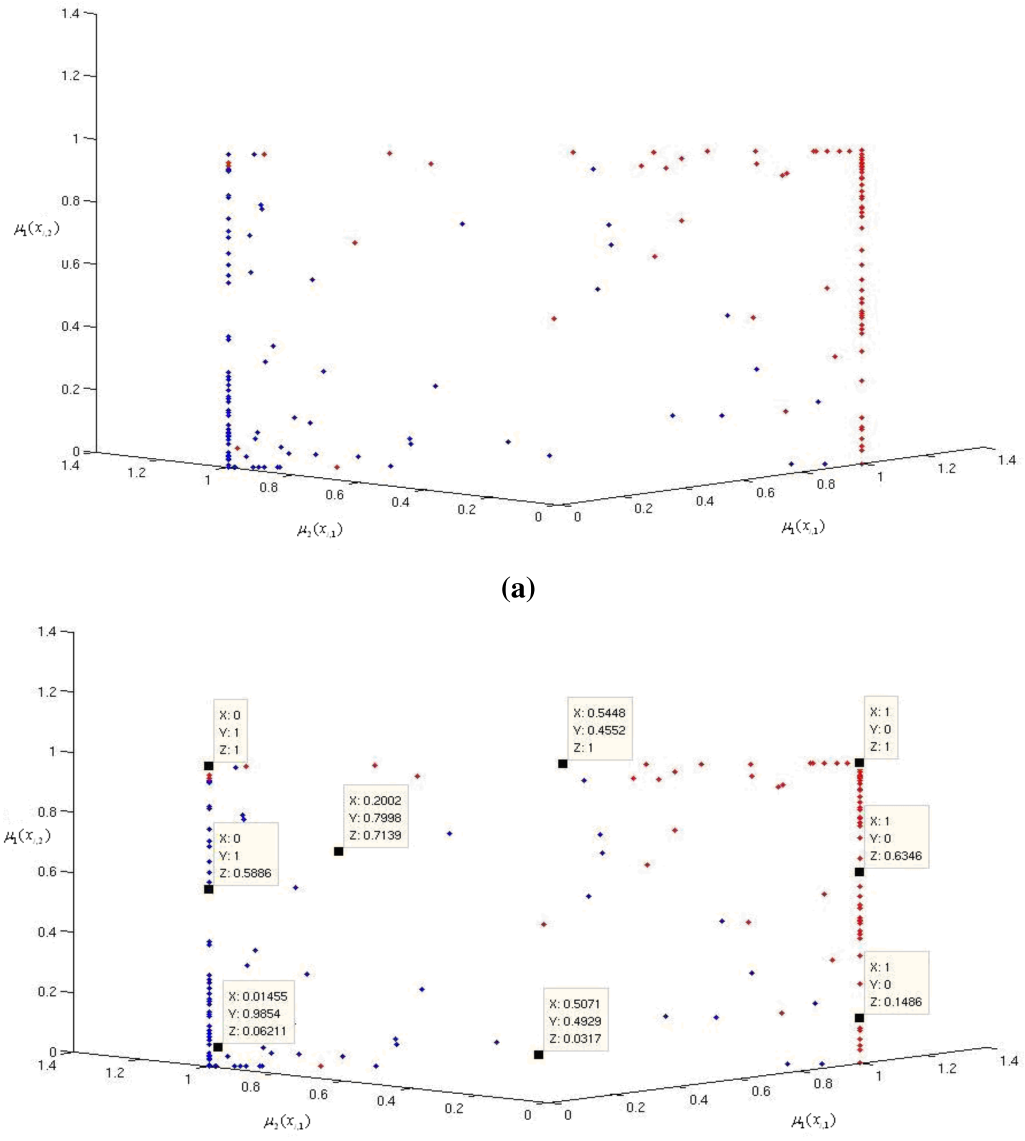
Figure 6.2 (Continued)

As can be seen from the figures, the samples are nicely mapped on a line for each descriptor. In the input space at the meta-layer, the membership vectors are concatenated. In order to visualize the concatenated feature space, the vectors are plotted in combination with 3 dimensions selected from the concatenated meta-layer feature vector $\mathrm{M}(s_i) = [\mu_c(\underline{x}_{i,k})]$, *c=1,2*, *k=1,2,3*, $\forall$ i .

In some of the figures, the values of some samples are introduced in order to sense the structure of the data space (Figure 6.3, Figure 6.4. and Figure 6.5).

In Figure 6.3, it is observed that the samples are mostly spread through the edges and are accumulated around the vertices of the cube. In Figure 6.4, similar separation and the accumulation is observed for different basis space consisting of different membership values, which are the membership values of the samples relative to the first class. Moreover, in Figure 6.4-c, 2D projection of the samples over $\mu_1(x_{i,1}) - \mu_2(x_{i,3})$ plane is visualized. In the figure, it is observed that the samples are accumulated through the corners of the plane, similar to the distribution in Figure 6.3.

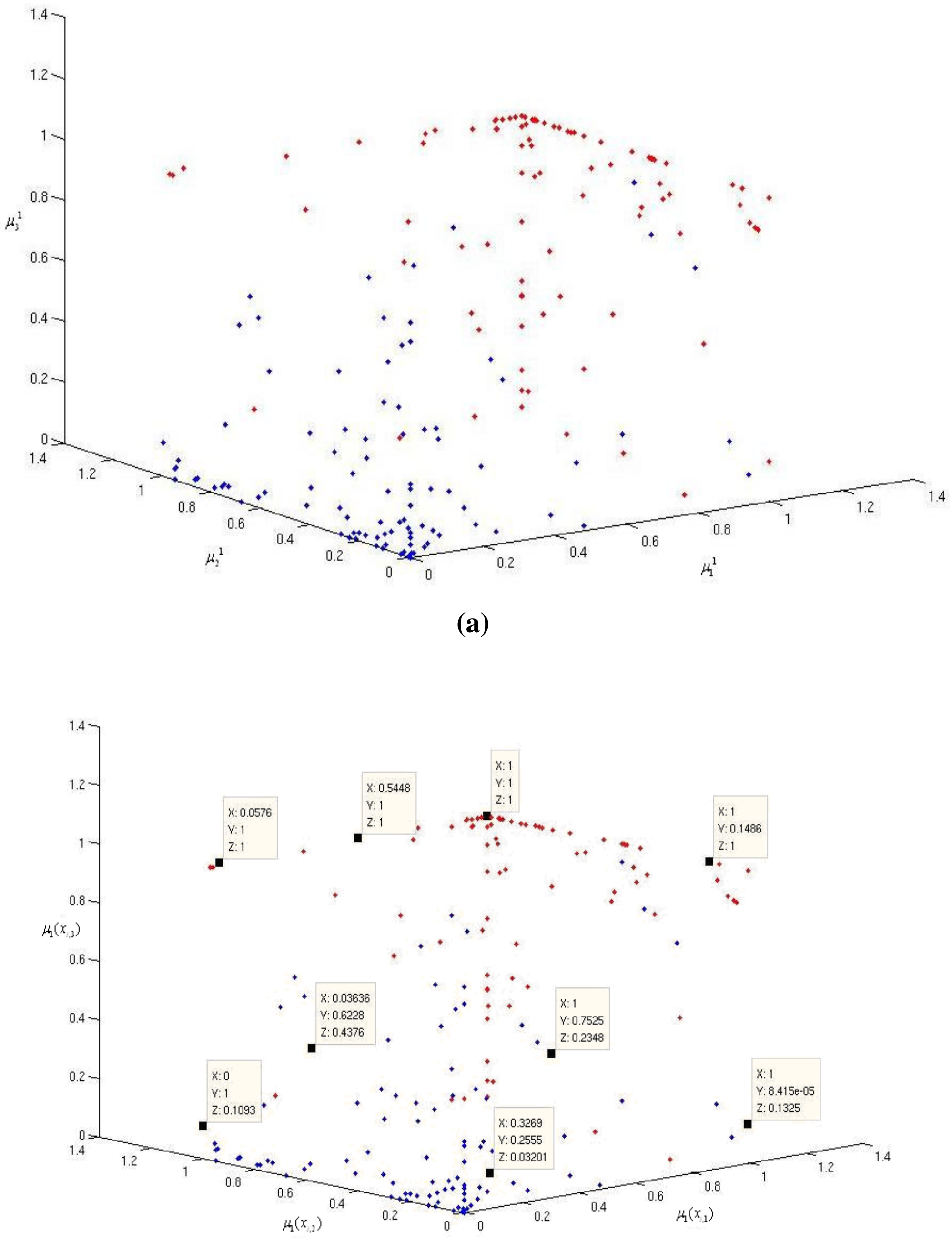
In Figure 6.5, the samples are distributed with the membership values for the second class and scattered with similar arrangement in Figure 6.4, even if the basis vectors that span the spaces are orthogonal.



(a)

(b)

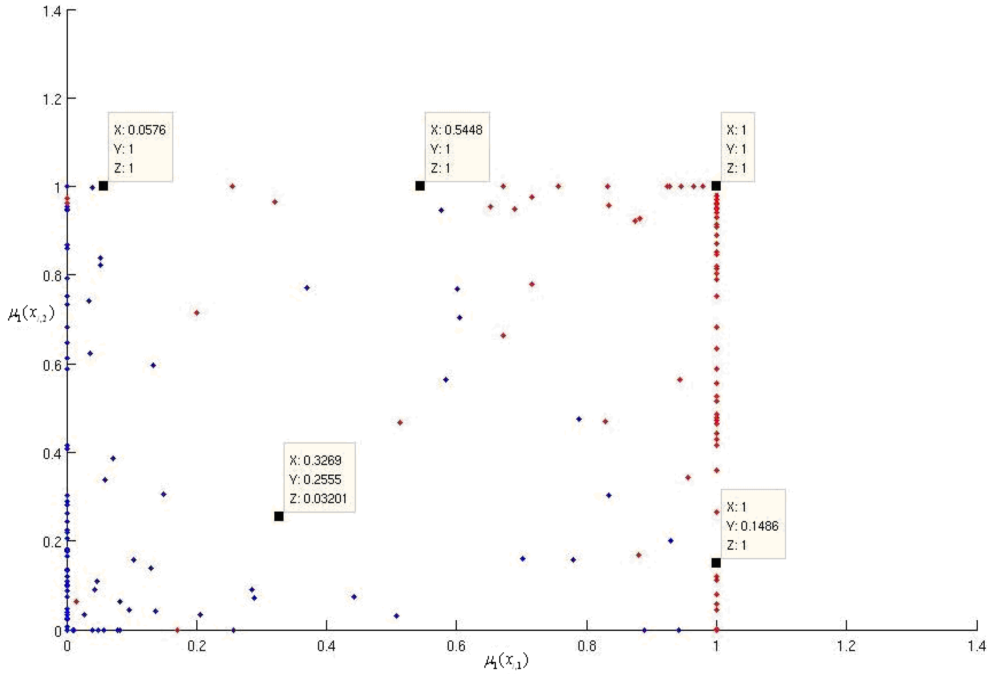
Figure 6.3: (a) Concatenation of base layer membership vectors, **(b)** with membership values of some samples. The concatenated dimensions are $\mu_1(x_{i,1}), \mu_2(x_{i,1}), \mu_1(x_{i,2})$



(a)

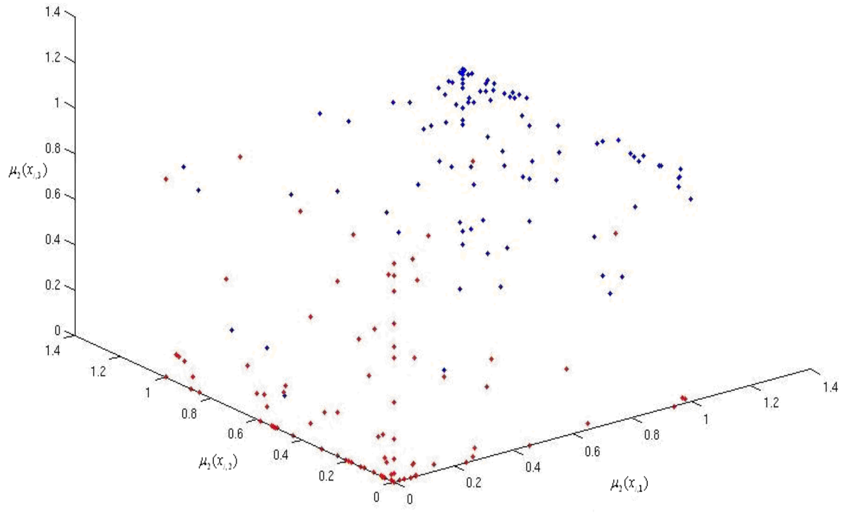
(b)

Figure 6.4: (a) Concatenation of base layer membership vectors over the dimensions $\mu_1(x_{i,1}), \mu_1(x_{i,2}), \mu_1(x_{i,3})$, **(b)** with data values of some samples , **(c)** the projection on $\mu_1(x_{i,1}) - \mu_1(x_{i,3})$ plane.



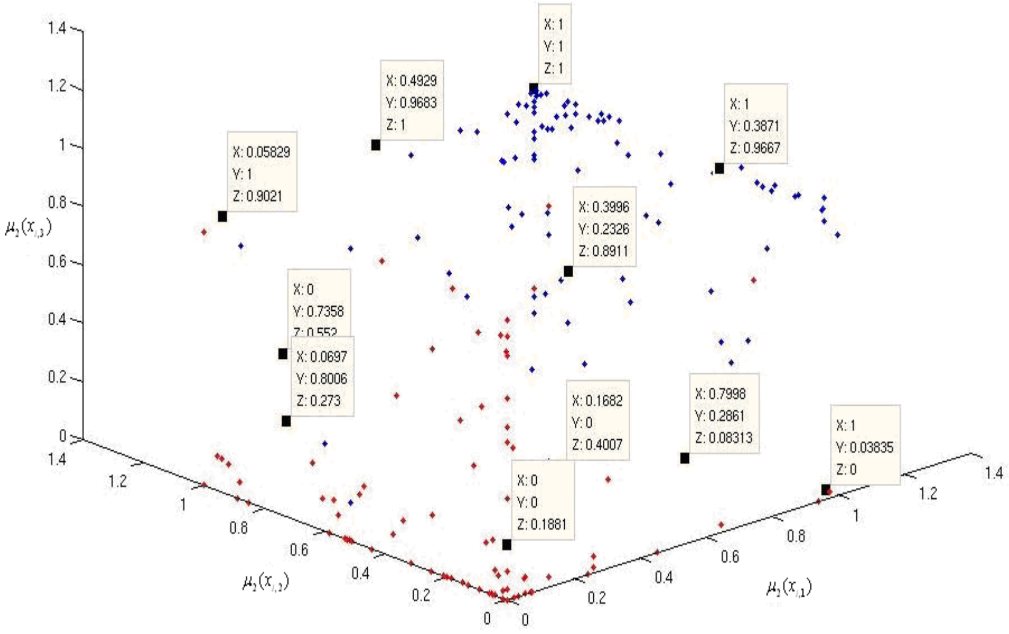
(c)

Figure 6.4 (Continued)



(a)

Figure 6.5: (a) Concatenation of base layer membership vectors over the dimensions $\mu_2(x_{i,1}), \mu_2(x_{i,2}), \mu_2(x_{i,3})$, **(b)** with membership values of some samples



(b)

