12-2017

Optimal Combiners for Multiple Classifier Systems

Mohammed Falih Hassan
Western Michigan University, mufalh@yahoo.com

OPTIMAL COMBINERS FOR MULTIPLE CLASSIFIER SYSTEMS

by

Mohammed Falih Hassan

A dissertation submitted to the Graduate College
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
Electrical and Computer Engineering
Western Michigan University
December 2017

Doctoral Committee:

Ikhlas Abdel-Qader, Ph.D., Chair
Bradley Bazuin, Ph.D.
Azim Houshyar, Ph.D.

OPTIMAL COMBINERS FOR MULTIPLE CLASSIFIER SYSTEMS

Mohammed Falih Hassan, Ph.D.

Western Michigan University, 2017

A Multiple Classifier System (MCS) is designed to combine classification results of an ensemble of different classifiers and consequently to produce the highest possible classification output. MCS has recently drawn growing attention and has become a necessity, especially when a problem involves a large class of noisy data or when using a single pattern classifier that has serious drawbacks in its results. A wide range of pattern recognition applications have benefited from the implementation of MCS, these include areas such as handwriting recognition, incremental learning, data fusion, feature selection, and a large variety of medical applications.

To achieve optimal ensemble performance, two design components must be optimized carefully which are diversity and the selection of combining rule. This dissertation is focused on designing an ensemble decision combining rule which leads the MCS to deliver the highest possible accuracy. Several models for decision combining rules, using an ensemble system of N classifiers and M classes, are developed. The proposed system can be considered as a unifying framework that works with any algebraic decision combining rule. While the results affirm that there is no single decision combining rule that can outperform in every classification problem, they clearly present the framework to design an optimum decision combining rule based on the statistics of the classifiers. Based on the predication extracted from the theoretical models, a novel algorithm that achieves optimal classification accuracy is presented in this study.

The proposed algorithm is tested on six datasets, the experimental results agree with the trend predicted by theoretical derivations. Results based on the proposed algorithm show that the performance of an ensemble always achieves at least the performance of the best performing individual classifier and evades selecting the least performing classifier. In addition, the results of the proposed algorithm show a comparable performance in classification accuracy compared to the random forest with less computational operations which makes it a good candidate for real time classification problems. Finally, the proposed model serves as an in-depth exploration into the performance of MCS and brings to the forefront of classification research significant insights.

ACKNOWLEDGEMENTS

All thanks go to he whom created me from nothing to a human being. I faced many challenges and difficult times and he has always been behind me to give me the patience and strength and the PhD study is just one of those many phases.

My deep thanks go to my supervisor, Dr. Ikhlas Abdel-Qader, for her continuous support and encouragement during all phases of my PhD research. I am so very thankful for her kindness and support which has played a pivotal rule in the many publications that we have worked on together. Also, I would like to thank my committee members, Dr. Bradley Bazuin and Dr. Azim Houshyar, for their support during the dissertation work.

Two women who play a crucial role in my life; my mother and my wife. I am grateful for their support, kindness and encouragement during my life. I'm dedicating my dissertation to them.

Mohammed Hassan

TABLE OF CONTENTS

Table of Contents—Continued

CHAPTER

Table of Contents—Continued

LIST OF TABLES

LIST OF FIGURES

List of Figures- Continued

List of Figures- Continued

List of Figures- Continued

CHAPTER I

INTRODUCTION

1.1. Background

The history of the works in Multiple Classifier Systems (MCS) goes back to the 1979 paper by Dasarathy and Sheela [1]. In their work, the training space is partitioned into several combined classifiers. In 1990, the work given by Hansen and Salamon [2] improved the generalization error by combining several neural networks classifiers. The main contribution into ensemble systems is done by Schapire [3]. In his work, a strong classifier is generated from combining several weak classifiers using boosting. From there, research in ensemble systems grew rapidly under different names. The short summary of names and algorithms that have described ensemble systems are a combination of multiple classifiers [4] – [8], classifier fusion [9] – [11], classifiers ensembles (ensemble systems) [12], [13], mixture of experts [14], [15] and many others. In addition, several books have already been written that focus on topics that deal with the development of ensemble systems such as [16], [17] and from 2000 until now there are series of annual workshops on multiple classifier systems [18]. The purpose of these workshops is to organize and improve progress in the area of combined classifier systems.

1.2. Advantages of Using Multiple Classifier Systems (MCS)

There are many advantages of using multiple classifiers over single classifier systems [19]: **Statistical Reasons:** A single classifier system may perform perfectly on classification of training data but perform poorly on test data. In this case, the resulting classifiers are said to have a high generalization error. In comparison, training an ensemble of classifiers and then taking the average of their outputs reduces the generalization error and improves the classification accuracy. Although the ensemble performance may not be better than the best performance classifier, it aids in avoiding the choice of a poor classifier.

Large volume of data: In cases of a large volume of data it is impractical to train a single classifier on this data, instead the training data space is divided into several sectors or partitions. Combining their outputs appropriately proved to be more effective compared to a single classifiers system.

Too little data: Ensemble systems are also proven to be an effective application on small training data sizes. For example, resampling the original data to generate different copies, then training different classifiers on these copies creates an ensemble that has proven to be very effective.

Divide and Conquer: In cases of a complex and highly nonlinear decision boundary, it is difficult to train a single classifier. Instead if the training space divided into an appropriate number of partitions and train different classifiers on these partitions, then the complex decision boundary breaks into smaller pieces that can be handled by individual classifiers. In other words, the individual classifiers that constitute an ensemble complete the complex decision boundary from combining the smaller pieces.

Data fusion: Data fusion is a term that is used to describe data from different sources in order to make a formal decision. Therefore, in cases of data training spaces that comes from different sources, it is impractical to train a single classifier on such data. These types of features are called heterogeneous features. A better solution to this issue is to partition the training data into subsets in order to train different classifiers that are used to construct an ensemble.

1.3. Architecture of Ensemble Systems

Figure 1.1 shows a typical structure for an ensemble system. A collection of a group of classifiers is called an ensemble and the fuser is a predefined rule that combines outputs of classifiers. Two famous topologies which are used to construct an ensemble are serial and parallel configuration figure 1.2 and figure 1.3 respectively. In both cases the classifiers group is called an ensemble. In serial topology, the individual classifiers are connected in a cascade manner, and the benefit of this structure is the feature's spaces are classified sequentially. When a primary classifier is uncertain about classifying a given instance then the data is fed to the next classifier that specializes in certain difficult instances and the process continues. Using this approach, it becomes possible to build a stronger ensemble system based on weak classifiers. Schapire [3] showed that it can boost the weak classifiers into a strong one by focusing on the subsets that are difficult to classify. On the other hand, the parallel structure is the most used and studied ensemble,

in this topology a feature vector is fed to all classifiers and each classifier makes its decision independently, then the fuser combine the classifier's outputs to give the final class label. In addition, there are ensemble systems which are based on hybrid topology, i.e. a mixture of parallel and serial configurations which are rarely used in practice.

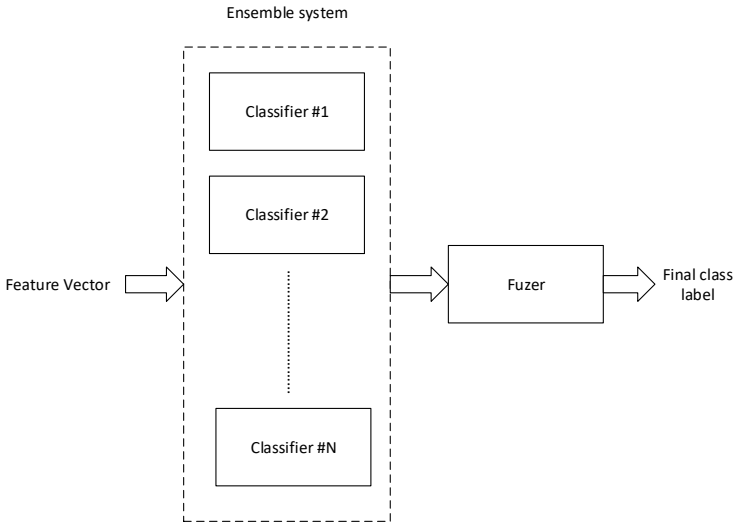

Figure 1.1. Structure of Ensemble Systems

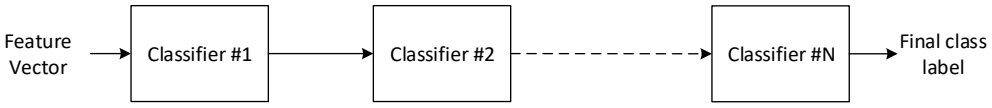

Figure 1.2. Serial Structure of Ensemble Systems

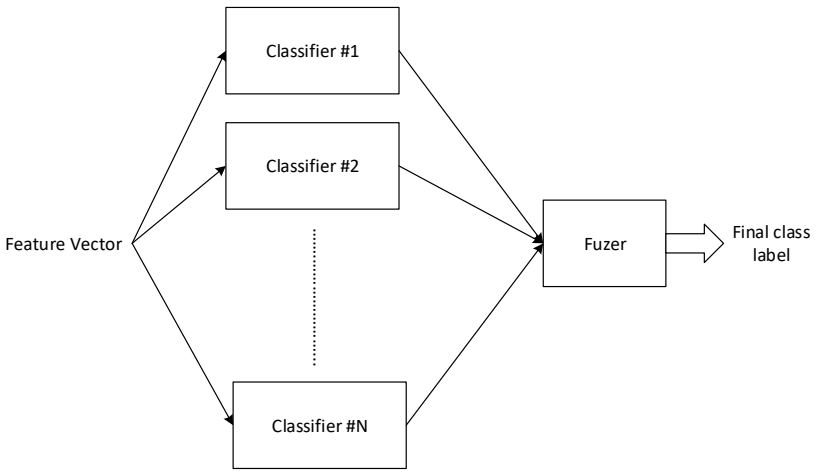

Figure 1.3. Parallel Structure of Ensemble Systems

To get a better understanding of ensemble systems, figure 1.4 shows a comprehensive picture, where the ensemble structure can be divided into four layers. The function of each layer is described as follows:

1. **Features generation:** this layer is a preprocessing stage which is related to the raw data. The purpose is to generate features that achieve high individual accuracy and diversity among base classifiers.
2. **Feature manipulation:** during the training phase features manipulation is necessary in order to achieve diversity among base classifiers. Diversity means all classifiers have complementary information which is directly related to the improvement of classification accuracy.
3. **Classifiers:** In this layer, there are many parameters that optimize the ensemble performance such as how to determine the number of base classifiers that are used to build an ensemble and which is the best method to train the base classifiers. It is better to train them simultaneously or iteratively by adding and removing classifiers. In addition, the chosen ensemble topology is very important. Mainly two topologies are used, and these are parallel and serial. In addition, creating diversity among base classifiers is very important, in which each individual classifier learns part of the training space. There are many methods used to create diversity among base classifiers, for example:

a. Creating an ensemble that is based on a different classifier model.
b. Using different parameters in training individual classifiers.
c. Partitioning the training space into different sectors, so that each sector is used to train a different base classifier.
d. Dividing classification labels among different classifiers, in order to ensure that classifiers are trained on different classification tasks.

4. **Combiner:** a combination or fusion process is the last stage in the classifiers combination, and it can be classified as follows:
    a. A non-trainable combiner is a combiner that is not related to the training data structure. An example of this is the simple arithmetic combiner such as the average and majority vote.

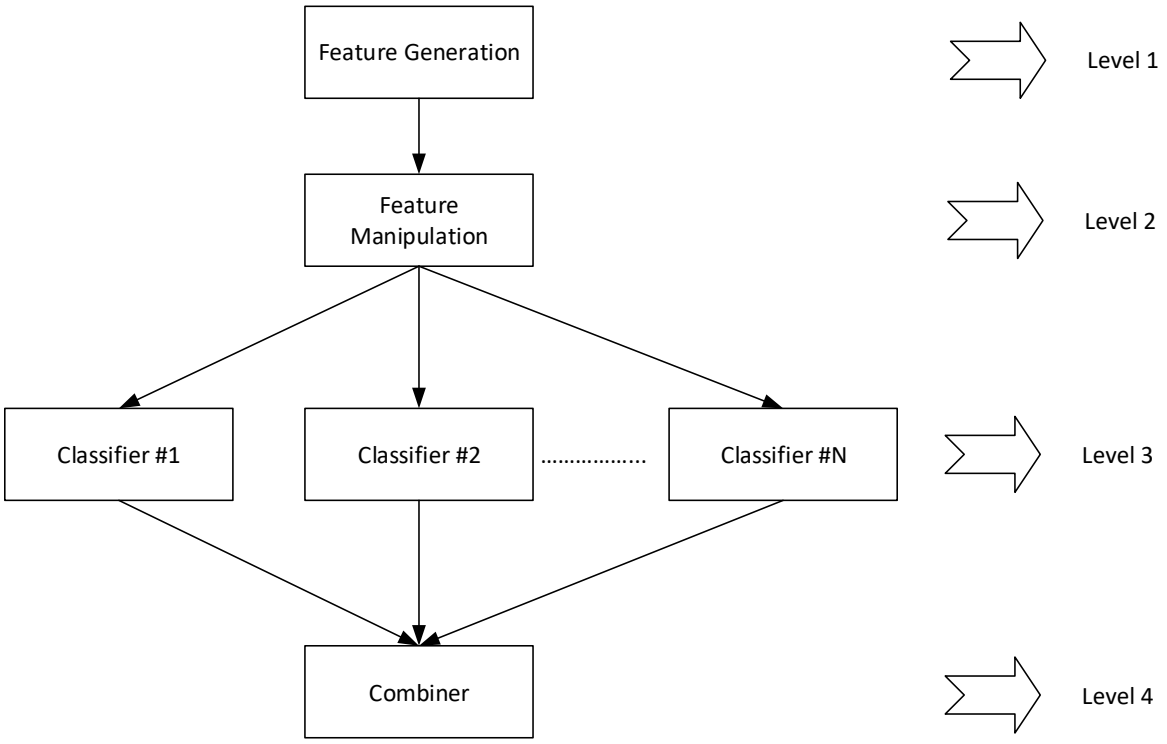

Figure 1.4. Design Stages for Ensemble Systems

b. A trainable combiner is the combiner that is generated using a special training algorithm. An example of this are the weighted combining rules and the combining rules that are based on classifier selection methods, in this case a single classifier is authorized to classify data for a given feature vector.
c. A meta classifier utilize the output of individual classifiers as training inputs for a new classifier. This approach is called stack generalization.

1.4. Literature Survey

For many theoretical and practical reasons, multiple classifier systems show improvements in classification accuracy compared to a single pattern classifier [16], [17], [19], [20], [21], [22]. Improvements come mainly from combining several classifiers that have diverse characteristics. For example, a single classifier system may give a high recognition rate on training data but performs poorly on new data, especially when the trained data is noisy. Consequently, the resulting classifier has poor generalization performance. It is better practice to design a classifier with good generalization performance. Training an ensemble of classifiers creates individual classifiers with different generalization errors, and taking the average performance of these classifiers minimizes overall generalization errors [23], [24].

Multiple classifier systems that use parallel structures are composed mainly of three stages. The first stage consists of a group of individual classifiers (base classifiers) that are already trained to recognize new data. In the second stage, the outputs of these classifiers are combined using a predefined combining rule. In the third stage, a class that has a maximum membership value among others is chosen as a correct class. The first and second stages are considered as crucial parts in the MCS design process. The major research areas in optimizing ensemble performance are focusing on diversity among individual classifiers (stage one) and the method used to combine their outputs (stage two). Combining multiple classifiers that have the same knowledge about feature space would not improve classification accuracy, and some form of diversity is necessary to minimize generalization errors [25], [26], [27], [28]. On the other hand, combining methods are used to fuse outputs of classifiers, which have been under the spotlight, and several researchers have presented different algorithms attempting to improve classification accuracy [16], [17], [18],

[29]. Combining rules are divided into three categories depending on the level of the outputs of classifiers. The category levels are: abstract level, rank level, and continuous output level. The last category is studied extensively since it contains more information about class as compared to the others. Combining rules also may consider either simple (unweighted) or weighted rules. In simple rules, all base classifiers in the ensemble are considered equally in the combination process since all classifiers have equal strengths (classification accuracies). In practice, base classifiers may use different classification algorithms or be trained by different data sectors leading to different classification accuracies. Therefore, weighting the output of each classifier according to its strength guarantees improvements in classification accuracy, and consequently, the weighted combining rule is a natural solution to combine base classifiers.

Another issue related to the combination process is the assumption regarding the correlation between outputs of classifiers. As the correlation coefficient increases among a group of base classifiers, they become redundant and result in lower classification accuracy. Correlation level among outputs of classifiers is used to model the diversity level among them [19]. The literature has many practical and theoretical works focused on recognition rates under the assumption of independent versus dependent classifiers and models that use simple versus weighted combining rules as in [30], [31], [32], [33], [34], [35], [36], [37].

In [30], a theoretical and experimental work for combining classifiers is presented. They assumed independence between classifiers and considered simple combining rules (product, sum, max, min, median and majority vote). Their results show that the sum rule outperforms others. In [31], analytical models are derived for six fusion rules. Derivations are based on assumptions that outputs of classifiers produce an estimation of the class posterior probability, independent and identically distributed for two class distributions: normal and uniform. Results showed that the ensemble performance depends on class distribution. The work presented in [32] compares the performance of the sum versus majority vote. They proposed a model based on error estimation for each classifier. They assumed classifiers are equal in strengths and distribution of classifiers outputs are normal, independent and identically distributed. Their results show that the sum always outperforms majority vote except for long tail distribution in which majority vote gives superior performance over the sum rule. In [33], a theoretical and experimental analysis is done

using simple average and weighted average rules. Their results show that the added error of the ensemble depends on the performance of individual classifiers and the correlation level between their outputs

1.5. Dissertation Purpose

This work is focused on optimizing performance of ensemble systems by managing diversity among individual classifiers and the methods used to combine their outputs. Combining multiple classifiers that have the same knowledge about features space would not improve the classification accuracy, so some form of diversity is necessary to minimize generalization errors. In practice, classifiers exhibiting a correlation among their outputs result in decreasing classification accuracy. In addition, base classifiers exhibit different classification accuracies on test data (each classifier has a different classification strength). To get a realistic model, it is necessary to account for the correlation effect and classifiers' weights into the fusion process. As shown in the survey of the previous studies, not all the conditions are considered in evaluating ensemble performance. Therefore, theoretical models are needed in order to get a better understanding of ensemble performance. One of the purposes of this study is to estimate the performance of an MCS that uses weighted combining rules for correlated classifiers.

On the other hand, methods that are used to combine outputs of classifiers are an interesting research area, and many experimental and theoretical studies have addressed this problem. The evolution of combining rule performance using experimental studies did not explain clearly the interrelated relationship among system parameters, and it is not leading to a deep understanding of the system behavior. As a result, mathematical models are needed to help in investigating why a specific combining rule works better than others for different classification problems. As shown in the literature survey, each combining rule works under a specific ensemble condition. It is uncleared which combining rule will work for a given classification problem. So, another purpose of this study is to design an optimal combining rule for a given classification problem.

The final dissertation goal is to optimize ensemble systems by creating an ensemble with maximum diversity and an optimal fusion rule. For the purpose of validation, the proposed algorithm will be tested on challenging classification problems.

1.6. Dissertation Contribution

This work is divided into two parts. The first part is focused on theoretical derivations. By assuming weighted and correlated classifiers, closed form expressions for probability of classification errors are estimated for four weighted combining rules which are: geometric mean, average, majority vote and harmonic mean. Theoretical results show there is no single combining rules that work for all classification problems, i.e. each combining rule works under a specified ensemble condition. In addition, results show that the ensemble performance (classification accuracy) degraded exponentially as correlation coefficient increases among individual classifiers. Based on the previous results, the derivation is generalized by estimating the classification error for generalized mean (power mean) rule. Power mean includes a spectrum of averaging functions, the results of theoretical derivations help to select an optimal fusion rule that minimized the classification error.

In the second part and guided by the results of theoretical derivations, an algorithm for combining classifiers is proposed. The algorithm is tested among six data sets. Experimental results agree with predication of theoretical derivations and ensemble classification always provide better classification accuracy of individual classifiers and allows for avoiding the worst performance classifiers. In addition, the classification results of the proposed algorithm show comparable classification results compared to the random forest over six data set under study.

1.7. Dissertation Overview

The rest of dissertation chapters are briefly described as follows: chapter 2 reviewed two theoretical frameworks for estimating the performance of ensemble systems which are the Kunchava [31] and Tumer-Ghosh [39] frameworks. In chapter 3, closed formulas for classification error for majority votes and geometric mean rules are estimated under assumption of unweighted classifiers. Chapter 4 extended the derivation presented in Chapter 3 to weighted and correlated classifiers and for four weighted fusion rules; geometric mean, majority vote, average and harmonic mean. Chapter 6 generalized the derivation for generalized mean rule under an assumption of N classifiers and M classes. Also, a novel ensemble algorithm is proposed which

provided classification results comparable to the standard ensemble classification algorithm. Finally, Chapter 6 presented a dissertation conclusion, contribution and future work.

CHAPTER II

THEORETICAL FRAMEWORKS FOR MULTIPLE CLASSIFIER SYSTEMS

2.1. Architecture of Multiple Classifier Systems

Figure 2.1 shows a typical configuration for an ensemble of classifiers that consists primarily of three stages. Stage one is related to the classification process on an individual classifier level, where a feature vector x_k defined in R^n is fed to N parallel classifiers. These classifiers may use the same training algorithm such as homogeneous classifiers or use different training algorithms such as heterogeneous classifiers. Each classifier in the ensemble is trained to recognize M classes. In case of continuous classifiers outputs and using an appropriate normalization, it can be assumed that each one produces at its output an estimation of the posterior class probability ($d_{i,j}$) for M classes, i.e. $d_{i,j} \in \{0,1\}$, where $i = 1,2, \dots, N$ and $j = 1,2, \dots, M$.

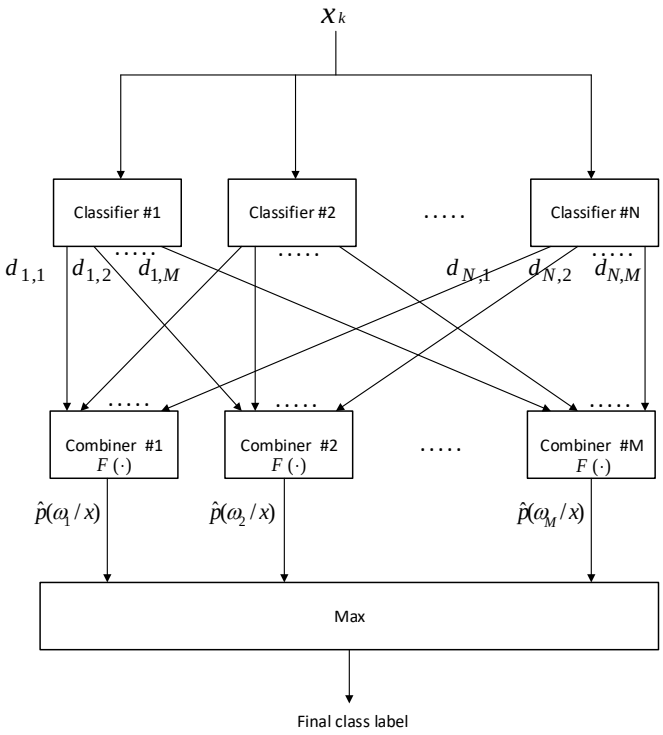

Figure 2.1. A Typical Configuration for a Multiple Classifier System

Outputs of classifiers are best described in terms of decision profile matrix ($Dp(x)$) [11], as in (2.1). The dimension of decision profile matrix is $N \times M$, its columns represent the support given from N classifiers to a single class. On the other hand, rows represent the support from a single classifier to M classes. In this work, a parallel classifiers structure is considered, which is the most widely used structure for combining ensembles of classifiers, other structures are also used in practice such as serial or hybrid [38].

$$Dp(x) = \begin{bmatrix} d_{1,1}(x) & d_{1,2}(x) & .. & d_{1,M}(x) \\ d_{2,1}(x) & d_{2,2}(x) & .. & . \\ . & . & .. & . \\ d_{N,1}(x) & d_{N,2}(x) & .. & d_{N,M}(x) \end{bmatrix}. \tag{2.1}$$

From (2.1), it can define two correlation domains in the ensemble, one in the vertical direction that is among different classifiers for the same class and another in the horizontal direction which is among different classes for the same classifier. In the derivation, it assumed two classes, thus the focused is on the correlation effect among different classifiers for the same class.

In stage two, a combining rule or fusion rule combines the output from each classifier (the columns of $Dp(x)$), i.e.

$$\hat{p}(\omega_j / x) = F(d_{1,j}, d_{2,j}, ..., d_{N,j}), \quad for\ j = 1,2,..,M\ , \tag{2.2}$$

where $\hat{p}(\omega_j/x)$ is the estimated class posterior probability for a given class ω_j and $F(\cdot)$ is the fusion rule used to combine classifiers outputs. Finally, in stage three, the combiner's output that has a maximum membership to a specific class is chosen as the correct class label, which means

$$\begin{aligned} &assign\ \ x \to \omega_k \qquad if \\ &\hat{p}(\omega_k / x) > \hat{p}(\omega_l / x) \qquad \forall k \neq l \end{aligned} \tag{2.3}$$

2.2. Review of Kuncheva Framework

Kuncheva [31] presented a framework for multiple classifiers systems that estimates the classification accuracies of combining several classifiers. The fusion rules that are used in the framework given in [31] are minimum, maximum, average, median, majority vote and oracle. In the derivation, it was assumed the following:

- There are N classifiers that work in parallel.
- Each classifier produces at its output an estimation of the posterior class probability which is denoted by $d_{i,j} \in [0,1]$ where

$$d_{i,j}(x) = p_i(\omega_j / x) + \eta(x), \tag{2.4}$$

  where $x \in R^n$ is the feature vector and $p_i(\omega_j/x)$ is the true class posterior probability and $\eta(x)$ is a random variable that could have any distribution, where $i = 1,2,\dots,M$ and $j = 1,2,\dots,N$.
- To simplify derivation, it was assumed two classes $\{\omega_1, \omega_2\}$, then $d_{1,j} + d_{2,j} = 1$ for $j = 1,2,\dots,N$.
- Classifiers outputs $d_{i,j}$ are independent and identically distributed.

In [31] two probability distributions are considered in the derivation which are normal and uniform distributions. The probability of classification error is calculated as

$$p_e = p(p_1 \le 0.5) = \int_0^{0.5} f_{p_1}(y)dy, \tag{2.5}$$

where $\hat{p}_1 = F\{d_{1,1}, d_{1,2}, \dots, d_{1,N}\}$, $\hat{p}_2 = F\{d_{2,1}, d_{2,2}, \dots, d_{2,N}\}$ and $f_{\hat{p}_1}(\cdot)$ is the probability density function of the random variable $\hat{p}_1$. For a single classifier the probability of classification error for normal distribution is

$$p_e = \Phi\left(\frac{0.5-m}{\sigma}\right), \tag{2.6}$$

and for uniform distribution is

$$p_e = \frac{0.5-m+b}{2b}, \tag{2.7}$$

where m, σ are the first and second moments of $\hat{p}_1$ when its distribution is normal while m and b are mean and period of $\hat{p}_1$ when its distribution is uniform. Based on the previous assumptions, Kuncheva derived and compared the classification errors for six fusion rules not including product rule which do not fit easily into the model presented in [31]. In addition, the assumption of independence between classifiers is unrealistic since in practice classifiers shows dependence among each other and as a result it is expected to lower the classification accuracy. Therefore, it can consider that the independence assumption provides an upper limit of the ensemble performance (optimistic estimation). It is important to model the correlation between classifiers since it's directly connected to diversity. Highly correlated classifiers represent less diversity while independence represents highly diverse classifiers. Diversity is considered as a cornerstone in the ensemble design and model it contributes significantly to the theory of multiple classifiers.

