DISCRIMINANT ENSEMBLES AND ERROR ANALYSIS OF CLASSIFIER FUSION RULES

by

Murat Semerci

BS, Electrical and Electronics Engineering, Boğaziçi University, 2005

BS, Computer Engineering, Boğaziçi University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2007

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Prof. Ethem Alpaydın for his guidance, patience and tolerance throughout the preparation of this thesis. He always supported and encouraged me during my studies. He not only guided me throughout this thesis but also counselled me on any subject I needed help. I am very grateful to him. It is a privilege to work with him.

I would like to express my gratitude to Prof. Fikret Gürgen and Prof. Günhan Dündar for taking part in my thesis committee.

I would like to thank my colleagues Assist. Prof. Olcay T. Yıldız and M. Aydın Ulaş for their cooperation and collaboration during my studies. I would also like to express my gladness to Mehmet and Esma for weekly discussions we had.

I do not know how to thank my family for their continuous support and encouragement. Without them, I wouldn't be who I am now. Their faith in me has always been the most significant factor in everything I have accomplished in my life. I also owe a great debt of gratitute to Elif. She was there to help me whenever I felt distressed.

The work in this thesis has been supported by Boğaziçi University Scientific Research Project 05HA101 and Turkish Scientific Technical Research Council TÜBİTAK EEEAG 104E079.

ABSTRACT

DISCRIMINANT ENSEMBLES AND ERROR ANALYSIS OF CLASSIFIER FUSION RULES

Each classification algorithm has its own underlying assumption and misclassifies different patterns and overall accuracy can be increased by a suitable fusion of multiple classifiers.

The combination is performed over the scores of classifiers, which are mostly posterior probabilities. Although the aim of classifier fusion is improved accuracy, there is no guarantee that this will be the case. In this study, we propose a new combination scheme which uses a subset of the classifier scores instead of using all of them. We experiment with three different methods for discriminant selection and combination, using decision trees and feature selection. We see that decision trees are better in choosing the best subset of features and accuracy is improved especially when the chosen discriminant outputs are combined with a trained linear model.

In trying to understand the behavior of the fixed rules, we apply the idea of decomposing a loss function into bias, variance and noise. This study gives a brief survey of the bias, variance and noise decompositions in the literature for squared and 0/1 loss. We show that they are unable to explain the error behaviour of fusion rules, especially for minimum and maximum rules. We give the reasons why some fusion strategies work better than others under the assumptions of uniform or Gaussian noise. We propose instead a measure based on the area of intersection to explain the behavior of the fixed rules.

ÖZET

AYIRTAÇ TOPLULUKLARI VE SINIFLANDIRICI BİRLEŞTİRME KURALLARININ HATA ÇÖZÜMLEMESİ

Her sınıflandırma algoritması veri hakkında farklı bir varsayımda bulunur ve farklı örüntüler üzerinde yanlış yapar; bu yüzden uygun bir kaynaşım ile genel doğruluk arttırılabilir.

Birleştirme, genellikle sonsal olasılık olan sınıflandırıcı sonuç değerleri üzerinde gerçekleştirilir. Sınıflandırıcı kaynaşım yöntemlerinin amacı doğruluk başarımını arttırmak olsa da her zaman başarılı olacaklarının teminatı yoktur. Bu çalışmada bütün sınıflandırıcı sonuç değerlerini kullanmak yerine, onların altkümelerini kullanan yeni bir birleştirme tasarısı öneriyoruz. Ayırtaç seçmek ve birleştirmek için karar ağaçları ve özellik seçme kullanan üç farklı yöntem deniyoruz. Karar ağaçlarının en iyi özellik altkümesini seçmede daha başarılı olduğunu ve seçilen ayırtaç sonuçları eğitilmiş bir doğrusal model ile birleştirildiğinde doğruluk başarımını daha da arttırılabildiğini görüyoruz.