Figure 6.5 (Continued)

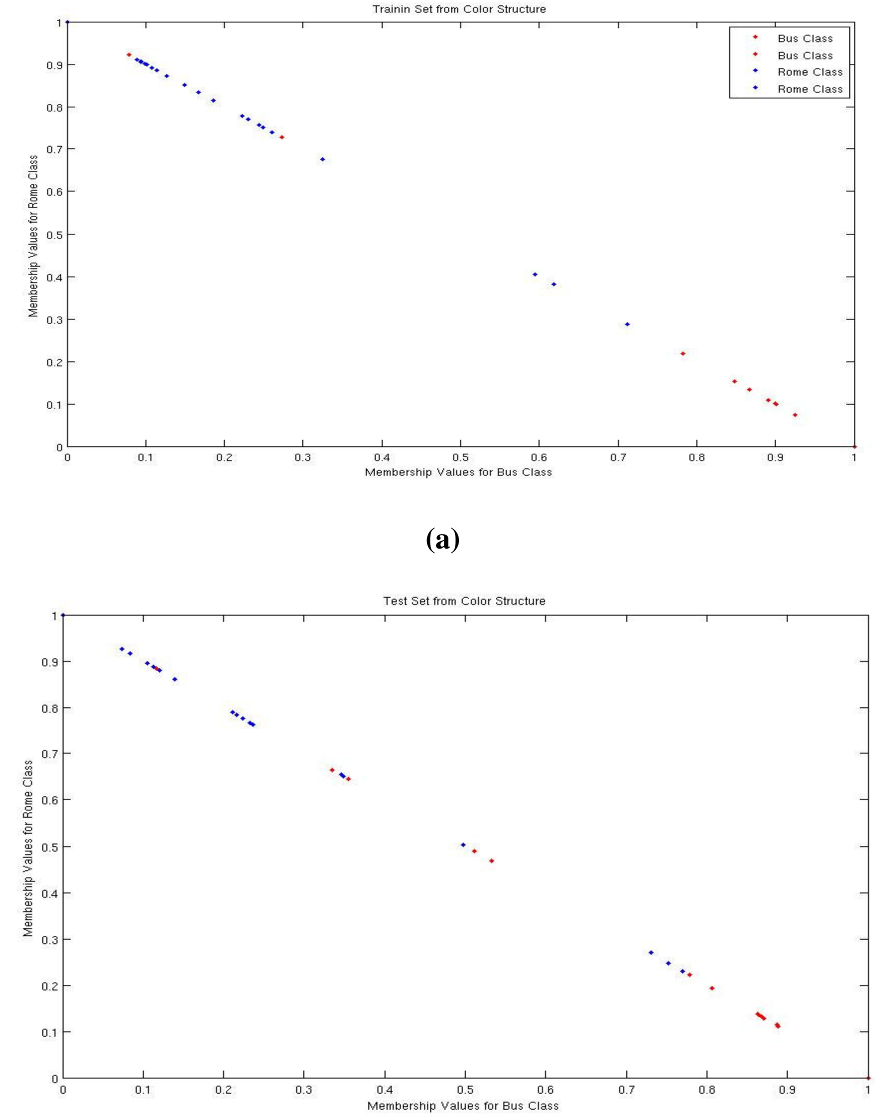
In the second type of the experiments, the concatenation operation is analyzed on the Corel Dataset. The classification is implemented on Rome and Buses datasets using Color Structure (32 dimensional) and Edge Histogram (80 dimensional) descriptors, with 50 samples for each class. The classification performances of the base layer fuzzy k-nn classifiers implemented on the Color Structure and Edge Histogram descriptors are introduced in (Table 6.1)

Table 6.1: Individual performances of the base-layer fuzzy k-nn implemented on Color Structure and Edge Histogram descriptors, respectively. The performance of SG is 100%

	Training Set	**Test Set**
Color Structure	94%	94%
Edge Histogram	95%	85%

In the experiment, it is observed that SG increases the classification performance by 6% relative to the individual classification performances.

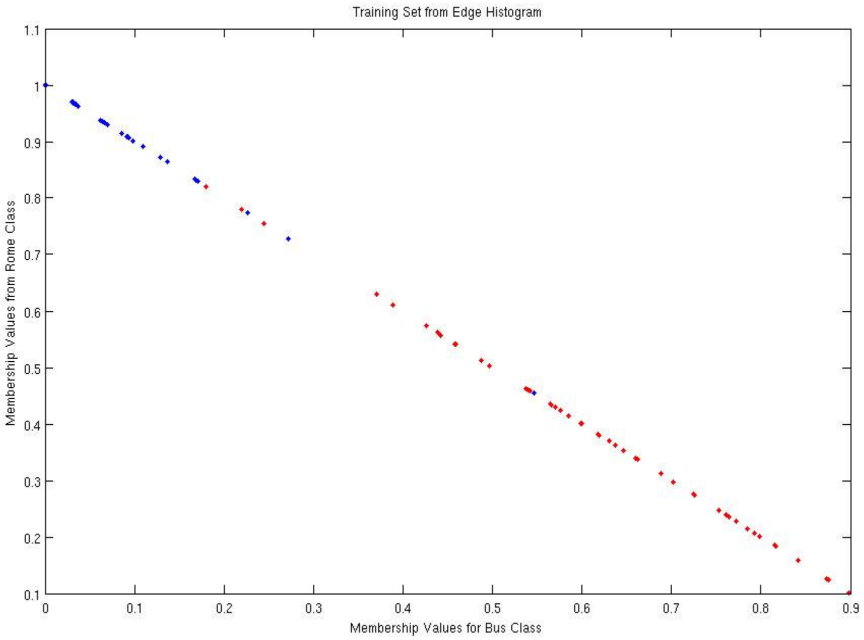
The membership vectors at the output space of the base layer classifiers are visualized in Figure 6.6 for (a) Color Structure training dataset (CS-tr), (b) Color Structure test dataset (CS-te), (c) Edge Histogram training dataset (EH-tr), (d) Edge Histogram test dataset, (EH-te) respectively.



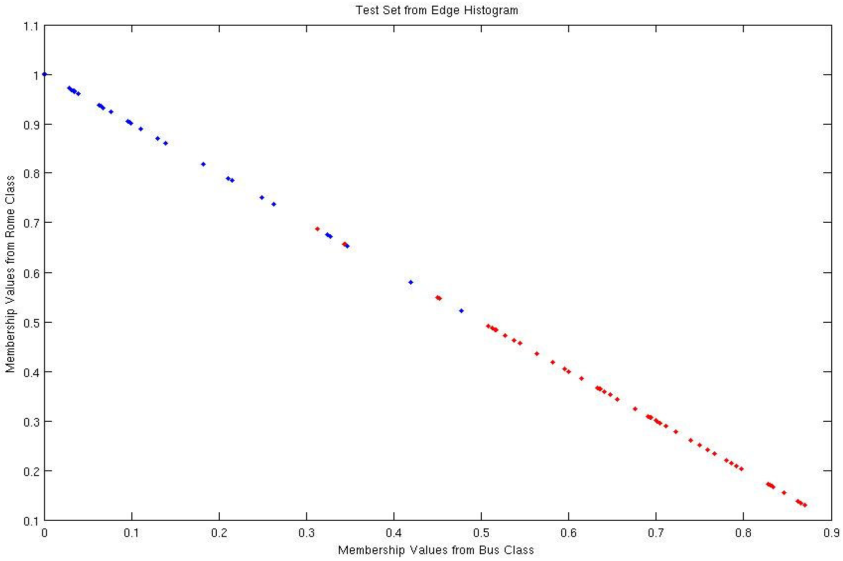
(a)

(b)

Figure 6.6: Visualization of the membership vectors obtained from fuzzy k-nn classifiers implemented on (a) CS-tr, (b) CS-te, (c) EH-tr , (d) EH-te



(c)



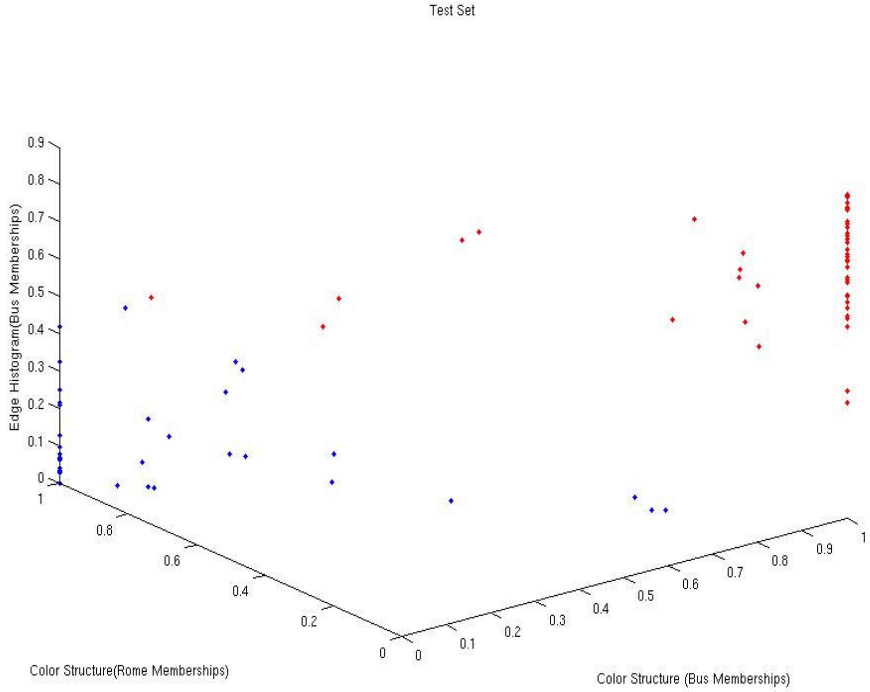
(d)

Figure 6.6 (Continued)

In Figure 6.6, the distribution of the samples at the base layer output space with the basis consisting of the membership values is observed. By examining the sample distributions in the figures, we can perceive the classification performances in Table

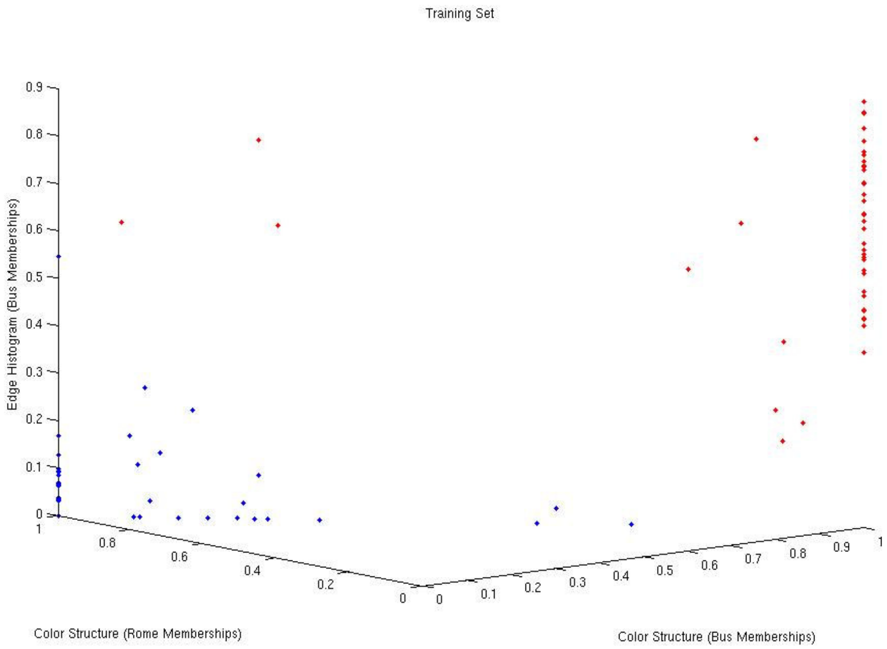
6.1. In Figure 6.6-a to Figure 6.6-c, the samples are well separated through the end points of the line and correspondingly, the classification performances on these spaces are relatively higher than in Figure 6.6-d. In Figure 6.6-d, the samples are scattered smoothly on the line, and the classification performance of the corresponding space is relatively low.

In the input space of the base layer, the membership vectors are concatenated. The proposed concatenated matrix is visualized in (Figure 6.7).

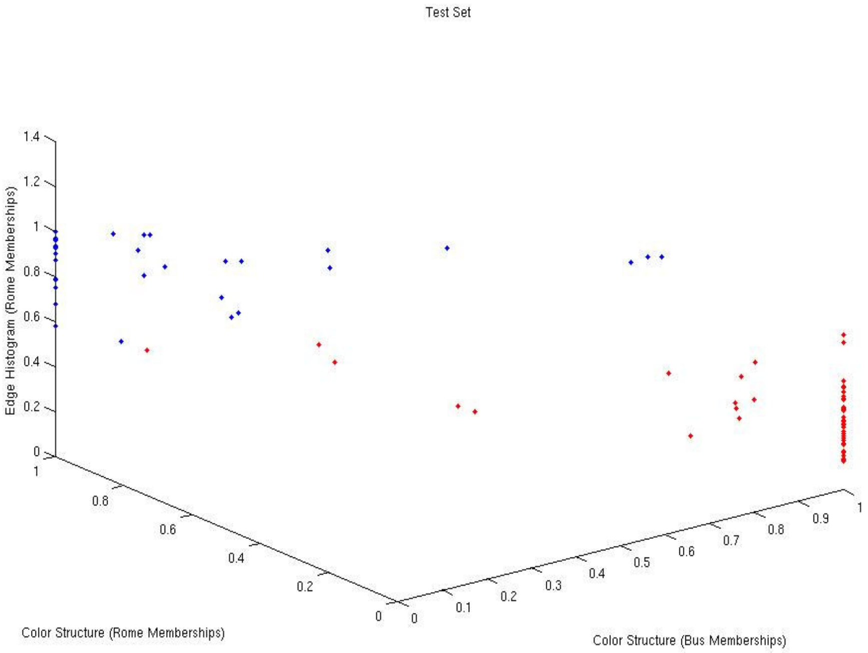


(a)

Figure 6.7: Concatenation of the membership vectors with the dimensions obtained from the descriptors **(a)** CS(Bus)-CS (Rome)- EH (Bus) for test dataset, **(b)** for training dataset, **(c)** CS(Bus)-CS (Rome)- EH(Rome) for test dataset, **(d)** for training dataset, **(e)** CS(Rome)- EH(Bus)- EH(Rome) for test dataset, **(f)** for training dataset**.**