2.3. Review of Tumer and Ghosh Framework

In the following, the framework given by [33] and [39] is briefly reviewed, in order to simplify the derivation, it was assumed $M = 2$ but it can be extended to any number. The probability density function of two classes are $p(\omega_1/x)$ and $p(\omega_2/x)$. In addition, the dimension of a feature vector is considered as a scalar for the same reason as above and it can be extended to k dimension. In practice, a trained classifier gives an estimate of the class probability density function ($f_j(x)$) that is deviated from true value by ε_j i.e.

$$f_j(x) = p(\omega_j / x) + \varepsilon_j(x), \;\; j = 1,2, \tag{2.8}$$

where ε_j is the error related to the *jth* classifier's output which is considered as a normal random variable with mean m_j (bias error) and variance σ_j^2 (variance error). Figure 2.2 shows the decision boundary for a single classifier with two class examples, as shown the overall classification error is broken down into two components. The gray area is regarded as a Bayes error while the dark area is defined as the added error which is related to the imperfection in the training process, either due to noisy data or an incomplete representation of the actual training data space. Due to the inaccuracies of trained classifiers, the decision boundary deviated or shifted from the optimum decision value (x_a) by a value b (shift parameter) resulting in the expected added error estimated as

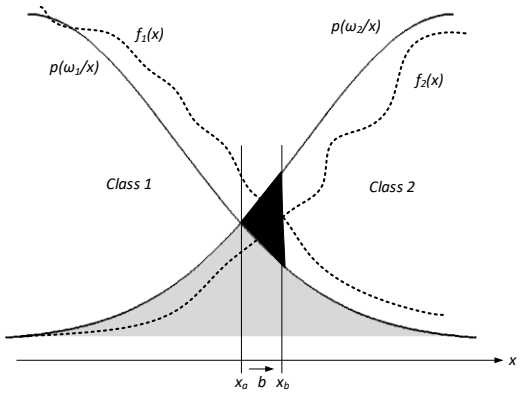

Figure 2.2. Definition of Bayes and Added Errors for a Single Classifier System and Two Class Problem

$$E_{add} = E[A(b)] = \int_{-\infty}^{\infty} A(b) f_b(b) db = \int_{-\infty}^{\infty} \int_{x_a}^{x_a+b} [p(\omega_1 / x) - p(\omega_2 / x)] p(x) f_b(b) dx db, \qquad (2.9)$$

where $A(b)$ is the added error region colored in a black as shown in figure 2.2, $p(x)$ is the probability distribution of the feature *x* and $f_b(b)$ is the probability density function of the random variable b. Using linear approximations $p(x)$ can be represented by a constant $p(x_a)$ and $A(b)$ is expressed as [33]

$$A(b) = \frac{p(x_a)}{2} b^2 s\,, \tag{2.10}$$

where $s = \acute{p}(\omega_1/x_a) - \acute{p}(\omega_2/x_a)$, and $\acute{p}(\cdot)$ is the first derivative of $p(\cdot)$, then (2.9) is modified into

$$E_{add} = \frac{p(x_a)}{2} sE[b^2]\,. \tag{2.11}$$

Expression (2.11) suggests that the key idea in calculating E_{add} is estimating the probability density function of b and its moments. From figure 2.2 and for a single classifier with two classes, the random variable b is estimated as ([39])

$$b = \frac{\varepsilon_1(x_b) - \varepsilon_2(x_b)}{s}\,. \tag{2.12}$$

Then the added error for a single classifier is defined as [33]

$$E_{add} = \frac{p(x_a)s}{2}(m_b^2 + \sigma_b^2),\ where\ m_b^2 = \frac{1}{s}(m_1 - m_2)\ and\ \sigma_b^2 = \frac{\sigma_1^2 + \sigma_2^2}{s^2}\,. \tag{2.13}$$

After the brief review of the framework defined in [39] and [33], the work is extended in the next chapter to derive an expression for the added error for combined classifier systems. The Tumer and Gosh framework is studied extensively for linear combining rules [34] but no studies have been done in nonlinear combining rules. Chapter four is focused on derive a close form expression for estimation error for the product rule based on Tumor and Gosh's framework. Kuncheva's framework presents an excellent derivation to model combining classifiers and attempts to estimate the error resulting from inherent interference among classes for given data, while Tumer and Gosh's framework attempts to model the bias and variance error that results from poor and over training of base classifiers. The idea of combining both frameworks into a single one is considered important toward unifying the theory of multiple classifier systems.

CHAPTER III

ANALYSIS OF A MULTIPLE CLASSIFIER SYSTEM USING PRODUCT AND MAJORITY VOTING RULES

In this chapter, the performance of product and majority voting rules is studied under idealized ensemble conditions then the derivations are generalized in the next chapter.

3.1. Analysis of Product Rule Using Kuncheva Framework

One of the key factors in designing a successful multiple classifier system (MCS) is choosing an appropriate combining rule. Many theoretical and experimental efforts have been focused on estimating the probability of classification error for different combining rules. In this work, assuming *N* classifiers and two independent and identically distributed classes, closed formulas for product and modified product rules are derived for estimating classification error probability under assumption of two class distributions, normal and uniform. The derivations are validated with computer simulations. The performance results of product, modified product, average, and majority vote rules are compared. The comparisons are done in term of probability of classification error as a function of class variance and number of classifiers. Results show that the modified product rule outperforms others while the product rule ranks last under the assumption of combining classifiers with good classification properties.

3.1.1. Probability of Classification Error

In this section, a closed form expression is derived for classification error probability using product and modified product rules. Assuming two class distributions (normal and uniform), N base classifiers $(i = 1,2, \dots, N)$, and two classes $j = 1,2$, then $p_i(\omega_1/x) = 1 - p_i(\omega_2/x)$. For simplifying expressions, it can set; $p_i = p_i(\omega_1/x)\ and\ \bar{p}_i = p_i(\omega_2/x) \rightarrow p_i = 1 - \bar{p}_i$.

A. Product Rule

The typical formula of product rule is defined as

$$\hat{p} = \prod_{i=1}^{N} p_i, \tag{3.1}$$

where $\hat{p}$ is the overall posterior class probability. The aim here is to estimate the probability density function for $\hat{p}$ and its moments. By taking the natural logarithm of both sides of (3.1), the multiplication process between random variables is converted into addition, that means

$$\log(\hat{p}) = \sum_{i=1}^{N} \log(p_i). \tag{3.2}$$

The purpose here is not estimating the probability density function of $\log(p_i)$ but rather in its first and second moments. From the statistic theory [40], if there is a function $f(x)$ of a random variable X provided that $f(x)$ is differentiable and the moments of X are finite, then the moments of $f(x)$ is approximated as

$$E[f(X)] \approx f(m_x) + \frac{\ddot{f}(m_x)}{2}\sigma_x^2, \tag{3.3}$$

$$VAR[f(X)] \approx (\dot{f}(m_x))^2\sigma_x^2, \tag{3.4}$$

where m_x and σ_x^2 are the mean and variance of the random variable X. $\dot{f}(x)$ and $\ddot{f}(x)$ are the first and second derivatives of $f(x)$ respectively. In the following, two formulas are derived for classification error, one for normal distribution and another for uniform distribution. All random variables (p_i) are normal, independent and identically distributed, then from (3.3) and (3.4), the moments of each one is approximated as

$$E[log(p_i)] = log\left(m_g\right) - \frac{\sigma_g^2}{2m_g^2}, \tag{3.5}$$

$$VAR[log(p_i)] = \frac{\sigma_g^2}{m_g^2}. \tag{3.6}$$

According to the central limit theorem, the probability density function of the sum of k random variables approaches normal distribution as k become large. Then the resulting mean and variance of the random variable ($\log(p_i)$) are

$$E[log(\hat{p})] = N[log(m_g) - \frac{\sigma_g^2}{2m_g^2}], \tag{3.7}$$

$$VAR[log(\hat{p})] = E[(log(\hat{p}) - E[log(\hat{p})])^2] = \frac{N\sigma_g^2}{m_g^2}. \tag{3.8}$$

To find the distribution of $\hat{p}$, take the exponent of both sides of (3.2). From the probability theory, if X and Y are random variables where $\mathrm{Y} = log(\mathrm{X})$ and Y has a normal distribution with mean m and variance σ^2 , then the random variable X has a lognormal distribution with probability density function defined as

$$f_{\hat{p}}(x) = \frac{1}{x\,\sigma\sqrt{2\pi}}\, exp\left[-\frac{1}{2}\left(\frac{log(x)-m}{\sigma}\right)^2\right], \tag{3.9}$$

$where \;\; 0 < x < \infty$, with

$$E[X] = e^{m+\sigma^2/2}, \tag{3.10}$$

$$VAR[X] = e^{2m+\sigma^2}\,(e^{\sigma^2} - 1). \tag{3.11}$$

The cumulative distribution is given by

$$F_{\hat{p}}(x) = \Phi\left(\frac{log(x)-m}{\sigma}\right). \tag{3.12}$$

Then the probability of classification error is calculated as follows

$$p_e = F(\hat{p} < 0.5) = \Phi\left(\frac{log(0.5) - N[log(m_g) - \frac{\sigma_g^2}{2m_g^2}]}{\frac{\sqrt{N}\,\sigma_g}{m_g}}\right). \quad (3.13)$$

It is also assumed that the posterior classifier probabilities have uniform distribution with mean (m_u) and variance $\sigma_u^2 = w^2/3$, where w is defined within a period of $|m_u\text{-w}, m_u\text{+w}|$ [31]. The rest of the previous assumptions and derivations also hold for uniform distribution with minor changes. Then the probability of classification error for uniform distribution is can be written as

$$p_e = F(\hat{p} < 0.5) = \Phi\left(\frac{log(0.5) - N\,[log(m_u) - \frac{W^2}{6m_u^2}]}{(\sqrt{N}W)/(\sqrt{3}m_u)}\right). \quad (3.14)$$

B. Modified Product Rule

A closer look at (3.7) and (3.8) reveals that the mean and variance of $\log(p_i)$ grows linearly with N. That means the performance of product rule degrades rapidly with increasing N. If the right side of (3.2) is divided by N, this makes (3.7) independent on N as well as reduces the variance as defined in (3.8) by a factor of $1/N$. Therefore, the modified version of product rule becomes

$$\hat{p} = (\prod_{i=1}^{N} p_i)^{1/N}. \quad (3.15)$$

Equation (3.15) is usually referenced as the geometric mean. In parallel steps of the derivations from (3.2) to (3.8), it can get the following

$$E[log(\hat{p})] = [log(m_g) - \frac{\sigma_g^2}{2m_g^2}], \quad (3.16)$$

$$VAR[log(\hat{p})] = \frac{\sigma_g^2}{Nm_g^2}. \quad (3.17)$$

The results of derivations defined in (3.16) and (3.17) confirmed the conclusions in the previous section. The probability of classification error for normal distribution is

$$p_e = \Phi\left(\frac{log(0.5)-[log(m_g)-\frac{\sigma_g^2}{2m_g^2}]}{\frac{\sigma_g}{\sqrt{N}\,m_g}}\right), \tag{3.18}$$

and for the uniform distribution is

$$p_e = \Phi\left(\frac{log(0.5)-[log(m_u)-\frac{W^2}{6m_u^2}]}{W/(\sqrt{3N}m_u)}\right). \tag{3.19}$$

The formulas defined in (3.13), (3.14), (3.18) and (3.19) are very valuable since it helps in predicting the performance of product and modified product rules versus class mean and class variance as well as understanding the impact of varying the number of base classifiers.

3.1.2. Results and Discussion

To validate the derivations for estimating the probability density function (see (3.9)), a computer program is generated that uses 10 classifiers, each classifier gives an estimate of the posterior probability density (p_i). Each estimated p_i is considered as a random variable with normal distribution that has $m_g = 10$ and σ_g^2=2. The product rule is implemented and estimated the overall posterior probability ($\hat{p}$) by multiplying the individual random variable probabilities p_i for each classifier and computed the pdf of the result. Figure 3.1 shows the two density functions from the results of the simulation program and from the mathematical derivations (3.9), the x-axis is normalized for the purpose of clarity. The similarity between the empirical and theoretical results is clearly evident. There are noticeable small difference between the two graphs as may be expected since equations (3.3) and (3.4) used in the derivation are an approximation to exact values.

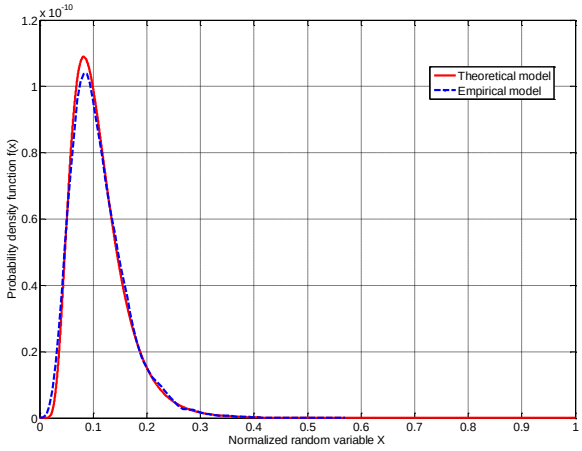

Figure 3.1. A Comparison of Probability Density Functions of ($\hat{p}$) Between Theoretical Model and Computer Simulations $m_g = 10$, σ_g=$\sqrt{2}$ and N=10

Another computer experiment is designed to verify the derivation of the probability of classification error as defined in (3.13). The set up in this experiment is similar to the previous one, except that there are 20 classifiers and each has a normal distribution with $m_g = 1$ and σ_g as a variable. Figure 3.2 shows the probability of classification error as a function of σ_g. The figure clearly shows that the computer simulation and theoretical model are in agreement. The small difference between theoretical and practical results again is due to the approximations made in (3.3) and (3.4) as well as the limited data distribution generated by a simulation program. Figure 3.3 displays a two-dimensional plot between m_g and σ_g as a function of classification error for 9 classifiers.

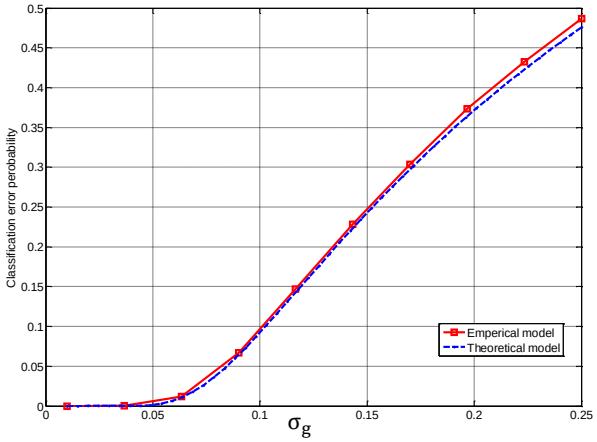

Figure 3.2. A Comparison in Term of the Probability of Classification Error Against σ_g Between Theoretical Model and Computer Simulation, $m_g = 1$ and N=20

As shown the operating characteristic with low classification error probability is limited to a small region $\{ m_g > 0.93 \,\&\, \sigma_g < 0.1\}$. Also a careful investigation on the m_g axis, shows that the abrupt change at $m_g \approx 0.93$ exhibits a smooth change on the σ_g axis. It is now clear that the behavior of the product rule is very sensitive to changes in m_g and less to σ_g variations. This is due to the fact that the probability density function of $\hat{p}$ (as shown in figure 3.1) is concentrated into a small region; therefore, a small change in m_g results in a large change in the mean as well as in the variance of the random variable $\hat{p}$ (see (3.10) and (3.11)). Such a behavior among variables can cause a significant abrupt degradation or improvement in the system performance.

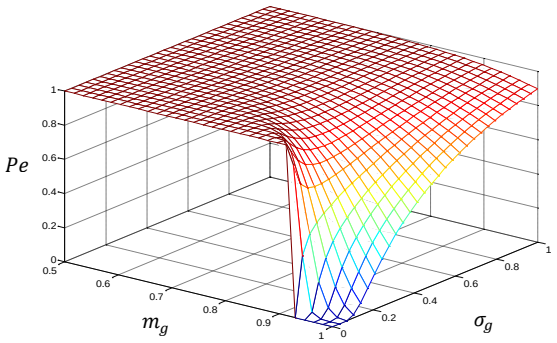

Figure 3.3. Probability of Classification Error of Product Rule as a Function of σ_g , m_g and N=9

If the condition that all random variables have the same mean and variance is removed, and assign different values to each one, then is expected to get a more robust performance. Figure 3.4 shows a two-dimensional plot for probability of classification error using modified product rule as a function of m_g and σ_g for 9 classifiers. It is clear the modified product rule exhibits better performance than the product rule since it displays a smoother behavior against changes in m_g and σ_g as well as it is having a larger region with low classification error compared to the product rule performance.

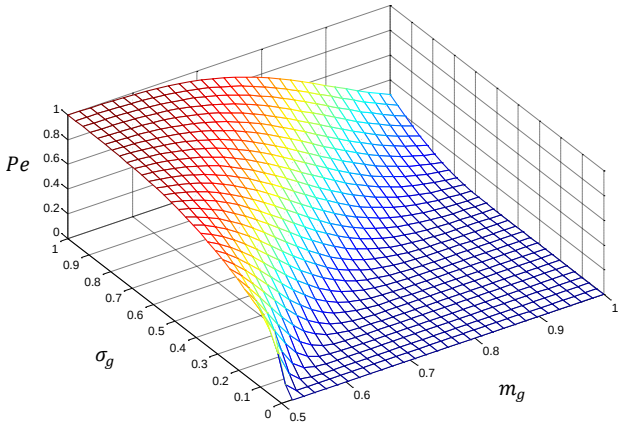

Figure 3.4. Probability of Classification Error for Modified Product Rule as a Function of σ_g , m_g and N=9

Figure 3.5 and figure 3.6 show the performance of the product, modified product, average and majority vote rules in term of classification error as a function of σ_g and w respectively. (Formulas for average and majority vote rules are taken from [31]), for $m_g = m_u = 1 \; and \; N = 7$. The comparison included two distributions, normal and uniform. As shown in figure 3.5 and figure 3.6, the product rule exhibits poor performance for σ values of 0.1 and higher, and its overall performance ranked last among the other combining rules. As can be seen, the modified product rule outperforms other rules, notably the uniform distribution. These results were expected, since figure 3.3 suggested that the low classification error region of the product rule is limited to $m > 0.93 \; and \; \sigma < 0.1$.

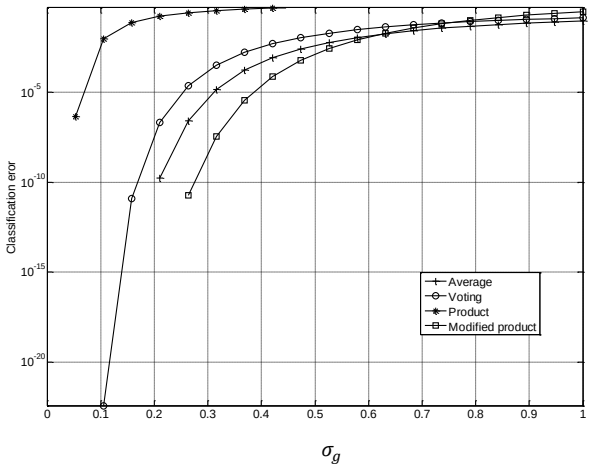

Figure 3.5. Classification Error for Different Combining Rules as a Function of σ_g for Gaussian Distribution, $m_g = 1$ and N=7

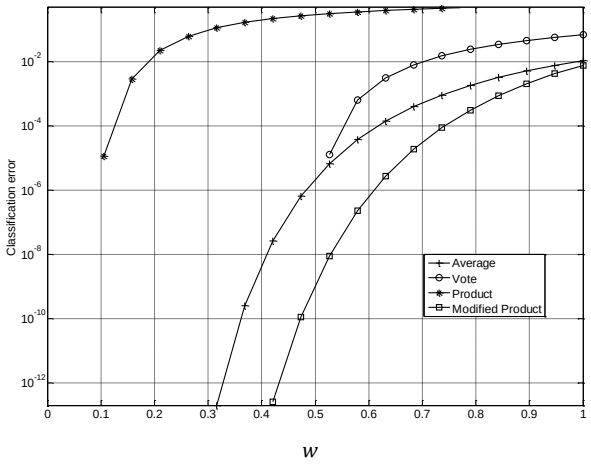

Figure 3.6. Classification Error for Different Combining Rules as a Function of w for Uniform distribution, $m_u = 1$ and N=7

Finally, figure 3.7 shows a comparison of the performance of modified product, average and majority vote rules as a function of classifier numbers for $m_g = 0.8$ and $\sigma_g = 0.3$. It is clear that the modified product rule gives superior performance compared to others. The product rule is not considered in the comparison because its performance degrades exponentially with the increase in classifiers number. This behavior results from the fact that the total class variance of product rule increases linearly with the increase of the number of classifiers causing exponential performance degradation.

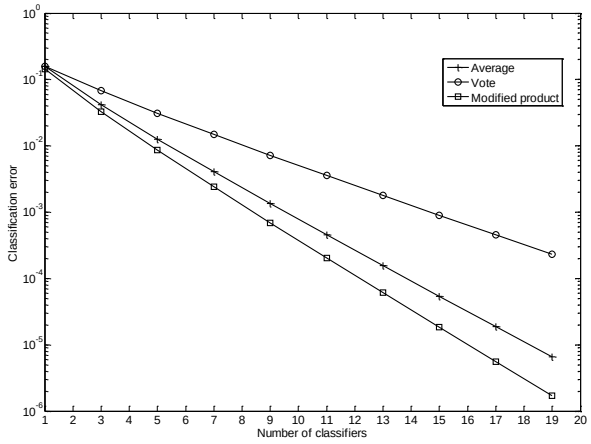

Figure 3.7. Classification Error for Different Combining Rules as a Function of Number of Classifiers for $m_g = 1$ and $\sigma_g = 0.3$

3.1.3. Section Conclusions

In this section, assuming *N* classifiers and two classes, a mathematical model is proposed for estimating the classification error probability of ensemble of classifiers operating under product and modified product rules. It was assumed the posterior class probability is independent, identically distributed and for two class distributions, normal and uniform. The derivations were verified using computer simulations. The system performance in terms of classification error as a function of mean and variance of posterior class probabilities was investigated. It also addressed the impact of posterior class variance and the number of classifiers on the probability of classification error. Results show that product rule ranks last among other combining rules, while the modified product rule outperforms them.

3.2. Analysis of Product Rule Using Tumor-Gush Framework

In order to improve classification accuracy, multiple classifier systems have provided better pattern classification over single classifier systems in different applications. The theoretical frameworks proposed in [33] and [39] present important tools for estimating and minimizing the added error of linearly combined classifier systems. In this section, a theoretical model is proposed that estimates the added error using the geometric mean rule which is a nonlinear combining rule. In the derivation, it assumed assume classifier outputs are uncorrelated and have identical distributions for a given class case. It was shown by setting the number of classifiers to one (a single classifier system), the derived formula is modified and matches the results given in [33]. Derivations are validated with computer simulations and compared with the analytical results. Due to the nonlinearity of the geometric mean, theoretical results show that the bias and variance errors are mixed together in their contribution to the added error. It was also shown that the bias error dominated the contribution to the added error compared to the variance error. It is possible to minimize the variance error by increasing the ensemble size (number of classifiers) while the bias error is minimized under specific conditions. The proposed theoretical work can help in investigating the added error for other nonlinear arithmetic combining rules.

3.2.1. Estimation the Added Error for Geometric Mean Rule

In this section, a formula that estimated the added error for geometric mean rule is derived. Due to the inaccuracy of the individual classifiers, the estimated decision boundary is deviated from the ideal one. At the decision boundary of the combined classifier outputs, the posterior probabilities of two classes are defined as

$$f_1^g(x_a + b_g) = f_2^g(x_a + b_g). \tag{3.20}$$

The subscript g refers to the geometric mean fusion process, $f_1^g(\cdot)$ and $f_2^g(\cdot)$ are the combined posterior probabilities for classes ω_1 and ω_2 respectively. Using the mathematical principles defined in (2.12), the shift parameter is written as

$$b_g = \frac{[p^g(\omega_1 / x_a) + \varepsilon_1^g(x_b)] - [p^g(\omega_2 / x_a) + \varepsilon_2^g(x_b)]}{p'^g(\omega_1 / x_a) - p'^g(\omega_2 / x_a)} = \frac{\eta_1^g - \eta_2^g}{s^g}, \tag{3.21}$$

where $\acute{p}^g(\omega_j/x_a)$ is the first derivative of $p^g(\omega_j/x)$ for $j = 1,2$. Since the combination process is averaged, then the slope of $\acute{p}^g(\omega_j/x_a)$ is assumed not changed by combination which results in $s^g = s$. In order to estimate the probability density function of b_g, the distribution of the term $\eta_1^g - \eta_2^g$ should be estimated. The Geometric Mean (GM) rule is defined as $GM = \prod_{i=1}^{N} x_i^{1/N}$ where x_i is a real positive number, then it can represent η_j^g as

$$\eta_j^g = \prod_{i=1}^{N} \left(p_i(\omega_j / x_a) + \varepsilon_{i,j}(x_b)\right)^{\frac{1}{N}} = \prod_{i=1}^{N} \eta_{i,j}^{\frac{1}{N}}, \text{ where } j = 1,2. \tag{3.22}$$

To simplify (3.22), the natural logarithm is applied to the both sides, as a result the multiplication operations is converted into addition as defined below

$$\log(\eta_j^g) = \frac{1}{N} \sum_{i=1}^{N} \log(\eta_{i,j}), \quad j = 1,2. \tag{3.23}$$

To estimate the added error of the geometric mean rule, it should estimate the probability density function of η_j^g and its moments. Based on the previous assumptions the probability density function of $\eta_{i,j}$ is a normal with mean and variance are $m_{\eta_{i,j}}$ and $\sigma^2_{\eta_{i,j}}$ respectively, then the moments of $\log(\eta_{i,j})$ are written as

$$E[\log(\eta_{i,j})] = \frac{1}{\sqrt{2\pi\sigma^2_{\eta_{i,j}}}} \int_{-\infty}^{\infty} \log(\eta_{i,j}) \exp\left(-\frac{1}{2} \frac{(\eta_{i,j} - m_{\eta_{i,j}})^2}{\sigma^2_{\eta_{i,j}}} \right) d_{\eta_{i,j}}, i = 1,2,...,N \text{ and } j = 1,2 \qquad (3.24)$$

$$VAR[\log(\eta_{i,j})] = \left[\frac{1}{\sqrt{2\pi\sigma^2_{\eta_{i,j}}}} \int_{-\infty}^{\infty} \left(\log(\eta_{i,j})\right)^2 \times \exp\left(-\frac{1}{2} \frac{(\eta_{i,j} - m_{\eta_{i,j}})^2}{\sigma^2_{\eta_{i,j}}} \right) - \left(E[\log(\eta_{i,j})]\right)^2 \right] d_{\eta_{i,j}},$$

$$i = 1,2,...,N \text{ and } j = 1,2. \qquad (3.25)$$

For a given class, the random variables $\eta_{i,j}$ are assumed independent, then the moments of $\log(\eta_j^g)$ are expressed as follows

$$E[\log(\eta_j^g)] = \frac{1}{N} \sum_{i=1}^{N} E[\log(\eta_{i,j})], \qquad (3.26)$$

$$VAR[\log(\eta_j^g)] = \frac{1}{N^2} \sum_{i=1}^{N} VAR[\log(\eta_{i,j})]. \qquad (3.27)$$

Since the term $\log(\eta_j^g)$ involves the addition of N random variables, then it can approximate its distribution as a normal as N become large. In order to get the distribution of η_j^g taking the exponent of both sides of (3.23). From the probability theory if there are two random variables X and Y and they are defined as $X = \exp(Y)$ and Y has a normal distribution then X (which is this case η_j^g) has lognormal distribution with moments defined as follows

$$E[\eta_j^g] = \exp\left(E[\log(\eta_j^g)] + \frac{1}{2} VAR[\log(\eta_j^g)] \right), \tag{3.28}$$

$$VAR[\eta_j^g] = \exp\left(2E[\log(\eta_j^g)] + VAR[\log(\eta_j^g)]\right) \times \left(\exp\left(VAR[\log(\eta_j^g)]\right) - 1\right). \tag{3.29}$$

There is no analytical solution for (3.28) and (3.29) but they can be solved numerically; however, an analytical expression helps us understanding the interrelationship roles of the ensemble parameters on the overall performance. Using Taylor series it is possible to approximate the moments of $\log(\eta_j^g)$.If the moments of $\eta_{i,j}$ are finite then (3.28) and (3.29) can be rewritten as follows:

$$E[\eta_j^g] \approx \exp\left(\frac{1}{N} \sum_{i=1}^{N} \left(\log\left(p_i(\omega_j / x_a) + m_{i,j} \right) - \frac{\sigma_{i,j}^2}{2\left(p_i(\omega_j / x_a) + m_{i,j} \right)^2} \right) \right.$$

$$\left. + \frac{1}{N^2} \sum_{i=1}^{N} \left(\frac{\sigma_{i,j}^2}{2\left(p_i(\omega_j / x_a) + m_{i,j} \right)^2} \right) \right), \quad j = 1,2, \tag{3.30}$$

$$VAR[\eta_j^g] \approx \exp\left(\frac{2}{N} \sum_{i=1}^{N} \left(\log\left(p(\omega_j / x_a) + m_j \right) - \frac{\sigma_j^2}{2\left(p(\omega_j / x_a) + m_j \right)^2} \right) + \frac{1}{N^2} \sum_{i=1}^{N} \frac{\sigma_j^2}{\left(p(\omega_j / x_a) + m_j \right)^2} \right)$$

$$- \frac{\sigma_j^2}{2\left(p(\omega_j / x_a) + m_j \right)^2} \Bigg) + \frac{1}{N^2} \sum_{i=1}^{N} \frac{\sigma_j^2}{\left(p(\omega_j / x_a) + m_j \right)^2} \Bigg) \times \left[\exp\left(\frac{1}{N^2} \sum_{i=1}^{N} \frac{\sigma_j^2}{\left(p(\omega_j / x_a) + m_j \right)^2} \right) - 1 \right], \quad j = 1,2. \tag{3.31}$$

In order to simplify the derivation, errors that corrupt the N classifiers for a specific class are assumed identical, then $m_{i,j} = m_j$ and $\sigma_{i,j}^2 = \sigma_j^2$ for $i = 1,2,\dots,N$ and $j = 1,2$, then (3.30) and (3.31) evolve into

$$E[\eta_j^g] \approx \exp\left(\log\left(p(\omega_j / x_a) + m_j \right) + \frac{1}{2} \frac{\sigma_j^2}{\left(p(\omega_j / x_a) + m_j \right)^2} \left(\frac{1}{N} - 1 \right) \right), \quad j = 1,2, \tag{3.32}$$

$$VAR[\eta_j^g] \approx \exp\left(\log\left(p(\omega_j / x_a) + m_j\right)^2 + \frac{\sigma_j^2}{\left(p(\omega_j / x_a) + m_j\right)^2}\left(\frac{1}{N} - 1\right)\right)$$

$$\times\left[\exp\left(\frac{\sigma_j^2}{N\left(p(\omega_j / x_a) + m_j\right)^2}\right) - 1\right],\ j = 1,2. \tag{3.33}$$