Sabit kuralların davranışlarını anlamaya çabalarken, hata fonksiyonunu yanlılık, değişinti ve gürültü bileşenlerine ayırma fikrini uyguluyoruz. Bu çalışmada yazındaki kare ve 0/1 hata tanımlamalarının yanlılık, değişinti ve gürültü ayrıştırmalarını kısaca gözden geçiriyor, bu bileşenlerin toplulukların hata davranışını, özellikle en küçük ve en büyük kuralları kullanıldığında, açıklamada yetersiz kaldığını gösteriyoruz. Bazı kaynaşım yöntemlerinin, veri kümesinin üstünde birbiçimli veya Gauss gürültü varsayımları altında, ötekilerden daha iyi çalışmasının nedenlerini veriyoruz. Sabit kuralların davranışını açıklamak için kesişim alanına dayanan bir ölçü öneriyoruz.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

C_i	i^{th} class in a classification problem
d	Number of attributes of a data instance
D	A classifier ensemble
T	A discriminant ensemble
D_j	j^{th} classifier in a classifier ensemble
D^*	The set of optimal classifiers for a classification problem
K	Number of classes in a classification problem
k	Number of nearest instances to a specified instance
L	Number of all candidate classifiers in a classifier ensemble
n	Number of classifiers in a set of optimal classifiers
M	Number of discriminants in a discriminant based ensemble
m	Number of classifiers in a classifier ensemble
T^*	The set of optimal discriminants for a classification problem
T_i^j	Discriminant of i^{th} class of j^{th} classifier
X_{tra}	Training Dataset
X_{valA}	Validation Dataset A
X_{valB}	Validation Dataset B
X_{test}	Test Dataset
DT	Decision Tree
FSS	Forward Subset Selection
LIN	Linear Perceptron Model

1. INTRODUCTION

People try to automate many processes either to ease life or eliminate errors that can occur due to carelessness or insufficient knowledge. Machines are used for many processes in many areas in daily life, i.e. medical operations, signaling systems, etc.

We believe that the past knowledge or experience on a phenomenon can be used in detecting and identifying possible patterns, which can be used for future predictions. Many research studies in many fields, including computer engineering, try to construct systems that can learn from past data of the given problem.

Machine learning is programming computers to estimate the optimal response from example samples (training data) or past experience (Alpaydın, 2004). The response can be the class label in the case of classification, or any numeric value in the case of regression.

Perfect classification is often impossible, and the more general task is to determine the posterior probability of each candidate class given the input (Duda *et al.*, 2001). Discriminant functions can also be used for seperating classes from each other if only the label, not the posterior probability, matters. Discriminant scores can have different meanings and ranges depending on the application.

There are many classification algorithms and each has its own underlying assumption about the distribution of the data. These assumptions can either become a strength when they hold or a weakness if they do not. It is not an easy task to determine which classification method is suitable to use for a given dataset. Therefore, the general approach is to construct many classifiers from the training dataset and then design a strategy to improve the accuracy by suitably combining them (Kuncheva, 2004).

1.1. Classifier Combination

There is no universal classifier that works as the most accurate in all datasets. Each classification algorithm has its own inductive bias and makes different assumptions about the data. Most of the cases, the different classifiers err on distinct samples. If they deliver a set of classification results, then a combination method might be applied to combine these diverse results such that the accuracy of the ensemble is higher than the accuracy of any individual classifier on the ensemble (Rahman and Fairhust, 2003).

The combination works well only when the individual classifiers misclassify different samples, that is, when there is diversity. Otherwise, there can be no improvement by combination. If the classifiers produce diverse results, then the next question will be how to combine them. In the literature, there are many proposed methods, each with its own advantages and disadvantages (Kuncheva, 2004).

In a multiple learner system, the models to be used in the ensemble should be selected carefully to increase the accuracy. Increasing the size of the ensemble improves the accuracy only up to a point. If a new expert does not contribute to diversity, it will only unnecessarily increase the cost of the ensemble in terms of both time and space. It is therefore essential that given a large set of classifiers, a subset needs to be found. An algorithm proposed for this purpose constitutes a contribution of this thesis.

1.2. Bias–Variance Analysis

The source of an expert's error is decomposed into three components. Bias is a measure of fitness of the algorithm to the data. For instance, if the classes are not linearly separable, a linear model will have high bias and perform poorly. The parameters learnt during training will be clearly dependent on the samples in the training set. The models constructed from different training sets will be different from each other, unless our model is constant, which means no learning is done. Instable algorithms such as decision trees, will be more influenced by a change in the training set.