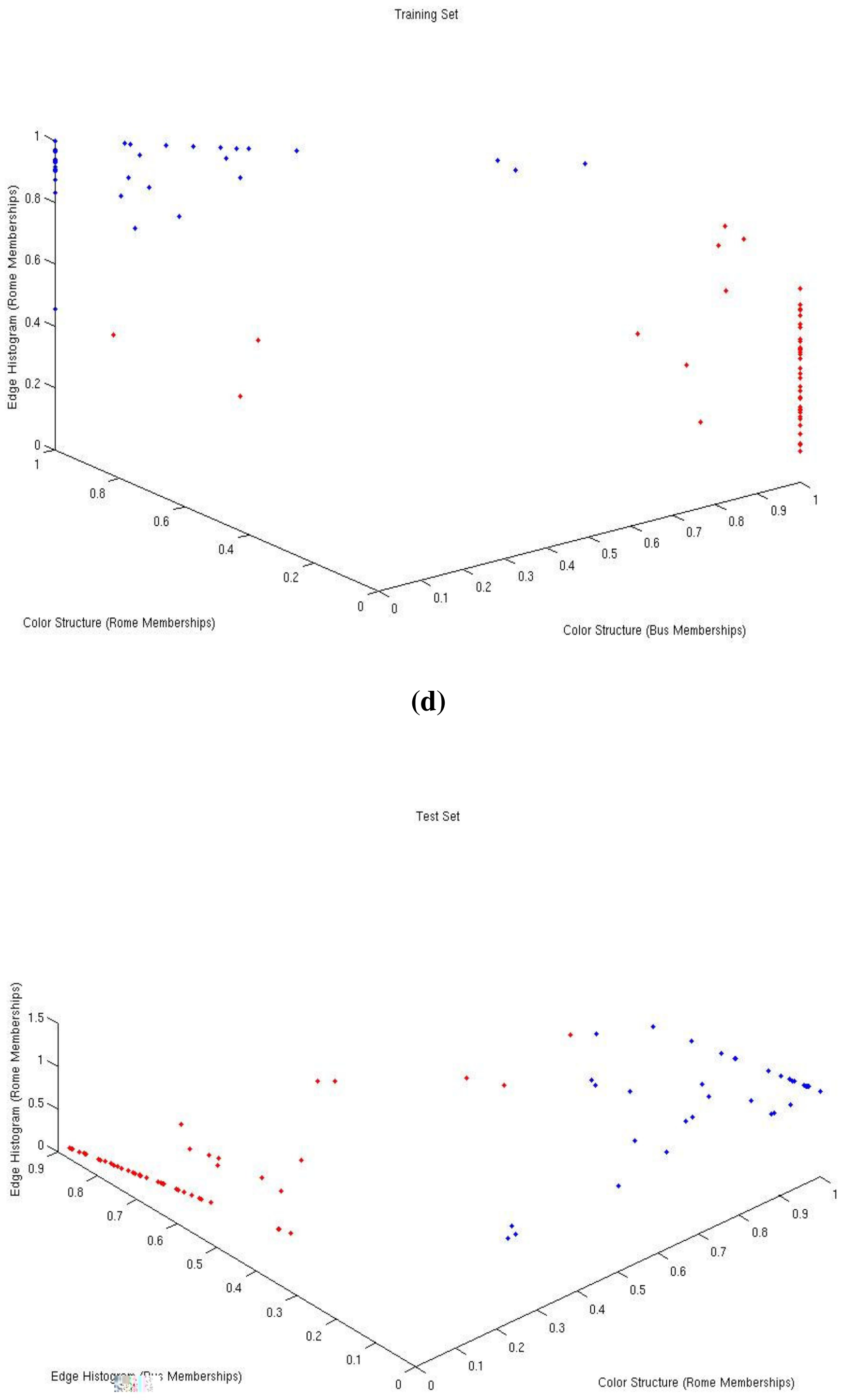


(b)



(c)

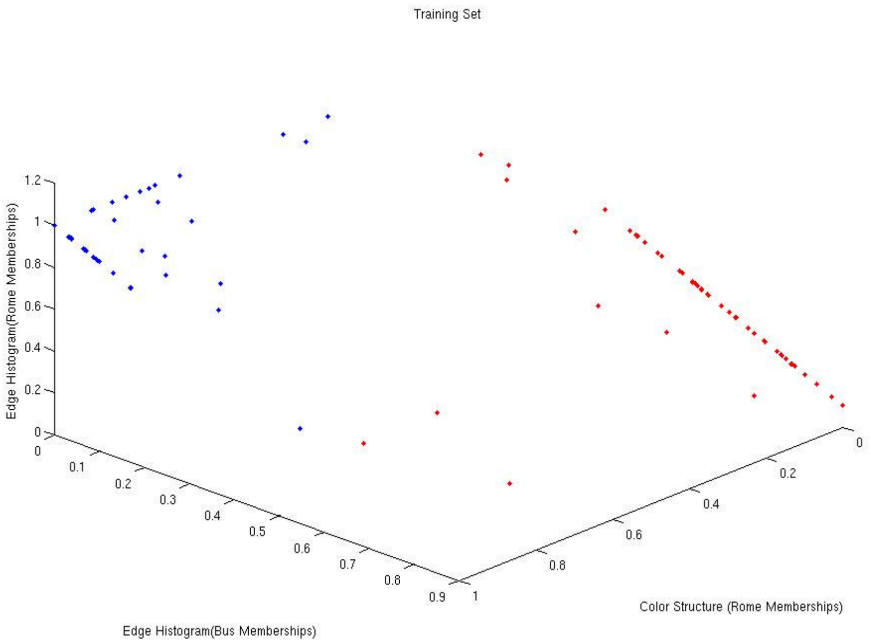
Figure 6.7 (Continued)



(d)

(e)

Figure 6.7 (Continued)



(f)

Figure 6.7 (Continued)

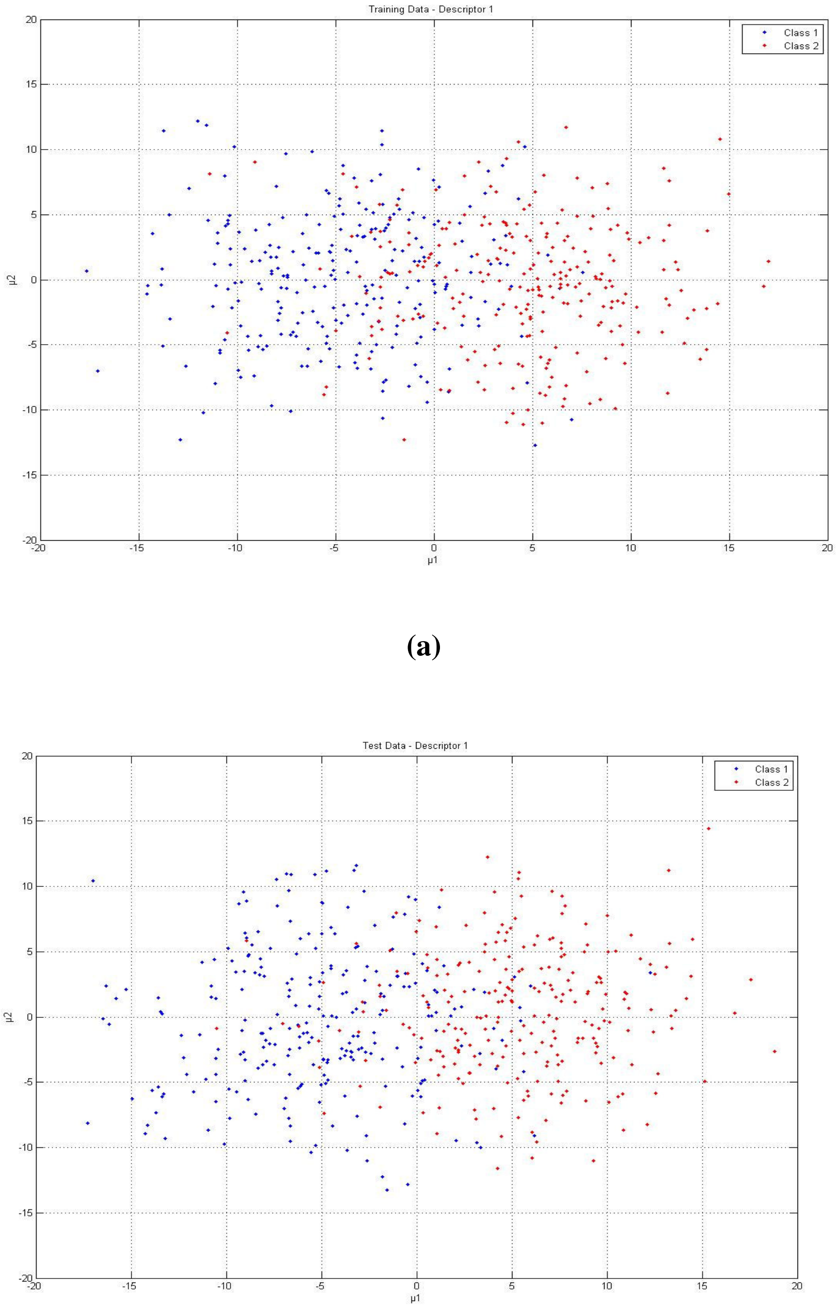
In Figure 6.7, we examine that the number of red points, which represent the samples belonging to bus class, that reside on the edges are more than the blue points, which represent the Rome class. Therefore, we observe that the samples from the bus class are clustered better than the samples from the Rome class.

In Figure 6.7-a and Figure 6.7-b it is observed that the red points are mostly accumulated on the edge that is represented by the membership values obtained from the classifier which is fed by Edge Histogram descriptor and the samples are relatively well separated compared to the other spaces. Hence, it is considered that the samples are classified better in that space and the samples from the bus class are classified better via the feature vectors obtained from Edge Histogram.

Meanwhile, in Figure 6.7-f, it is observed that the samples from Rome class mostly reside on the edges represented by the memberships vectors obtained from Color Structure descriptor indicating that they are described better by Color Structure descriptor than Edge Histogram descriptor. Consequently, it can be considered that the bus class is described better by Edge Histogram descriptor and Rome class is described better by Color Structure descriptor.

Furthermore, it is investigated that the samples reside on the planes and tends to be separated linearly in all the subspaces of Figure 6.7. These experiments provide a strong clue for the effect of the concatenation of the memberships vectors, for an increased the linear separability.

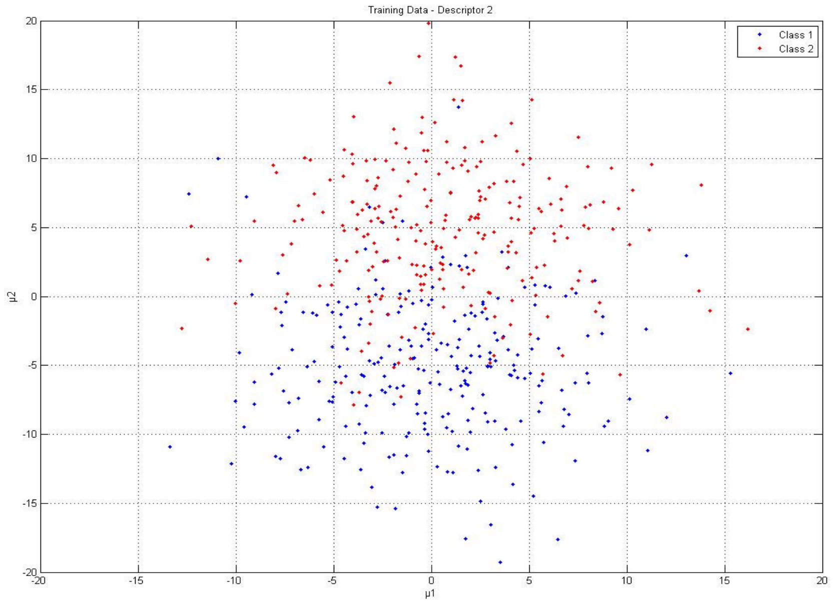
In the third type of the experiments, synthetic datasets, consisting of 2 classes and 3 descriptors are constructed. In base layer feature space, the classes are distributed by Gaussian distribution with the covariance and variance matrices, $M_1 = \begin{pmatrix} -5 & 0 \\ 5 & 0 \end{pmatrix}$, $\mathrm{T}_1 = \begin{pmatrix} 5 & 5 \\ 5 & 5 \end{pmatrix}$, $M_2 = \begin{pmatrix} 0 & -5 \\ 0 & 5 \end{pmatrix}$, $\mathrm{T}_2 = \begin{pmatrix} 5 & 5 \\ 5 & 5 \end{pmatrix}$ and $M_3 = \begin{pmatrix} 5 & 5 \\ 0 & 0 \end{pmatrix}$, $\mathrm{T}_3 = \begin{pmatrix} 5 & 5 \\ 5 & 5 \end{pmatrix}$, for the first, second and third descriptor, respectively (Figure 6.8) . Each class consists of 250 samples for each test and training datasets, therefore, training and test datasets consist of 500 samples for each descriptor, respectively. The κ values for the fuzzy κ-nn algorithms at the base layer and the meta-layer are chosen as 10, for simplicity.



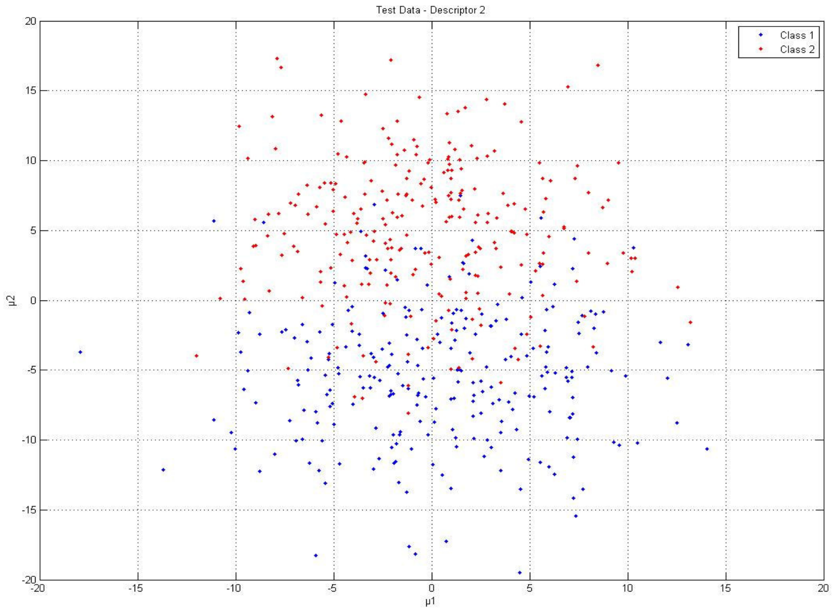
(a)

(b)

Figure 6.8: Artificial **(a)** feature training dataset in descriptor 1, **(b)** feature test dataset in descriptor 1, **(c)** feature training dataset in descriptor 2, **(d)** feature test dataset in descriptor 2, **(e)** feature training dataset in descriptor 3, **(f)** feature test dataset in descriptor 3, consisting of 2 classes, where blue samples belong to the first class and red samples belong to the second class.