If class ω_1 and ω_2 are assumed independent then based on the simulation test given in section 3.2.2, the distribution of the difference of two independent lognormal distributions (b_g) can be approximated as a normal distribution. Using (3.21) the moments of b_g are defined as follows

$$E[b_g] = m_{bg} = \frac{1}{s}\left(E[\eta_1^g] - E[\eta_2^g]\right), \tag{3.34}$$

$$VAR[b_g] = \sigma_{bg}^2 = \frac{1}{s^2}\left(VAR[\eta_1^g] + VAR[\eta_2^g]\right). \tag{3.35}$$

Using (2.9) the added error (E_{add}^g) for geometric mean is defined as

$$E_{add}^g = E[A(b_g)] = \frac{p(x_a)s}{2\sqrt{2\pi\sigma_{bg}^2}} \int_{-\infty}^{\infty} b_g^2 \exp\left(-\frac{1}{2}\frac{(b_g - m_{bg})^2}{\sigma_{bg}^2}\right) db_g. \tag{3.36}$$

It is possible to approximate (3.36) if the moments of b_g are finite. By using Taylor series expansion for $A(b_g)$ around m_{bg}, the added error is written as

$$E_{add}^g = E\left[A(b_g)\right] \approx A(m_{bg}) + \frac{A''(m_{bg})}{2}\sigma_{bg}^2, \text{ where } A(b_{bg}) = \frac{s}{2}b_g^2, \tag{3.37}$$

where $A''(b_g)$ is the second derivative of $A(b_g)$. Using (3.36) and (3.37) the added error is written as

$$E_{add}^{g} \approx \frac{p(x_a)s}{2}(m_{bg}^2 + \sigma_{bg}^2). \tag{3.38}$$

The added error defined in (3.38) is decomposed into two components m_{bg}^2 and σ_{bg}^2. Due to the nonlinearity of the geometric mean, the bias (m_j) and variance (σ_j^2) errors are mixed together in their contribution to the added error. It can be noted that as *N* increases, the variance component (σ_{bg}^2) of b_g as defined in (3.33) and (3.35) is gradually diminished, and the only limiting term is the bias component (m_{bg}) of b_g. As part of the validation of the previous derivation, if the number of classifiers in (3.38) is set to 1, then the added error from (3.38) should match the added error for a single classifier as expressed in (2.13). Substitute $N = 1$ into (3.34) given that $p(\omega_1/x_a) = p(\omega_2/x_a)$, then

$$m_{bg} = \frac{1}{s}\left(\exp\left(\log(p(\omega_1/x_a) + m_1\right) - \exp\left(\log(p(\omega_2/x_a) + m_2\right)\right) = \frac{1}{s}(m_1 - m_2) = m_b, \tag{3.39}$$

and for (3.35)

$$\sigma_{bg}^2 = \frac{1}{s^2}\left(\left(p(\omega_1/x_a) + m_1\right)^2 \times \left[\exp\left(\frac{\sigma_1^2}{\left(p(\omega_1/x_a) + m_1\right)^2}\right) - 1\right]\right.$$

$$\left. + \left(p(\omega_2/x_a) + m_2\right)^2 \times \left[\exp\left(\frac{\sigma_2^2}{\left(p(\omega_2/x_a) + m_2\right)^2}\right) - 1\right]\right). \tag{3.40}$$

In order to simplify (3.40) Maclaurin series expansion is used for $exp(\cdot)$, where $exp(x) \approx 1 + x$, when $x \ll 1$, i.e. $p(\omega_j/x_a) + m_j \gg \sigma_j$ then (3.40) modifies to

$$\sigma_{bg}^2 \approx \frac{1}{s^2}\left(\sigma_1^2 + \sigma_2^2\right) = \sigma_b^2. \tag{3.41}$$

From previous results (3.39) and (3.41), it is clear that the approximation in (3.32) is valid over wide range of m_j and σ_j values while the approximation in (3.33) is valid when $p(\omega_j/x_a) + m_j \gg \sigma_j$. However, the exact calculation for $E[\eta_j^g]$ and $VAR[\eta_j^g]$ are obtained by solving the numerical integrations given in (3.24) and (3.25).

3.2.2. Results and Discussion

The discussion in this section is divided into two parts. In the first part, the purpose is to validate the derivations with computer simulations. In the second part, the performance analysis of the geometric mean rule is studied in terms of ensemble parameters $m_1, m_2, \sigma_1,\ \sigma_2$ and N. Since $p(x_a)$ is a constant that scales the added error and for comparison, the value of the added error is normalized by $p(x_a)$. To validate derivations, a computer simulation model is proposed. The purpose is to estimate the probability density function of the shift parameter b_g experimentally and compare it with the predication of the theoretical model. In the simulation model, the system behavior shown in figure 2.2 is simulated, in which the posterior probability of two classes are approximated with linear lines that have slopes {1,-1}, each line is corrupted with a normal random variable that represents class error ($\varepsilon_j, j = 1,2$). The mean and variance of ε_j are m_j and σ_j^2 respectively, the setting of other parameters are $f_1(x) = 1 - x$, $f_2(x) = x$, s $= \acute{p}(\omega_1/x_a) - \acute{p}(\omega_2/x_a) = 2,\ m_1 = 0.2,\ m_2 = 0.1,\ \sigma_1 = \sigma_2 = 0.1$ and $N = 10$. The steps is repeated for $N = 10$ classifiers, then the geometric mean combining rule is applied to combine classifier outputs in order to get the final estimate of the final posterior probabilities $f_j^g(x),\ j = 1,2$. The shift parameter b_g is calculated when $f_1^g(x_b) = f_2^g(x_b)$, the previous algorithm is repeated 100,000 times in order to get an accurate estimate of b_g. In figure 3.8, an example image of an iteration of the procedure is presented, showing the two posterior probabilities deviating from the optimum decision boundary ($x = 0.5$) due to errors.

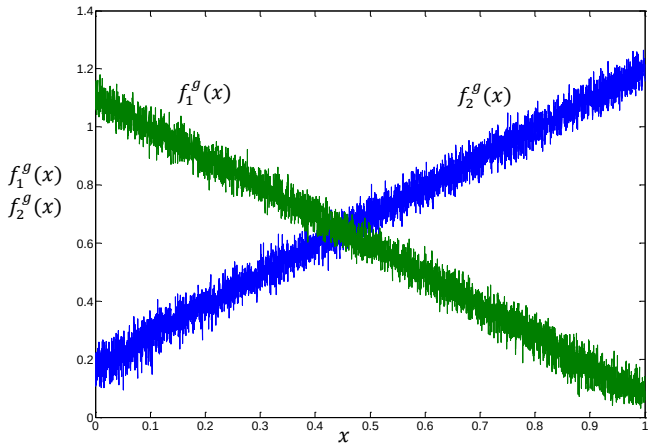

Figure 3.8. Simulation of Posterior Probabilities of the Combined Classifiers That Used Geometric Mean Rule

The bias component of a class error shifts the posterior probabilities up and down depending on the level of bias, while the variance component represents a random fluctuation about the average value of $f_j(x)$. Figure 3.9 shows a comparison in terms of the probability density functions of b_g between the empirical and theoretical models for the geometric mean rule. The parameters used in comparison are $\mathrm{s} = \acute{p}(\omega_1/x_a) - \acute{p}(\omega_2/x_a) = 2$, $m_1 = 0.2$, $m_2 = 0.1$, $\sigma_1 = \sigma_2 = 0.1$ and $N = 10$, the figure is clearly shown the matching between both models. Figure 3.10 shows the error components of E_{add}^g (σ_g and m_g) are plotted as a function of *N*.

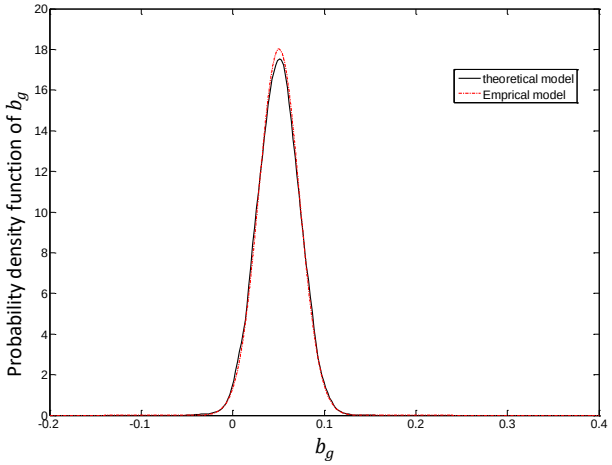

Figure 3.9. Probability Density Function of b_g where $m_1 = 0.2$, $m_2 = 0.1$, S=2, $\sigma_1 = \sigma_2 = 0.1$ and $N = 10$

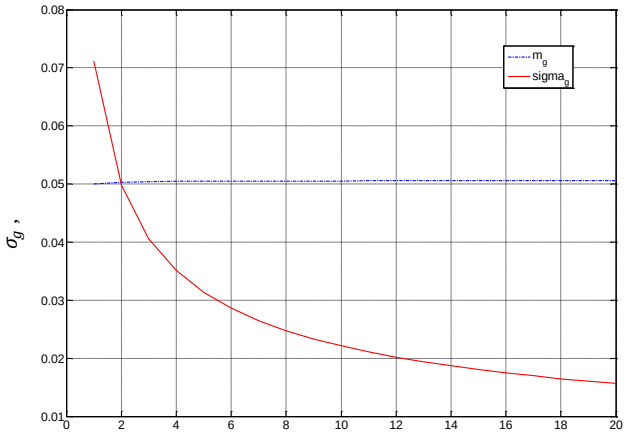

Figure 3.10. Added Error Components (σ_g and m_g) as a Function of N Where $m_1 = 0.2$, $m_2 = 0.1$, S=2, and $\sigma_1 = \sigma_2 = 0.1$

As shown the variance component is minimized gradually as expanding the ensemble size while the bias component remains unaffected. A Closer look at (3.32), (3.33) and (3.38) reveals that as $N \to \infty$ then $\lim_{N\to\infty} E_{add}^{g} = (s/2)m_{bg}^2$, where $\sigma_{bg}^2 = 0$. The possible ways to minimize m_{bg}^2 is either by generating classifiers with very low individual bias error or for a given class, training classifiers that have nearly identical bias errors i.e. making $m_1 \approx m_2$ given that $m_j \gg \sigma_j$. The performance is tested as a function of the relative mean $m_1 - m_2$ and under equal standard deviation error $\sigma_1 = \sigma_2$. The reason for this choice is because these definitions match the expressions for the bias and variance errors defined in (2.13), (3.34) and (3.35). Figure 3.11 and figure 3.12 show the added error E_{add}^{g} plotted as a function of σ_j and $(m_1 - m_2)$ respectively. As shown, the performance degraded severely against the relative bias error while it exhibited smoother change against variance error.

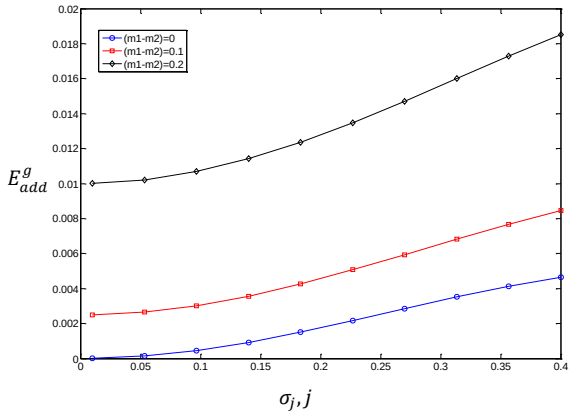

Figure 3.11. Added Error as a Function Of $\sigma_1 = \sigma_2$ for Different Relative Bias Values $(m_1 - m_2)$, Where $s = 2$, and $N = 10$

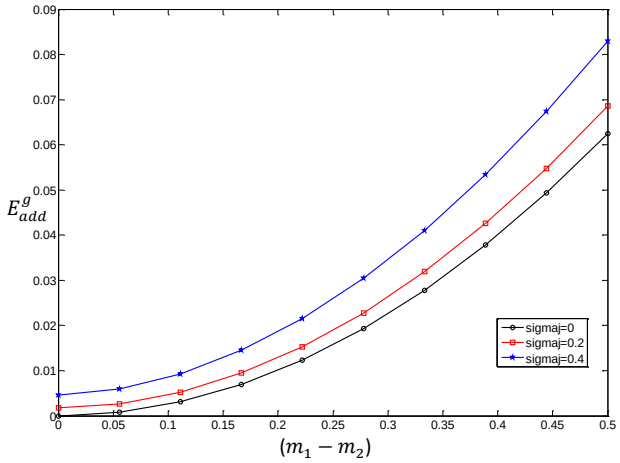

Figure 3.12. Added Error as a Function of Relative Bias Values $(m_1 - m_2)$ for Different Values of $\sigma_1 = \sigma_2$, Where $s = 2$ and $N = 10$

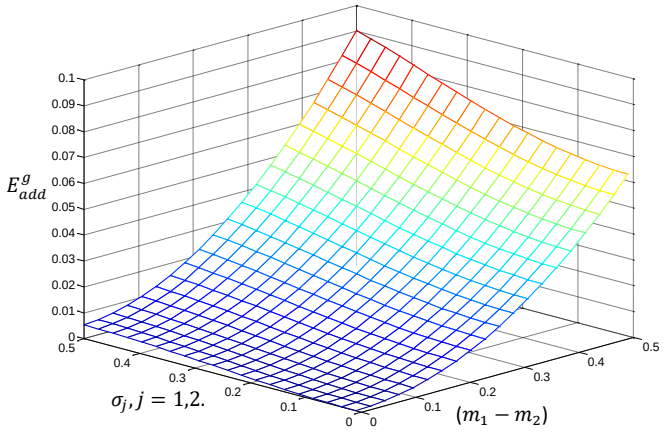

Figure 3.13. Added Error as a Function of Relative Bias Values $(m_1 - m_2)$ and $\sigma_1 = \sigma_2$, Where $s = 2$ and $N = 10$

The reason behind this behavior as defined in (3.32) and (3.33) the variance error is scaled by the number of classifiers used in the ensemble while there is no control on the bias error in the combination process. Another possible solution for minimizing the bias error is to weigh each classifier before the fusion process. The weight should be directly related to the bias level of each classifier since the variance error is already scaled by the number of classifiers. In order to get a better vision of the system performance, figure 3.13 shows a two-dimensional plot in terms of added error as a function of relative bias and variance errors. As shown the system performance grows exponentially with the level of bias error. For the given system parameters

$s = 2$ and $N = 10$, the region of low added error (≤ 0.01) is bounded by $(m_1 - m_2) < 0.1$ and $\sigma_j < 0.5$.

3.2.3. Section Conclusion

The purpose of combining several classifiers is to minimize the added error from each one and improve the overall classification accuracy. In this work, a theoretical model is developed for estimating the added error of combining N classifiers using a nonlinear geometric mean rule. The assumptions used in the derivations are based on the framework given by [33] and [39] where classifier outputs are considered as an estimation of posterior class probability functions which behave as monotonic functions around the decision boundary. The imperfection in the training process is modeled as a normal random variable superimposed on classifier outputs whose first and second moments represent the bias and variance errors respectively. The purpose of the combination process is to minimize the level of the error components. It was shown that the variance error is minimized by increasing the ensemble size, while there is no control on bias error. Yet, under certain conditions, the effect of bias error can be reduced if classifiers are optimized to satisfy the condition of $m_1 \approx m_2$ and $m_j \gg \sigma_j, j = 1,2$. Also, the results show that the bias error dominates the contribution into overall error compared to the variance error. One possible solution for minimizing the effect of bias error is to weigh classifiers according to the level of bias error before the fusion process. The developed framework for the geometric mean rule gives more intuition into estimating the added error for other nonlinear combining rules.

3.3. **Performance Analysis of Majority Vote**

Combining rules in Multiple Classifier Systems (MCS) play a central role in shaping their performance. Many theoretical works are developed to predict the performance using different combining rules. Some of the developed works assumed that classifier outputs are independent; however, in practice an ensemble of classifiers shows dependent behavior between each other. In this work, a theoretical model is derived for estimating the misclassification error probability of MCS based on the majority vote combiner. In the derivation, it assumed each

classifier produces at its output an estimation of the posterior class probability that has a normal distribution. In addition, each classifier assumed to has two classes, and the outputs of classifiers are dependent and identically distributed. The model is validated using computer simulations. Results show that the ensemble performance is highly sensitive to class variance while exhibits a smoother behavior against class mean. Also, results show that as the correlation among classifier outputs increases, the probability of classification error decreases exponentially. The trend continues until the performance reaches the behavior of a single classifier regardless of the number of base classifiers used in the ensemble. The proposed model provides a better understanding of the behavior of majority vote combiner in MCS.

3.3.1. Majority Vote Rule

The majority voting rule is considered one of the most commonly used rules in MCS [16]. There are three types of majority voting based on a method used in decision making. The first method called "Unanimous" voting selects a class that all classifiers are in agreement. The second type is called simple majority, in this case, a class is chosen if it is at least one more than half number of classifiers are agreed on that class. The third type is called "majority voting", in this type, the class received the heights number of votes will be chosen. The majority voting rule is the most popular rule used. The class selection procedure is described as follows. A class ω_j will be chosen if

$$\sum_{i=1}^{N} d_{i,j} = \max_{j=1}^{M} \sum_{i=1}^{N} d_{i,j} \tag{3.42}$$

where N is the number of classifiers in the ensemble, M is the number of classes and $d_{i,j} \in \{0,1\}$ is the decision of the *ith* classifier for the *jth* class. Theoretical results shown that as $N \to \infty$ the probability of classification error approaches 0, when a single base classifier error probability is less than 0.5 ($p_e < 0.5$) and approaches 1 when $p_e > 0.5$, [19].

3.3.2. Probability of Classification Error for Majority Vote Combiner

In this section, a mathematical model is derived to estimate the performance of MCS using majority vote rule for N correlated classifiers. It was considered two classes' problem ($M = 2$) [31], [32], [33], since it is presumed that more assumptions about system variables will be needed for classification problems with $M > 2$ [31]. Also, it is assumed that classifiers' outputs produce an estimation of the class posterior probability ($d_{i,j}$) that have identical normal distribution with mean m and variance σ^2. In order to simplify derivations, let $p_{\mathrm{i}} = d_{i,1},\ \bar{p}_i = d_{i,2} = 1 - p_i \ \ for\ i = 1,2,\dots,N$. For details on these assumptions, one can refer to work done in [31] and [41].

Additionally, correlation coefficients $\rho_{k,l}$ between any pair of classifiers are assumed identical, i.e. $\rho_{k,l} = \rho$ for all $k \neq l$. Consequently, the covariance between each classifiers pair is defined as

$$\mathrm{cov}(p_k, p_l) = \rho_{k,l}\ \sigma_k\ \sigma_l = \rho\sigma^2, \quad \text{for} \quad k \neq l \tag{3.43}$$

According to the previous assumptions and from the probability theory, it is possible to express the joint normal probability density function of classifiers' outputs as follows

$$f(\mathbf{P}) = f(p_1, p_2, \dots .. p_N) = \frac{1}{(2\pi)^{N/2}\sqrt{|\mathbf{\Sigma}|}} \exp\left[-\frac{1}{2}(\mathbf{P}-\mathbf{\Omega})^T \mathbf{\Sigma}^{-1}(\mathbf{P}-\mathbf{\Omega})\right] \tag{3.44}$$

where P is a vector of random variables which representing classifiers' outputs, $\mathbf{\Omega}$ is the mean vector of the random variable P, $(\cdot)^T$ is a matrix transpose operator and $\mathbf{\Sigma}$ is the covariance matrix that should be symmetric and positive definite. P, $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ are defined as follow

$$\mathbf{P}=\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ . \\ . \\ p_N \end{bmatrix}, \quad \mathbf{\Omega}=\begin{bmatrix} E[p_1] \\ E[p_2] \\ E[p_3] \\ . \\ . \\ E[p_N] \end{bmatrix}=\begin{bmatrix} m \\ m \\ m \\ . \\ . \\ m \end{bmatrix}, \mathbf{\Sigma}=\begin{bmatrix} \sigma^2 & \rho\sigma^2 & . & . & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 & . & . & \rho\sigma^2 \\ . & . & . & . & . \\ . & . & . & . & \rho\sigma^2 \\ \rho\sigma^2 & \rho\sigma^2 & . & . & \sigma^2 \end{bmatrix}. \tag{3.45}$$

To calculate the probability of classification error, it should count all events in which the number of base classifiers in error which is equal or more than $(N+1/2)$, i.e.

$$p_e=\sum_{k=\frac{N+1}{2}}^{N}\binom{N}{k}F\left[\mathbf{a}(k),\mathbf{b}(k)\right]. \tag{3.46}$$

$F[\boldsymbol{a},\boldsymbol{b}]$ is the cumulative distribution function for jointly normal random variable vector P. $\boldsymbol{a}=[a_1,a_2,a_3,\ldots,a_N]$ and $\boldsymbol{b}=[b_1,b_2,b_3,\ldots,b_N]$ are defining the lower and upper integration limits respectively, using (3.44), $F[\boldsymbol{a},\boldsymbol{b}]$ is written as follows

$$F\left[\mathbf{a},\mathbf{b}\right]=\frac{1}{(2\pi)^{N/2}\sqrt{|\mathbf{\Sigma}|}}\int_{a_1}^{b_1}\int_{a_2}^{b_2}\int_{a_3}^{b_3}\ldots\int_{a_N}^{b_N}\exp\left[-\frac{1}{2}(\mathbf{P}-\mathbf{\Omega})^T\mathbf{\Sigma}^{-1}(\mathbf{P}-\mathbf{\Omega})\right]dp_1dp_2dp_3\ldots dp_N \ . \tag{3.47}$$

To clarify how to use integral limits defined in (3.47). it was defined two domains; the first one is related to the probability of misclassification that is calculated over $\{-\infty,0.5\}$ and the second is the probability of correct classification that is calculated over $\{0.5,\infty\}$. Using (3.44) through (3.47), the derived formula for the probability of classification error is described as follows

$$p_e=\frac{1}{(2\pi)^{N/2}\sqrt{|\mathbf{\Sigma}|}}\sum_{k=\frac{N+1}{2}}^{N}\binom{N}{k}\underbrace{\left(\prod_{i=1}^{k}\int_{-\infty}^{0.5}\right)}_{k-terms}\underbrace{\left(\prod_{\mathrm{j=k+1}}^{\mathrm{N}}\int_{0.5}^{\infty}\right)}_{(N-k)-terms} \quad ,$$

$$\times \exp\left[-\frac{1}{2}(\mathbf{P}-\mathbf{\Omega})^T \mathbf{\Sigma}^{-1}(\mathbf{P}-\mathbf{\Omega})\right] dp_1 dp_2 dp_3 \ldots dp_N, \tag{3.48}$$

where P, $\mathbf{\Omega}$ and $\mathbf{\Sigma}$ are defined in (3.46). The k terms defined in (3.48) stand for misclassification probability and $(N-k)$ terms correspond to the correct classification probability. The $(N-k)$ terms only exist when $k+1 \leq N$ otherwise their value are considered to be unity. Proposed works in [42], [43], suggested algorithms for numerical computation of the multivariate normal distribution function defined in (3.48). In order to investigate the effect of the correlation parameter (ρ) on the classification error probability, expression (3.48) is expanded in terms of ρ as follows

$$p_e = \frac{1}{(2\pi)^{N/2}\sqrt{|\mathbf{\Sigma}|}} \sum_{k=\frac{N+1}{2}}^{N} \binom{N}{k} \underbrace{\left(\prod_{i=1}^{k} \int_{-\infty}^{0.5}\right)}_{k-terms} \underbrace{\left(\prod_{\mathrm{j=k+1}}^{\mathrm{N}} \int_{0.5}^{\infty}\right)}_{(N-k)-terms}$$
$$\times \exp\left[-\frac{1}{2}\left(\kappa_1 \sum_{\lambda=1}^{N} \frac{(p_\lambda - m)^2}{\sigma^2} + \kappa_2 \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \sum_{\beta=1}^{N} 2\frac{(p_\alpha - m)(p_\beta - m)}{\sigma^2}\right)\right] dp_1 dp_2 dp_3 \ldots dp_N, \tag{3.49}$$

where $|\mathbf{\Sigma}|$, κ_1 and κ_2 are defined by (3.50), (3.51) and (3.52) respectively

$$|\mathbf{\Sigma}| = \sigma^{2N}[(1-\rho)^N + N\rho(1-\rho)^{N-1}] \tag{3.50}$$

$$\kappa_1 = \frac{[(1-\rho)^{N-1} + (N-1)\rho(1-\rho)^{N-2}]}{[(1-\rho)^N + N\rho(1-\rho)^{N-1}]} \tag{3.51}$$

$$\kappa_2 = \frac{-\rho(1-\rho)^{N-2}}{[(1-\rho)^N + N\rho(1-\rho)^{N-1}]} \tag{3.52}$$

It is worth noticing that using (3.49) the correlation coefficient can be decomposed into two parts. The first part κ_1 contributes to the independent random variable components of P while the second part κ_2 contributes to the joint components. For $N \gg 1$, κ_1 and κ_2 can approximate as follows

$$\tilde{\kappa}_1 \approx \kappa_1 \approx \frac{1}{1-\rho} \quad , \quad \tilde{\kappa}_2 \approx \kappa_2 \approx \frac{-1}{N(1-\rho)} \tag{3.53}$$

Substitute the values of κ_1 and κ_2 defined in (3.53) into (3.49) results in

$$p_e \approx \frac{1}{(2\pi)^{N/2}\sqrt{|\boldsymbol{\Sigma}|}} \sum_{k=\frac{N+1}{2}}^{N} \binom{N}{k} \underbrace{\left(\prod_{i=1}^{k} \int_{-\infty}^{0.5} \right)}_{k-terms} \underbrace{\left(\prod_{\mathrm{j=k+1}}^{\mathrm{N}} \int_{0.5}^{\infty} \right)}_{(N-k)-terms}$$

$$\times \exp\left[-\frac{\tilde{\kappa}_1}{2} \left(\sum_{\lambda=1}^{N} \frac{(p_\lambda - m)^2}{\sigma^2} - \frac{2}{N} \sum_{\substack{\alpha=1 \\ \alpha \neq \beta}}^{N} \sum_{\beta=1}^{N} \frac{(p_\alpha - m)(p_\beta - m)}{\sigma^2} \right) \right] dp_1 dp_2 dp_3 \ldots dp_N \ . \tag{3.54}$$

The expression defined in (3.54) holds only for $N \gg 1$. Since the value of correlation coefficient varies as $0 \leq |\rho| \leq 1$, then as ρ increases, its value contributes exponentially into (3.54). Therefore, it can anticipate that the performance of MCS based majority vote combiner grows at an exponential rate as the correlation level between classifiers increases. For a special case, when $\rho = 0$ (outputs of classifiers are uncorrelated) the covariance matrix ($\boldsymbol{\Sigma}$) become a diagonal matrix with diagonal elements equal to σ^2, i.e.

$$\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}, \tag{3.55}$$

where I is $N \times N$ identity matrix then

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma^2}\mathbf{I} \quad \text{and} \quad \sqrt{|\boldsymbol{\Sigma}|} = \sigma^N . \tag{3.56}$$

Substituting (3.55) and (3.56) in (3.48) results in

$$p_e = \frac{1}{(2\pi)^{N/2}\sigma^N} \sum_{k=\frac{N+1}{2}}^{N} \binom{N}{k} \underbrace{\left(\prod_{i=1}^{k} \int_{-\infty}^{0.5} \right)}_{k-terms} \underbrace{\left(\prod_{\mathrm{j=k+1}}^{\mathrm{N}} \int_{0.5}^{\infty} \right)}_{(N-k)-terms},$$

$$\times \exp\left[-\frac{1}{2}\left(\frac{(p_1 - m)^2}{\sigma^2} + \frac{(p_2 - m)^2}{\sigma^2} + \ldots + \frac{(p_N - m)^2}{\sigma^2} \right) \right] dp_1 dp_2 dp_3 \ldots dp_N \quad (3.57)$$

It is also possible to get (3.57) by substituting $\rho = 0$ into (3.49). By making a little arrangement in (3.57), results in

$$p_e = \frac{1}{(2\pi)^{N/2}\sigma^N} \sum_{k=\frac{N+1}{2}}^{N} \binom{N}{k} \underbrace{\left(\prod_{i=1}^{k} \int_{-\infty}^{0.5} \exp\left(-\frac{1}{2} \frac{(p_i - \mathrm{m})^2}{\sigma^2} \right) dp_i \right)}_{k-terms} \underbrace{\left(\prod_{\mathrm{j=k+1}}^{\mathrm{N}} \int_{0.5}^{\infty} \exp\left(-\frac{1}{2} \frac{(p_j - \mathrm{m})^2}{\sigma^2} \right) dp_j \right)}_{(N-k)-terms}, \quad (3.58)$$

equation (3.58) describes the performance of MCS for $\rho = 0$, which is equivalent to the model derived in reference [31] (eq. 25). The expression derived in (3.40) or (3.49) can be viewed as a generalized version of the formula derived in reference [31]. When removing dependency condition among the classifiers' outputs, both models match in performance as presented in the next section. Therefore, the model defined in (3.49) is considered as a tool that helps in the analysis of MCS using majority voting combiner.