Variance is the component that accounts for this sensitivitiy. Noise is the unpreventable component and is the variance of the noise added to the data.

These concepts are helpful in expressing the causes of the error, however, their evaluations vary with loss functions used. Although these concepts are straightforward, there is no unique decomposition for 0/1 loss. There are different decompositions proposed in the literature, which are discussed in Section 3.3. Evaluating different combination rules using the bias–variance decomposition constitutes a second contribution of this thesis.

1.3. Objectives and Approach

This study is made of two separate sections. In the first section, we are concerned with developing an ensemble that is constructed with a subset of discriminants instead of the whole classifier set.

In our methodology, firstly L base classifiers, $D_1, \ldots, D_L$, from different families are trained. Then, from $L \cdot K$ discriminants of the base classifiers, where K is the number of classes, a subset of discriminants, $M \leq L \cdot K$, are selected and used. As a result, a more cost-effective ensemble can be constructed, since only the discriminants that give us information are used and not all.

The second part focuses on the bias–variance analysis of fusion strategies. The bias–variance decompositions of the ensembles are evaluated both empirically and theoretically for different loss functions, and the link between loss and misclassification rate is investigated. We are interested in expressing the performance of the ensemble in terms of bias and variance as the size of the ensemble and the distribution of the noise is varied.

1.4. Thesis Organization

This thesis has the following outline. In Section 2, we discuss discriminant ensembles. A brief survey of classifier fusion will be given at first. Then the proposed discriminant based ensemble construction methods are explained, together with case studies. A comparison with other classifier selection methods are also given. Section 3 concerns with bias–variance analysis of fusion rules. It begins with a summary of the proposed bias–variance decompositions and ensemble misclassification rate evaluations in the literature. The applications of the bias–variance decomposition to the ensembles are theoretically explained. The relation between them and misclassification error is discussed. Conclusions and possible future work are stated in Chapter 4.

2. DISCRIMINANT ENSEMBLES

Discriminants are functions that return scores whose values can be used as membership measures of an input sample. The scores indicate which class the sample most likely belongs to. A discriminant score can be any value in any range interval. Many classification algorithms use discriminants for decision and we view each classifier as being composed of discriminants. Since the classifiers are combined to have a more accurate classifier, it is possible that only some of the discriminants (a part/subset of classifier) need to be used in combination. This chapter focuses on why and how to use discriminants in an ensemble.

2.1. Classifier Ensembles

Classifier combination has been very popular recently. The idea behind is that any single classifier might not be the optimal classifier, and an ensemble of classifiers can complement each other and reduce the error (Figure 2.1).