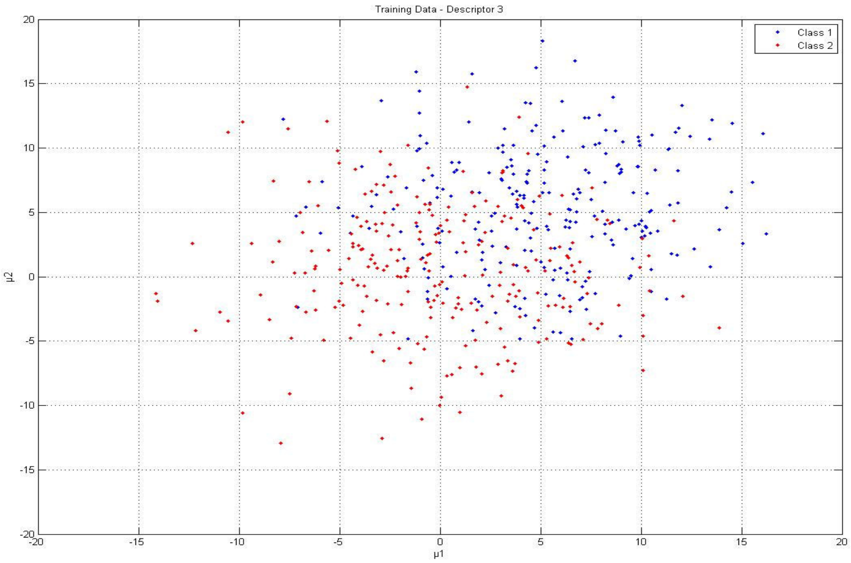


(c)

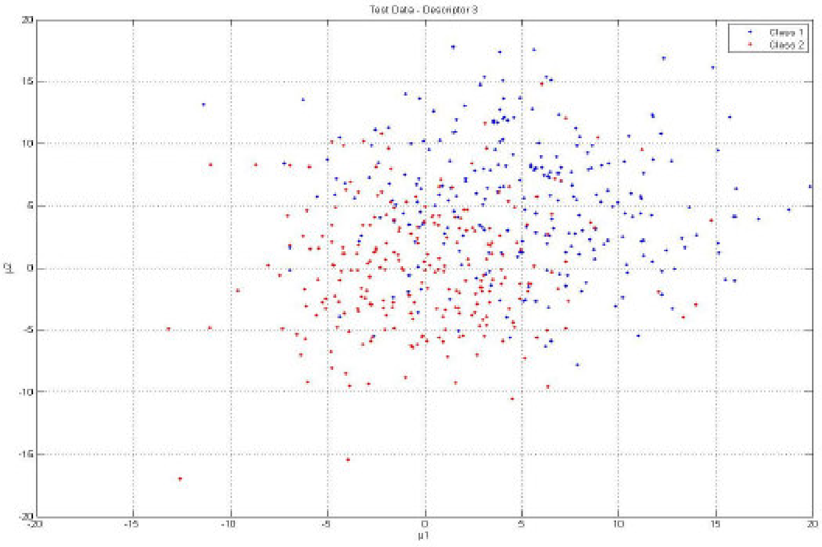


(d)

Figure 6.8 (Continued)



(e)



(f)

Figure 6.8 (Continued)

The samples are distributed with equal $\sigma_{BC}^{1,2}$ in Figure 6.8-a,b and Figure 6.8-c,d however, with different $\sigma_{BC}^{1,2}$ in Figure 6.8-e,f. It is noticed that in all of the figures, the samples are distributed with vertical, horizontal and diagonal symmetry, respectively.

After the classification, at each base layer classifiers, the following classification performances are obtained,

Table 6.2: Fuzzy SG classification performances of the classifiers

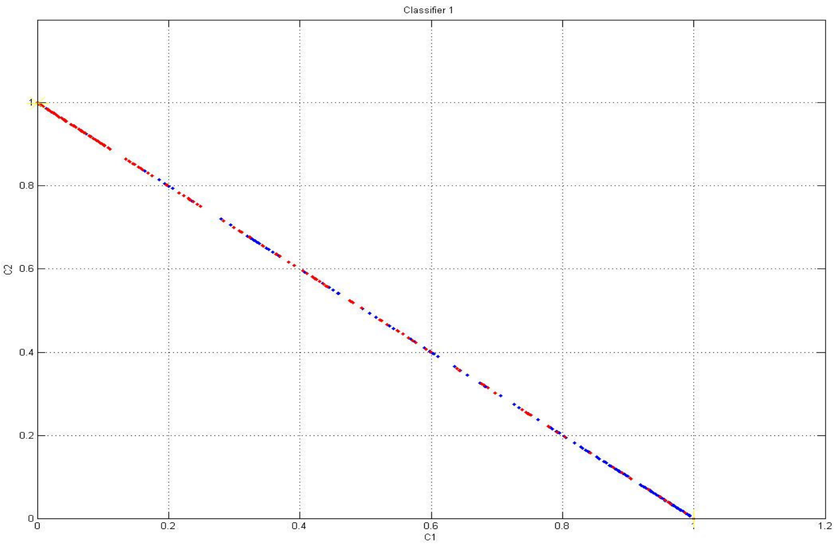
	Class 1	**Class 2**	**Overall Performance**
Classifier 1	80.0%	82.8%	81.4%
Classifier 2	83.2%	84.0%	83.6%
Classifier 3	69.2%	69.6%	72.8%
Meta-Classifier	87.6%	90.8%	89.2%

In Table 6.2, it is observed that the classification performance of meta-classifier is approximately 9% better than that of the individual classifiers. Therefore, we can infer that different samples are classified by different classifiers and provide complementary information on the samples to the meta classifier.

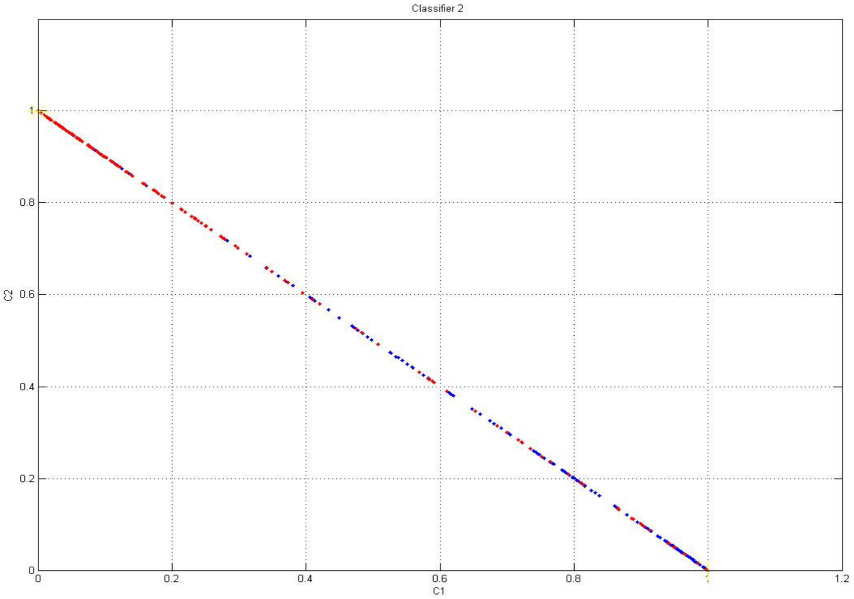
In addition, meta classifier perform 10% better for Class 1 and 12% for Class 2, in average. Therefore, it can be stated that the number of samples from class 2 which are classified by different classifiers are more than the samples from class 1.

In the experiments, 497 samples are classified by at least one classifier, 249 samples from the first class, and 248 samples from the second class. In addition, 247 samples are classified by both of the classifiers.

In Figure 6.9, the membership vectors obtained from each classifier for the test sets are visualized. The samples from the first class are represented by blue, from the second class are represented by red. In Figure 6.10, the samples from class 1 and class 2, which are correctly classified by at least one classifier, $S = \bigcup_{k=1}^{3} \hat{S}_k$, are visualized and represented by black and green, respectively. The vertices where the correct labels of the classes take place are represented by yellow markers.

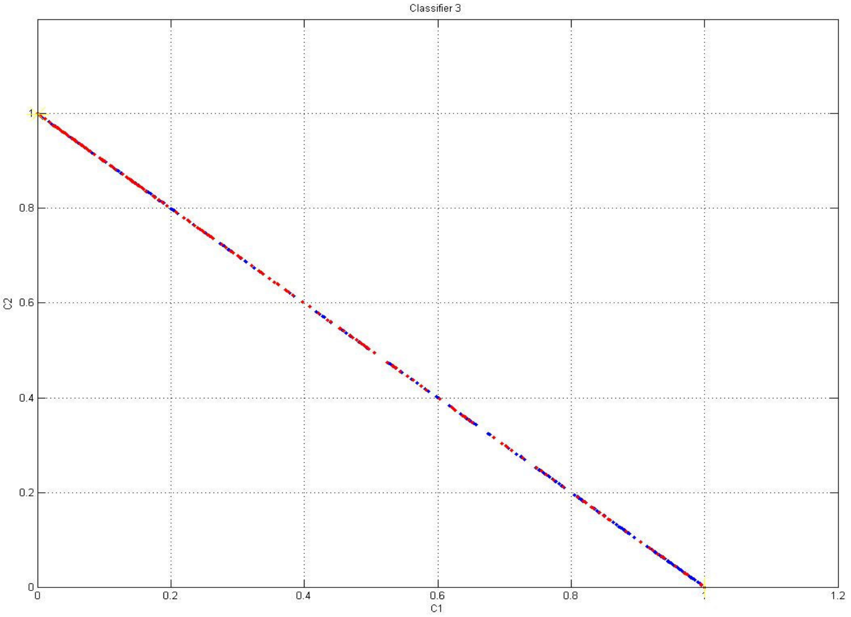


(a)



(b)

Figure 6.9: Membership vectors obtained from **(a)** Classifier 1, **(b)** Classifier 2, **(c)** Classifier 3

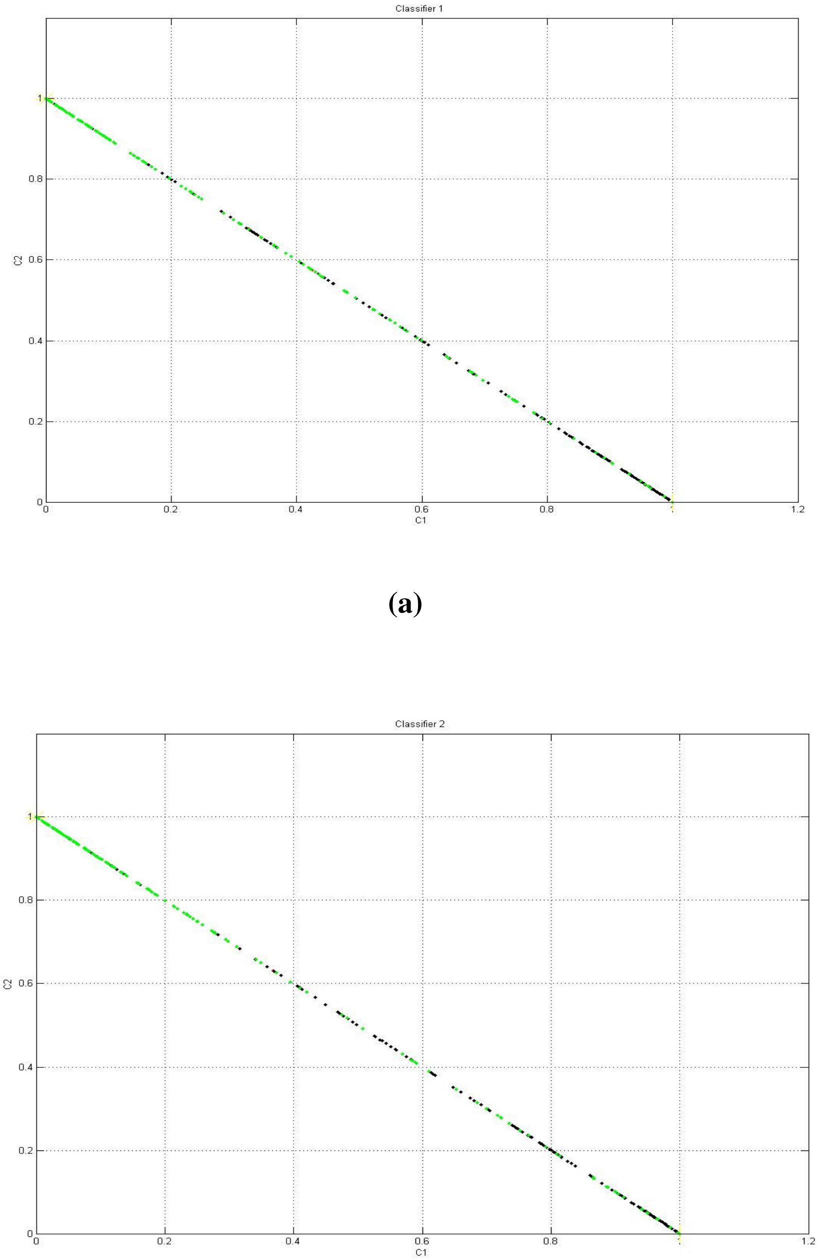


(c)

Figure 6.9 (Continued)

In Figure 6.9-b**,** the samples are well separated through the end points of the line relative to the other sub-figures. Therefore, we can conclude that Classifier 2 performs better than the other classifiers. However, the samples are mostly overlapped in Figure 6.9-c shows that classifier three has the worst classification performance.

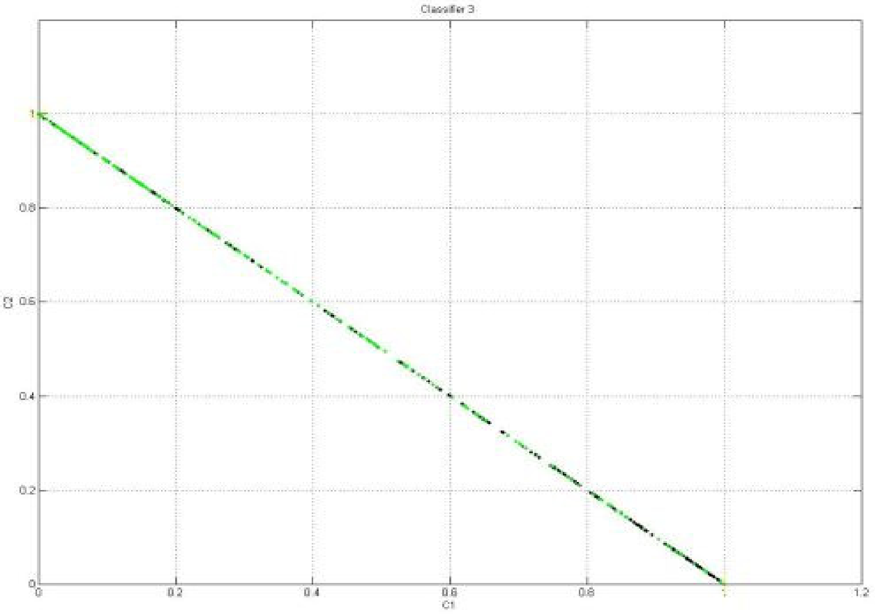
Furthermore, the red points, which represent the first class, scatter through the blue points, which represent the second class, mostly. On the other hand, the blue points scatter up to the middle of the line. Consequently, the samples from the first class causes a decrease in the classification performance more than the samples from the second class, and also affect the classification ability of the samples from the second class. Hence, if the samples from the first class can be separated from the samples of the second class in a higher dimensional feature space, the increase in the classification performance of the samples from the second class would be higher.



(a)

(b)

Figure 6.10: Membership vectors obtained from **(a)** Classifier 1, **(b)** Classifier 2, **(c)** Classifier 3, of the samples classified by at least one classifier**.**



(c)

Figure 6.10 (Continued)

In Figure 6.11, the concatenated membership vectors are visualized including the samples which are classified by at least one classifier.