3.3.3. Results and Concluding Remarks

In this section, a model is verified in two stages and then discuss the results. In the first stage, the model defined in (3.48) is compared with the model proposed in [31]. Since expression defined in [31] is derived for uncorrelated classifiers, and for the comparison to be fair, the correlation coefficient in (3.48) is set to zero ($\rho = 0$). Figure 3.14 shows two plots one for the

expression defined in [31] and another is for the model defined in (3.48) by setting $\rho = 0$ where $N = 9$ and $m = 0.8$. As shown, both models give identical behavior. In the second stage, the derived model is needed to validate for different correlation values. To achieve this goal, a computer simulation program is built that depicts the stochastic behavior of MCS based majority voting combiner. N jointly normal random variables are generated, the level of dependence between these random variables is controlled by the covariance matrix given in (3.45). Each random variable is considered as an estimation of the posterior class probability, and each has normal distribution with a mean m and variance σ^2. Then outputs of base classifiers are combined using the majority voting rule. According to the voting rule, the ensemble outcome is considered as misclassifying the class if the number of classifiers in error is equal or more than $(N + 1)/2$. The probability of classification error is calculated as the ratio of the number of times that ensemble is in error to the total number of trials. In order to get accurate results with reasonable execution speed, the total number of iterations is chosen to be 100,000 times. Figure 3.14 shows a comparison between empirical and theoretical models. The comparison is done in terms of classification error probability against class standard deviation (σ) for $\rho = \{0, 0.25, 0.5\}$, $m = 0.8,$ and $N = 9$. It is clear a match in performance between computer simulation and the proposed theoretical model using three values of correlation coefficients.

In order to evaluate the performance of majority voting combiner, three plots are generated in figures 3.15, 3.16 and 3.17. These figures show the classification error probability as a function of the correlation coefficient for different m, σ and N values. Careful inspection of these figures, one can deduce that the performance of classifier ensemble degrades exponentially as the correlation coefficient increases. These results are expected and agree with the prediction given by (3.54). As the correlation among classifiers' outputs increases, more classifiers share the same information. As $\rho \to 1$, and for a given m and σ values the overall ensemble performance approaches the performance of a single classifier regardless of the number of base classifiers used in the ensemble (as shown in figure 3.17). Figures (figure 3.15, figure 3.16 and figure 3.17) also show that the sensitivity of ensemble performance against $m, \sigma \, or \, N$ varies. The ensemble is more sensitive to changes in σ (figure 3.15), i.e. the performance degraded exponentially with a linear increase in σ while it exhibits smoother behavior versus m and N.

Therefore, to get a low classification ensemble error, the value of σ must be kept at low as possible.

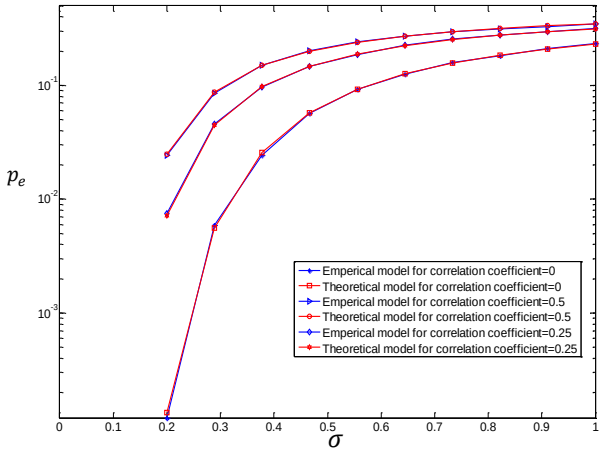

Figure 3.14. A Comparison in Term of Probability of Classification Error Between Model Derived in (3.48) and Simulated Model for Three Correlation Coefficient Values {0, 0.25, 0.5}, m=0.8 and N=9

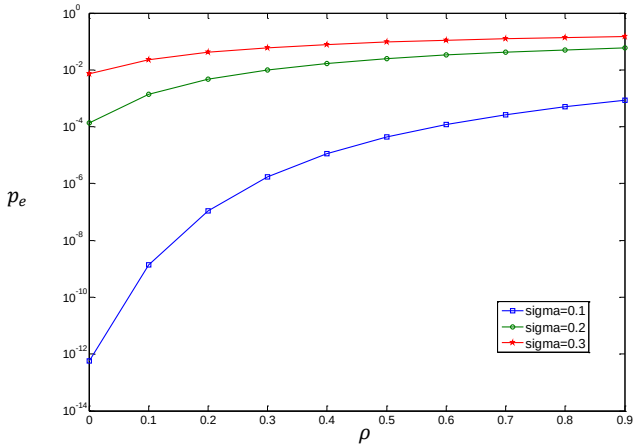

Figure 3.15. Probability of Classification Error as a Function of ρ , for σ = {0.1, 0.2, 0.3}, m=0.8 and N=9

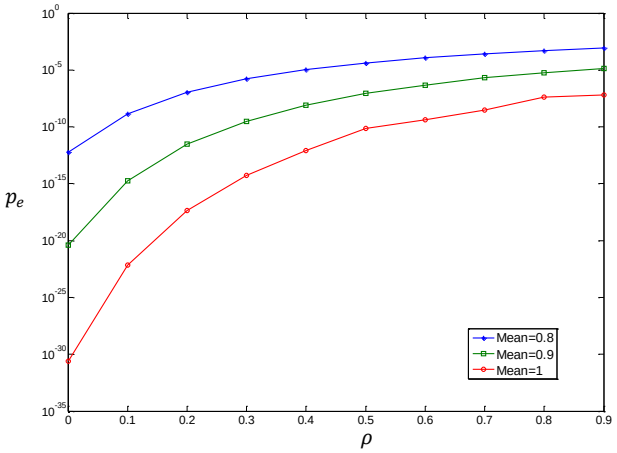

Figure 3.16. Probability of Classification Error as a Function of ρ, for m= {0.8, 0.9, 1} , $\sigma = 0.1$ and N=9

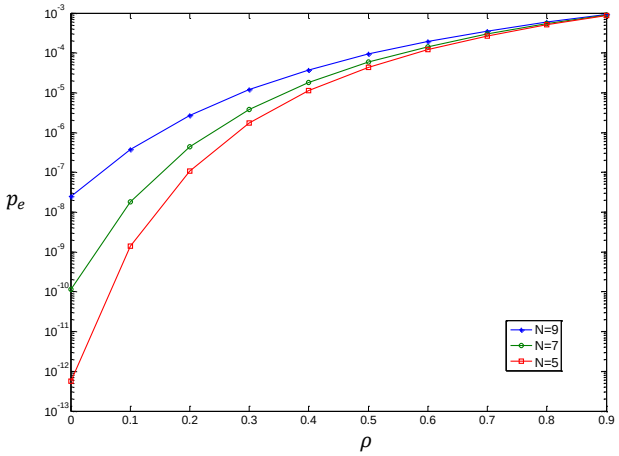

Figure 3.17. Probability of Classification Error as a Function of ρ, For N= {5, 7, 9} , M=0.8 and $\sigma = 0.1$

To provide a generalized judgment on the performance of MCS based majority voting rule, figure 3.18 and figure 3.19 are generated which are two-dimensional plots of the ensemble performance in terms of misclassification error against the class mean and class standard deviation for two correlation values $\rho = \{0, 0.5\}$. As shown in figure 3.19 ($\rho = 0$) a poor ensemble performance for $\sigma > 0.2$ region is obvious. It is also clear from figure 3.18 that the ensemble shows sensitive performance against varying σ while it exhibits a smoother behavior with variations in m. Similarly, figure 3.19 shows the performance of the ensemble for $\rho = 0.5$. In which it is evident that the effect of correlation between classifiers output increases the level of the region of low classification error probability to higher values.

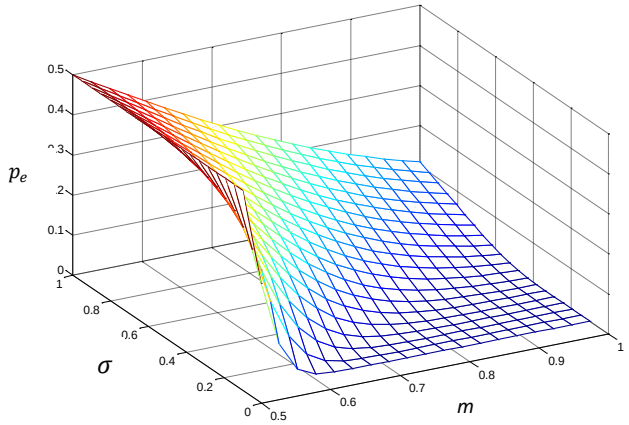

Figure 3.18. Two Dimensional Plot of Probability of Classification Error as a Function of σ and m for ρ=0 and N=9

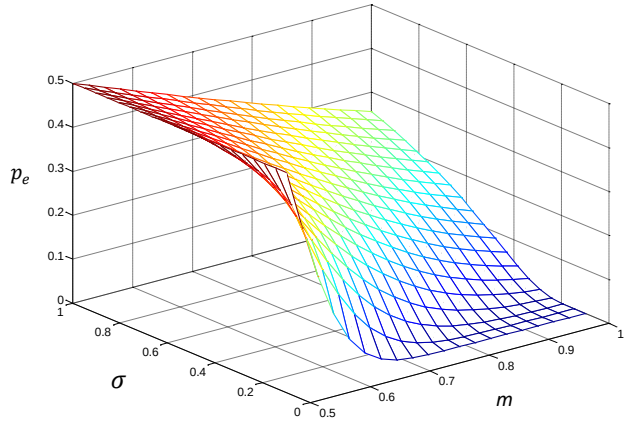

Figure. 3.19. Two-Dimensional Plot of Probability of Classification Error as a Function of σ and m for ρ=0.5 and N=9

3.3.4. Section Conclusion

Combining rules are considered as one of the crucial layers in the design process of a multiple classifier system. The estimation of classification error probability has been proposed through many models assuming independence among classifiers' outputs. However, since classifiers exhibit dependent behavior, a theoretical model is derived for estimating the performance of MCS using majority voting as a combining rule. The derivation considered that the classifiers' outputs are dependent, normal and identically distributed and for two classes case.

To validate the mathematical model, computer simulations is used and verifications result confirmed the validity of the proposed model. It also has shown that the performance of the classifier ensemble decreases exponentially as the correlation coefficient increases. To get the benefit of using MCS in a correlated condition, many ensemble parameters should be optimized properly. Values of $\rho, \sigma \; and \; m$ are the key factors in the ensemble design. For example, to get acceptable classification accuracy ($< 10^{-3}$) in a highly correlated environment with $\rho = 0.5$. The class mean should be more than 0.7 ($m > 0.7$), and the class standard deviation must be less than 0.2 ($\sigma < 0.2$). The proposed model serves as an investigation of the performance of MCS using majority voting rule and brings significant insights.

CHAPTER IV

AN ANALYTICAL FRAMEWORK FOR WEIGHTED FUSION RULES IN COMBINED CLASSIFIER SYSTEMS

4.1. Introduction

Fusion rules in a Multiple Classifier System (MCS) are considered one of the major design components in improving its classification accuracy. In this work, performance results of four weighted combining rules are estimated and compared which are: geometric mean, majority vote, average and harmonic mean. The derivations are based on assumptions adapted from Kuncheva framework [31] that stated individual classifiers outputs are an estimation of their individual posterior class probability, and they are correlated and normally distributed. Also, a systematic way to estimate the weights of individual classifiers is developed based on their class mean and variance. The results show that the ensemble performance degrades exponentially as increases the correlation level among classifiers outputs. In fact, as the correlation coefficient value approaches one, the ensemble reaches the performance of a single classifier, which is shown mathematically. Upon comparing the ensemble performance against class mean, class variance, correlation coefficient and number of classifiers, it found that the ensemble performance agrees with the principle of the no-free-lunch theorem in which each combining rule works for a given set of ensemble parameter. For a given ensemble condition, this study allow us to better understand of the strengths and weaknesses of each rule and thus to choose an appropriate rule that minimized the ensemble error. The proposed analytical models can be used as tools for the analysis and prediction of the performance of multiple classifier systems.

4.2. Analytical Analysis

In this section, analytical expressions for estimating the probability of classification error is developed using four weighted combining rules. It assumed that there are N classifiers that work in parallel $\{C_1, C_2, \dots, C_N\}$, each classifier classifies data into M classes and each classifier's output is considered as an estimation of the class posterior probability represented by $d_{i,j}(x) \in [0,1]$ with weight $w_{i,j}$ related directly to a classifier's accuracy where $i = 1, 2, \dots, N$ and

$j = 1,2, \dots, M$. In addition, it assumed that $d_{i,j}(x)$ has a normal distribution in which the class mean is $p(\omega_j/x)$, and so we have

$$d_{i,j} = p_i(\omega_j/x) + \eta_{i,j}(x). \tag{4.1}$$

The term $p_i(\omega_j/x)$ represents the true class posterior probability density function for a given input feature vector in which $x \in R^K$ and $\eta_{i,j}(x)$ is a normal random variable with zero mean and σ^2 variance, the justifications for the previous assumptions are given in [31] and [41]. To simplify the derivations, it considered the case of two classes ($M = 2$), then (2) is reduced to a column vector i.e. $\mathbf{W} = [w_1, w_2, \dots, w_N]^T$, $d_{i,1}(x) + d_{i,2}(x) = 1$, for $i = 1,2, \dots N$. It is possible to extend derivations for M classes, but it requires including more assumptions about other variables. Also let $p_i = d_{i,1}$ and $\bar{p}_i = d_{i,2} = 1 - d_{i,1} = 1 - p_i$, and then the fused output for class ω_1 is $\hat{p}_1 = \hat{p}(\omega_1/x) = \Psi\{(p_1, w_1), (p_2, w_2), \dots, (p_N, w_N)\}$ and $\hat{p}_2 = \hat{p}(\omega_2/x) = \Psi\{(\bar{p}_1, w_1), (\bar{p}_2, w_2), \dots, (\bar{p}_N, w_N)\}$ for ω_2. It also considered that classifiers outputs are normally distributed since it is the most common distribution that arises in many stochastic systems. However, the previous assumptions are applicable to any other distributions. In this section, the study is focused on the effects of correlation among N classfiers.

To define a systematic way for incorporating classifiers' weights into analytical models, it assumed that the sum of the weights of the classifiers to be 1, i.e. $\sum_{i=1}^{N} w_i = 1$ and w_i>0, then the weight of individual classifier is defined as

$$w_i = Q_i \Big/ \sum_{k=1}^{N} Q_k \,, \quad for\ i = 1,2, \dots, N \,, \tag{4.2}$$

where Q_k is the probability of correct classifications for the *kth* classifier, thus $Q_k = 1 - P_k$, P_k is the probability of classification error. For a normal posterior class distribution, the probability of classification error for base classifier is defined as, [31]

$$P_i = P(p_i < 0.5) = \Phi\left(\frac{0.5 - m_i}{\sigma_i}\right), \ where\, i = 1,2,...,N\,, \tag{4.3}$$

where m_i and σ_i^2 are the mean and variance of the posterior class random variable p_i and $\Phi(\cdot)$ is the cumulative distribution function of p_i. Using (4.3), it can rewrite (4.2) as

$$w_i = \left(1 - \Phi\left(\frac{0.5 - m_i}{\sigma_i}\right)\right) \Big/ \sum_{k=1}^{N}\left(1 - \Phi\left(\frac{0.5 - m_k}{\sigma_k}\right)\right), \text{ where } \ i = 1,2,....,N\,. \tag{4.4}$$

The expression in (4.4) defines the weight of the *ith* classifiers based on its class variance (σ_i^2) and mean (m_i). In the following derivations, a formula for each combining rule and under independent classifiers condition is proposed and then generalized the results for the correlated classifiers outputs. analytical models for each rule is briefly described. In the subsequent derivations SG, SM, SA and SH are referred to as Simple Geometric, Simple Majority vote, Simple Average and Simple Harmonic mean respectively, while WG, WM, WA and WH are referred to Weighted Geometric, Weighted Majority vote, Weighted Average and Weighted Harmonic mean respectively.

4.3. Weighted Geometric Mean Rule (WG)

The work defined in [36], showed that the product rule exhibits poor performance compared to the geometric mean. Therefore, it would not be included in the discussion. In order to propose a suitable definition for WG, the weight is chosen to be the power of estimated class posterior probability (p_i). This definition means that the classifiers' weights scale the contribution of each classifier's output to the overall performance according to their individual accuracies. The rule used to define the weighted geometric mean for two classes is defined as follows

$$\hat{p}_1 = \prod_{i=1}^{N} p_i^{\,w_i} \quad . \tag{4.5}$$

In case of equal classifiers' weights, i.e. $w_i = 1/N$, the expression in (4.5) is modified to $\hat{p}_1 = \prod_{i=1}^{N} p_i^{1/N}$, which represents a simple geometric mean. In order to calculate the classification error of the weighted geometric rule, the probability density function of $\hat{p}_1$ should be estimated. The problem of finding the probability density function of the product of N random variables is extensively studied in different fields such as statistics and engineering [44], [45]. In this work, an accurate probabilistic method is proposed that can handle the product of N dependent random variables and estimate the resulting distribution. By taking the natural logarithm of both sides of (4.5), results in

$$\log(\hat{p}_1) = \sum_{i=1}^{N} w_i \log(p_i). \tag{4.6}$$

The probability density function of $\log(\hat{p}_1)$ defined in (4.6) approaches a normal distribution as N becomes larger, and the mean and variance of $\log(\hat{p}_1)$ are given by

$$E[\log(\hat{p}_1)] = \sum_{i=1}^{N} w_i E[\log(p_i)], \tag{4.7}$$

$$VAR[\log(\hat{p}_1)] = \sum_{i=1}^{N} w_i^2 VAR[\log(p_i)]. \tag{4.8}$$

where

$$E[\log(p_i)] = m_{\log(p_i)} = \int_0^{\infty} \log(p_i) f(p_i) dp_i, \text{ and} \tag{4.9}$$

$$VAR[\log(p_i)] = \sigma^2_{\log(p_i)} = \left[\int_0^{\infty} (\log(p_i))^2 f(p_i) dp_i - m^2_{\log(p_i)} \right] \tag{4.10}$$

where $f(p)$ is the probability density of p_i. There is no compact expression for $m_{\log(p_i)}$ and $\sigma^2_{\log(p_i)}$ but they can be solved numerically. In order to get useful analytical expressions that explain how ensemble's parameters optimize its performance, the moments defined in (4.7) and (4.8) are approximated using Taylor series expansion around m_i as follows

$$E[\log(\hat{p}_1)] = \sum_{i=1}^{N} w_i m_{\log(p_i)} \approx \sum_{i=1}^{N} w_i \left[\log(m_i) - \frac{\sigma_i^2}{2m_i^2} \right], \tag{4.11}$$

$$VAR[\log(\hat{p}_1)] = \sum_{i=1}^{N} w_i^2 \sigma_{\log(p_i)}^2 \approx \sum_{i=1}^{N} w_i^2 \frac{\sigma_i^2}{m_i^2}. \tag{4.12}$$

The previous approximation in (4.11) is held over a wide range of values for m_i and σ_i, while (4.12) is valid when $\sigma_i \ll m_i$, [36]. The exact value for $m_{\log(pi)}$ and $\sigma_{\log(pi)}^2$ are given in (4.9) and (4.10) respectively and the approximated values are $m_{\log(p_i)} \approx [\log(m_i) - \sigma_i^2/2m_i^2]$ and $\sigma_{\log(p_i)}^2 \approx \sigma_i^2/m_i^2$. To find the distribution of $\hat{p}_1$, take the exponent of both sides of (4.6). It is known from probability theory that if X and Y are two random variables defined as $Y = \log(X)$ and Y has a normal distribution, then X has lognormal distribution, and its cumulative distribution is given by

$$F_X(x) = \Phi\left(\frac{\log(x) - m}{\sigma} \right). \tag{4.13}$$

Then using (4.7) – (4.13), the average probability of classification error for weighted geometric mean is defined as

$$P_e = P(\hat{p}_1 < 0.5) = \Phi\left(\left[\log(0.5) - \sum_{i=1}^{N} w_i m_{\log(p_i)} \right] \Big/ \sqrt{\sum_{i=1}^{N} w_i^2 \sigma_{\log(p_i)}^2} \right)$$

$$\approx \Phi\left(\left[\log(0.5) - \sum_{i=1}^{N} w_i \left(\log(m_i) - \frac{\sigma_i^2}{2m_i^2} \right) \right] \Big/ \sqrt{\sum_{i=1}^{N} w_i^2 \frac{\sigma_i^2}{m_i^2}} \right). \tag{4.14}$$

The previous derivation is based on the assumption that classifiers are uncorrelated, in the case of correlated classifiers, the correlation coefficient among classifier outputs is defined as $\rho_{k,l}$ where $k = 1,2,\dots,N$ and $l = 1,2,\dots,N$ given that $k \neq l$. The mean of the random variable

$(\log(\hat{p}_1))$ remains the same as defined in (4.11) while the variance is calculated as follows

$$VAR\left[\log(\hat{p}_1)\right] = E[(\log(\hat{p}_1) - E[\log(\hat{p}_1)])^2],$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j E[(\log(p_i) - E[\log(p_i)]) \times (\log(p_j) - E[\log(p_j)])],$$

$$= \sum_{k=1}^{N} w_k^2 \sigma^2_{\log(p_k)} + \sum_{\substack{i=1 \\ i\neq j}}^{N}\sum_{j=1}^{N} w_i w_j \rho_{i,j} \sigma_{\log(p_i)} \sigma_{\log(p_j)}. \tag{4.15}$$

Then the average classification error probability is

$$P_e = \Phi\left(\frac{\left[\log(0.5) - \sum_{i=1}^{N} w_i m_{\log(p_i)}\right]}{\sqrt{\sum_{k=1}^{N} w_k^2 \sigma^2_{\log(p_k)} + \sum_{\substack{i=1 \\ i\neq j}}^{N}\sum_{j=1}^{N} w_i w_j \rho_{i,j} \sigma_{\log(p_i)} \sigma_{\log(p_j)}}} \right),$$

$$\approx \Phi\left(\frac{\left[\log(0.5) - \sum_{i=1}^{N} w_i \left(\log(m_i) - \frac{\sigma_i^2}{2m_i^2}\right)\right]}{\sqrt{\sum_{k=1}^{N} w_k^2 \frac{\sigma_k^2}{m_k^2} + \sum_{\substack{i=1 \\ i\neq j}}^{N}\sum_{j=1}^{N} w_i w_j \rho_{i,j} \frac{\sigma_i \sigma_j}{m_i m_j}}} \right). \tag{4-16}$$

The expression in (4.16) provides us with an estimate of the ensemble performance based on weighted geometric mean and correlated classifiers. A closer look at (4.14) or (4.16) reveals that the class mean is inversely related to the variance by a factor of $1/m^2$. A key point in the ensemble design is as follows: In classifier training phase, if there is no control over minimizing the class variance, then it is better to construct an ensemble with classifiers that have as large a class mean as possible. This strategy will reduce the overall ensemble error. Then, it expected the

performance of geometric mean to improve as $m_i \rightarrow 1$, where the improvement rate is $1/m_i^2$. To consider a special case when all base classifiers have equal strengths then $w_i = 1/N$, $m_1 = m_2 = m_i \ldots = m_N = m$, $\sigma_1^2 = \sigma_2^2 = \sigma_i^2 \ldots = \sigma_N^2 = \sigma^2$, $\rho_{i,j} = \rho$ (where $i \neq j$), then (20) simplifies to

$$P_e \approx \Phi\left(\frac{\left[\log(0.5) - \left(\log(m) - \frac{\sigma^2}{2m^2}\right)\right]}{\sqrt{\frac{1}{N}\frac{\sigma^2}{m^2} + \frac{1}{N}(N-1)\rho\frac{\sigma^2}{m^2}}}\right). \tag{4.17}$$

Based on the assumption which stated that $m \gg \sigma$ and for fully correlated classifiers ($\rho = 1$), it can rewrite (4.17) as follows

$$P_e \approx \Phi\left(\frac{\log(1/2m)}{\sigma/m}\right). \tag{4.18}$$

In order to simplify (4.18) more $\log(x = 1/2m)$ is expanded using Taylor series around $x = 1$ then $\log(x) = (x-1) - (1/2)(x-1)^2 + (1/3)(x-1)^3 + \cdots$. If the value of x is chosen close to 1 i.e. $m \approx 0.5$ then $\log(x) \approx (\mathrm{x} - 1)$ and (4.18) modified to

$$P_e \approx \Phi\left(\frac{0.5 - m}{\sigma}\right). \tag{4.19}$$

This is an interesting result since (4.19) is identical to (4.3) which is the probability of classification error for a single base classifier. The equation defined in (4.19) suggests that there is no benefit in creating an ensemble of identical classifiers because the overall ensemble performance will be equivalent to a performance of a single classifier.