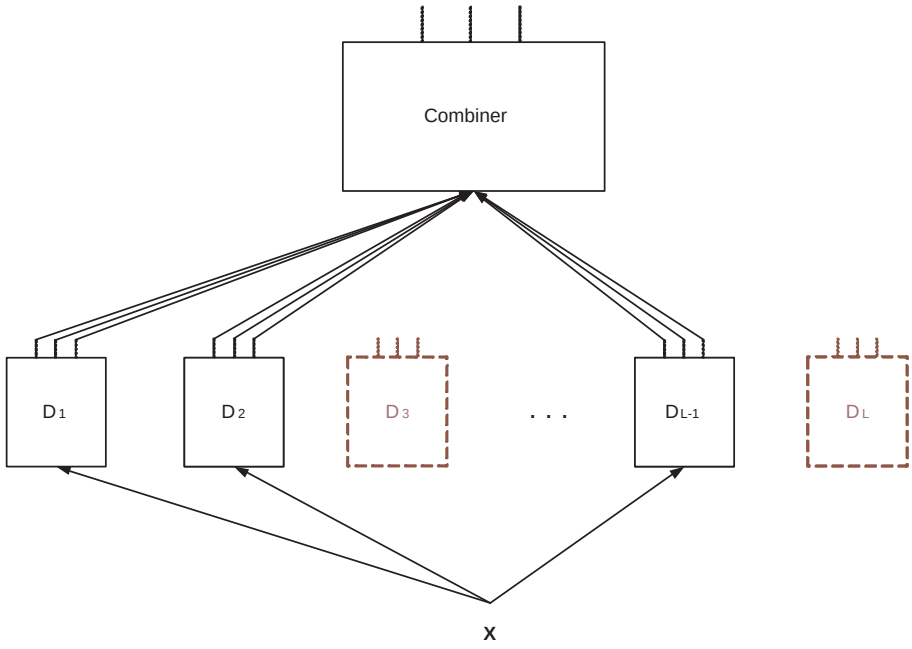


Figure 2.1. Ensemble of Classifiers

2.1.1. Terminology

The following notation is used throughout next section:

- It is assumed that we have L base classifiers (experts, learners, models, ...) in a classifier ensemble:
  $D = [D_1, \ldots, D_L]$.
- All base classifiers D_j, $j = 1, \ldots, L$, are trained with the training dataset X_{tra}.
- Output of any base classifier D_j is a set of discriminant scores for input x, $T_i^j(x)$, $i = 1, \ldots, K$, for example, posterior probabilities.
- Discriminants are selected and their outputs are combined using a trained meta-classifier by using a validation dataset X_{valA}.
- The optimal subset $D^* \subseteq D$ includes $n \leq L$ of the classifiers for a given test instance x.
- The optimal subset $T^* \subseteq T$ includes $M \leq (K \cdot L)$ of the discriminants for a given test instance x.
- The final output is the class label s that has the highest overal discriminant value:

$$s = \arg\max_i T_i(x) \tag{2.1}$$

- The misclassification error is calculated when the estimated class does not match the true class, r:

$$ERR(x) = \begin{cases} 1 & \text{if } \arg\max_i T_i(x) \neq r \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

2.1.2. Previous Works on Classifier Combination

Classifiers can be categorized into three with respect to their output scores:

- *Labels*: The classifier returns s, the label of class it assigns to the sample.
- *Ranks*: The candidate labels are returned in the order of their plausibility of

being the correct class.

- *Scores*: The classifier returns a vector of K dimensions, whose elements represent the support values for each class.

In this study, only combination of scores is of concern. There are many combination rules proposed in the literature. Fixed rules do not need any extra training phase, once the base classifier scores are obtained. There is no cost overhead except the complexity of the base classifiers. Voting and other rules (i.e. average, sum, product, minimum, median and maximum) (Kittler *et al.*, 1998) belong to this group. In stacking, the final decision is given by a trained meta-classifier on the second-level, which processes the scores of the base classifiers as its inputs (Wolpert, 1992). It needs an extra training set for the upper-level learner, otherwise it would be biased with the training set used by the base classifiers. The mixture of experts architecture constructs local models and a gating network which choses the experts with respect to their location in the input space (Jacobs *et al.*, 1991).

There are other methods based on resampling. The rationale is not combining different algorithms, but averaging over perturbations of the traning set. In bagging (Breiman, 1996), the instable classifiers, commonly decision trees, trained by bootstrap replicates of the training set are combined. In this way, diversity is improved. AdaBoost (Freund *et al.*, 1996) is an incremental ensemble construction technique where in each step, a new classifier is trained on the data selectively sampled from the training set X_{tra}. The selected data contains the samples which the previous classifier in the ensemble misclassifies.

Composite models of combination and selection can also be implemented. Kılıç (2007) proposes a composite system which selects dynamically an optimal subset of classifiers from an ensemble for a given test input. Only those classifiers competent in the region of the input are selected and combined to give the final decision.

The sample size of the training set is also influential in deciding which fusion rule to use. Raudys emphasizes in his recent studies (Raudys, 2006a; Raudys, 2006b) that

when the training data size is small, fixed rules should be preferable over trainable rules. Raudys also concludes that if the base classifiers can work well in separate regions of the input space, a simple rule, such as Behaviour Knowledge Space, can work successfully with label outputs in case the dataset size is large. Otherwise a trainable rule using continuous outputs of the base classifier should be preferred.

Diversity is a related notion. Intuitively, learners should be diverse, that is, they should not err on the same instances otherwise there would be no gain through combination. AdaBoost is an example of how diversity influences the accuracy of the ensemble. Although conceptually diversity is needed for multi-classifier system, the relation between diversity and accuracy is difficult to define precisely (Kuncheva and Whitaker, 2003).