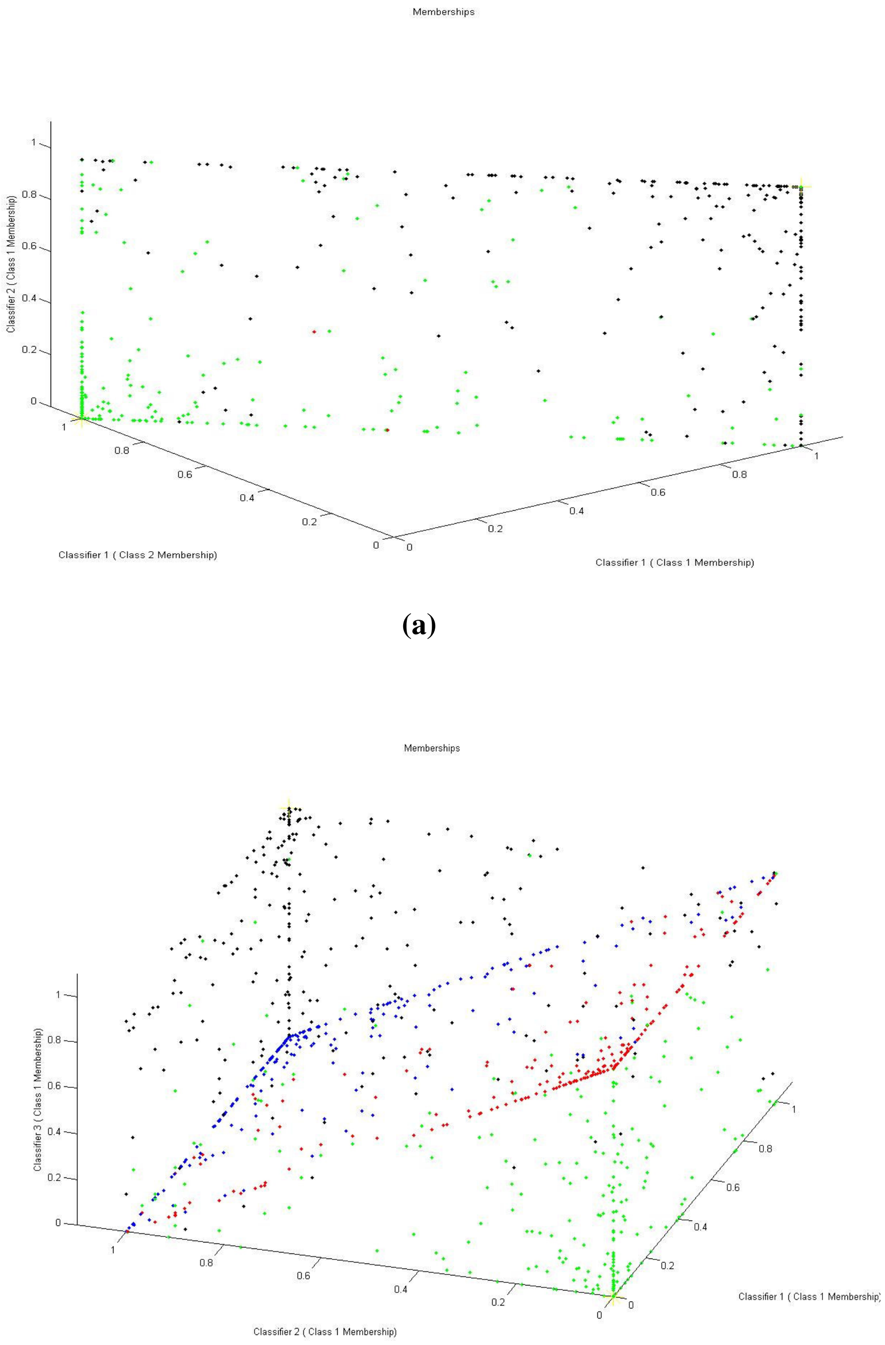
Since the first 2 classifiers perform with higher classification performances, it is observed that most of the samples are scattered through the two dimensional plane described by the predictions of Classifier 1 and Classifier 2 in Figure 6.11-a. Meanwhile, it is considered that the samples from $\hat{S}$ accumulate towards the vertices of the cube.

On the other hand, since $\hat{S}_2 < \hat{S}_1$ and $\hat{S}_2 < \hat{S}_3$, the samples are scattered through different planes in Figure 6.11-b. While, the samples from $\hat{S}$ scatter towards the vertices and edges, and reside on the surfaces of the cube, the remaining samples reside on the diagonal surface of the cube, towards the edges at the inverse of the edges which represent the actual class labels. The reason of the diagonal distribution is that most of the misclassified samples belong to the first class and are affected by the predictions of the third classifier, which performs worse, that lies on the z-axis and causes the diagonalization of the distribution through bottom-up directions.

In Figure 6.11-c and Figure 6.11-d, it is observed that most of the samples from $\hat{S}$ accumulate towards the edges and scatter through the faces smoothly. Since the basis spaces in these figures are constructed by the membership vectors that contain the membership values belonging to the second class and there are relatively less samples which are classified by none of the classifiers in the figures, we observe that most of the misclassified samples belong to the first class.

In Figure 6.11-f, the samples from $\hat{S}$ are scattered on the diagonal plane of the cube. Since the diagonalization is through the basis vectors described by the membership vectors obtained from the second classifier, we can state that the second classifier form the topology of the distribution of the samples by having the higher classification performance.

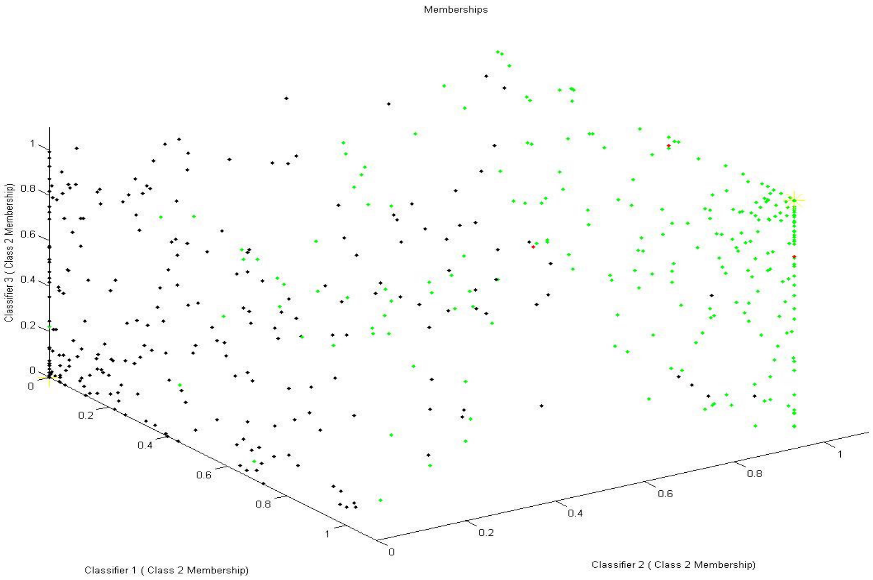
In order to visualize the affect of concatenation of the feature vector in meta layer, we take various combinations of classifiers and class memberships. For this purpose, we design Figure 6.11, throughout the concatenated membership vectors with the axis (a) Classifier 1 - Class 1 Membership ($\mu_1(x_{i,1})$),Classifier 1 - Class 2 Membership ($\mu_2(x_{i,1})$), Classifier 2 - Class 1 Membership ($\mu_1(x_{i,2})$), (b) Classifier 1 - Class 1 Membership ($\mu_1(x_{i,1})$), Classifier 2 - Class 2 Membership ($\mu_2(x_{i,2})$), Classifier 3 - Class 1 Membership ($\mu_1(x_{i,3})$), (c) Classifier 1 - Class 2 Membership ($\mu_2(x_{i,1})$),Classifier 2 - Class 2 Membership ($\mu_2(x_{i,2})$), Classifier 3 - Class 2 Membership ($\mu_2(x_{i,3})$), (d) Classifier 1- Class 1 Membership ($\mu_1(x_{i,1})$), Classifier 2 - Class 2 Membership ($\mu_2(x_{i,2})$), Classifier 3 - Class 2 Membership ($\mu_2(x_{i,3})$), (e) Classifier 1 - Class 1 Membership ($\mu_1(x_{i,1})$), Classifier 2 - Class 1 Membership ($\mu_1(x_{i,2})$), Classifier 2 - Class 2 Membership ($\mu_2(x_{i,2})$), are visualized.



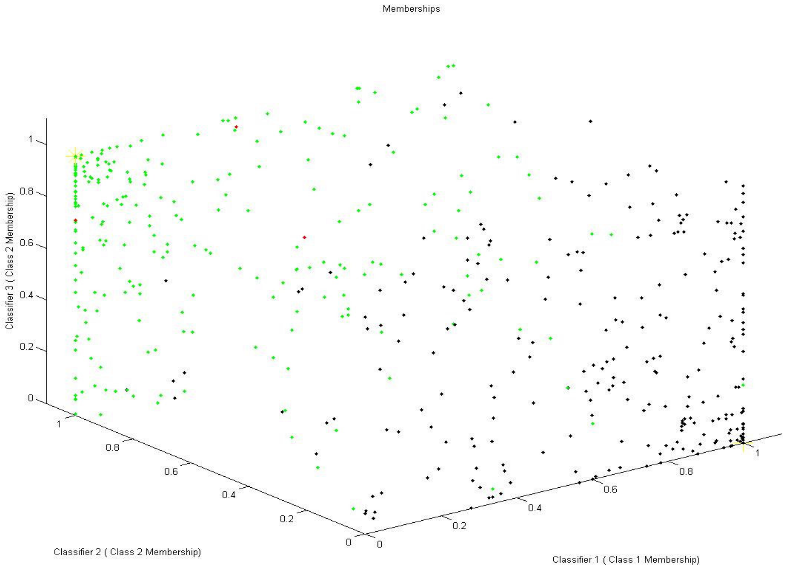
(a)

(b)

Figure 6.11: The relationship among **(a)** $(\mu_1(x_{i,1}))$, $(\mu_2(x_{i,1}))$,$(\mu_1(x_{i,2}))$, **(b)** $(\mu_1(x_{i,1}))$, $(\mu_2(x_{i,2}))$, $(\mu_1(x_{i,3}))$, **(c)** $(\mu_2(x_{i,1}))$, $(\mu_2(x_{i,2}))$, $(\mu_2(x_{i,3}))$, **(d)** $(\mu_1(x_{i,1}))$, $(\mu_2(x_{i,2}))$, $(\mu_2(x_{i,3}))$, **(e)** $(\mu_1(x_{i,1}))$, $(\mu_1(x_{i,2}))$, $(\mu_2(x_{i,2}))$ are visualized.

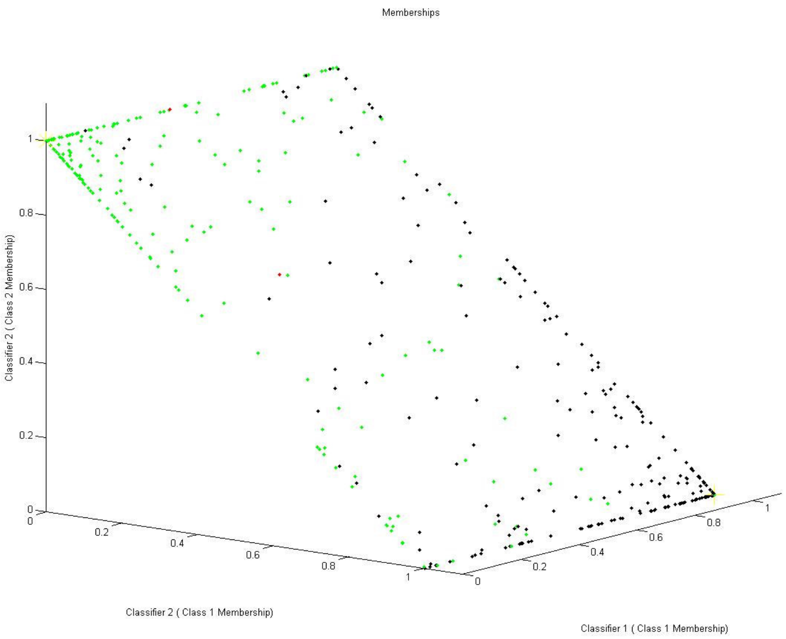


(c)



(d)

Figure 6.11 (Continued)



(e)

Figure 6.11 (Continued)

As can be observed from the experiments, the concatenation operations over the membership vectors sort the samples through the edges of the polygons, in a linearly separable space provided that a sample is correctly classified by at least one classifier. In the experiments, it is observed that as the dimension of the samples increase by expanding the feature space, the samples tend to accumulate around the solution points, which are vertices, of the polygons. Therefore, the tendency increases as the dimension by complementary feature spaces, and the performance increases, relatively.

Note that concatenation of the memberships vectors at the output of the base layer classifiers, transforms the feature spaces of the input of the individual classifiers to a fixed *CK* dimensional vector space, where the samples which are correctly classified by at least one classifier lies on hyper-lines.

6.3.3 The Functional Analysis of Meta-layer Concatenation

In Chapter 2, one of the fundamental problems of the pattern recognition is discussed by the minimization of the error function,

$$error = \sum_{i=1}^{N} \left\| \hat{y}_{i,k} - y_i \right\|^2 \qquad \text{(Equation 6.31)}$$

Where $\hat{y}_{i,k} = h(\underline{x}_{i,k})$ is the estimated label of the actual label y_i for the sample $\underline{x}_{i,k}$, through the hypothesis function h_k. Additionally, the classification performance is defined based on the hypothesis function using equation (2.7).

In Section 6.3.1, *Theorem 1* states that the concatenation process provides solutions for the linear discrimination equation constructed by the concatenated matrixes and in section 6.3.2, *Theorem 1* is examined experimentally, on both synthetic and Corel datasets. In other words, it is observed that the concatenation process separate the feature space linearly under the conditions defined by *Theorem 1.*

Moreover, in Section 6.3.2, it is observed that, as the dimension of the feature space increases during the concatenation process, the samples tend to approach to the vertices where the actual labels of the samples reside and decrease the error function described in equation (6.13).

One of the conditions for increasing the performance of SG is to expand the feature space with additional column spaces that will decrease the margin to the vertices and the lines where the samples lay, which are described in the previous section. Following *Theorem 1,* the second condition is that as the feature space increases by the increasing the number of columns (expanding the column space), the label space should also be expanded such that the label space will lay in the span of the column space of the features.

The proposed conditions are consistent with *Hypothesis 1* , which state that, as the column space is increased by the way that the margins to the edges where the correctly classified samples lay decrease, the general performance of SG increases, in other words, the classification performance on the concatenated feature space and the error function decreases.

Lemma 2: If the set of feature vectors G_l is correctly classified by each classifier Υ_k, $\exists\, k$, mutually, then G_l is G_k. Therefore, $l = k$, $\gamma_l = \Upsilon_k$, $K' = K$ and $\varepsilon_l = \varepsilon_k$.

Definition 6:

The transpose of the membership matrix $\mathrm{M}(s_i) = \left[\underline{\mu}(\underline{x}_{i,1}) \ldots \underline{\mu}(\underline{x}_{i,k})\right]$, $k = 1,2,...,K$, $i = 1,2,...,N$ samples, is defined by M^T, such that, $\mathrm{M}^T(s_i) = \left[\underline{\mu}^T(\underline{x}_{i,1}) \ldots \underline{\mu}^T(\underline{x}_{i,k})\right]$.