4.4. Weighted Majority Voting Rule (WM)

In the majority voting rule, three existing algorithms are available. The first, called unanimous voting, requires that all classifiers agree on a chosen class. The second, known as simple majority, requires that at least one more than half of the number of classifiers agree. The third, called majority voting, in which a class is chosen if it gets the largest number of votes. Among these algorithms, majority voting is considered an optimal rule compared to the others. In this section, it assumed classifiers have different accuracies, therefore a modified version of SM is used called weighted majority voting. The formula for WM given the weight of each classifier is w_i is defined as follows

$$\sum_{i=1}^{N} w_i d_{i,j} = \max_{j=1}^{M} \sum_{i=1}^{N} w_i d_{i,j} \,. \tag{4.20}$$

The probability of classification error for the simple voting rule which have equal weights and independent classifiers is derived in [31]. In case of WM, the situation becomes more complicated. For N base classifiers, $\mathbf{W} = [w_1, w_2, \ldots, w_N]^T$ and $\mathbf{P} = [P_1, P_2, \ldots, P_N]^T$, where $\mathbf{W}$ is the weight vector of individual classifiers in an ensemble and $\mathbf{P}$ is the vector of misclassification probability for each classifier as defined in (4.3). In order to estimate the probability of classification error for an ensemble based on weighted majority vote, a formula that generates a probability distribution for N classifiers is proposed. The following expression achieves this goal

$$P(k,N) = \sum_{l=0}^{\binom{N}{k}-1} \left(\prod_{i=l}^{l+k-1} P_{(i \oplus N)+1} \prod_{j=l+k}^{l+N-1} \left(1 - P_{(j \oplus N)+1}\right) \right), \tag{4.21}$$

where $P(k,N)$ is the sum of product of individual classifications' accuracies for given values of k and N and the symbol $\oplus$ is a modulo N addition. The expression defined in (4.21) covers all permutations of individual classifiers' accuracies; k represents the number of individual classifiers in the ensemble that are in error. Then using (4.3), and (4.21) to compute the

probability of classification error for N classifiers when the number of base classifiers in error is greater than or equal to α i.e.,

$$\alpha = \begin{cases} \frac{N}{2}+1 & when\ N\ is\ even \\ \frac{N+1}{2} & when\ N\ is\ odd \end{cases}, \tag{4.22}$$

then the probability of classification error is written as

$$P_e = \sum_{k=\alpha}^{N} P(k,N) = \sum_{k=\alpha}^{N} \left[\binom{N}{k}^{-1} \sum_{l=0}^{} \left[\prod_{i=l}^{l+k-1} \Phi\left(\frac{0.5 - m_{(i\oplus N)+1}}{\sigma_{(i\oplus N)+1}} \right) \times \prod_{j=l+k}^{l+N-1} \left(1 - \Phi\left(\frac{0.5 - m_{(j\oplus N)+1}}{\sigma_{(j\oplus N)+1}} \right) \right) \right] \right]$$

$$\times \prod_{j=l+k}^{l+N-1} \left(1 - \Phi\left(\frac{0.5 - m_{(j\oplus N)+1}}{\sigma_{(j\oplus N)+1}} \right) \right) \Bigg] \Bigg], \tag{4.23}$$

The expression given in (4.23) is applicable to the simple majority vote in case of different classifiers' weights. It requires to modify (4.23) in order to be applicable to the weighted majority vote rule. In order to clarify the idea, an example of three classifiers ($N = 3$) is considered with a weight vector $\mathbf{W} = [0.6\ 0.3\ 0.1]^T$, and then from (4.23), the terms that account for classification error are

$$P_e = P_1P_2P_3 + P_1P_2Q_3 + P_2P_3Q_1 + P_3P_1Q_2, where \quad Q_i = 1 - P_i, \tag{4.24}$$

Based on weighted majority rule, the term $P_2P_3Q_1$ should be excluded from the summation in (4.24), since the weight of classifier#1 is 0.6 and the total weights of classifier#2 and classifier#3 is 0.4. Therefore, the majority decision goes to classifier#1, which represents the correct classification. Therefore, it is necessary to include an extra function that identifies these terms and removes them from the overall probability of classification error. This function is

labeled as $\delta_{k,l}(w)$ and defined as

$$\delta_{k,l}(w)=\begin{cases}0 & if \quad \sum_{m=l+k}^{l+N-1} w_{(m\oplus N)+1} \geq \frac{1}{2} \\ 1 & \text{elsewhere}\end{cases}, \text{where } l=0,1,2,\ldots,\binom{N}{k}-1, \tag{4.25}$$

Based on the previous discussion and using (4.22) and (4.25), the overall average probability of classification error for the weighted majority vote under the condition of uncorrelated classifiers is defined as

$$P_e=\sum_{k=\alpha}^{N}\left[\sum_{l=0}^{\binom{N}{k}-1}\left[\prod_{i=l}^{l+k-1}\Phi\left(\frac{0.5-m_{(i\oplus N)+1}}{\sigma_{(i\oplus N)+1}}\right)\right.\right.$$

$$\left.\left.\times\prod_{j=l+k}^{l+N-1}\left(1-\Phi\left(\frac{0.5-m_{(j\oplus N)+1}}{\sigma_{(j\oplus N)+1}}\right)\right)\delta_{k,l}(w)\right]\right]. \tag{4.26}$$

To derive an expression for the probability of classification error in case of correlated classifiers, a different approach is followed because the previous method is not applicable to an ensemble that used majority voting rule. For N correlated classifiers, the probability of classification error can be written as

$$P_e=\sum_{k=\alpha}^{N}\sum_{l=0}^{\binom{N}{k}-1}F\left(\underbrace{p_{(l\oplus N)+1}\leq 0.5,\ p_{(l+1\oplus N)+1}\leq 0.5,\ldots,p_{(l+k-1\oplus N)+1}\leq 0.5,}_{k\text{-terms}}\right.$$

$$\left.\times F\left(\underbrace{p_{(l\oplus N)+1}\leq 0.5,\ p_{(l+1\oplus N)+1}\leq 0.5,\ldots,p_{(l+k-1\oplus N)+1}\leq 0.5,}_{k\text{-terms}}\right.\right.$$

$$\underbrace{p_{(l+k\oplus N)+1} > 0.5,\ p_{(l+k+1\oplus N)+1} > 0.5, \ldots, p_{(l+N-1\oplus N)+1} > 0.5}_{(N-k)\text{-}terms}\Bigg)\delta_{k,l}(w), \quad (4.27)$$

where $F(\cdot)$ is the joint cumulative distributions for N classifiers outputs and $\delta_{k,l}(w)$ is defined as in (4.25). If it is assumed that the classifiers outputs are normally distributed, then the joint probability density function of N classifiers is defined as follows

$$f(\mathbf{P}) = f(p_1, p_2, \ldots, p_N) = \frac{\exp\left(-\frac{1}{2}(\mathbf{P}-\boldsymbol{\Delta})^T \mathbf{K}^{-1}(\mathbf{P}-\boldsymbol{\Delta})\right)}{(2\pi)^{N/2}|\mathbf{K}|^{1/2}}, \quad (4.28)$$

where $\mathbf{P}$ is a vector of random variables that represents classifiers outputs, Δ is the mean vector of $\mathbf{P}$ and $\mathbf{K}$ is the covariance matrix of $\mathbf{P}$. $\mathbf{P}$, Δ and $\mathbf{K}$ are defined in (4.29) and (4.30) respectively

$$\mathbf{P} = \begin{bmatrix} p_{(l\oplus N)+1} \\ p_{(l+1\oplus N)+1} \\ \cdot \\ p_{(l+k-1\oplus N)+1} \\ \cdot \\ p_{(l+N-1\oplus N)+1} \end{bmatrix}, \ \boldsymbol{\Delta} = \begin{bmatrix} E[p_{(l\oplus N)+1}] \\ E[p_{(l+1\oplus N)+1}] \\ \cdot \\ E[p_{(l+k-1\oplus N)+1}] \\ \cdot \\ E[p_{(l+N-1\oplus N)+1}] \end{bmatrix} = \begin{bmatrix} m_{(l\oplus N)+1} \\ m_{(l+1\oplus N)+1} \\ \cdot \\ m_{(l+k-1\oplus N)+1} \\ \cdot \\ m_{(l+N-1\oplus N)+1} \end{bmatrix} \quad (4.29)$$

The dimension of the covariance matrix defined in (4.30) is $N \times N$, as a result its size expands excessively with N. For the purpose of implementing (4.27) efficiently a matrix for a given k and l values is generated then using modulo N property to get other matrices as k and l values varies.

$$\mathbf{K}_{k,l} = \begin{bmatrix} \sigma^2_{(l\oplus N)+1} & \rho_{(l\oplus N)+1,(l+1\oplus N)+1}\sigma_{(l\oplus N)+1}\sigma_{(l+1\oplus N)+1} & \cdots & \rho_{(l\oplus N)+1,(l+N-1\oplus N)+1}\sigma_{(l\oplus N)+1}\sigma_{(l+N-1\oplus N)+1} \\ \rho_{(l+1\oplus N)+1,(l\oplus N)+1}\sigma_{(l+1\oplus N)+1}\sigma_{(l\oplus N)+1} & \sigma^2_{(l+1\oplus N)+1} & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \rho_{(l+N-1\oplus N)+1,(l\oplus N)+1}\sigma_{(l+N-1\oplus N)+1}\sigma_{(l\oplus N)+1} & \cdot & \cdots & \sigma^2_{(l+N-1\oplus N)+1} \end{bmatrix}_{N\times N}$$

(4.30)

Using (4.27) through (4.30), the average probability of the classification error for N correlated classifiers is estimated as follows

$$P_e = \frac{1}{(2\pi)^{N/2}} \sum_{k=\alpha}^{N} \sum_{l=0}^{\binom{N}{k}-1} \frac{1}{\sqrt{|\mathbf{K}_{k,l}|}} \underbrace{\left(\prod_{i=l}^{l+k-1} \int_{-\infty}^{0.5} \right)}_{k-terms} \underbrace{\left(\prod_{j=l+k}^{l+N-1} \int_{0.5}^{\infty} \right)}_{(N-k)-terms}$$

$$\times \exp\left[-\frac{1}{2} (\mathbf{P} - \boldsymbol{\Delta})^T \mathbf{K}_{l,N}^{-1} (\mathbf{P} - \boldsymbol{\Delta}) \right] \delta_{k,l}(w)$$

$$\times dp_{(l \oplus N)+1} dp_{(l+1 \oplus N)+1} \ldots dp_{(l+N-1 \oplus N)+1}, \tag{4.31}$$

where $\delta_{k,l}(w)$, $\mathbf{P}$, Δ and $\mathbf{K}_{k,l}$ are defined in (4.25), (4.29) and (4.30) respectively. The k integration terms in (4.31) represent the probability of classification error, and $(N-k)$ terms represent the probability of correct classification. $(N-k)$ terms are considered to be in unity when $(l+k) > (l+N-1)$. Many algorithms in the literature, such as in [42], [43], are used to compute the integration of the multivariate normal function that was given in (4.31). For a special case, when classifiers outputs are uncorrelated, then substituting $\rho = 0$ in (4.31) results in

$$P_e = \frac{1}{\left((2\pi)^{N/2} \prod_{m=1}^{N} \sigma_m \right)} \sum_{k=\alpha}^{N} \sum_{l=0}^{\binom{N}{k}-1} \underbrace{\left(\prod_{i=l}^{l+k-1} \int_{-\infty}^{0.5} \right)}_{k-terms} \underbrace{\left(\prod_{j=l+k}^{l+N-1} \int_{0.5}^{\infty} \right)}_{(N-k)-terms}$$

$$\times \exp\left[-\frac{1}{2} \left(\left(\frac{(p_{(l \oplus N)+1} - m_{(l \oplus N)+1})}{\sigma_{(l \oplus N)+1}} \right)^2 + \left(\frac{(p_{(l+1 \oplus N)+1} - m_{(l+1 \oplus N)+1})}{\sigma_{(l+1 \oplus N)+1}} \right)^2 \right.\right.$$

$$\left.\left. + \ldots + \left(\frac{(p_{(l+N-1 \oplus N)+1} - m_{(l+N-1 \oplus N)+1})}{\sigma_{(l+N-1 \oplus N)+1}} \right)^2 \right) \right]$$

$$\times \delta_{k,l}(w) \times dp_{(l \oplus N)+1} dp_{(l+1 \oplus N)+1} \ldots dp_{(l+N-1 \oplus N)+1}. \tag{4.32}$$

Rewrite (4.32) results in

$$P_e = \frac{1}{\left((2\pi)^{N/2}\prod_{m=1}^{N}\sigma_m\right)}\sum_{k=\alpha}^{N}\sum_{l=0}^{\binom{N}{k}-1}$$

$$\times \underbrace{\left[\prod_{i=l}^{l+k-1}\int_{-\infty}^{0.5}\exp\left(-\frac{1}{2}\left(\frac{(p_{(i\oplus N)+1}-\mathrm{m}_{(i\oplus N)+1})}{\sigma_{(i\oplus N)+1}}\right)^2\right)dp_{(i\oplus N)+1}\right]}_{k-terms}$$

$$\times \underbrace{\left[\prod_{\mathrm{j}=l+k}^{l+N-1}\int_{0.5}^{\infty}\exp\left(-\frac{1}{2}\left(\frac{(p_{(j\oplus N)+1}-\mathrm{m}_{(j\oplus N)+1})}{\sigma_{(j\oplus N)+1}}\right)^2\right)dp_{(j\oplus N)+1}\right]}_{(N-k)-terms}\times\delta_{k,l}(w). \qquad (4.33)$$

The expression defined in (4.33) is identical to (4.26) since both are derived for uncorrelated classifiers, though they use different methodologies. In case of equal classifiers' weights then $w_i = 1/N$, $m_{\mathrm{i}} = m$, $\sigma_{\mathrm{i}}^2 = \sigma^2$, , $\delta_{k,l}(w) = 1$, and $p_i = p$ for $i = 1,2,\ldots,N$, then (4.33) is modified to

$$P_e = \frac{1}{(2\pi)^{N/2}\sigma^N}\sum_{k=\alpha}^{N}\binom{N}{k}\left(\int_{-\infty}^{0.5}\exp\left(-\frac{1}{2}\frac{(p-\mathrm{m})^2}{\sigma^2}\right)dp\right)^k$$

$$\times\left(\int_{0.5}^{\infty}\exp\left(-\frac{1}{2}\frac{(p-\mathrm{m})^2}{\sigma^2}\right)dp\right)^{N-k}. \qquad (4.34)$$

Equation (4.34) is identical to the formula (25) derived in reference [31] since both are derived for equal strength and uncorrelated classifiers. As a result, (4.31) can be considered a useful formula in analyzing the performance of a MCS based on weighted majority voting and under the condition of correlated classifiers.

4.5. Weighted Average Rule (WA)

SA rule is simply taking the average sum of classifiers outputs. The benefit of this structure is to reduce the random fluctuations in classifiers outputs and to get an estimated value that is close to the actual class posterior probability. This rule works when all classifiers have equal strength. If classifiers have different weights, then the weighted average rule would be the best choice for improving classification accuracy.

The weighted average rule for two classes' problem is defined as

$$\hat{p}_1 = \sum_{i=1}^{N} w_i p_i \ , \tag{4.35}$$

where $\hat{p}_1$ is the estimated posterior probability for class ω_1 and w_i is the *ith* classifier weight.

The moments of $\hat{p}_1$ for independent classifiers' output are defined as follows

$$E[\hat{p}_1] = \sum_{i=1}^{N} w_i m_i \ , where\ E[p_i] = m_i \ , \tag{4.36}$$

$$VAR[\hat{p}_1] = \sum_{i=1}^{N} w_i^2 \sigma_i^2 , \ where\ VAR[p_i] = \sigma_i^2 . \tag{4.37}$$

The distribution of $\hat{p}_1$ approaches a normal distribution as N become large, therefore the probability of classification error is calculated as

$$P_e = \Phi\left((0.5 - \sum_{i=1}^{N} w_i m_i) \Big/ \sqrt{\sum_{i=1}^{N} w_i^2 \sigma_i^2} \right). \tag{4.38}$$

To find a formula for classification error under correlated classifiers outputs, the variance of $\hat{p}_1$ is modified to

$$VAR[\hat{p}_1] = \sum_{k=1}^{N} w_k^2 \sigma_k^2 + \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} w_i w_j \rho_{i,j} \sigma_i \sigma_j \tag{4.39}$$

Using (4.39), the average classification error probability is estimated for correlated classifiers as

$$P_e = \Phi\left(\frac{\left(0.5 - \sum_{i=1}^{N} w_i m_i \right)}{\left(\sqrt{\sum_{k=1}^{N} w_k^2 \sigma_k^2 + \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} w_i w_j \rho_{i,j} \sigma_i \sigma_j} \right)} \right). \tag{4.40}$$

In (4.40), a case study is considered in which all classifiers have equal strengths then (4.40) is reduced to

$$P_e = \Phi\left((0.5 - m) \Big/ \left(\sqrt{\frac{1}{N}\sigma^2 + \frac{1}{N}(N-1)\rho\sigma^2} \right) \right). \tag{4.41}$$

If all classifiers assumed to be correlated with $\rho = 1$, then (4.41) simplifies as

$$P_e = \Phi\left(\frac{0.5 - m}{\sigma} \right). \tag{4.42}$$

The result given in (4.42) confirms the conclusion given in section 4-3 in which creating an ensemble of identical classifiers will not improve the classification accuracy.

4.6. Weighted Harmonic Mean Rule (WH)

The harmonic mean is a member of the Pythagorean means family which also includes the average mean and geometric mean. The SH is defined as follows [46]

$$SH = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{1}{p_i} \right)^{-1}. \tag{4.43}$$

If the value of p_i is limited to positive real values, then it is possible to express the harmonic mean in terms of average and geometric mean as follows [46]

$$SH = \frac{N \prod_{i=1}^{N} p_i}{\sum_{i=1}^{N} \frac{\prod_{j=1}^{N} p_j}{p_i}} = \frac{SG^N}{SA\left(\frac{\prod_{i=1}^{N} p_i}{p_1}, \frac{\prod_{i=1}^{N} p_i}{p_2}, \ldots, \frac{\prod_{i=1}^{N} p_i}{p_N} \right)}. \tag{4.44}$$

As shown in (4.44) the harmonic mean is closely related to the SG and SA. For $p_i > 0$ the averaging process of the harmonic mean rank always the last among others i.e. $p_{min} \leq SH \leq SG \leq SA \leq p_{max}$. The weighted harmonic is defined as

$$\hat{p}_1 = \left(\sum_{i=1}^{N} \frac{w_i}{p_i} \right)^{-1}. \tag{4.45}$$

Assuming the distribution of p_i as normal, the moments of $(1/p_i)$ are derived as follows

$$E\left[\frac{1}{p_i} \right] = m_{1/pi} = \int_{-\infty}^{\infty} \frac{1}{p_i} f(p_i) dp_i \approx \left[\frac{m_i^2 + \sigma_i^2}{m_i^3} \right], \tag{4.46}$$

$$VAR\left[\frac{1}{p_i} \right] = \sigma_{1/pi}^2 = \int_{-\infty}^{\infty} \frac{1}{p_i^2} f(p_i) dp_i - m_{1/p_i}^2 \approx \frac{\sigma_i^2}{m_i^4}, \tag{4.47}$$

where $f(p_i)$ is the probability density function of p_i. The approximation in (4.46) and (4.47) is

calculated using the Taylor series at m_i, where m_i and σ_i^2 are the first and second moments of p_i. The purpose of the approximations is to get more understanding about the behavior of WH as m_i and σ_i^2 varies. In order to find the distribution of $\hat{p}_1$, the distribution of the sum of (w_i/p_i) is approximate as a normal random variable as N become larger with first and second moments defined as

$$m_y = E\left[Y = \sum_{i=1}^{N} \frac{w_i}{p_i}\right] = \sum_{i=1}^{N} w_i m_{1/pi}\,, \quad \sigma_y^2 = VAR\left[Y = \sum_{i=1}^{N} \frac{w_i}{p_i}\right] = \sum_{i=1}^{N} w_i^2 \sigma_{1/pi}^2 \qquad (4.48)$$

The random variable $\hat{p}_1$ is inversely related to Y, then the event $\{Y \leq y\}$ occurs when $Y^{-1} \leq \hat{p}_1$ or $\hat{p}_1 \geq Y^{-1}$. Thus, the cumulative distribution of $\hat{p}_1$ is written as

$$F_{\hat{p}_1} = P[Y \geq \hat{p}_1^{-1}] = 1 - F_Y(\hat{p}_1^{-1})\,, \qquad (4.49)$$

and probability density function for $\hat{p}_1$ is

$$f_{\hat{p}_1} = \hat{p}_1^{-2} f_Y(\hat{p}_1^{-1})\,. \qquad (4.50)$$

Since the distribution of Y is normal, using (4.50) the probability density function of $\hat{p}_1$ is defined as follows

$$f_{\hat{p}_1} = \frac{1}{\hat{p}_1^2 \sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{1}{2}\frac{\left(\hat{p}_1^{-1} - m_y\right)^2}{\sigma_y^2}\right), \qquad (4.51)$$

where m_y and σ_y are defined in (4.48), using (4.49) and for independent classifiers, the probability of classification error is calculated as follows

$$P_e = P[\hat{p}_1 \leq 0.5] = 1 - \Phi\left(\left(2 - \sum_{i=1}^{N} w_i m_{1/pi}\right) \Big/ \sqrt{\sum_{i=1}^{N} w_i^2 \sigma_{1/pi}^2}\right)$$

$$\approx 1-\Phi\left(\left(2-\sum_{i=1}^{N} w_i\left[\frac{m_i^2+\sigma_i^2}{m_i^3}\right]\right)\Big/\sqrt{\sum_{i=1}^{N} w_i^2\,\frac{\sigma_i^2}{m_i^4}}\right). \qquad (4.52)$$

In case of correlated classifiers (4.52) is modified as

$$P_e = P[\hat{p}_1 < 0.5] = 1-\Phi\left(\frac{\left(2-\sum_{i=1}^{N} w_i m_{1/p_i}\right)}{\sqrt{\sum_{k=1}^{N} w_k^2\sigma_{1/p_k}^2 + \sum_{i=1}^{N}\sum_{\substack{j=1\\ i\neq j}}^{N} w_i w_j \rho_{i,j}\sigma_{1/p_i}\sigma_{1/p_j}}}\right),$$

$$\approx 1-\Phi\left(\frac{\left(2-\sum_{i=1}^{N} w_i\left[\frac{m_i^2+\sigma_i^2}{m_i^3}\right]\right)}{\sqrt{\sum_{k=1}^{N} w_k^2\,\frac{\sigma_k^2}{m_k^4} + \sum_{i=1}^{N}\sum_{\substack{j=1\\ i\neq j}}^{N} w_i w_j \rho_{i,j}\,\frac{\sigma_i}{m_i^2}\,\frac{\sigma_j}{m_j^2}}}\right). \qquad (4.53)$$

From (4.52) or (4.53), it can expect the performance of harmonic mean outperforms other rules for combining classifiers with high class mean ($m \approx 1$) because the overall variance is reduced at a rate of $1/m^4$, while the rule performs poorly for low class mean values ($m \approx 0.5$).

4.7. Results and Discussion

To provide a comprehensive assessment of the previous derivations the study is separated into two parts. The first part dealt with simple (unweighted) and correlated classifiers, i.e. the study is focused on the effects of correlation, class mean, class variance and number of base classifiers on the different combining rules. In the second part, the performance of weighted and correlated classifier conditions is evaluated. Regarding the first part, the un-weighted classifiers implies that all classifiers have equal weights which result in $w_i = 1/N$ for $i = 1,2,\ldots,N$. This means that classifiers outputs are identical random variables with equal mean and variance i.e.

$\sigma_1^2 = \sigma_2^2 = \sigma_i^2 \ldots = \sigma_N^2 = \sigma^2$, $m_1 = m_2 = m_i \ldots = m_N = m$ and $\rho_{i,j} = \rho$ (where $i \neq j$). For comparison purposes the ensemble parameters (σ, m, N, ρ) are chosen in a way that reflect the distinguishing behavior among different rules under comparison.

Figure 4.1 shows the performance of the four-combining rule (SA, SH, SG and SM) as a function of σ for $m = 0.7$, $\rho = 0.1$ and $N = 9$. As shown, the role of getting the best performance is changed over the range of σ values ($\sigma = [0.1, 0.5]$). For low σ values ($\sigma < 0.1$), SH achieved the best performance followed by SG, SA and SM.

The situation is changed when σ increases ($\sigma > 0.3$), where the SA outperforms all others. Figure 4.2 shows the performance as a function of m for $\sigma = 0.2$, $\rho = 0.1$ and $N = 9$. As shown, the performance gets better as m improved. This occurs because the error region between classes decreases. Again, a similar behavior for figure 4.1 appears in figure 4.2, where the best performance role of different rules is changed as class mean improves. These results agree with the predictions given in the previous sections in which the overall variance is reduced by a factor of $1/m^4$ and $1/m^2$ for SH and SG respectively. From figure 4.3 it's evidence that as correlation coefficient increases, the performance degrades exponentially. This is because as correlation coefficient increases, more classifiers share the same information about each other, resulting in degradation of ensemble performance. This behavior continues until $\rho = 1$, at this point all classifiers are similar to each other and overall performance mimics a single classifier system. It also found these results are consistent with predictions given by (4.19) and (4.42).

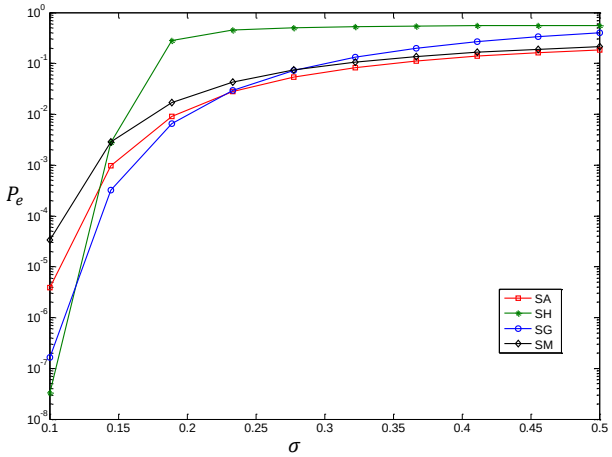

Figure 4.1. Probability of Classification Error as a Function of σ for SA, SH, SG and SM where $m = 0.7, N = 9$ and $\rho = 0.1$

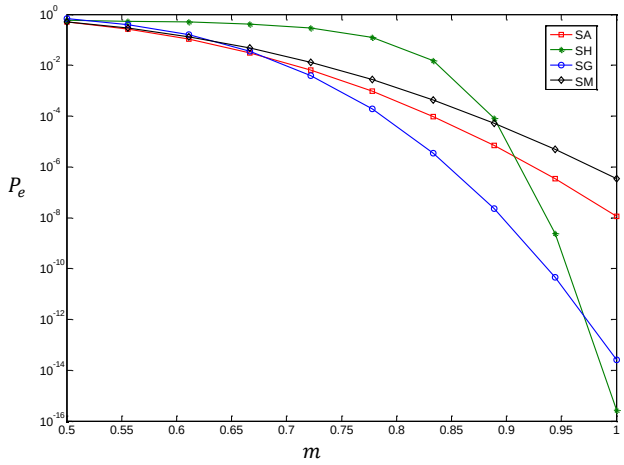

Figure 4.2. Probability of Classification Error as a Function of m for SA, SH, SG and SM where $\sigma = 0.2, N = 9$ and $\rho = 0.1$

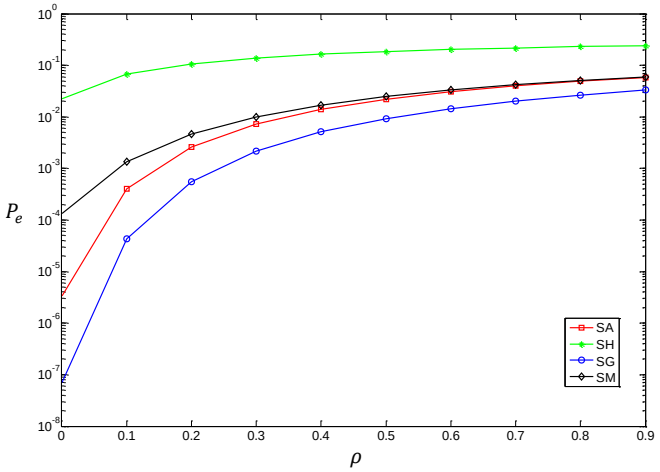

Figure 4.3. Probability of Classification Error as a Function of ρ for SA, SH, SG and SM where $\sigma = 0.2, N = 9$ and $m = 0.8$

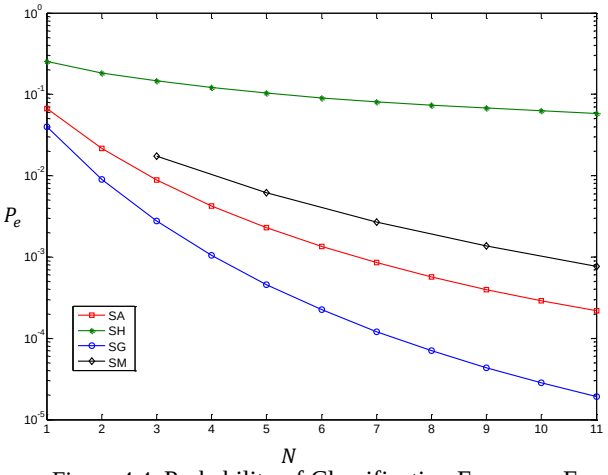

Figure 4.4. Probability of Classification Error as a Function of N for SA, SH, SG and SM where $\sigma = 0.2$, $\rho = 0.1$ and $m = 0.8$

Figure 4.4 shows the performance against a number of classifiers, where the performance improved as a function of N. This is because the total variance is reduced by a factor of $1/N$. However, the rate of improvement varies among combining rules which depends strictly on the chosen ensemble parameters. Expanding the ensemble size does not always result in improving performance, where other ensemble parameters (σ, m, and ρ) have their effects on the improving rate. The comprehensive study shown in the previous figures (figure 4.1, figure 4.2, figure 4.3 and figure 4.4) helps in choosing the appropriate combining rule for a given set of ensemble conditions. In the this section, the performance of SG, SM, SA and SH is investigated in a three dimensional (3D) view. Four 3D plots are generated in terms of probability of classification error as a function of σ and m for $\rho = 0.4$ and $N = 9$. These plots are shown in figure 4.5, figure 4.6, figure 4.7 and figure 4.8 for SG, SM, SA and SH respectively. As shown from these figures, the performance behavior decreases exponentially against σ while maintaining approximately a smooth behavior against m.

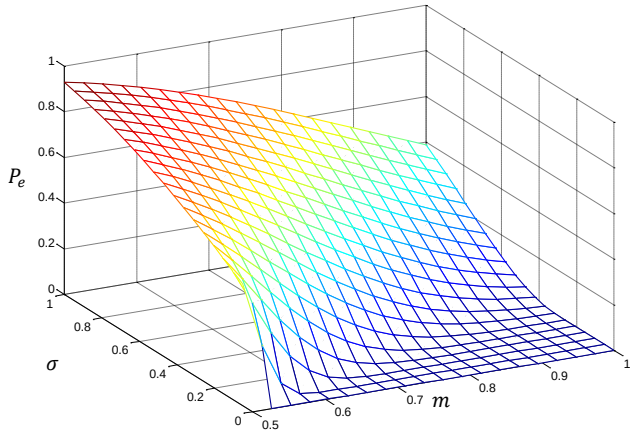

Figure 4.5. Probability of Classification Error as a Function of σ and m for SG Where $N = 9$ and $\rho = 0.4$

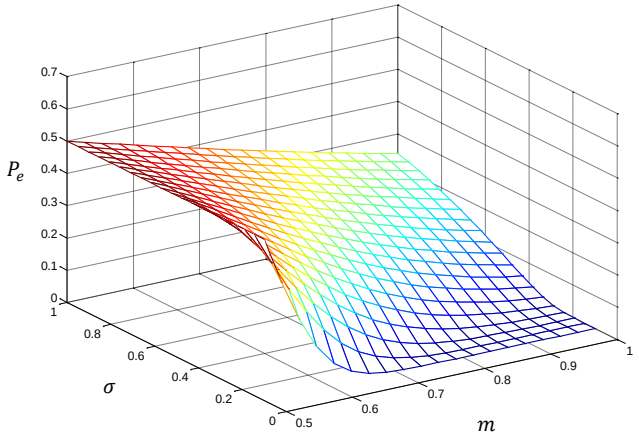

Figure 4.6. Probability of Classification Error as a Function of σ and m for SM Where $N = 9$ and $\rho = 0.4$

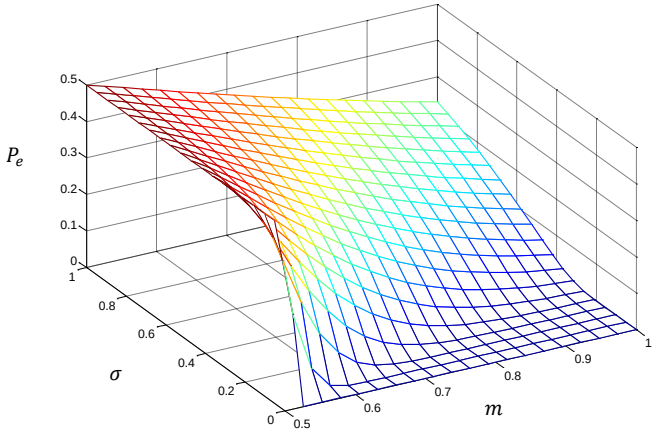

Figure 4.7. Probability of Classification Error as a Function of σ and m for SA Where $N = 9$ and $\rho = 0.4$

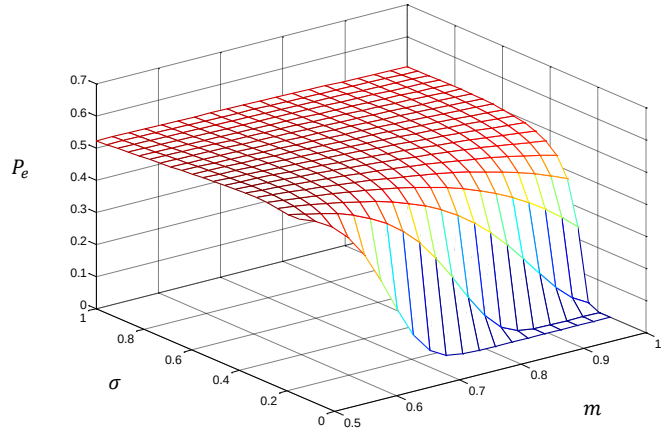

Figure 4.8. Probability of Classification Error as a Function of σ and m for SH Where $N = 9$ and $\rho = 0.4$

In addition, the region (in terms of σ and m) of low classification error for SH is smaller than others under the same ensemble condition, which means the SH rule gives limited options in optimizing the ensemble performance as compared to others. SA and SM rules gives approximately similar performances while the SG rule exhibits better performance as $m \to 1$ with low class variance $(\sigma \leq 0.1)$. In the second part of this section, the ensemble performance when classifiers' strengths are unequal and correlated is studied. From (4.4), it is clear that classifiers' weights are directly related to the class mean and variance. To make the comparison fair and based on average performance among different combining rules, it considered that both

σ_i and m_i are independent uniform random variables with periods $[a_1, b_1]$ and $[a_2, b_2]$ for σ_i and m_i respectively. This makes the probability of classification error as defined in (4.16), (4.31), (4.40) and (4.53) random variables and thus, it is reasonable to get their average or expectation values of P_e ($E[P_e]$) for comparison purposes. Figure 4.9 – figure 4.11 show the performance comparison as a function of N for average, geometric mean and majority vote. Harmonic mean is not considered in this comparison because the probability of low classification error is limited to a small area (in terms of σ and m), so it will not fit appropriately into our comparison. Each combining rule is studied for equal and different classifiers' weights, with the purpose to find how much improvements can be obtained from using a weighted combining rule compared to an unweighted. The periods of class mean and standard deviation are chosen to be $m = [0.7,1]$ and $\sigma = [0.1,0.3]$ for average and geometric mean while the range is expanded to $m = [0.5,1]$ and $\sigma = [0.1,0.5]$ for majority vote. The range of values is chosen whenever improvements in classification accuracy are visible. As shown, for all cases the improvement is negligible for small classifier numbers and improve as the ensemble size increases. This happens because as N increases, more individual classifiers with good properties are considered in the final decision process.