During combination, most of the rules, particularly fixed rules, assume either that the scores are posterior probabilities or that they have same the measure levels (same meanings and ranges). If this is not the case, normalization should be applied (Liu, 2005). But the normalization method should be chosen carefully, otherwise, there will be information loss (Jain *et al.*, 2005).

During construction of the ensembles, the models must be chosen carefully to reduce the error. In particular, model combination through averaging reduces variance, and hence error, but it is valid only if bias does not increase, or if the concomitant increase in bias is small with respect to the decrease in variance. Consequently it is essential that only those models which contribute to accuracy are included and the poorly performing ones are weeded out. Additional to its effect on statistical accuracy, each additional model means an increase in the space and computational complexity. A new model may also be sensing/extracting a costly representation which is an extra burden if the model is redundant. Methods therefore have been proposed to choose a small subset from a large set of candidate models. L candidate models can constitute 2^L possible subsets of it is not feasible to try for all possible subsets unless L is small, and various algorithms have been proposed to get a reasonable subset of size $m < L$ in reasonable time (Ulaş *et al.*, 2007).

A method by proposed Ulaş (2007), ICON (Incremental CONstruction) algorithm, chooses m out of L base classifiers greedily. It starts with the empty set and incrementally constructs an ensemble where at each iteration, it selects among all possible classifiers the one that best improves the performance when added to the current ensemble (Demir and Alpaydın, 2005). The performance evaluation criterion used are accuracy, Q statistics, correlation coefficient and cross-validation. The procedure is repeated until no more improvement is observed.

Despite the fact that trained rules have lower bias, fixed rules are mostly preferred owing to their advantages. (1) Fixed rules have no extra cost of storing and processing. (2) The training set can be used for completely training the base classifiers, which means better-trained base classifiers. The second-layer learner needs its own training set and hence the training data must be divided into two parts, which can cause undertrained base classifiers. (3) A trained rule requires extra time for training.

An experimental study done by Alkoot and Kittler (1999) shows that, among the fixed rules, sum and median rules seem more robust to noise. Kuncheva (2002) also investigates performance of six fusion strategies assuming that the outputs are independent and follow either Gaussian or uniform distribution. She concludes that in the two–class case, the minimum and maximum rules (which are the same for two–class cases) performs drastically better when the outputs are from a uniform distribution. Similar conclusions are drawn for Gaussian output distribution. Although in these studies the base classifiers are assumed to be independent, really they are not. They have correlations, and that must be kept in mind during the construction of an ensemble (Ulaş *et al.*, 2007). For example, instead of using variants from the same family (i.e., support vector machines with different polynomial kernels), using members from different families (i.e. trees and multilayer perceptrons) will result in a more accurate ensemble.

2.2. Ensembles of Discriminants

2.2.1. Rationale

The methods previously proposed in the literature focus on construction of classifier ensembles. They combine the classifiers which maximize the overall accuracy. In this work, we view a classifier as a set of discriminants, and are interested in constructing an ensemble of carefully selected discriminants.

In the ensemble of classifiers, m classifiers from L candidate classifiers are selected. In the ensemble of discriminants where K is the number of classes, $M \geq K$ discriminants from $L \cdot K$ are chosen (Figure 2.2). We consider a new intermediate $L \cdot K$ dimensional space in which we do feature selection and then classification.