Theorem 3 (Linear Separability): In a 2-layer Homogenous Stacked Generalization architecture consisting of C-classes and K-classifiers, fed by distinct K-descriptors; if $\underline{\mu}(\underline{x}_{i,k})$ obtained from each classifier γ_k for the feature vectors $\underline{x}_{i,k}$ of samples $s_i \in G_k$, is merged to the concatenated membership matrix space, then the space becomes linearly separable.

Proof:

By considering M^T, if $\underline{\mu}(\underline{x}_{i,k}) \geq 0.5$, and $\mu_c(\underline{x}_{i,k}) = \max(\underline{\mu}(\underline{x}_{i,k}))$ the sample s_i is assigned to ω_c, for *c=1,2,…,C* classes, and if its actual class labels is ω_c, than it is correctly classified. Therefore, the correctly classification of the sample $\underline{x}_{i,k}$ assures that

$$\underline{\mu}(\underline{x}_{i,k}) \geq 0.5 \qquad \text{(Equation 6.32)}$$

Since $\sum_{k=1}^{K} \underline{\mu}(\underline{x}_{i,k}) = K$, concatenation of each membership vector of $\underline{x}_{i,k}$, discriminates the space by the equation $\sum_{k=1}^{K} \underline{\mu}(\underline{x}_{i,k}) = K$, then,

$$\sum_{k=1}^{K} \underline{\mu}(\underline{x}_{i,k}) = K + 0.5 \qquad , \qquad \text{(Equation 6.33)}$$

and

$$\sum_{k=1}^{K} \underline{\mu}(\underline{x}_{i,k}) > K \qquad . \qquad \text{(Equation 6.34)}$$

So, $\underline{x}_{i,k}$ is assigned to ω_c.

As a result, merging the concatenated membership vector of $\underline{x}_{i,k}$ linearly separates the sample $\underline{x}_{i,k}$ in the concatenated space and, hence, the space is linearly separated $\forall i$, $i = 1,2,...,N$.

Theorem 4 (Performance Relation): In a 2-layer Homogenous Stacked Generalization architecture consisting of C-classes and K-classifiers, fed by distinct K-descriptors; if the classification is implemented on a group of samples $s_i \in G_k$, then, $Perf(SG) \geq Perf(\gamma_k)$, $\forall$ k .

Proof:

Following equation (4.4), $Perf(\gamma_k)$ is,

$$Perf(\gamma_k) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{y}'_{i,k}}(Y_k) \qquad \text{(Equation 6.35)}$$

,where $\underline{\mu}(\underline{x}'_{i,k}) = h(\underline{x}'_{i,k})$ is the membership vector for test sample $\underline{x}'_{\mathrm{i,k}}$ by h_k of γ_k and $\hat{y}'_{i,k} = \max(\underline{\mu}(\underline{x}'_{i,k}))$ is the estimated class label value.

At the meta layer, $\mathrm{M}'(k)$,which is the meta layer membership matrix for test data ,consists of the estimated membership values of $\underline{x}'_{\mathrm{i,k}}$ by h_k obtained from k classifiers, such that,

$$\begin{aligned} \mathrm{M}'(k) &= \left[\underline{\mu}(\underline{x}'_{i,1})\ \underline{\mu}(\underline{x}'_{i,2})\ .\ .\ .\ \underline{\mu}(\underline{x}'_{i,k})\right] \\ &= \left[h(\underline{x}'_{i,1})\ h(\underline{x}'_{i,2})\ .\ .\ .\ h(\underline{x}'_{i,k})\right] \end{aligned} \qquad \text{(Equation 6.36)}$$

,and the classification performance of the meta layer classifier is,

$$Perf(\gamma_{meta}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{y}'_{i,meta}}(Y(k)) \qquad \text{(Equation 6.37)}$$

where

$$\begin{aligned} \hat{y}'_{i,meta} &= \mathrm{M}^{'}(k)X(k) \\ &= \left[h(\underline{x}'_{i,1})\ h(\underline{x}'_{i,2})\ .\ .\ .\ h(\underline{x}'_{i,k})\right]X(k) \end{aligned} \qquad \text{(Equation 6.38)}$$

is the estimated class label of s_i at the meta layer.

At the base classifiers, while the samples belonging to ω_c, $\forall\ c$, are not correctly classified, $\underline{\mu}(\underline{x}'_{i,k})$ are wrongly estimated and become relatively joint, then $Perf(\gamma_k)$ is low. However, following *Theorem 3*, the concatenation operation linearly separates the meta layer feature space. Therefore, it is obvious that, $Perf(\gamma_{meta})$, which is the classficiation performance of SG, which is $Perf(SG)$, is higher than $Perf(\gamma_k)$.

Theorem 5 (Error Bound): In a 2-layer Homogenous Stacked Generalization architecture consisting of C-classes and K-classifiers, fed by distinct K-descriptors, $Perf(SG)$ increases as $CK \to N$ for the meta-layer feature matrix $\mathrm{M} \in \Re^{NxCK}$, and $e(k)$ reaches to the its optimum error bound ε_{meta}, provided that M consists of the membership vectors obtained from supplementary classifiers γ_k .

Proof:

Let's define the hypothesis function for the k^{th} classifier of the feature vector $\underline{x}_{i,k} \in \Re$ from the feature set $S_k = \{\underline{x}_{i,k}, y_i\}_{i=1}^{N}$, by $h(\underline{x}_{i,k})$, $h(\underline{x}_{i,k}) \in H$, for K classifiers, which is bounded by $0 < h(\underline{x}_{i,k}) < 1$, and for some error function $\varepsilon > 0$, we may state that,

$$\left\| h(\underline{x}_{i,k}) - y_i \right\| < \varepsilon \qquad \text{(Equation 6.39)}$$

where y_i is the actual class label of the feature vector $\underline{x}_{i,k}$.

The classifier γ_k with the hypothesis function $h(G_k)$ satisfies,

$$\left\| h(G_k) - Y_k \right\| < \varepsilon_k \quad , \quad \forall\ \varepsilon_k \geq 0 \qquad \text{(Equation 6.40)}$$

Meanwhile, following *Theorem 2*, since $e(k)>0$and $\mathrm{M}(k)\mathrm{M}(k)^t$is positive semi-definite, $\left\| e(k) \right\|^2 > \left\| e(k+1) \right\|^2$, while $0 < \upsilon(t) < 1$. Therefore, the error function of the meta-layer classifer $e(k)$ is monotonically decreasing, and converges to some limiting value $\varepsilon_{meta} \geq 0$, such that,

$$\left\| \mathrm{M}^{'}(k) X(k) - Y(k) \right\| = e(k) \to \varepsilon_{meta} \qquad \text{(Equation 6.41)}$$

where, $\mathrm{M}'(k)$ is the membership matrix for test data.

As the number of classifiers γ_k and k increases, $k \to K$ and the dimension of the meta-layer feature matrix M increases such that,

$$Ck \to CK \qquad \text{(Equation 6.42)}$$

When *N=CK*, which is an upper bound for *CK*, the matrix M becomes a square matrix by implying the uniqueness and existence of the solution matrix *X* [93] and $e(k) \to \varepsilon_{meta}$.

On the other hand, if $CK \geq N$, the system becomes overdetermined, and perturbs the robustness and the integrity of the feature space.

In *Theorem 6*, the boundary condition for the performance of SG is stated.

Theorem 6 (Performance Condition): In a 2-layer Homogenous Stacked Generalization architecture consisting of C-classes and K-classifiers, fed by distinct K-descriptors; the $Perf(SG)$ is bounded by the percentage of correctly classified samples by at least one classifier, provided that training and test sets are both statistically stable and consistent.

Proof:

Let's consider two sets of samples belonging to two sets of feature vectors, $s_g \in G$, which is the set of feature vectors which are classified by at least one classifier, and $s_{mc} \in MC$, which is the set of feature vectors which can not be classified by at least one classifier, such that, $MC = S - G$.

The previous theorems state that, as the membership vectors of the samples $s_g \in G$ are augmented to the meta-layer feature space, the space becomes linearly separable. However, the membership vectors of s_{mc} damages the linear separability and the samples become inseparable at the feature space and can not be correctly classified by the meta-layer linear classifier.

Therefore, for a finite set of classifiers, $Perf(SG)$ is limited by $|G|$, in other words, limited by the percentage of the samples that can be correctly classified by at least one classifier.

Following the theorems, we can conclude that, for the present architecture, the concatenation operation provides a linearly separable space as the membership vectors belonging to the samples that are classified by at least one classifier are concatenated.

6.4 A new Algorithm for Stacked Generalization: Meta-Fuzzified Yield Values

In this section, following the conclusions and the theorems proposed in the previous sections, we will introduce a meta-layer classification algorithm, which is called Meta-layer Fuzzified Yield Values (Meta-FYV) with a feature space composition-decomposition method.

In *Theorem 1*, we have stated that while the label space of the features at the meta-layer spans the column space of the features, then, a solution for the linear equation $\mathrm{M}X = Y$ can be obtained. In other words, while the matrix Y consists of the elements that are basis of the feature matrix M, then a linear discriminate classifier can be used for the meta-layer classification.

Applying *Theorem 1* for the classification of two layer fuzzy SG architecture consisting of *K* fuzzy k-nn base layer classifiers and *C* classes, and defining the class membership values for training and testing set of features , S_k^{tr} and $S_k\,'$, respectively, obtained from each *k* classifier, $\{\underline{\mu}(\underline{x}^{tr}{}_{i,k})\}_{k=1}^{K}$ and $\{\underline{\mu}(\underline{x}'_{i,k})\}_{k=1}^{K}$, respectively. Then we can define the meta-layer training membership matrix consisting of membership vectors of training dataset, such that, $\mathrm{M}^{tr} = [\underline{\mu}(\underline{x}^{tr}{}_{i,k}) \ . \ . \ . \ \underline{\mu}(\underline{x}^{tr}{}_{i,K})]$ and meta-layer test membership matrix consisting of membership vectors of test dataset, such that, $\mathrm{M}^{te} = [\underline{\mu}(\underline{x}'_{i,1}) \ . \ . \ . \ \underline{\mu}(\underline{x}'_{i,K})]$. Then, we state the Meta-FYV-1 and Meta-FYV-2 algorithms.

Algorithm 6. 1: Meta-FYV-1 algorithm

```
for $k=1,2,\ldots,K$
    Calculate $\underline{\mu}(\underline{x}^{tr}_{i,k})$ and $\underline{\mu}(\underline{x}'_{i,k})$
end
    Concatenate $\underline{\mu}(\underline{x}^{tr}_{i,k})$, such that, $\mathrm{M}^{tr} = [\underline{\mu}(\underline{x}^{tr}_{i,k}) \ldots \underline{\mu}(\underline{x}^{tr}_{i,K})]$
    Concatenate $\underline{\mu}_{te}(\underline{x}'_{i,k})$, such that, $\mathrm{M}^{te} = [\underline{\mu}(\underline{x}'_{i,1}) \ldots \underline{\mu}(\underline{x}'_{i,K})]$
for $k=1,2,\ldots,K$
    for $c=1,2,\ldots,C$
        Define the label vector of the training samples, such that,
    while $\underline{x}^{tr}_{i,k}$ belongs to the $\omega_l$,
    $Y_c(\underline{x}^{tr}_{i,k}) = \begin{Bmatrix} 1, \; if \; \underline{x}^{tr}_{i,k} \in \omega_c \\ 0, \; otherwise \end{Bmatrix}$
    end
    Concatenate the label vector for the meta –layer label matrix,
 such that,
 $\underline{Y}(\underline{x}^{tr}_{i,k}) = \left[ Y_1(\underline{x}^{tr}_{i,k}) \ldots Y_c(\underline{x}^{tr}_{i,k}) \ldots Y_C(\underline{x}^{tr}_{i,k}) \right]$
end
    Concatenate $Y(\underline{x}_{i,k})$, such that, $Y^{tr} = \left[ \underline{Y}(\underline{x}^{tr}_{i,1}) \ldots \underline{Y}(\underline{x}^{tr}_{i,K}) \right]$,
 $k=1,2,\ldots,K$
    Solve the equation $\mathrm{M}^{tr} X^{tr} = Y^{tr}$ by $X^{tr} = pinv(\mathrm{M}^{tr}) Y^{tr}$, where
 $pinv(\mathrm{M}^{tr})$ is the pseudo-inverse of $\mathrm{M}^{tr}$
    Calculate $Y^{te}$ by, $Y^{te} = \mathrm{M}^{te} X^{tr}$,
    Decompose $Y^{te} = \left[ \underline{Y}(\underline{x}'_{i,1}) \ldots \underline{Y}(\underline{x}'_{i,K}) \right]$ by separating into the
 column vectors $\underline{Y}(\underline{x}'_{i,k})$, $k=1,2,\ldots,K$
    Apply majority voting over the vectors $\underline{Y}(\underline{x}'_{i,k})$, that is, calculate
 the maximum values of the each sample over row-space, such that,
        if $\omega_l = \max(\underline{Y}(\underline{x}'_{i,k}))$, $\underline{x}'_{i,k}$ belongs to the $\omega_l$ class.
```

In Meta-FYV-1 algorithm (Algorithm 6.1), firstly, the membership vectors are obtained from the base layer classifiers, for both training and test datasets. Then, the membership vectors are concatenated in order to construct the meta-layer feature matrix M^{tr} and M^{te}.

Then, the class labels are represented as the base vectors in consistent with the base-layer membership vectors by applying a new representation algorithm (Step 8). In the representation, the samples are coded with the Boolean values relative to the classes which they belong to. In the composed label matrix Y^{tr}, the row vectors represent the samples and the column vectors represent the classes. In the next step, the label vector that span the membership space of each classifier is concatenated in correspondence with meta-layer feature space in order to completely span the meta-layer feature space.

After constructing the membership and the label matrixes, $\mathrm{M}^{tr} X^{tr} = Y^{tr}$ is solved. The solution matrix X^{tr} which is obtained from the training data is, then, applied to the membership matrix of the test data. The resulting label matrix Y^{te} is decomposed to its column vectors. It should be noticed that, the column vectors may be considered as the predictions of different classifiers in the decomposed label matrix. Therefore, majority voting is applied on each vector in order to obtain class label of each sample.

In Algorithm 6.2, the concatenation of the label matrix is not applied since the label matrix corresponding to different classifiers are equal and sufficient to represent the basis vectors that span the space of the meta-layer feature matrix, M^{tr}, which is the concatenated membership matrix, different from Algorithm 6.2.