The previous results provided comprehensive assessments of different combining rules under study. For given ensemble parameters (σ, m, N and ρ) and based on derived formulas ((4.16, (4.31), (4.40) and (4.53))) it is possible to choose the appropriate fusion rule that achieves a minimum classification error.

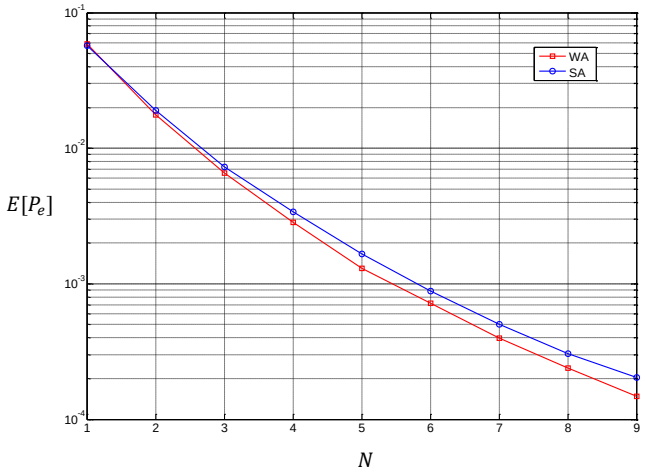

Figure 4.9. Probability of Classification Error as a Function of N For WA, and SA where $\rho = 0.1$, $m = [0.7, 1]$ and $\sigma = [0.1, 0.3]$

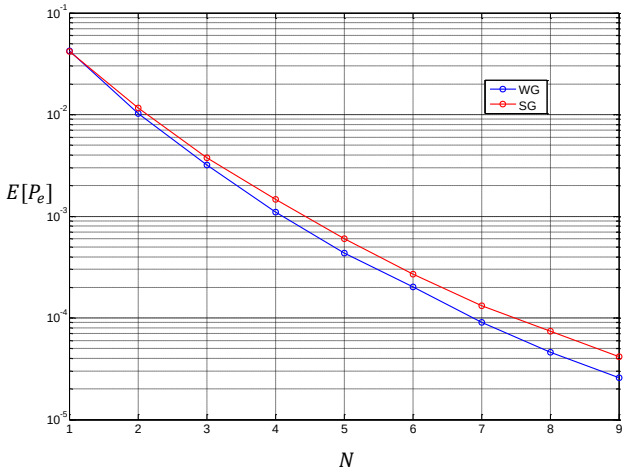

Figure 4.10. Probability of Classification Error as a Function of N For WG, and SG Where $\rho = 0.1, m = [0.7, 1]$ and $\sigma = [0.1, 0.3]$

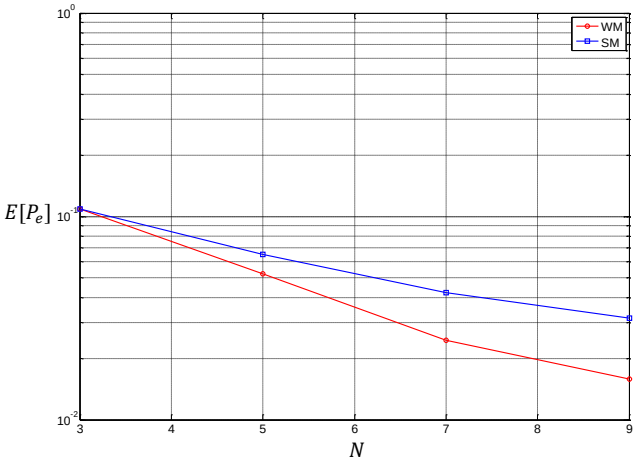

Figure 4.11. Probability of Classification Error as a Function of N for WM, and SM where $\rho = 0.1, m = [0.5, 1]$ and $\sigma = [0.1, 0.5]$.

4.8. Conclusion

In this chapter, an analytical framework for estimating the classification error of a MCS using four weighted combining rules is presented. The rules that considered for study were geometric mean, majority vote, average and harmonic mean. In the derivations, it assumed that classifiers have different strengths and classifiers outputs are correlated and normally distributed. A correlation assumption is used to model a diversity among classifiers. For large N and for

Pythagorean mean rules, the derivations are held over wide range of class distributions other than normal distribution. This is true since the distribution of averaging N random variables approach a normal distribution as N become large. Results show that the ensemble performance degrades exponentially as the correlation coefficient increases. It was expected from the theoretical results that the ensemble performance improves as a function of the class mean (m) by reducing the total variance by a factor of $1/m^2$ for geometric mean and $1/m^4$ for harmonic mean. Most previous studies have shown that the average rule outperforms others. However, a comparable study of different combining rules is investigated for different ensemble parameters ($\sigma, m, N\ and\ \rho$). Results show that the ensemble performance follows the principle of no free lunch theorem in which of combining classifiers with a high individual classification accuracy, harmonic mean gives the best classification accuracy, followed by geometric mean, average and majority vote. On the other hand, in case of combining classifiers with a low individual classification accuracy, average rule works best, followed by geometric mean, majority vote and harmonic mean. If the condition of individual classifiers is unknown, then on average, the geometric mean is the best candidate for combined classifier systems. Also, it shows that the weighted combining rule always improved classification accuracy in comparison to the un-weighted rule and that improvements get better with the expansion of the ensemble size.

CHAPTER V

A NOVEL APPROACH FOR SELECTING AN OPTIMAL ALGEBRAIC FUSION RULE FOR A MULTIPLE CLASSIFIER SYSTEM

5.1. Introduction

Multiple classifier or ensemble systems find wide applications due to the performance limitations of single classifier systems. One of the key factors in creating a successful ensemble is to design an optimal fusion rule that minimizes classification errors. Finding an optimal fusion rule that maximizes ensemble performance is a challenge, due to the lack of a strong foundation theory. It is obvious from literature that there is not a single combining rule that will work for all classification problems. In this chapter, the problem of selecting an optimal fusion rule for a given classification problem is studied. A mathematical model is proposed for estimating the classification accuracy for an ensemble made up of N individual classifiers and M classes. The performance trend that is predicted by the mathematical model is validated through six real datasets. Results show that over all spectrums of algebraic combining rules, there is always a set of fusion rules where the ensemble gives poor performance as the worst individual base classifier, while other sets give superior performance as the best classifier in the ensemble. As a result, performance is strictly dependent on individual classifier output statistics. Based on theoretical predication a novel method is developed for constructing an ensemble that produces classification accuracy equal or better than the best performing individual classifier. In addition, ensemble design shows robust performance against the overfitting problem. Derivations results presented in this chapter bring significant insights into the performance of combined classifier systems.

5.2. Background

Much work done in modeling of Multiple Classifier Systems (MCS) tries to answer the question of how to construct an ensemble that minimizes classification error [16], [17], [19], [20]. To answer this question, some literature is focused on experimental implementation to

optimize ensemble parameters that minimize classification error, however, a robust mathematical theory can provide more insights. In an ensemble design the most crucial designing phases are creating diversity among base classifiers and choosing an optimal fusion rule. The purpose of diversity is to ensure that each classifier contains complementary information i.e. making base classifiers independent. Independence means that base classifiers are uncorrelated with maximum diversity while dependence means classifiers are fully correlated with minimum diversity. One way to model the diversity among base classifiers is to calculate the correlation coefficient among their outputs [19]. Another important phase in the ensemble design is to choose an optimal combiner, and there are many developed works targeting this problem such as [30], [31], [32], [33], [35], [36], [37], [39], and [47].

Kuncheva in [31] proposed a framework to evaluate the performance of combined classifier systems. [31] presented estimation of the classification errors for six fusion rules which are minimum, maximum, average, median and majority vote. Results show that there is no best combiner rule and the performance varies and depends on the distribution of the posterior class probability. In Tumer and Gosh's work given in [39], a framework is developed which is based on a decision boundary analysis of the posterior class probability. They modeled the inaccuracy in training classifiers and decomposed it into bias and variance errors then quantified the error and referenced it as the added error. The work presented in [33] takes the idea further and extends the work given in [39] by including weighted combining rules under the condition of correlated classifiers. They show that performance depends on correlation levels among base classifiers. In [35], an extensive theoretical study on majority vote combiners is given for a binary classification problem. The estimation given in [35] is for the lower and upper bounds of an ensemble performance. A comparison between sum and majority rules are given in [32] based on assumptions of independent and identically distributed classifiers outputs with a normal distribution. It was shown that the sum rule always outperforms majority vote except under a certain condition when majority vote outperforms sum. The work in [30] presented a theoretical and experimental comparison for different fusion rules, their comparison shows that the sum rule always outperforms others (product, max, min, median and majority vote). The work in [36] and [37], presented a closed form expression for estimating classification error using product and majority vote combiners. The work showed the modified product rule outperforms sum and

majority under certain conditions when combining classifiers with good individual classification accuracies. In the work shown in [47], the framework presented in [33] is extended for nonlinear combiner rules such as product and geometric mean rules.

Based on the previous literature survey there is no guarantee that a given combiner rule would always provide superior performance among others. The paper cited in [48] confirms the fact that there is no perfect fusion algorithm that works for all classification problems. As a result the combination of multiple classifiers is lacking a strong foundation theory in order get a better understanding of how to optimize ensemble performance. All previous fusion rules mentioned in the literature survey are included in what is called the generalized mean rule. Based on assumptions given in [31] and [41] a general formula is derived for estimating the classification error based on a generalized mean rule. The derivation results enable us to estimate the classification error for a whole spectrum of algebraic combining rules and to choose the best combiner for a given classification problem.

5.3. Configuration of Ensemble Systems

Figure 5.1 shows the block diagram of a multiple classifiers system, the input features vector **x** defined in $\Re^n$ space is fed to N parallel classifiers that already learned features statistics. Each classifier produces at its output an estimation of the posterior class probabilities of M classes, and inputs to the classifiers are multivariate feature vectors **x** with a posterior probability density function of $p(\omega_j/\mathbf{x})$. If the output of a trained classifier is normalized appropriately between {0,1}, then each individual classifier transforms the multidimensional feature vector into a one-dimensional variable $d_{i,j}(\mathbf{x})$, where $i = 1,2,\dots,N$ and $j = 1,2,\dots,M$. A compact way to describe classifier outputs is in terms of a decision profile matrix ($Dp(x)$) which is defined as follows [11]

$$Dp(\mathbf{x}) = \begin{bmatrix} d_{1,1}(\mathbf{x}) & \cdots & d_{1,M}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ d_{N,1}(\mathbf{x}) & \cdots & d_{N,M}(\mathbf{x}) \end{bmatrix}. \tag{5.1}$$

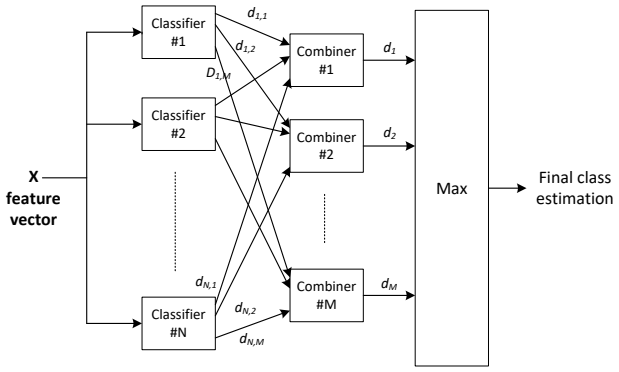

Figure 5.1. Structure of a Combined Classifier

Each column of the decision profile matrix is fed to the jth combiner that fuses classifier outputs for the jth class as follows

$$d_j(\mathbf{x}) = \mathcal{F}(d_{1,j}(\mathbf{x}), d_{2,j}(\mathbf{x}), \dots, d_{N,j}(\mathbf{x})), where\ j = 1,2, \dots, M, \tag{5.2}$$

where $d_j(\mathbf{x})$ is the jth combiner output and $\mathcal{F}(\cdot)$ is the fusion function. Finally, the max rule chooses the class label that has the maximum membership among M combiners

$$j = \arg\max(d_j(\mathbf{x})), for\ j = 1,2, \dots, M. \tag{5.3}$$

5.4. Generalized Geometric Mean Rule

Sometimes called power mean, the generalized geometric mean rule is a function that aggregates a spectrum of arithmetic fusion operations that includes a variety of functions such as arithmetic, geometric and harmonic means. For positive real numbers $k_1, k_2, \dots, k_N$ and a real number α, the generalized mean rule is defined as follows

$$\mathcal{M}_\alpha(k_1, k_2, \dots, k_N) = \left(\frac{1}{N}\sum_{i=1}^{N} k_i^\alpha\right)^{1/\alpha}, where -\infty \leq \alpha \leq \infty. \tag{5.4}$$

Some special cases for generalized mean are defined below for different α values

$$\mathcal{M}_{-\infty}(k_1, k_2, \dots, k_N) = min\{k_1, k_2, \dots, k_N\}, minimum\ rule,$$

$$\mathcal{M}_{-1}(k_1, k_2, \dots, k_N) = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{k_i}\right)^{-1}, harmonic\ mean\ rule,$$

$$\mathcal{M}_{0}(k_1, k_2, \dots, k_N) = \left(\frac{1}{N}\prod_{i=1}^{N} k_i\right)^{1/N}, geometric\ mean\ rule,$$

$$\mathcal{M}_{1}(k_1, k_2, \dots, k_N) = \frac{1}{N}\sum_{i=1}^{N} k_i, average\ rule,$$

$$\mathcal{M}_{\infty}(k_1, k_2, \dots, k_N) = max\{k_1, k_2, \dots, k_N\}, maximum\ rule.$$

5.5. Estimation of Classification Error for Generalized Mean Rule

In this section, the classification error is estimated for the generalized mean rule by changing the parameter α. This enables us to get different formulas for a whole spectrum of aggregating functions. To proceed, it is necessary to set up assumptions to derive the final formulas. Generalized the assumptions that are given in [31] to be more realistic by considering different weight base classifiers and removing the condition of independence among them. By introducing a parameter ρ, known as a correlation coefficient, which measures dependency among classifiers. Modeling dependency among classifiers is crucial in ensemble systems design since it is a direct measure of the ensemble diversity.

Each base classifier transforms a multidimensional posterior class probability into a single dimensional probability defined as follows

$$p(\omega_j/p_i) = \Im\big(p(\omega_j/\mathbf{x})\big) + \epsilon_{i,j}, for\ \ j = 1,2, and\ i = 1,2, \dots, N, \tag{5.5}$$

where $\Im(\cdot)$ is the transform operator that is directly related to the classifier characteristics and $\epsilon_{i,j}$ is the error added by *ith* classifier for the *jth* class. For a single classifier based on two classes of problems, equal prior classes probabilities ($p(\omega_1) = p(\omega_2) = 1/2$) and identical distribution of the classifiers outputs, the probability of classification error is defined in [31] as

$$P_e = \Phi\left(\frac{0.5 - m}{\sigma}\right), \tag{5.6}$$

where m and σ^2 are the moments of the classifiers outputs. The study is started by assuming there are N classifiers with a binary classification problem ($M = 2$), then extend the number of classes to M. In this case $d_{i,1} + d_{i,2} = 1$ or $d_{i,1} = 1 - d_{i,2}$ where $i = 1,2, \dots, N$. In order to simplify derivation, the following is assumed

$$p_i = d_{i,1} = 1 - d_{i,2}, for\ i = 1,2, \dots, N,$$

$$p = \mathcal{F}(p_1, p_2, \dots, p_N), \tag{5.7}$$

where p_i is the output of the *ith* individual classifier outputs and p is the combiner's output. p_i is assumed to be a normal random variable. The generalized mean rule is written as

$$p = \left(\frac{1}{N}\sum_{i}^{N} p_i^{\alpha}\right)^{1/\alpha}. \tag{5.8}$$

After clearly defining the required assumptions, the statistic of the random variable, should be estimated, i.e. its distribution and moments when α is changed over the range $-\infty \leq \alpha \leq \infty$.

Case 1: when $\alpha \geq 0$

The conditional mean and variance of the random variable p_i are $m_{p_i|\omega_j}$ and $\sigma^2_{p_i|\omega_j}$ respectively. Using Taylor series expansion around m_i [36], it can approximate the statistics of p_i^{α} as defined in (5.8) as follows

$$m_{p_i^{\alpha}|\omega_j} = E\left[p_i^{\alpha}|\omega_j\right] \approx \sum_{n=0}^{\infty} \frac{f^{(n)}(m_{p_i|\omega_j})}{n!} E\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right]$$

$$= m^{\alpha}{}_{p_i|\omega_j} + \sum_{n=2}^{\infty} \frac{f^{(n)}(m_{p_i|\omega_j})}{n!} E\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right], \tag{5.9}$$

$$\sigma^2_{p_i^{\alpha}|\omega_j} = VAR\left[p_i^{\alpha}|\omega_j\right] \approx \sum_{n=1}^{\infty} \left(\frac{f^{(n)}\left(m_{p_i|\omega_j}\right)}{n!}\right)^2 VAR\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right]. \tag{5.10}$$

Using binomial theorem, it can be approximated as $E\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right]$ and $VAR\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right]$ as follows

$$E\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right] = \sum_{k=0}^{n} \binom{n}{k} \left(-m_{p_i|\omega_j}\right)^{n-k} E\left[p_i^k\right], \tag{5.11}$$

$$VAR\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right] = E\left[\left(p_i - m_{p_i|\omega_j}\right)^{2n}\right] - E\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right]^2, \tag{5.12}$$

where $f^{(n)}$ represents the *nth* derivative of function $f(\cdot)$. A good approximation is achieved to the moments of p_i^{α} with $n \approx 10$. The variables $p_i,\ i = 1,2,\dots,N$ are assumed independent and identically distributed, that means $m_{p_1|\omega_j} = m_{p_2|\omega_j} = \cdots = m_{p_N|\omega_j} = m_{\omega_j}$, and $\sigma^2_{p_1|\omega_j} = \sigma^2_{p_2|\omega_j} = \cdots = \sigma^2_{p_N|\omega_j} = \sigma_{\omega_j}{}^2$, then

$$m_{\beta|\omega_j} = E\left[\left(\beta = \frac{1}{N}\sum_i^N p_i^{\alpha}\right)|\omega_j\right],$$

$$= m^{\alpha}{}_{p_i|\omega_j} + \sum_{n=2}^{\infty} \frac{f^{(n)}(m_{p_i|\omega_j})}{n!} E\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right],\ j = 1,2, \tag{5.13}$$

$$\sigma^2_{\beta|\omega_j} = VAR\left[\left(\beta = \frac{1}{N}\sum_i^N p_i^{\alpha}\right)|\omega_j\right],$$

$$= \frac{1}{N}\sum_{n=1}^{\infty}\left(\frac{f^{(n)}(m_{p_i|\omega_j})}{n!}\right)^2 VAR\left[\left(p_i - m_{p_i|\omega_j}\right)^n\right], \quad j = 1,2, \tag{5.14}$$

where $m_{\beta|\omega_j}$ and $\sigma^2_{\beta|\omega_j}$ are conditional moments of the random variable β. From (5.14) one obvious advantage of combining N classifiers are the overall variance is reduced by a factor $(1/N)$. To finalize derivation, the distribution of p should be estimated, the distribution of the random variable β approaches a normal distribution as N become large. The relation between p and β is defined as $p = \beta^{1/\alpha}$, then the conditional cumulative distribution of p can be written as

$$F\left(p|\omega_j\right) = Pr[\beta \leq p^{\alpha}] = G\left(p^{\alpha}|\omega_j\right), j = 1,2, \tag{5.15}$$

where $F(\cdot)$ and $G(\cdot)$ are cumulative functions for p and β respectively, using (5.13) the probability density function of p is estimated as follows

$$f\left(p|\omega_j\right) = \alpha\, p^{\alpha-1} g\left(p^{\alpha}|\omega_j\right), j = 1,2, \tag{5.16}$$

where $f(\cdot)$ and $g(\cdot)$ are probability density functions for p and β respectively, since the distribution of β is normal then

$$f\left(p|\omega_j\right) = \alpha \frac{p^{\alpha-1}}{\sqrt{2\pi\sigma^2_{\beta|\omega_j}}} \exp\left(-\frac{\left(p^{\alpha} - m_{\beta|\omega_j}\right)^2}{2\sigma^2_{\beta|\omega_j}}\right), j = 1,2, \alpha \geq 0\,, \tag{5.17}$$

then using (5.15) the probability of classification error is written as

$$P_e^{\alpha^+} = P(\omega_1)\left[1 - \Phi\left(\frac{\mu - m_{\beta|\omega_1}}{\sigma_{\beta|\omega_1}}\right)\right] + P(\omega_2)\Phi\left(\frac{\mu - m_{\beta|\omega_2}}{\sigma_{\beta|\omega_2}}\right), \alpha \geq 0, \tag{5.18}$$

where μ is the optimum decision value that minimized the overall classification error and $P_e^{\alpha^+}$ refers to the classification error over the range $\alpha \geq 0$.

Case 2: when $\alpha < 0$

The derivation defined in (5.13) and (5.14) hold but (5.15) through (5.17) need to be modified. So $p = \beta^{-1/\alpha}$ or $\beta \geq p^{-\alpha}$ that means

$$F(p|\omega_j) = \Pr(\beta \geq p^{-\alpha}) = 1 - G(p^{-\alpha}|\omega_j). \tag{5.19}$$

The probability density function for the random variable p is defined as

$$f(p|\omega_j) = \alpha \frac{p^{-\alpha-1}}{\sqrt{2\pi\sigma^2_{\beta|\omega_j}}} \exp\left(-\frac{\left(p^{-\alpha} - m_{\beta|\omega_j}\right)^2}{2\sigma^2_{\beta|\omega_j}} \right), \alpha < 0. \tag{5.20}$$

Then probability of classification error is defined as

$$P_e^{\alpha^-} = P(\omega_1)\Phi\left(\frac{\mu - m_{\beta|\omega_1}}{\sigma_{\beta|\omega_1}}\right) + P(\omega_2)\left[1 - \Phi\left(\frac{\mu - m_{\beta|\omega_2}}{\sigma_{\beta|\omega_2}}\right)\right], \alpha < 0, \tag{5.21}$$

where $m_{\beta|\omega_j}$ and $\sigma_{\beta|\omega_j}$ are defined in (5.13) and (5.14) respectively and $P_e^{\alpha^-}$ refers to the classification error over the range $\alpha < 0$. Using (5.13), (5.14), (5.18) and (5.21) the derived formula for classification error for generalized mean rule is

$$P_e^g = \begin{cases} P_e^{\alpha^+} & when\ \alpha \geq 0 \\ P_e^{\alpha^-} & when\ \alpha < 0 \end{cases}. \tag{5.22}$$

The subscript g refers to the generalized mean rule. The formulas defined in (5.22) are the probability of the classification error for the whole spectrum of the aggregating functions generated from the generalized mean rule. These results help in selecting an optimal fusion rule

that minimize the classification error for a given ensemble condition. In order to calculate the optimum threshold, the likelihood ratio test is used which is defined as

$$P(\omega_1|p) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} P(\omega_2|p). \tag{5.23}$$

Expression (5.23) states that the class ω_1 is chosen if $P(\omega_1|p)$ is greater than $P(\omega_2|p)$ otherwise ω_2 is selected. Using Bayes theory equation (5.23) is modified to

$$\frac{P(p|\omega_1)}{P(p|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{P(\omega_2)}{P(\omega_1)}. \tag{5.24}$$

Equation (5.24) suggests a decision should be based on measurements at the combiner's output. The decision test is based on a chosen class with maximum probability. Therefore, this rule is called the maximum a posteriori criterion. It is also called minimum error criterion since, on average, it minimizes the classification error. Using (5.24) it can be rewritten as the likelihood ratio test

$$\frac{\sigma_{\beta|\omega_2}\, exp\left(-\frac{\mu^2}{2\sigma^2_{\beta|\omega_1}}\right) exp\left(-\frac{m^2_{\beta|\omega_1}}{2\sigma^2_{\beta|\omega_1}}\right) exp\left(\frac{2\mu m_{\beta|\omega_1}}{2\sigma^2_{\beta|\omega_1}}\right)}{\sigma_{\beta|\omega_1} exp\left(-\frac{\mu^2}{2\sigma^2_{\beta|\omega_2}}\right) exp\left(-\frac{m^2_{\beta|\omega_2}}{2\sigma^2_{\beta|\omega_2}}\right) exp\left(\frac{2\mu m_{\beta|\omega_2}}{2\sigma^2_{\beta|\omega_2}}\right)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{P(\omega_2)}{P(\omega_1)}, \tag{5.25}$$

where $\mu = p^{\alpha}$, in order to simplify (5.25) assume $\sigma_{\beta|\omega_1} = \sigma_{\beta|\omega_2} = \sigma_{\beta}$ then (5.25) rewritten as

$$\mu = p^{\alpha} = \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{2\sigma^2_{\beta} \ln\left(\frac{P(\omega_2)}{P(\omega_1)}\right) + m^2_{\beta|\omega_1} - m^2_{\beta|\omega_2}}{2\left(m_{\beta|\omega_1} - m_{\beta|\omega_2}\right)} \quad or \tag{5.26}$$

$$p_{th} = \underset{\omega_2}{\overset{\omega_1}{\gtrless}}{}^{\alpha}\sqrt{\frac{2\sigma_\beta^2 \ln\left(\frac{P(\omega_2)}{P(\omega_1)}\right) + m_{\beta|\omega_1}^2 - m_{\beta|\omega_2}^2}{2\left(m_{\beta|\omega_1} - m_{\beta|\omega_2}\right)}} \tag{5.27}$$

If prior class probabilities are equal, then

$$p_{th} = \underset{\omega_2}{\overset{\omega_1}{\gtrless}}{}^{\alpha}\sqrt{\frac{m_{\beta|\omega_1} + m_{\beta|\omega_2}}{2}} = \sqrt{\beta_{th}} \tag{5.28}$$

For a single classifier system (5.28) modified into

$$p_{th} = \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{m_{\omega_1} + m_{\omega_2}}{2} \tag{5.29}$$

When $\sigma_{\beta|\omega_1} \neq \sigma_{\beta|\omega_2}$, the optimal threshold is deviated from estimated values in (5.27), then using (5.25) the optimal threshold is calculated using

$$\left(\frac{1}{\sigma_{\beta|\omega_2}^2} - \frac{1}{\sigma_{\beta|\omega_1}^2}\right)\mu^2 + 2\left(\frac{m_{\beta|\omega_1}}{\sigma_{\beta|\omega_1}^2} - \frac{m_{\beta|\omega_2}}{\sigma_{\beta|\omega_2}^2}\right)\mu$$

$$+\frac{m_{\beta|\omega_2}^2}{\sigma_{\beta|\omega_2}^2} - \frac{m_{\beta|\omega_1}^2}{\sigma_{\beta|\omega_1}^2} - 2log\left(\frac{\sigma_{\beta|\omega_1}P(\omega_2)}{\sigma_{\beta|\omega_2}P(\omega_1)}\right) = 0,\ where\ p_{th} = \sqrt[\alpha]{\mu} \tag{5.30}$$

When $\alpha = 1$, expression (5.27) is converted to the arithmetic mean and the probability of classification is defined as

$$P_e^a = P_e^g\,\big|_{\alpha=1} = \Phi\left(\frac{0.5 - m}{\sigma/\sqrt{N}}\right). \tag{5.31}$$

The subscript a refers to the arithmetic mean. The expression defined in (5.31) is identical to the formula (20) derived in reference [31].