Algorithm 6. 2: Meta-FYV-2 Algorithm

```
for $k=1,2,\ldots,K$
    Calculate $\underline{\mu}(\underline{x}^{tr}{}_{i,k})$ and $\underline{\mu}(\underline{x}'_{i,k})$
end
    Concatenate $\underline{\mu}(\underline{x}^{tr}{}_{i,k})$, such that, $\mathrm{M}^{tr}=[\underline{\mu}(\underline{x}^{tr}{}_{i,k})\ldots\underline{\mu}(\underline{x}^{tr}{}_{i,K})]$
    Concatenate $\underline{\mu}(\underline{x}'_{i,k})$, such that, $\mathrm{M}^{te}=[\underline{\mu}(\underline{x}'_{i,1})\ldots\underline{\mu}(\underline{x}'_{i,K})]$
for $k=1,2,\ldots,K$
    for $c=1,2,\ldots,C$
        Define the label vector of the training samples, such that,
        while $\underline{x}^{tr}{}_{i,k}$ belongs to the $\omega_l$,
        $Y_c(\underline{x}^{tr}{}_{i,k})=\begin{Bmatrix}1, & if\ \underline{x}^{tr}{}_{i,k}\in\omega_c\\ 0, & otherwise\end{Bmatrix}$
    end
    Concatenate the label vector for the meta –layer label matrix, such that,
    $\underline{Y}(\underline{x}^{tr}{}_{i,k})=\left[Y_1(\underline{x}^{tr}{}_{i,k})\ldots Y_c(\underline{x}^{tr}{}_{i,k})\ldots Y_C(\underline{x}^{tr}{}_{i,k})\right]$
end
    Assign $\underline{Y}(\underline{x}^{tr}{}_{i,k})$ to $Y^{tr}$, such that, $Y^{tr}=\underline{Y}(\underline{x}^{tr}{}_{i,k})$,
    since $\underline{Y}(\underline{x}^{tr}{}_{i,k})=\underline{Y}(\underline{x}^{tr}{}_{i,j})$,      for any $k\neq j$
    Solve the equation $\mathrm{M}^{tr}X^{tr}=Y^{tr}$ by $X^{tr}=pinv(\mathrm{M}^{tr})Y^{tr}$,
    where $pinv(\mathrm{M}^{tr})$ is the pseudo-inverse of $\mathrm{M}^{tr}$
    Calculate $Y^{te}$ by, $Y^{te}=\mathrm{M}^{te}X^{tr}$.
    Apply majority voting over the vectors $\underline{Y}(\underline{x}'_{i,k})$, that is calculate the
maximum values of the each sample over row-space, such that,
        if $\omega_l=\max(\underline{Y}(\underline{x}'_{i,k}))$, $\underline{x}'_{i,k}$ belongs to the $\omega_l$ class
```

In Table 6.3, Table 6.4 and Table 6.5, the experiments implemented on 4 to 8 descriptor with different set of classes are introduced. The performances of Fuzzy SG algorithms, Meta-FYV-1 algorithm, Meta-FYV-2 algorithm and the performance gain of Meta-FYV-2 over Fuzzy SG are provided in the tables.

In the first set of the experiments, New Guinea, Beach, Rome, Bus, Dinosaurs, Elephant, Roses, Horses, Mountain, and Dining classes are classified using the feature sets obtained by 4 (Color Structure , Color Layout, Edge Histogram, Region-based Shape), 5 (Color Structure , Color Layout, Edge Histogram, Region-based Shape, Haar) , 6 (Color Structure , Color Layout, Edge Histogram, Region-based Shape, Haar, Dominant Color) , 7 (Color Structure , Color Layout, Edge Histogram, Region-based Shape, Haar, Dominant Color, Scalable Color), and 8 (Color Structure , Color Layout, Edge Histogram, Region-based Shape, Haar, Dominant Color, Scalable Color, Homogenous Texture) of the MPEG 7 descriptors (Table 6.3) .

In the second set of the experiments, 5 more classes, Autumn, Bhutan, California Sea, Canada Sea, and Canada West classes are added to the data set. In that case, the performances of 5-8 descriptor experiments are provided in the (Table 6.4)

In the third set of the experiments, 5 more classes; China, Croatia, Death Valley, Dogs and England are added to the present classes (Table 6.5).

Table 6.3: Performances of 10-Class Experiments

10-Class Experiments	**Fuzzy SG**	**Meta-FYV-1**	**Meta-FYV-2**	**Performance Gain** ***(Fuzzy SG and Meta-FYV2)***
4 Descriptors	85.6%	80.4%	88.0%	2.4%
5 Descriptors	86.8 %	81.2%	88.6%	1.8%
6 Descriptors	85.6%	80.4%	87.4%	1.8%
7 Descriptors	85.8%	80.8%	88.2 %	2.4%
8 Descriptors	86.4%	81.6%	89.0.%	2.6%

In 10 class experiments, Meta-FYV-1 performed worse than Fuzzy SG algorithm, however, Meta-FYV-2 perform better than the Fuzzy SG algorithm. In addition Meta-FYV-2 algorithm provided 2.2% performance gain over Fuzzy SG, in average. It is also observed that as the number of descriptors (classifiers) increases, the classification performance and the relative performance gain increase.

Table 6.4: Performances of 15-Class Experiments

15-Class Experiments	**Fuzzy SG**	**Meta-FYV-1**	**Meta-FYV-2**	**Performance Gain** ***(Fuzzy SG and Meta-FYV2)***
5 Descriptors	65.3%	66.4%	69.5%	4.2%
6 Descriptors	62.3 %	65.6%	68.3%	6.0%
7 Descriptors	62.8%	65.7%	68.5%	5.7%
8 Descriptors	64.5%	66.3%	69.1 %	4.7%

In 15-Class experiments, both algorithms performed better than Fuzzy SG. In addition, the relative performance gain of Meta-FYV-2 over Fuzzy-SG is increased by 5.5%. The better performance gain is obtained from 6-Descriptor experiments.

Table 6.5: Performances of 20-Class Experiments

20-Class Experiments	**Fuzzy SG**	**Meta-FYV-1**	**Meta-FYV-2**	**Performance Gain** ***(Fuzzy SG and Meta-FYV2)***
4 Descriptors	52.4%	54.8%	57.5%	5.1%
5 Descriptors	50.7%	54.1%	56.2%	5.5%
6 Descriptors	49.9%	53.8%	55.8.%	5.9%
7 Descriptors	50.9%	54.4%	56.0%	5.1%
8 Descriptors	52.9%	54.4%	56.5%	3.6%

In 20-Class experiments, Meta-FYV-2 performed better than Fuzzy SG with 5% performance gain. The performance of 5-Descriptor experiment increases while the others decrease, concluding that the additional classes are not well described by Dominant Color, Scalable Color and Homogenous Texture descriptors. In addition, the Meta-FYV-1 provided 3% performance gain over Fuzzy SG.

In addition, the algorithms are tested over randomly selected 50 classes from Corel dataset with 8 descriptors which are introduced above. Fuzzy SG provided 47.0% classification performance while Meta-FYV-1 provided 51.2% performance and Meta-FYV-2 provided 52.2% performance.

One of the reasons of the differences between the performance gains of Meta-FYV-1 and Meta-FYV-2 is that the additional concatenation operation of the label matrix in Meta-FYV-1 requires additional majority voting operation, and the majority voting could not succeed at the prediction of the class labels.

Another reason of the performance difference is that as the dimension of Y^{tr} increases, the dimension of the solution matrix X^{tr} and Y^{te} increases. Therefore, the majority voting for the matrix decomposition becomes an inadequate and insufficient method for high dimensional matrix decomposition.

The reason of the failure of Fuzzy SG is that the concatenation operations on the membership vectors construct a relatively linearly separable feature space at meta-layer input space (*Theorem 3*). Therefore, linear classifiers perform better than the non-parametric classifier, fuzzy k-nn. In addition, *Theorem 4* introduced in the promises a solution for the constructed linear equation via corresponding composition-decomposition technique.

Meanwhile, there exists a classification performance limit that can be achieved in Stacked Generalization algorithm introduced by *Theorem 5* and *Theorem 6.* Therefore, the limit should be analyzed over the base layer classifiers and should be considered in the design of the architecture.

It should be noticed that as the number of samples that can be correctly classified by at least one classifier increases, in other words, and as the number of classifiers that can correctly classify the samples provided by the descriptors increases, the performance of SG increases. In that case, the dimensions of the concatenated membership vector M^{tr} and the label vector Y^{tr} increase. It should be remarked that, in order to recover the under-determination of the linear equation, the solution matrix should have the dimension as the number of samples which require that the dimension of Y^{tr} should be the number of samples. Therefore, as the dimension of Y^{tr} goes to the number of samples, that is, as $CK \rightarrow N$, the performance reaches to its limit.

6.5 Discussion

In the present chapter, we investigate the concatenation operation in classification problems, which is a *black box* technique for feature space fusion of pattern recognition.

In Section 6.2, the concatenation problem for the pattern recognition is analyzed. In Section 6.3, the concatenation operation is analyzed from three perspectives; linear system analysis, geometrical analysis, and functional analysis.

In the analysis of the concatenation by linear systems, the theorem which states the existence of the solutions for the concatenated feature spaces, is introduced, namely, *Theorem 1.* Following *Theorem 1,* the ratio of error decrease at the meta-layer feature space is stated by *Theorem 2.*

In the geometrical analysis of the concatenation, the structures of the spaces and the transformations between the spaces are visualized and *Theorem 1* is verified by the experiments implemented on synthetic and Corel Draw datasets.

In the functional analysis, the concatenation is investigated in terms of error functions and hypothesis functions by introducing *Theorem 3,* which states that the space of the concatenated membership vectors is linearly separable. *Theorem 4* states the performance relations between base layer classifiers and the meta-layer classifier by proving the hypothesis proposed in Chapter 4. *Theorem 5* states the performance limit of SG and *Theorem 6* states the conditions in order to obtain higher performances in two layer fuzzy SG architectures.

In Section 6.4, the analysis and the theorems are applied to the classification problems in two-layer fuzzy SG architecture and a new classification algorithm, which is Meta-FYV, is introduced with its two variations.

CHAPTER 7

EPILOGUE

“The computers are useless, they can only give answers”

Pablo Picasso

7.1 Observations and Interpretations

This study investigates the conditions of the general improvements for the performances of SG classifiers by analyzing the behavior of the individual classifiers to learn the data. For this purpose, a great variety of experiments are performed on both real and synthetically generated data sets. The experiments are restricted to control the critical parameters of the SG architecture, which directly and significantly affect the overall performance.

In the first group of experiment, the synthetic data is generated in such a way that the samples can be labeled by at least one classifier correctly, at the base layer. It is observed that if one assures this condition, the classification performance of SG is significantly higher than that of the individual classifier performances. This observation shows that the performance of the SG architecture depends on the share of detection of the samples rather than the performance of individual classifiers. It is well known that high individual classification performances are practically not possible to achieve, especially, when the class numbers are high. However, SG allows us to reach a substantially high performance even if the performances of the individual classifiers are rather low. This high performance is attributed to the following factors

i) The ability of SG to share the correct-labeling of the samples among the classifiers at the base layer.

ii) The ability of meta-layer classifier to learn the mistakes of the base layer classifiers.

In the next sets of the synthetic data experiments, the results show that there is a nonlinear sigmoid relationship between the performances of the base layer classifiers and that of the overall performance of SG. The parameters of the sigmoid function, *k, l, m* and *n* depend on the properties of training data and need to be explored further. Since the observations can only be made in the restricted experiments, the generalization of the sigmoid function is highly difficult. Furthermore, the nonlinearity of SG architecture complicates the problem of estimating the above mentioned parameters.

Another problem in SG is the complexity of the controlled data set construction. As the dimension of the data sets increases, the alternative values of mean value vectors and covariance matrices increase, the degree of freedom in the Euclidean space increases, resulting a decrease in the degree of control in the experimentation.

In the second part of the experiments, the relation between the behavior of the training data at the base layer classifiers and the performance of the SG is investigated. In order to obtain a meta-layer input data set, which consists of the well-separated samples, the samples in the training set that could not be correctly labeled by at least one individual classifier are eliminated from the feature space.

It is observed that, as the number of classifiers increase, the number of the samples to be eliminated decreases. As we add more classifiers at the base layer, the space become more and more linearly separable. If we have sufficient number of classifiers, it is observed that at least one individual classifier can label a sample correctly in the base layer and the clustered samples can preserve the topology in the meta-layer input space by concatenation, in some manner. However, the preservation is not perfect since still some lacks of performance is observed. The characteristics of the mapping that will perform with the perfect preservation is still an open and unsolved problem, which is one of the black art problems.

In the experiments, construction of meta-layer input space by eliminating misclassified samples from the base layer output space, improves the classification performance of SG. However, one may employ different methods to deal with the violation of hypothesis 1 and 2. For example, one may construct the base layer classifiers in such a way so that the Hypothesis 1 is satisfied as much as possible. Therefore, Hypothesis 1 provides a sufficient condition for the improvement of the overall performance of SG. Implementation to assure this condition is yet a separate issue.

7.2 Imminent Route: Dialogue

In the present work, the analyses of the structure and behavior of the two layer fuzzy SG architecture have been established. The result of the analyses reveals new conjunctures and the conjectures of the architecture have been introduced.

During the theoretical investigations of SG structure, we have noticed the relations between the base layer feature space and the classification performance of the architecture and, also, between the performances of the individual classifiers and that of the general SG architecture. We formulate these relationships as two *black art* problems of SG.

The proposed complementary hypotheses state rigorous explanations on the architecture as the milestones on the pathway of the development and the consideration of new Stacked Generalization architectures. In the feature work, the hypothesis will be enforced with the supplementary hypothesis in order to construct a theoretical framework for Stacked Generalization architecture.

Another conjecture acquired from the theoretical and the experimental studies is the sigmoid function that represents the relationship between the base layer feature space and the performance of the architecture. Although its controllable parameters, in other words, the degree of freedom, is high, the studies that may provide a solution to the sigmoid equation would provide very imperative and essential information on the architecture. In addition, the equation would present a new sight on the feature selection paradigm, which is another *black art* of pattern recognition.

In this study, the *black art* problems of SG are extended beyond the definitions of Wolpert, Witten and Tang. In that perspective, new challenging conjunctures of SG are introduced to the pattern recognition community that may lead to new approaches in order to consider and develop the ensemble learning and data fusion architectures.

On the other perspective, the work on the future pathway would be accomplished under the new learning paradigm where learning can be formulated as uniting two available information to create “new” information. That is why we symbolized the concept of learning by “love”, saying “learning is love”. Extending this paradigm through the available tools that are provided by the other fields of science to the general argument of the Universe, not only the pattern recognition paradigms but also most of the, may be all of the, new born “babies” of computer science, would be stated in more robust platforms. Additionally, this would enable computer science to be considered as the big brother of mathematics, which is reflection and the production of human brain, with an additional power, which is the ability of the unlimited creativity and implementation process.

Moreover, the development will be two-fold, in other words, the development of the ideas and the paradigms of computer science, would enable the development of the corresponding models in the other fields, such as the developments of Quantum Information Theory and Quantum Computation Theory.

The only concern and the conjuncture of that interpretation is that it is based on the meaning of the process, not on the result. Therefore, the work that will be realized should take care of that consideration. Using the tools of science obtained from other grandparents and the parents without any analyses of the process, but focusing on the results, with the natural instincts that take the ego under control to achieve the results, would be nothing but the waste of time.

In conclusion, following the light of Athena, the men would focus on the process without the concern of accomplishing the results, and would place near Athena at Artemis. The *Artist* enlighten that in order to satisfy the desires of the instincts, one should leave out the instincts for a while and feel the delight of information acquisition and creativity, which is *love*, and the ego and the idea would be satisfied with the delight.

REFERENCES

1. Valiant, L. G. (1984). A theory of the learnable. *Communications of ACM, 27(11),* 1134-1142.

2. Stephen Hawking (2002). *The Theory of Everything: The Origin and Fate of the Universe*, New Millennium Press.

3. Steven Weinberg (2005). *The Quantum Theory of Fields, Volume 1: Foundations*. Cambridge University Press.

4. Joseph Polchinski (2005). *String Theory, Volume 1*. Cambridge University Press.

5. Green, M. B., Schwarz J. H., and Witten, E. (1988). *Superstring Theory: Volume 1, Introduction.* Cambridge University Press.