In the following, the derivation in (5.22) is generalized for correlated base classifiers with different weights and M classes. Each classifier is assumed to have a weight w_i, where $i = 1,2,\dots,N$. The weight generalized mean is defined as

$$p = \left(\sum_{i=1}^{N} w_i p_i^{\alpha} \right)^{1/\alpha} . \tag{5.32}$$

The condition of independence is removed so the correlation coefficient among classifier outputs is defined as $\rho_{k,l}$ where $k = 1,2,\dots,N$ and $l = 1,2,\dots,N$ given that $k \neq l$. Using (5.13) and (5.14) the moments of $\theta = \sum_{i=1}^{N} w_i p_i^{\alpha}$ are defined as follows

$$m_{\theta|\omega_j} = E\left[\left(\theta = \sum_{i=1}^{N} w_i p_i^{\alpha} \right) | \omega_j \right] = \sum_{i=1}^{N} w_i m_{p_i^{\alpha}|\omega_j}, \tag{5.33}$$

$$\begin{aligned} \sigma^2_{\theta|\omega_j} &= VAR\left[\left(\theta = \sum_{i=1}^{N} w_i p_i^{\alpha} \right) | \omega_j \right] \\ &= \sum_{i=1}^{N} w_i^2 \sigma^2_{p_i^{\alpha}|\omega_j} + \sum_{k=1}^{N} \sum_{l=1}^{N} w_k w_l \rho_{k,l} \sigma^2_{p_k^{\alpha}|\omega_j} \sigma^2_{p_l^{\alpha}|\omega_j}. \end{aligned} \tag{5.34}$$

Using (5.33) and (5.34) the probability of classification error for N classifiers and M classes is defined as

$$P_e^g = \begin{cases} \mathrm{p}(\omega_1)\mathrm{Q}\left(\dfrac{\mu_{1,2} - m_{\theta|\omega_1}}{\sigma_{\theta|\omega_1}} \right) + \displaystyle\sum_{j=2}^{M-1} \mathrm{p}(\omega_j) \left[\Phi\left(\dfrac{\mu_{j,j+1} - m_{\theta|\omega_{j+1}}}{\sigma_{\theta|\omega_{j+1}}} \right) + \mathrm{Q}\left(\dfrac{\mu_{j+1,j+2} - m_{\theta|\omega_{j+1}}}{\sigma_{\theta|\omega_{j+1}}} \right) \right] \\ \qquad + \mathrm{p}(\omega_M)\mathrm{Q}\left(\dfrac{\mu_{M-1,M} - m_{\theta|\omega_M}}{\sigma_{\theta|\omega_M}} \right) \qquad when\ \alpha \geq 0 \\ \mathrm{p}(\omega_1)\Phi\left(\dfrac{\mu_{1,2} - m_{\theta|\omega_1}}{\sigma_{\theta|\omega_1}} \right) + \displaystyle\sum_{j=2}^{M-1} \mathrm{p}(\omega_j) \left[\mathrm{Q}\left(\dfrac{\mu_{j,j+1} - m_{\theta|\omega_{j+1}}}{\sigma_{\theta|\omega_{j+1}}} \right) + \Phi\left(\dfrac{\mu_{j+1,j+2} - m_{\theta|\omega_{j+1}}}{\sigma_{\theta|\omega_{j+1}}} \right) \right] \\ \qquad + \mathrm{p}(\omega_M)\Phi\left(\dfrac{\mu_{M-1,M} - m_{\theta|\omega_M}}{\sigma_{\theta|\omega_M}} \right) \qquad when\ \alpha < 0, \end{cases} \tag{5.35}$$

where $\Phi(x) = 1 - Q(x)$, expression defined in (5.35) is the generalized version of (5.22) and the optimum threshold $\mu_{k,l}$, is defined as

$$\mu_{k,l} = \underset{\omega_l}{\overset{\omega_k}{\gtrless}} \frac{2\sigma_\beta^2 \ln\left(\frac{P(\omega_l)}{P(\omega_k)}\right) + m_{\beta|\omega_k}^2 - m_{\beta|\omega_l}^2}{2\left(m_{\beta|\omega_k} - m_{\beta|\omega_l}\right)}, k, l = 1,2, \dots, M \; given \; k \neq l, l > k. \quad (5.36)$$

The expression defined in (5.36) holds for each pair of classes if tails contributions from $M - 2$ distributions are assumed negligible. For identical class distribution with equal priori class probabilities (5.35) simplified to

$$P_e^g = \begin{cases} \frac{2(\mathrm{M}-1)}{M} \Phi\left(\frac{\mu - m_\theta}{\sigma_\theta}\right) & when \; \alpha \geq 0 \\ \frac{2(\mathrm{M}-1)}{M} \mathrm{Q}\left(\frac{\mu - m_\theta}{\sigma_\theta}\right) & when \; \alpha < 0 \end{cases}. \quad (5.37)$$

For a special case when all classifiers have equal strengths, equal prior probabilities, $\alpha = 1$, $M = 2$, $w_i = 1/N$, $m_1 = m_2 = \cdots = m_N = m$, $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_N^2 = \sigma^2$ and $p_{i,j} = \rho = 1$. Then (5.37) simplify into

$$P_e = \Phi\left(\frac{0.5 - m}{\sigma}\right). \quad (5.38)$$

The expression defined in (5.38) is identical to equation (8) derived in reference [31] which represents the classification error for a single classifier. Therefore (38) suggested that there is no benefit of combining identical classifiers with $\rho = 1$ since the ensemble performance reaches the performance of a single one. So, creating diversity among classifiers is crucial to improve the classification accuracy.

5.6. Results and Discussions

a) Theoretical Results

In this section, theoretical results are discussed in terms of how to select the optimal fusion rule for a given ensemble condition. Since the focus is on studying the effect of parameter α on ensemble error. Classifiers assumed to have equal weights, independent and identically distributed. Studying the effect of other parameters such as correlation and unequal classifier strengths are considered in others works. For example, the study in [37] show a correlation increases among classifiers outputs, the ensemble error increases exponentially, while the study in [33] show the weighted combining rules achieve negligible improvement compared to the unweighted one. Thus, expression (5.22) is considered for the comparison study. In all comparison cases, the number of classifiers is chosen to be as large as 10. The reason behind this choice is to satisfy the assumption made in the derivations in which the final distribution of adding N classifiers is a normal distribution.

Since the Taylor series approximation defined in 5.10 is not very accurate for estimating the class variance for higher α values ($\alpha \gg 1$), so the derivation is simulated using MATLAB. Figure 5.2, figure 5.3 and figure 5.4 shows the probability density function as a function combiner outputs (p) for the following ensemble parameters ($N = 10, m_1 = 0.4, m_2 = 0.6, M = 2$). The purpose is to evaluate the effect of class statistics (σ_1, σ_2) on combining N classifiers. By considering three cases ($\sigma_1 = \sigma_2 = 0.2$), ($\sigma_1 = 0.1, \sigma_2 = 0.3$), ($\sigma_1 = 0.3, \sigma_2 = 0.1$), the results are shown in figure 5.2, figure 5.3 and figure 5.4 respectively. To study the effect of each combining rule on minimizing classification error, three combining rule cases are considered; $\alpha = 10, \alpha = 1$ and $\alpha = -10$. In the case of symmetrical class distribution ($\sigma_1 = \sigma_2 = 0.2$), the average rule displays the best results as compared to others ($\alpha = 10$ and $\alpha = -10$), because the error is minimum. While in case of $\sigma_1 = 0.1, \sigma_2 = 0.3$ (figure 5.3) the system improves as $\alpha \rightarrow \infty$ and the behavior is reversed when $\sigma_1 = 0.3, \sigma_2 = 0.1$ (figure.5.4) where the performance improves as $\alpha \rightarrow -\infty$.

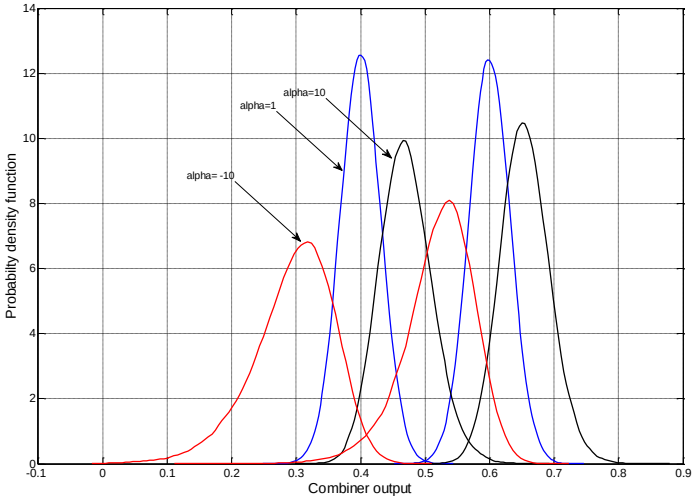

Figure 5.2. Probability Density Function For M=2 as a Function of Combiner Outputs for $\alpha = 10, \alpha = 1$ and $\alpha = -10$ and $\sigma_1 = \sigma_2 = 0.2$

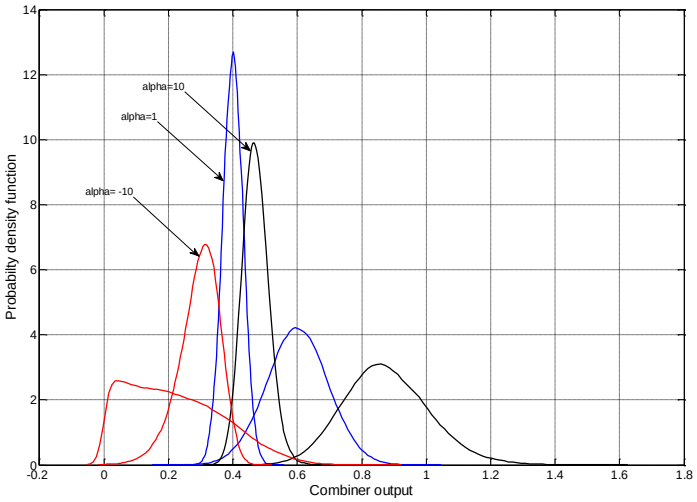

Figure 5.3. Probability Density Function for M=2 as a Function of Combiner Outputs for $\alpha = 10, \alpha = 1$ and $\alpha = -10$ and $\sigma_1 = 0.1, \sigma_2 = 0.3$

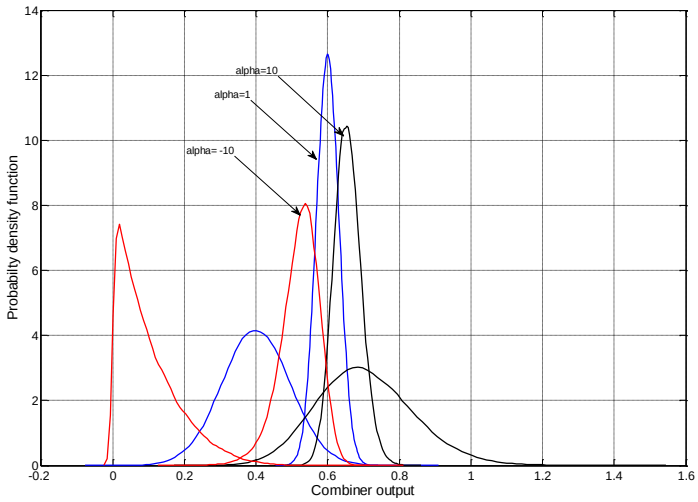

Figure 5.4. Probability Density Function for M=2 as a Function of Combiner Outputs for $\alpha = 10,\ \alpha = 1$ and $\alpha = -10$ and $\sigma_1 = 0.3, \sigma_2 = 0.1$

These results are quite interesting since they confirmed that there is no a single combining rule that work for all classification problems and every combining rule works under specific conditions. The pervious results matched the principle of the no free lunch theorem which states that there is not a single fusion rule which works for all classification problems and there is always a worst scenario for each fusion rule when it gives poor performance. As shown, each combing rule scales the resulting posterior class probability differently depend on their statistics. When $\sigma_1 = \sigma_2$ the best improvements are achieved using linear combiner ($\alpha = 1$) while in the case of $\sigma_1 < \sigma_2$ the improvement is achieved as $\alpha \to \infty$ and for $\sigma_1 > \sigma_2$ the improvement is in the direction of $\alpha \to -\infty$. So, the previous results help in the predication of what is the best combiner rule based on classifiers outputs statistics. Figure. 5.5 shows classification error as a function of α for different σ_1 and σ_2 values. As shown in the case of symmetrical classes variance ($\sigma_1 = \sigma_2 = 0.05$) a minimum classification error is achieved around $\alpha = 1$. While in case ($\sigma_1 = 0.05, \sigma_1 = 0.1$) a minimum classification error occurs at $\alpha = -5$ and for ($\sigma_1 = 0.1\ , \sigma_1 = 0.05$) at $\alpha = 5$. It is clearly evident from these results that selecting an optimal combining rule is strongly dependent on classifiers' statistics. The next section attempts to apply these results on real datasets.

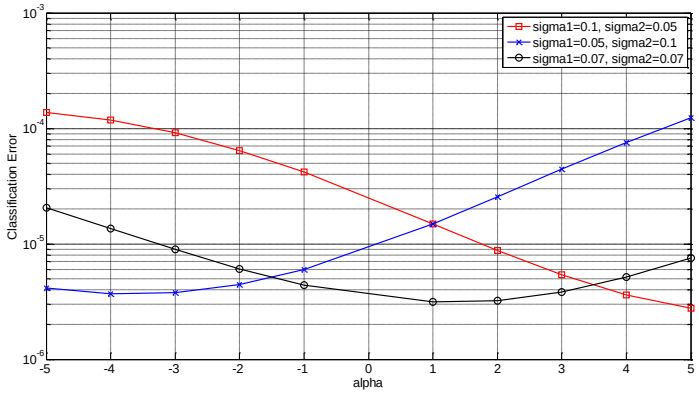

Figure 5.5. Classification error as a function of α for different σ_1 and σ_2 values

b) Experimental Results

In this section, the theoretical predications are verified using six datasets all from the UCI repository except diabetes which is from National Institute of Diabetes, Digestive and Kidney Diseases. The datasets that are considered for verification are breast cancer Wisconsin, magic gamma telescope, defaults of credit card clients, diabetes, ionosphere and diabetic retinopathy debrecen. Some features of the breast cancer dataset are extracted from a digital image of breast mass, and the generated attributes describe the cell nuclei characteristics that are presented in the image. The overall number of the generated attributes is 32, their values are real and the number of instances is 569. Features are classified into two classes (M: malignant, B: Benign). Magic gamma telescope dataset is generated for simulating the registration process of gamma particles using imaging techniques in a ground-based atmospheric Cherenkov gamma telescope. The dataset consists of 11 features in which their values are either integer or real valued and the total number of instance is 19020. A data set of credit card default clients is created to classify clients as credible or not credible for cases of customer default payments in Taiwan. The number of attributes in the dataset are 24 which could be integer and real valued and the number of instances are 30000. Diabetes consists from 8 features and 768 instances, the purpose of this dataset is to classify whether a patient shows symptoms of debates or not. Diabetic retinopathy debrecen data. The dataset contains 20 features and 1151 instances extracted from images to predict whether a patient shows symptoms of diabetic retinopathy or not.

Ionosphere consists from 34 features and 351 instances, the purpose of this data is to classify radar signals reflected from ionosphere to detect free electrons.

The purpose of the derivation is to predict the performance of an ensemble behavior as the combining rule varies. It is not expected that derivations would predict the exact behavior of the ensemble system trained on a real dataset, but rather it should estimate the performance trend, because the derivation are based idealized assumptions to come up with closed form expressions. The inaccuracies in the estimation results come from the following: assumptions of the normal posterior class probability are violated in practice since classifiers produce arbitrary distributions. Also, the derivations are based on an infinite train data size i.e. it was assumed the actual dataset distribution is known while in real cases the available training data is limited in size. The estimated error is based on the ratio of the number of times of an ensemble error to the total number of trails. Therefore, the expected error rate is biased from the actual value due to limited training data size. In addition, for simplification of the derivations it was assumed classes have an identical distribution with equal prior class probabilities these assumptions are usually violated in the actual dataset. However, the estimated model should predict the performance trend as fusion rules change which allow us to predict the best combining rules for a given dataset. The study is started by estimating the posterior class probabilities for the datasets under study to get an idea on how actual posterior class probability looks for the real datasets. Figure 5.6 - figure 5.11 show posterior class probabilities for six datasets. The dataset statistics for figure 5.6 - figure 5.11 are summarized in table (5.1)

Table 5.1: Statistics of Posterior Class Probabilities for Breast Cancer, Telescope, Credit Card, Diabetes, Ionosphere and Diabetic Retinopathy Datasets.

breast cancer dataset				telescope dataset				credit card dataset			
Class ω_1		Class ω_2		Class ω_1		Class ω_2		Class ω_1		Class ω_2	
mean	variance	mean	variance	mean	variance	mean	variance	mean	variance	mean	variance
0.2153	0.0078	0.7522	0.01	0.5435	0.0182	0.8388	0.0042	0.4989	0.0504	0.7782	0.0056
diabetes dataset				ionosphere dataset				Diabetic Retinopathy dataset			
Class ω_1		Class ω_2		Class ω_1		Class ω_1		Class ω_2		Class ω_1	
0.3015	0.0434	0.7986	0.0146	0.2154	0.0197	0.8501	0.0100	0.2683	0.0251	0.5995	0.0182

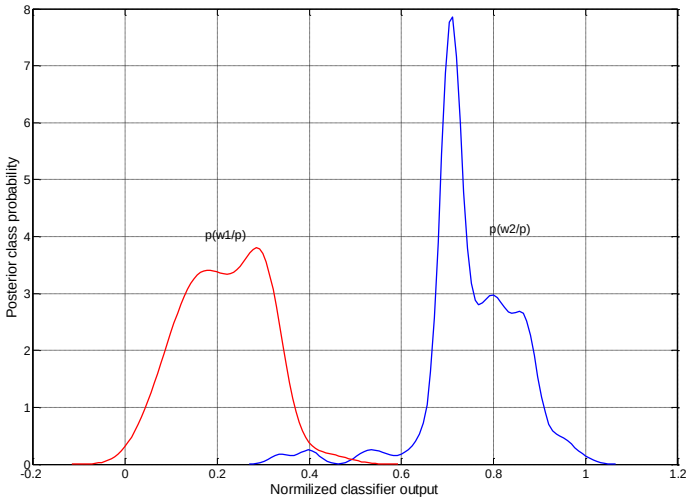

Figure 5.6. Posterior Class Probability for Breast Cancer Dataset

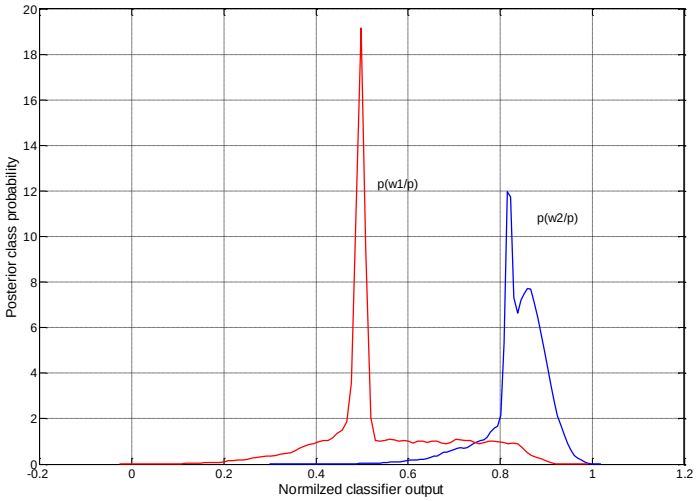

Figure 5.7. Posterior Class Probability for Telescope Dataset

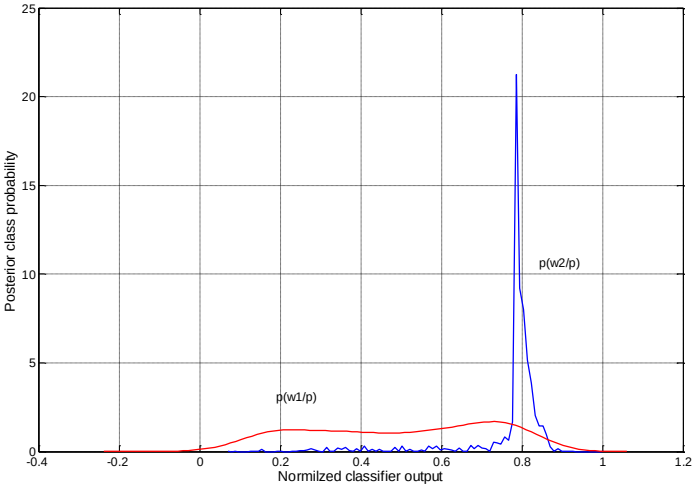

Figure 5.8. Posterior Class Probability for Credit Card Dataset

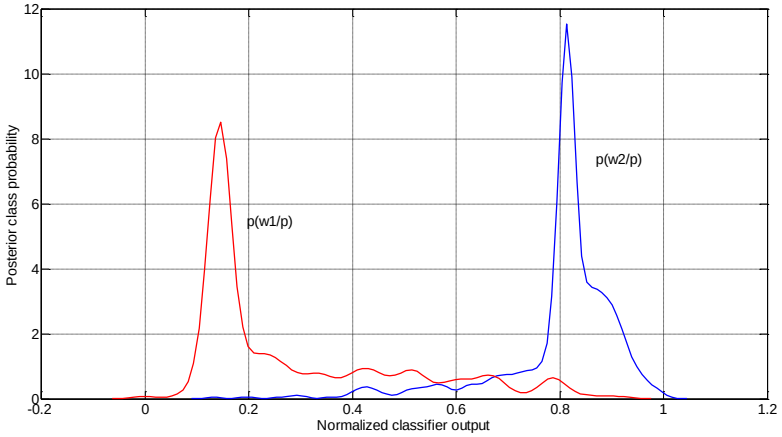

Figure 5.9. Posterior Class Probability for Diabetes Dataset

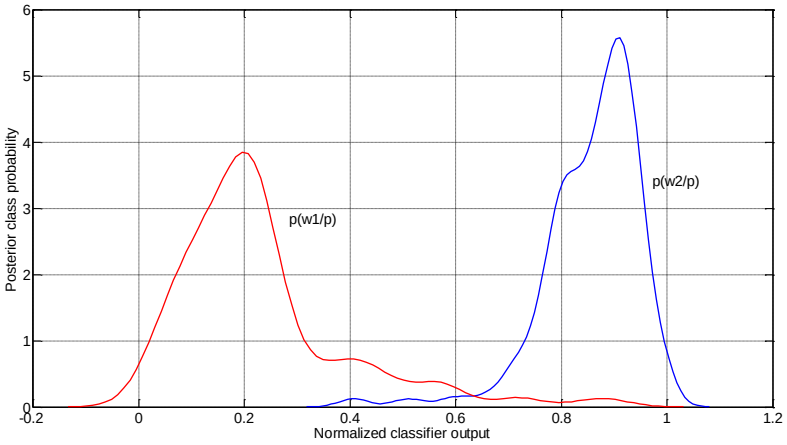

Figure 5.10. Posterior Class Probability for Ionosphere Dataset

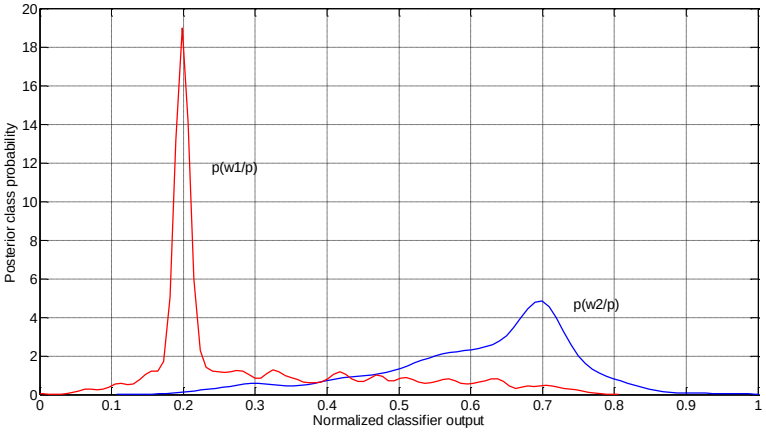

Figure 5.11. Posterior Class Probability for Diabetic Retinopathy Debrecen Dataset

As shown, the posterior class probabilities show an arbitrary probability density distribution for each dataset, the pdf of each class is plotted as a function of the normalized classifiers outputs. The classifier used to generate these plots is the Support Vector Machine (SVM) with a Gaussian radial basis as a kernel function. In each case, the whole dataset size is used in generating these plots (figure 5.6 - figure 5.11), the amount of classification error is indicated in the region of the intersection of the two class probabilities. As a result, these plots give us an idea about the generalization performance for a given classification task. For example, inspecting figure 5.8 shows that the performance is the worst compared to the others since posterior class probabilities highly interfere with each other. The classification error comes from two sources, the first source is from what is called the Bayes error which is inherent in the training data. It can be minimized by the proper selection of features that reduce the interference between the class probabilities. The other error source is the added error [39] which is made up of two components; bias and variance errors. Bias error results from training a classifier on a small training data size which results in a deviation in estimating the posterior class probability, while variance error results from over-training. Ensemble learning tries to overcome this problem by training a set of diverse classifiers by fusing their outputs. In the next section, an ensemble system is trained to compare the predicted results with results obtained from real datasets.

An ensemble of 10 SVM classifiers is created based on a Gaussian radial base as a kernel function. The datasets are divided into two parts, the first part is the training set which is comprised of 80% of the actual data size which leaves 20% for test purposes. To create diversity among base classifiers, a subspace training method is used in which each classifier is trained on a subset extracted from the available training set. The purpose of this method is to ensure that each classifier has complementary information which results in a reduction of correlation and increases the diversity among base classifiers. The classifiers outputs are fused using different combiners from {-200, 200}. To classify the fuzzed outputs, a decision threshold (μ) was set in which, when a fused output is equal or more than μ class ω_1 is chosen, otherwise ω_2 is selected. The optimal threshold is estimated using an adaptive algorithm that first estimates prior and posterior classes' probabilities, then the optimal threshold is calculated by minimizing the classification error using the following formula:

$$P_e = p(\omega_1)F_1(p,\mu) + p(\omega_2)F_2(p,\mu), \tag{5.39}$$

where P_e is the classification error, $p(\omega_i)$ is the classes prior probabilities, $F_i(p,\mu)$ is the cumulative distribution function of class ω_i, p is the combiner output and μ is the optimal threshold. $F_i(p,\mu)$ is estimated using the histogram technique during training phase then the calculated optimal threshold (μ) is used to classify data in the test phase.

Assessments for ensemble classification are based on classification error estimation which is calculated as a ratio of misclassification instances to the total number of instances in the test set. Figure 5.12 - figure 5.17, show the ensemble performance in terms of ensemble error as a function of the fusion rule parameter α. Figure 5.12 show that ensemble performance improves as $\alpha \to \infty$ i.e as fusion rule moves toward maximum. this trend trends are predicted by theoretical results shown in figure 5.3. The same conclusion can be applied to the telescope, credit card, diabetes, ionosphere and diabetic retinopathy datasets in which the system performance improves toward $\alpha = -1$ and $\alpha = 1$, $\alpha = -25$, $\alpha = 25$ and $\alpha = 50$ respectively. Tables (5.2) – table (5.7), show the ensemble performance as a function of α for 12 fusion rules and for six datasets. For all six cases, the best fused ensemble shows a performance equal to the best individual classifier in the ensemble. That means the optimal fused ensemble always achieves minimum classification accuracy among individual classifiers and shows a robust performance against the overfitting problem.

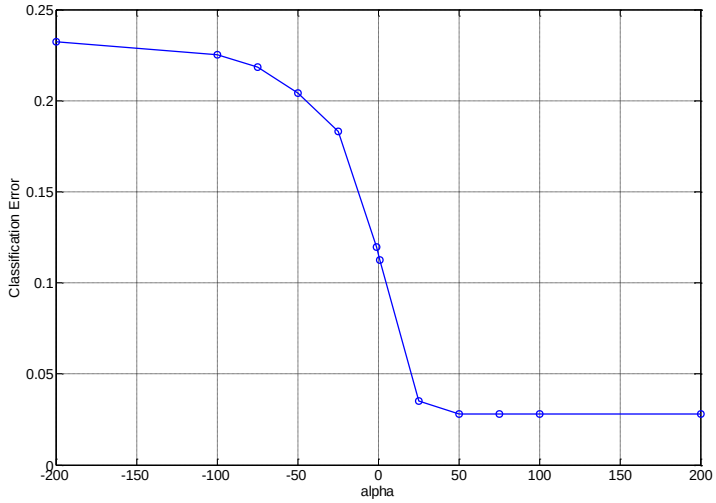

Figure 5.12. Classification Error as a Function of α for Breast Cancer Dataset

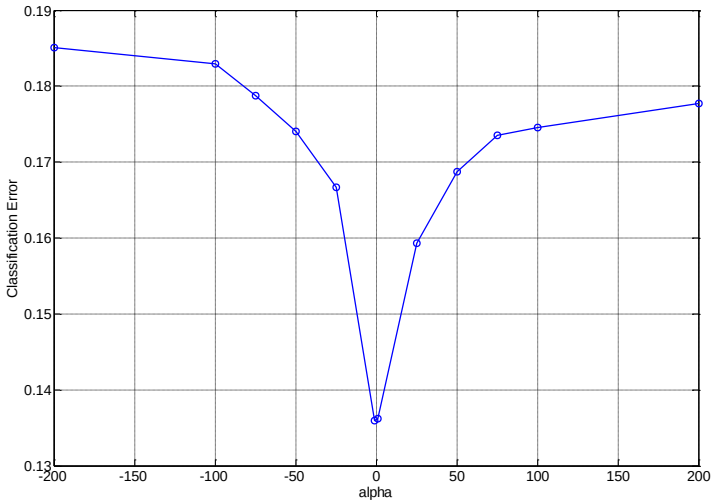

Figure 5.13. Classification Error as a Function of α for Telescope Dataset

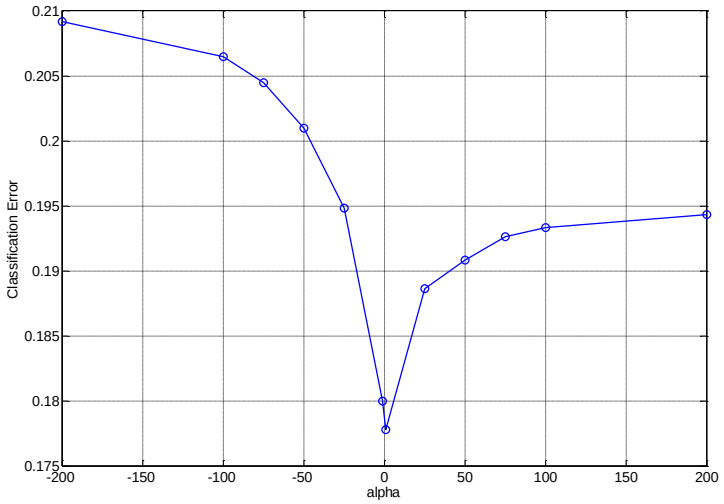

Figure 5.14. Classification Error as a Function of α for Credit Card Dataset

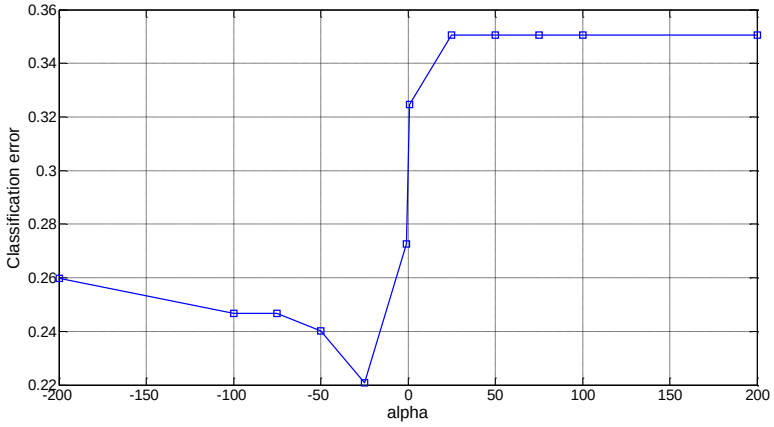

Figure 5.15. Classification Error as a Function of α for diabetic Dataset

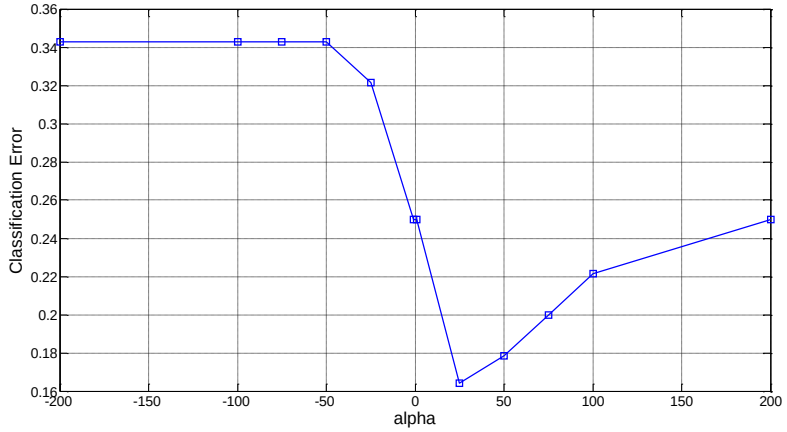

Figure 5.16. Classification Error as a Function of α for ionosphere Dataset

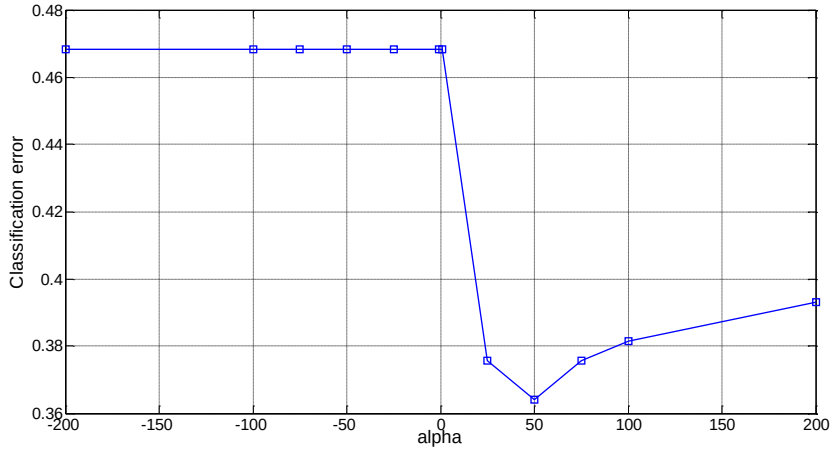

Figure 5.17. Classification Error as a Function of α for Diabetic Retinopathy Debrecen Dataset

It is clearly shown from tables 5.2 – 5.7 that selecting the appropriate fusion rules plays a crucial role in optimizing the ensemble performance and avoiding the worst-case scenario for selecting poor individual classifiers. For example, the worst classification accuracy was 76.76%, 81.49%, 79.08%, 64.94%, 65.71% and 53.18% while the best performance combining rule was 97.18%, 86.41%, 82.22%, 77.92%, 83.57% and 63.58% for breast cancer, telescope, credit card, diabetes, ionosphere and diabetic retinopathy datasets respectively. The improvement achieved over the six datasets are 20.42%, 4.92%, 3.14%, 17.86%, 13.43% and 10.4%. The improvement is calculated as a difference between the best and the worst ensemble performance. The best results achieved on the datasets are directly connected with the predication given in figure 5.6 - figure 5.11. For example, the tail probabilities shown in figure 5.6 are slightly interfere which

results in good classification accuracy (97.18%,) for the breast cancer dataset while in the case of telescope data and credit card data, the pdf tails penetration results in lower classification accuracy of 86.41% and 82.22% respectively. As a result, estimation of posterior class probabilities provided an indication of how datasets perform on trained classifiers. Removing features that causes these interferences among probability tails would improve the ensemble performance significantly. This is an interesting research area by training a classifier on a whole dataset then removing or transforming features that causes classification errors. These features are considered as a main source of errors and to remove them improves classification accuracy significantly.

Table 5.8 shows a classification results comparison between the proposed algorithm with standard ensemble classification algorithm called random forest for six data sets under study; breast cancer, telescope, credit card, diabetes, ionosphere and diabetic retinopathy. Random forest is a standard ensemble classification algorithm in machine learning which was first proposed by Ho in 1998 [49] and further improved by Leo Breiman [50]. The idea is based on training a set of decision tree classifiers using random subspace method, the classifiers results are combined using average combiner. The purpose of the algorithm is to minimize the variance of the classification results by combining large number of classifiers. In this work, random forest is choosing for comparison since its structure and the proposed algorithm are based on combining multiple classifiers. However, the combining process in random forest is fixed by average rule while the proposed algorithm is adaptive according to the base classifiers statistics. So, the random forest relies on using large number of base classifiers for a fixed combining rule while the proposed algorithm uses less base classifiers with flexible combining rule.

As shown the proposed algorithm achieved a comparable classification result with random forest with ensemble size of 10 base classifiers compare to 100 base classifiers in case of random forest. Since the proposed algorithm used less base classifiers, that results in faster execution time as compared to random forest. These results show that ensemble performance is strictly dependent on classifiers' output statistics. Therefore, selecting an optimal fusion rule plays an important role in optimized ensemble performance. Figure 5.18 summarized the proposed algorithm that is used in generating results given in tables 5.2 - 5.8.

Table 5.2: Ensemble Performance as a Function of α for Breast Cancer Dataset with Optimal Threshold = 0.699.

$\alpha = -200$	$\alpha = -100$	$\alpha = -75$	$\alpha = -50$	$\alpha = -25$	$\alpha = -1$	$\alpha = 1$	$\alpha = 25$	$\alpha = 50$	$\alpha = 75$	$\alpha = 100$	$\alpha = 200$
76.76%	77.46%	78.17%	79.58%	81.69%	88.03%	88.73%	96.48%	97.18%	97.18%	97.18%	97.18%

Table 5.3: Ensemble Performance as a Function of α for Telescope Dataset with Optimal Threshold = 0.5102.

$\alpha = -200$	$\alpha = -100$	$\alpha = -75$	$\alpha = -50$	$\alpha = -25$	$\alpha = -1$	$\alpha = 1$	$\alpha = 25$	$\alpha = 50$	$\alpha = 75$	$\alpha = 100$	$\alpha = 200$
81.49%	81.70%	82.12%	82.60%	83.33%	86.41%	86.38%	84.07%	83.12%	82.65%	82.54%	82.23%

Table 5.4: Ensemble Performance as a Function of α for Credit Card Dataset with Optimal Threshold = 0.6449.

$\alpha = -200$	$\alpha = -100$	$\alpha = -75$	$\alpha = -50$	$\alpha = -25$	$\alpha = -1$	$\alpha = 1$	$\alpha = 25$	$\alpha = 50$	$\alpha = 75$	$\alpha = 100$	$\alpha = 200$
79.08%	79.35%	79.55%	79.90%	80.52%	82.00%	82.22%	81.13%	80.92%	80.73%	80.67%	80.57%

Table 5.5: Ensemble Performance as a Function of α for Diabetes Dataset with Optimal Threshold = 0.3394.

$\alpha = -200$	$\alpha = -100$	$\alpha = -75$	$\alpha = -50$	$\alpha = -25$	$\alpha = -1$	$\alpha = 1$	$\alpha = 25$	$\alpha = 50$	$\alpha = 75$	$\alpha = 100$	$\alpha = 200$
74.03%	75.32%	75.32%	75.97%	77.92%	72.73%	67.53%	64.94%	64.94%	64.94%	64.94%	64.94%

Table 5.6: Ensemble Performance as a Function of α for Ionosphere Dataset with Optimal Threshold = 0.6104.

$\alpha = -200$	$\alpha = -100$	$\alpha = -75$	$\alpha = -50$	$\alpha = -25$	$\alpha = -1$	$\alpha = 1$	$\alpha = 25$	$\alpha = 50$	$\alpha = 75$	$\alpha = 100$	$\alpha = 200$
65.71%	65.71%	65.71%	65.71%	67.86%	75.00%	75.00%	83.57%	82.14%	80.00%	77.86%	75.00%

Table 5.7: Ensemble Performance as a Function of α Diabetic Retinopathy Dataset with Optimal Threshold = 0.7921.

$\alpha = -200$	$\alpha = -100$	$\alpha = -75$	$\alpha = -50$	$\alpha = -25$	$\alpha = -1$	$\alpha = 1$	$\alpha = 25$	$\alpha = 50$	$\alpha = 75$	$\alpha = 100$	$\alpha = 200$
53.18%	53.18%	53.18%	53.18%	53.18%	53.18%	53.18%	62.43%	63.58%	62.43%	61.85%	60.69%

Table 5.8: Classification Results Comparison Between Proposed Algorithm and Random Forest for Six Datasets; Breast Cancer, Telescope, Credit Card, Diabetes, Ionosphere and Diabetic Retinopathy datasets.

Dataset	breast cancer	telescope	credit card	Diabetes	ionosphere	diabetic retinopathy
Proposed algorithm	96.7 %	86.5 %	82.3 %	77.5%	82.0%	64.4%
Random Forest	95.9%	87.5%	81.6%	75.7%	64.9%	54.2%.

As shown from previous results, the proposed algorithm is not outperforming the random forest in all datasets cases but it provides a comparable classification accuracy. As known from machine learning different classifiers (such as svm, neural network, knn, etc.) exhibit different classification accuracy for a given data set. The overall performance of the proposed algorithm is dependent on ability of base classifiers to classify data which in this case svm. If these classifiers generate unsatisfying results on a certain data set it may affect the overall algorithm performance.

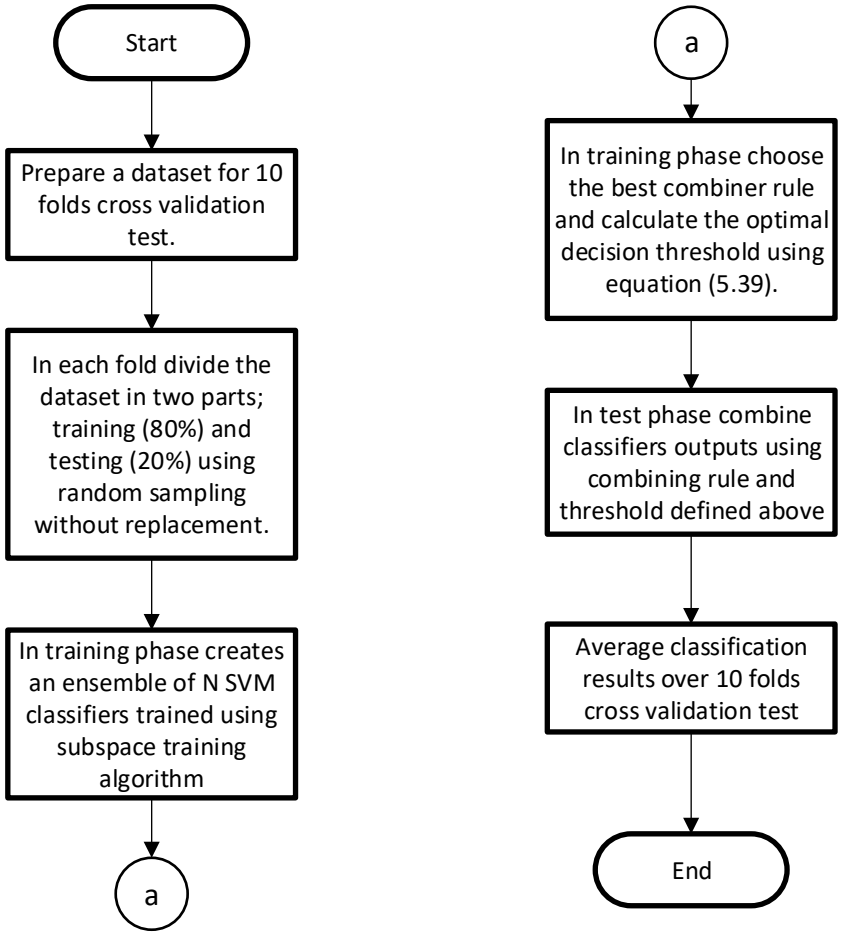

Figure 5.18. Flow Chart of the Proposed Algorithm

CHAPTER VI

SUMMARY AND CONCLUSIONS

6.1. Summary

The idea of combining N classifiers is a promising technique to achieve better classification performance and lower sensitivity to overfitting problems when compared to single classifier systems. In this work, a mathematical model is proposed that estimates the classification error for N classifiers and for M classes. A diversity among base classifiers is modeled using a correlation operator. Theoretical results show there is not a single combining rule that works for all classification problems. Theoretical predication on six real datasets is validated. Guided by theoretical derivations, a novel algorithm is developed that achieved optimal classification accuracy and avoided the scenario of choosing the worst performing classifiers. The proposed algorithm was tested on an ensemble of 10 SVM classifiers that was trained using a subspace training algorithm. Results show that the ensemble always gives a classification accuracy that is equal to or better than the best individual classifiers in the ensemble. Also, it shows a robust performance against the overfitting problem. The classification results of the proposed algorithm show a comparable performance against random forest which is a standard algorithm in ensemble classification.

6.2. Contributions

The dissertation purpose is to optimize the performance of ensemble systems by designing an optimal combining rule for a given classification problem. The contribution is divided into two parts; theoretical and experimental. In the theoretical part in chapter 3, analytical models are proposed for product and majority vote rules. The results show that for equal weight classifiers with a normal distribution, a modified product rule (geometric mean) outperforms sum, product and majority vote rules under the condition of independent classifiers. Also results show as the correlation among classifiers outputs increases, the probability of classification error degrades exponentially. The trend continues until the performance reaches the behavior of a single classifier regardless of the number of base classifiers used in the ensemble. In Chapter 4, four models are

proposed; weighted geometric mean, weighted majority vote, weighted average and weighted harmonic mean. Theoretical results show that there is not a single combining rule that works for all classification problems.

Part of Chapter 5 focused on designing an optimal ensemble combining rule for a given classification problem. A theoretical model is developed for estimating the performance of ensemble systems for M classes and N classifiers based on a generalized mean rule. Results show how to design an improved combining method based on classifier outputs statistics. The proposed theoretical models in Chapter 3,4 and 5 provide a better understanding of the behavior of MCS and bring significant insight.

In the experimental part in Chapter 5, a novel algorithm is proposed that predicts an optimal combining rule for a given classification problem. Six datasets are used to test the classification results of the proposed algorithm, which are breast cancer, telescope and credit card. Results show a comparable performance with the random forest. The benefit of the proposed algorithm is to use less base classifiers: 10 compared to 100 in the case of random forest which results in reducing the computational operation and makes it more suitable in real time classification problems.

6.3. Future Directions

- Dissertation results quantify the Bayes error resulting from combining N classifers which is directly related to the structure of the training data. The work given in [39] tried to estimate what is called the added error which is directly related to the imperfections in a classifier training process. The idea is to combine Bayes and added error into one formula and test the ensemble against different combining rules. As shown in the previous derivations in Chapter 4, different methodologies are applied to derive classification error probability. The purpose is to present a unified framework that would work with any combining rule. According to the current literature, there is no analytical solution which can achieve such a framework. However, a semi analytical method may provide the desired results. This would help in building a foundation theory for multiple classifier systems.
- By mathematical definition the generalized mean rule provides a limited search space for

an optimal combining rule so it can maximize the searching space by solving the following optimization problem. The key idea is to make an assumption about the joint probability density function $f(\mathbf{P})$ for N combined classifiers. The probability of classification error is calculated for N joint cumulative distribution function ($F(\mathbf{P})$) subjected to the constraint of the combining rule i.e,

$$F(p_1 \leq k_1,\, p_2 \leq k_2, \ldots,\, p_N \leq k_N)\,,$$

subject to

$$\Psi(k_1, k_2, \ldots, k_N) < c\,, \tag{6.1}$$

where c and k_i ($i = 1,2, \ldots, N$) are constants and $\Psi(\cdot)$ is the fusion function. The previous problem has N constraint integrals, solving (6.1) would help in finding an optimum combining rule for a given ensemble condition. On the other hand, the problem can be seen as an optimization puzzle, in which for any given distribution of the classifiers outputs, it can search for a target function (combining rules) that minimizes the ensemble error ($F(\mathbf{P})$).

6.4. Conclusion

In this work, the focused is on designing an optimal algebraic combining rule that minimized the classification error. The proposed algorithm is compared with random forest which shows a comparable classification accuracy with 10 base classifiers compared to 100 in random forest, which reduces the computational calculations significantly. Random forest uses s fixed average combining rule and relies on large number of classifiers. While, the proposed algorithm uses less classifiers number and flexible combining rule. This provides additional advantage for the proposed algorithm for finding the best combining rule that optimized the ensemble performance. The proposed algorithm is best suited for applications that are critical in classification time such as image processing, biometric applications and computer vision.

REFERENCES

[1] B.V. Dasarathy and B.V. Sheela, “Composite classifier system design: Concepts and methodology,” Proceedings of the IEEE, vol. 67, no. 5, pp. 708–713, 1979.

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

[3] R.E. Schapire, “The strength of weak learnability,” Machine Learning, vol. 5, no. 2, pp. 197–227, 1990.

[4] L. Xu, A. Krzyzak, and C.Y. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” IEEE Transactions on Systems, Man and Cybernetics, vol. 22, no. 3, pp. 418–435, 1992.

[5] T.K. Ho, J.J. Hull, and S.N. Srihari, “Decision combination in multiple classifier systems,” IEEE Trans. on Pattern Analy. Machine Intel., vol. 16, no. 1, pp. 66–75, 1994.

[6] G. Rogova, “Combining the results of several neural network classifiers,” Neural Networks, vol. 7, no. 5, pp. 777–781, 1994.

[7] L. Lam and C.Y. Suen, “Optimal combinations of pattern classifiers,” Pattern Recognition Letters, vol. 16, no. 9, pp. 945–954, 1995.

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

[9] I. Bloch, “Information combination operators for data fusion: A comparative review with classification,” IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans., vol. 26, no. 1, pp. 52–67, 1996.

[10] S.B. Cho and J.H. Kim, “Combining multiple neural networks by fuzzy integral for robust classification,” IEEE Transactions on Systems, Man and Cybernetics, vol. 25, no. 2, pp. 380–384, 1995.

[11] L.I. Kuncheva, J.C. Bezdek, and R. Duin, “Decision templates for multiple classifier fusion: An experimental comparison,” Pattern Recognition, vol. 34, no. 2, pp. 299–314, 2001.

[12] H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, and V. Vapnik, “Boosting and other ensemble methods,” Neural Computation, vol. 6, no. 6, pp. 1289–1301, 1994.

[13] L.I. Kuncheva, “Classifier ensembles for changing environments,” 5th Int. Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, F. Roli, J. Kittler, and T. Windeatt, Eds., vol. 3077, pp. 1–15, 2004.

[14] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, “Adaptive mixtures of local experts,” Neural Computation, vol. 3, pp. 79–87, 1991.

[15] M.J. Jordan and R.A. Jacobs, “Hierarchical mixtures of experts and the EM algorithm,” Neural Computation, vol. 6, no. 2, pp. 181–214, 1994.

[16] L. I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms. Second edition, New Jersey, NJ: Wiley, 2014.

[17] Z. H. Zhou, Ensemble Methods: Foundations and Algorithms. Boca Raton, FL: Chapman & Hall, 2012.

[18] Proc. 12th Int’l Workshop Multiple Classifier Systems (MCS 2015), F. Schwenkeret al., eds., 2015.

[19] R. Polikar, “Ensemble based systems in decision making,” IEEE Circuits and Systems Mag., vol. 6, no. 3, pp. 21-45, 2006.

[20] M.A. Bagheri, G. Qigang and S. Escalera, “A Framework towards the Unification of Ensemble Classification Methods,” IEEE Proc. 12th International Conference on Machine Learning and Applications (ICMLA), vol. 2, pp. 351 –355, 2013.

[21] M. Wozniak, M. Grana and E. Corchado, “A survey of multiple classifier systems as hybrid systems,” Information Fusion, vol. 16, pp. 3-17, 2014.

[22] A. Jain, P. Duin and Jianchang Mao, “Statistical pattern recognition: a review,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4-37, 2000.

[23] W. W. Y. Ng, A. P. F. Chan, D. S. Yeungand E. C. C. Tsang, “Quantitative study on the generalization error of multiple classifier systems,” IEEE International Conference on Systems, Man and Cybernetics, vol.1, pp. 889 -894, 2005.

[24] G. Zenobi and P. Cunningham, “Using Diversity in Preparing Ensembles of Classifiers Based on Different Feature Subsets to Minimize Generalization Error, “the series Lecture Notes in Computer Science, vol. 2167, pp. 576-587, 2001.

[25] T. Windeatt, “Accuracy/diversity and ensemble MLP classifier design,” IEEE Transactions on Neural Networks, vol. 17, no. 5, pp. 1194–1211, 2006.

[26] U. Johansson, T. Lofstrom and L. Niklasson, “The Importance of Diversity in Neural Network Ensembles -An Empirical Investigation,” IEEE Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007.

[27] L. Kuncheva, “Diversity in multiple classifier systems,” Information Fusion, vol. 6, no. 1, pp. 3-4, 2005.

[28] M. Haghighi, A. Vahedian and H. Yazdi, “Creating and measuring diversity in multiple classifier systems using support vector data description,” Applied Soft Computing, vol. 11, no. 8, pp. 4931-4942, 2011.

[29] G. Webb and Z. Zheng, “Multistrategy ensemble learning: reducing error by combining ensemble learning techniques,”IEEE Trans. Knowl. Data Eng., vol. 16, no. 8, pp. 980-991, 2004.

[30] J. Kittler, M. Hatef, R. Duin, and J. Matas, “On Combining Classifiers,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226–239, 1998.

[31] L. I. Kuncheva, “A theoretical study on six classifier fusion strategies,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp.281-286, 2002.

[32] J. Kittler and F. M. Alkoot,“Sum versus vote fusion in multiple classifier systems,” IEEE Transactions on Pattern Analysis and Machine Intelli-gence, vol. 25, no. 1, pp.110-115, 2003.

[33] G. Fumera and F. Roli,“A theoretical and experimental analysis of linear combiners for multiple classifier systems,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 6, pp.924-956, 2005.

[34] L. Louisa and C.Y. Suen, “Application of majority voting to pattern recognition: an analysis of its behavior and performance,” IEEE Trans-actions Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 27, no. 5, pp. 553 -568, 1997.

[35] N. Anand, “Theoretical bounds of majority voting performance for a binary classification problem,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 12, pp.1988-1995, 2005.

[36] M. Hassan and I. Abdel-Qader, “Analysis of Multiple Classifier Systems Using Product and Modified Product rules, ” Proceeding of IEEE International Conference on Electro/Information Technology, pp. 152 –157, 2015.

[37] M. Hassan and I. Abdel-Qader, “Performance Analysis of Majority Vote Combiner for Multiple Classifier Systems,” Proceeding of 14th IEEE International Conference on Machine Learning and Applications (IEEE ICMLA'15) Miami, Florida, USA, December 9-11, 2015.

[38] Wozniak, M., “Combining pattern recognition algorithms chances and limits,” Pro. 2nd International Conference on Computer Engineering and Technology (ICCET), pp. 111-115, 2010.

[39] K. Tumer and J. Ghosh, “Analysis of Decision Boundaries in Linearly Combined Neural Classifiers,” Pattern Recognition, vol. 29, pp. 341-348, 1996.

[40] Haym Benaroya, Seon Mi Han, and Mark Nagurka, Probability Models in Engineering and Science. CRC Press, 2005.

[41] F. Alkoot and J. Kittler, “Experimental Evaluation of Expert Fusion Strategies,” Pattern Recognition Letters, vol. 20, pp. 1361–1369, 1999.

[42] A. Genz and F. Bretz, “Numerical Computation of Multivariate t Probabilities with Application to Power Calculation of Multiple Contrasts,” Journal of Statistical Computation and Simulation. vol. 63, pp. 361–378, 1999.

[43] A. Genz and F. Bretz, “Comparison of Methods for the Computation of Multivariate t Probabilities,” Journal of Computational and Graphical Statistics. Vol. 11, no. 4, pp. 950–971, 2002.

[44] Y. Chen, K. G.K, H. Lu and N. Cao, “Novel Approximations to the Statistics of Products of Independent Random Variables and Their Applications in Wireless Communications,” IEEE Transactions on Vehicular Technology, vol. 61, no. 2, pp. 443-454, 2012.

[45] J. Salo, H. M. El-Sallabi and P. Vainikainen, “The distribution of the product of independent Rayleigh random variables,” IEEE Transactions on Antennas and Propagation, vol. 54, no.2, pp. 639-643, 2006.

[46] D. Xia, S. Xu, and F. Qi, “A proof of the arithmetic mean-geometric mean-harmonic mean inequalities,” RGMIA Research Report Collection, vol. 2, no. 1, 1999.

[47] Mohammed Falih Hassan and Ikhlas Abdel-Qader, “Improving Pattern Classification by Nonlinearly Combined Classifiers", Proceedings of the 15th IEEE International Conference on Cognitive Informatics & Cognitive Computing, Stanford University, USA, Aug. 22-23, 2016.

[48] R. Hu and R.I. Damper, “A No Panacea Theorem for classification combination,” Pattern Recognition, vol. 41, pp. 2665-2673, 2008.

[49] Tin Kam Ho, “The random subspace method for constructing decision forests”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 20, Issue: 8, Aug 1998.

[50] L. Breiman, “Random Forests”, Machine Learning, vol. 45, no. 3, pp. 261-277, 2001.