6. Feynman, R., Hey, A., Hey, T., Allen, R. W. (2000). *Feynman Lectures on Computation.* Westview Press.

7. Angluin, D. (1992). Computational learning theory: survey and selected bibliography. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on theory of Computing STOC '92, Victoria, British Columbia, Canada*. (pp. 351-369). ACM, New York, NY.

8. Pitas, Ioannis (1993). *Parallel Algorithms for Digital Image Processing, Computer Vision and Neural Networks*. John Wiley & Sons.

9. Jain, A. K., Duin, R. P. W. and Mao., J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4-37.

10. D. Wolpert. The relationship between pac, the statistical physics framework, the bayesian framework and the vc framework.

11. Sergios Theodoridis, Konstantinos Koutroumbas (2003). *Pattern Recognition,* USA:Elseiver Academic Press.

12. Vladimir Vapnik (2000). The Nature of Statistical Learning Theory, Springer-Verlag, New York, Inc.

13. R. O. Duda, P. E. Hart, and D. G. Stork (2000). *Pattern Classification*. Wiley- Interscience Publication.

14. V. N. Vapnik (1998). *Statistical Learning Theory*. New York: Wiley-Interscience Publication.

15. Ethem, Alpaydin(2004), *Introduction to Machine Learning.* Massachusetts Institute of Technology: MIT Press.

16. A. Engel, and C. Van den Broeck, (2001). Statistical Mechanics of Learning. Cambridge University Press.

17. I. Guyon, A. Elisseeff. (2003). An introduction to variable and feature selection*.* *Journal of machine learning research*, 3(1): 1157-118.

18. Christopher M. Bishop (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

19. David H. Wolpert (1992). How to Deal with Multiple Possible Generalizers in *Fast Learning and Invariant Object Recognition*, Ed. B. Soucek, Wiley and Sons.

20. T. K. Ho (2002). Multiple Classifier Combination: Lessons and the next steps. In A. Kandel and H. Bunke, editors, *Hybrid Methods in Pattern Recognition.* World Scientific Publishing, 171-198.

21. Mutlu Uysal, Emre Akbas, and Fatos T. Yarman-Vural (2006). A hierarchical classification system based on adaptive resonance theory. *ICIP,* 2913-2916.

22. Josef Kittler (2000). A Framework for Classifier Fusion: Is It Still Needed? , *SSPR/SPR, Vol: 1876. LNCS ,45-76,* Springer.

23. Kuncheva L.I. (2002). A theoretical study on six classifier fusion strategies, *IEEE Transactions on PAMI*, **24**, (2), 281-286.

24. K. Tumer, and J. Ghogh (1996). Classifier Combining: Analytical Results and Implications. *Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms, workshop at the Thirteenth National Conference on Artificial Intelligence*, Portland, OR, August.

25. K. Tumer, and J. Ghogh (1999). Linear and Order Statistics Combiners for Pattern Classification A. J. C. Sharkey, editors, *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, 127–162, Springer-Verlag, London.

26. K. Tumer, and J. Ghogh (1996) Error Correlation and Error Reduction in Ensemble Classifiers. *Connection Science*, 8(3/4):385–404.

27. Hu, Y., Park, J., and Knoblock, T. (1997). Committee Pattern Classifiers. In *Proceedings of the 1997 IEEE international Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Volume 4 - Volume 4* (April 21 - 24, 1997). ICASSP. IEEE Computer Society, Washington, DC, 3389.

28. Yu Hen Hu; Knoblock, T.; Jong-Ming Park **(**1997**).** Linear committee pattern classification. Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop Volume , Issue , 24-26 Sep 1997 Page(s):568 - 577

29. L. I. Kuncheva (2000). Fuzzy Classifier Design. Physica-Verlag, Heidelberg.

30. Dietterich, T. G. (2002). Ensemble Learning. In *The Handbook of Brain Theory and Neural Networks, Second edition,* (M.A. Arbib, Ed.), Cambridge, MA: The MIT Press, 2002. 405-408.

31. Ludmila I. Kuncheva (2004). *Combining pattern classifiers: methods and algorithms.* New Jersey:Wiley- Interscience Publication.

32. Thomas Bak, Lecture Notes – Estimation and Sensor Information Fusion, November 14, 2000. Aalborg University, Department of Control Engineering, Denmark.

33. A. Yilmaz (2007). Sensor Fusion in Computer Vision. IEEE GRSS/ISPRS *Joint Workshop on Remote Sensing and Data Fusion over Urban Areas.* 1-5, Paris.

34. M. Ozay, F.T. Yarman Vural (2008). "Yığılmış Genelleme Algoritmalarının Performans Analizi", *Sinyal İşleme ve İletişim Uygulamaları Kurultayı.*

35. M. Ozay, F.T. Yarman-Vural (2008). "On the Performance of Stacked Generalization Architecture", Lecture Notes in Computer Science, ICIAR 445-451.

36. K.M. Ting and I.H. Witten (1999). Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10:271-289.

37. Džeroski, S. and Ženko, B. (2004). Is Combining Classifiers with Stacking Better than Selecting the Best One?. *Mach. Learn.* 54, 3 (Mar. 2004), 255-273.

38. Ueda, N. and Nakano, R. (1996). "Generalization error of ensemble estimators," Proceedings of International Conference on Neural Networks (ICNN96), pp. 90-95.

39. Ali A. Ghorbani, Kiarash Owrangh (2001). Stacked Generalization in Neural Networks: Generalization on Statistically Neutral Problems. Proceedings. IJCNN apos;01. International Joint Conference on Volume 3, Issue , 2001 ,1715 - 1720 vol.3

40. Yoav Freund and Robert E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting**.** *Journal of Computer and System Sciences*, 55(1):119-139.

41. Robert E. Schapire. Theoretical views of boosting (1999**).** In *Computational Learning Theory: Fourth European Conference, EuroCOLT'99*, 1-10.

42. Cynthia Rudin, Ingrid Daubechies and Robert E. Schapire (2004) On the dynamics of boosting**.** In *Advances in Neural Information Processing Systems 16.*

43. Rosa M. Valdovinos, J. Salvador Sanchez, and Ricardo Barandela (2005). Dynamic and Static Weighting in Classifier Fusion, *LCNS* 3523, pp: 59-66, Springer-Verlag Berlin Heidelberg.

44. Ronald W. Potter (2000). *The art of measurement: theory and practice*. Prentice Hall PTR.

45. Michael Kirby (2001). Geometric Data Analysis: An empirical approach to dimensionality reduction and the study of patterns. John Wiley & Sons, Inc.

46. Eidenberger, H. (2003). How good are the visual MPEG-7 features?, SPIE & IEEE Visual Communications and Image Processing Conference, Lugano, Switzerland.

47. Eidenberger, H. (2007). Descriptor Evaluation for Visual Information Retrieval using Self-Organising Maps and other Statistical Methods, Multimedia Tools and Applications, Vol. 5, No. 3, 241-258.

48. Eidenberger, H.(2004). Statistical analysis of MPEG-7 image descriptions, ACM Multimedia Systems Journal, Springer, Vol. 10, No. 2, 84-97.

49. Eidenberger, H., and Breiteneder, C. (2002). Semantic Feature Layers in Content-based Image Retrieval: Implementation of Human World Features, *IEEE International Conference on Control, Automation, Robotic and Vision*, Singapore, Singapore.

50. Bertoni, R. Folgieri, G. Valentini (2005). Feature selection combined with random subspace ensemble for gene expression based diagnosis of malignancies, In: (B.Apolloni, M.Marinaro and R. Tagliaferri, eds) *Biological and Artificial Intelligence Environments*, 29-36, Springer.

51. A. Tsymbal, P. Cunningham, M. Pechinizkiy, S. Puuronen (2003). Search strategies for ensemble feature selection in medical diagnostics, in: M. Krol, S. Mitra, D.J. Lee (eds.), Proc. 16 *IEEE Symp. on Computer-Based Medical Systems CBMS'*2003.

52. A. Tsymbal, M. Pechenizkiy, and P. Cunningham. (2003). *Diversity in ensemble feature selection*. Technical report, Trinity College Dublin.

53. Wolpert, D.H. (1992). “On the Connection Between In-Sample Testing and Generalization Error”, *Complex Systems*, **6**, 47-94.

54. Wolpert, D.H., Knill, E., and Grossman, T. (1998). “Some results concerning off-training-set and IID error for the Gibbs and Bayes optimal generalizers”, *Statistics and Computing*, **8**(1), March 1998, 35—54.

55. Wolpert, D.H. (1996). “Determining Whether Two Data Sets are from the Same Distribution”, in *Maximum Entropy and Bayesian Methods 1995*, Ed. K. Hanson and R. Silver, Kluwer Academic press.

56. Kohavi, R., and Wolpert, D.H. (1996). “Bias Plus Variance Decomposition for Zero- One Loss Functions”, *Proceedings of the International Machine Learning Conference 13*, Ed. Lorenza and Saiita, Morgan Kauffman.

57. Valentini, G., Dietterich, T. G. (2002). Bias-Variance Analysis and Ensembles of SVM. In J. Kittler and F. Roli (Ed.) *Third International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, 2364*. (222-231) New York: Springer Verlag.

58. G. Valentini (2003). *Ensemble methods based on bias-variance analysis, Ph.D. thesis*, DISI - Dipartimento di Informatica e Scienze dell' Informazione - Universita` di Genova - Tech. Rep. TR-03-04.

59. Wolpert, D.H. (1997). “On Bias plus Variance”, *Neural Computation*, **9**.

60. Pedro Domingos (2000). Proceedings of the Seventeenth National Conference on Artificial Intelligence (pp. 564-569), 2000. Austin, TX: AAAI Press. A Unified Bias-Variance Decomposition for Zero-One and Squared Loss, 564-569.

61. W. H. E. Day. Consensus methods as tools for data analysis (1988). In H. H. Bock, editor, *Classification and Related Methods for Data Analysis*, Elseiver Science Publishers, 317-324.

62. L. Lam, and C. Y. Suen.(1997) Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics*, 27(5):553-568.

63. K.M. Ting and I.H. Witten (1997). Stacked generalization: when does it work? In *Proc International Joint Conference on Artificial Intelligence*, 866-871, Japan.

64. Wolpert, D.H. (1992), “Stacked Generalization”, *Neural Networks*, **5**, 241-259.

65. Alpaydin, E. (1998) “Techniques for combining multiple learners”, *Engineering of Intelligent Systems EIS’98,* Spain, February 1998.

66. Peter Savicky and Johannes Fürnkranz. Combining Pairwise (2003). Classifiers with Stacking In *Proceedings of the 5th International Symposium on Intelligent Data Analysis (IDA-03)*, Berlin, Germany.

67. Seewald A.K (2003). Towards Understanding Stacking. PhD Thesis, Vienna University of Technology.

68. Seewald A.K (2002). Exploring the Parameter State Space of Stacking. In Proceedings of International Conference on Data Mining (ICDM-2002), Maebashi TERRSA, Maebashi City, Japan. IEEE Computer Society Press, Los Alamitos, California.

69. Dietterich, T. (1999). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging boosting and randomization. Machine Learning,40, 139–158.

70. Ludmila I. Kuncheva (2004). *Combining pattern classifiers: methods and algorithms.* New Jersey:Wiley- Interscience Publication.

71. T. K. Ho. Multiple Classifier Combination (2002). Lessons and the next steps. In A. Kandel and H. Bunke, editors, *Hybrid Methods in Pattern Recognition.* World Scientific Publishing, 171-198.

72. Mutlu Uysal (2006) , *A Hierarchical Object Localization and Image Retrieval Framework.* PhD thesis, Middle East Technical University, Ankara, Turkey.

73. Akbaş, E., Özcanli, Ö.C., and Yarman-Vural, F. (2005), "A comparision of fuzzy ARTMAP and adaboost methos in image retrieval problems", *Signal Processing and Communications Applications Conference, Proceedings of the IEEE 13th.*, 691-694.

74. Smyth, P., and Wolpert, D. H. (1998), "Stacked density estimations", *Neural Information Processing Systems 10,* MIT Press.

75. Mertayak, C. (2007), "Toward the frontiers of stacked generalization architecture for learning", *MSc. Thesis*, Middle East Technical University, Ankara, Turkey.

76. C. Mulcahy (1997). Image compression using the haar wavelet transform. Spelman College Science and Mathematics, 1:22-31, April 1997.

77. International Organization for Standardization: Coding of Moving Pictures and Audio, Multimedia content description interface, part 3 visual. Technical Report, ISO/IEC JTC1/SC9/WG!!/N4062, 2001.

78. MPEG (Moving Picture Experts Group). MPEG-7 overview. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

79. B. S. Manjunath, Philippe Salembier, Thomas Sikora (2001). Introduction to MPEG-7: Multimedia Content Description Interface. New York, Cichhester: Wiley.

80. X. Lu, Y. Wang and A. K. Jain (2003). Combining Classifiers for Face Recognition. In IEEE International Conference on Multimedia and Expo (ICME), volume 3, 13-16, Baltimore, USA.

81. G. L. Marcialis and F. Roli (2004). Fusion of Appearance-based Face Recognition Algorithms. Pattern Analysis and Applications, 7(2):151-163.

82. T. Kinnunen, V. Hautamaki, and P. Franti (2004). Fusion of Spectral Feature Sets for Accurate Speaker Identification. In Ninth Conference on Speech and Computer, 361-365, Saint-Petersburg, Russia.

83. C. Sanderson and K. K. Paliwal (2001). Information Fusion for Robust Speaker Verification. In Seventh European Conference on Speech Communication and Technology, 755-758, Aalborg, Denmark.

84. G. Feng, K. Dong, D. Hu, and D. Zhang (2004). When Faces are Combined with Palm-prints: A Novel Biometric Fusion Strategy. In First International Conference on Biometric Authentication (ICBA), 701-707, Hong Kong, China.

85. K. Chang, K. W. Bowyer, S. Sarkar, and B. Victor (2003). Comparison and Combination of Ear and Face Images in Appearance-based Biometrics. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(9):1160-1165.

86. A. Ross and R. Govindarajan (2005). Feature Level Fusion Using Hand and Face Biometrics. In Proceedings of SPIE Conference on Biometric Technology for Human Identification II, volume 5779, 196-204, Orlando, USA.

87. P. Somervuo (2003). Experiments with Linear and Nonlinear Feature Transformations in HMM Based Phone Recognition, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) , Hong Kong, vol. I, 52-55.

88. C. Glasser and H. Schmitz (2000). Concatenation Hierarchies and Forbidden Patterns, Technical Report No. 256, University of Wuerzburg.

89. Afzal, Godil, Sandy Ressler and Patrick Grother (2004). "" Face Recognition using 3D surface and color map information: Comparison and Combination"" to the SPIE's symposium on " Biometrics Technology for Human Identification", Orlando, FL

90. G. D. Forney, Jr. (1966). Concatenated Codes, Cambridge, MA: MIT Press.

91. Kenneth Hoffman, Ray Kunze (1971). Linear Algebra, 2nd Edition, Prentice Hall.

92. X. Zhang (2004). A remark on common solutions of a pair of matrix equation. Acta Mathematica Universitatis Comenianae, 151-154, Vol 73, 2, Bratislava, Slovak Republic.

93. Gilbert Strang (1988), *Linear algebra and its applications*. Brooks Cole, Thomson Learning Inc.