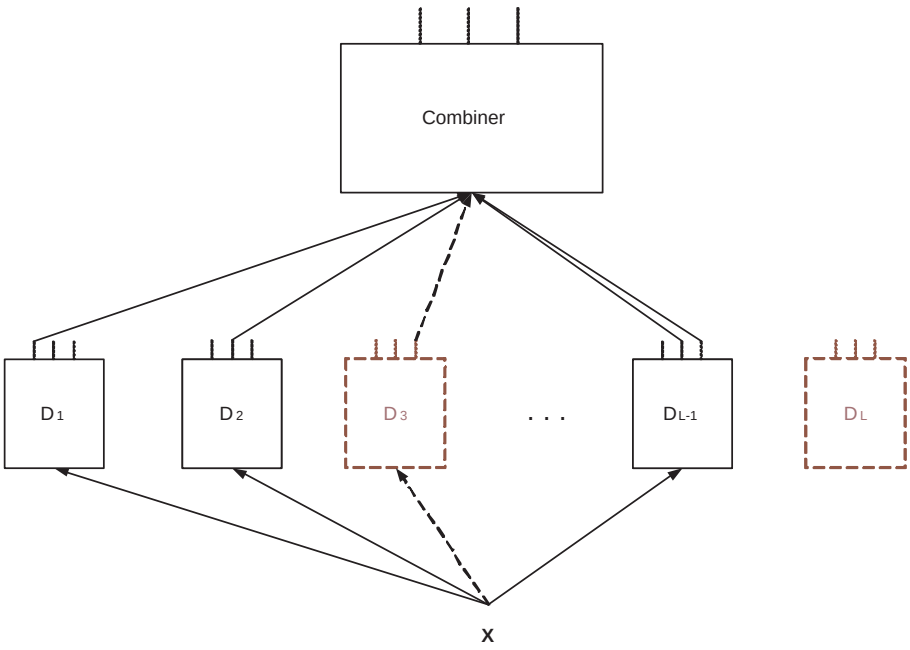


Figure 2.2. Ensemble of Discriminants

2.2.2. Discriminant Selection

Similar to the classifier ensemble construction, adding a new discriminant to an ensemble does not necessarily contribute to overall accuracy. An elaborative selection of a discriminant subset can result in a simpler and more accurate ensemble. Since base

classifier discriminants constitute a feature space for the second level fusion expert, we can use feature selection methods. We use three discriminant selection and combination algorithms:

- *Forward subset selection* (FSS): This is an incremental algorithm that adds one discriminant at a time until there is no further improvement in accuracy (Figure 2.3). There is a linear combiner to calculate the overall output from the discriminants and this combiner is trained on a set which is different from the one on which the base-discriminants are trained. The final linear model includes the discriminants as features in the intermediate space which are enough for a more accurate classification. The redundant discriminants are weeded out.

1. Given $X_{valA} \in R^{(L \cdot K)}$, initialize $T^{(0)} = \emptyset$ and $t = 0$.
2. do{
   $j = \arg\min_i \text{ERR}(\text{LIN}(T^{(t)} \cup X_{valA(i)}))$
   $t = t + 1$
   $T^{(t)} = T^{(t-1)} \cup X_{valA(j)}$
   } while($\text{ERR}(\text{LIN}(T^{(t)})) < \text{ERR}(\text{LIN}(T^{(t-1)}))$)
3. Return $\text{LIN}(T^{(t-1)})$

Figure 2.3. Forward Subset Selection Algorithm

- *Decision tree* (DT): A decision tree is trained to learn the final output from the $L \cdot K$ dimensional discriminant values (Figure 2.4). The decision tree acts both as a feature selector and a classifier. The trained tree gives the final decision for a given sample.

1. Given $X_{valA} \in R^{(L \cdot K)}$, initialize $T^{(all)} = X_{valA}$.
2. Return $\text{DT}(T^{(all)})$

Figure 2.4. Decision Tree Algorithm

- DT *with linear output* (DT.LIN): A decision tree is trained as above but instead of using it also for decision, it is used for feature selection only (Figure 2.5). That is, we first train the tree, take the features it uses, and give them as input to a linear model. The features used by decision tree indicate which discriminants are the best at separating a class from the others.

1. Given $X_{valA} \in R^{(L \cdot K)}$, initialize $T^{(all)} = X_{valA}$.
2. $T^{(chosen)} = \text{FEATURES}(\text{DT}(T^{(all)}))$
3. Return $\text{LIN}(T^{(chosen)})$

Figure 2.5. Decision Tree with Linear Output Algorithm

2.2.3. Features of Discriminant Ensembles

There are some interesting points to note about discriminant ensembles. They are simple both in terms time and memory. They do not contain any redundancy. Unlike classifier ensembles which usually need posteriors as base classifiers' scores, they do not make any assumptions about the output scores of the base classifier. They also inform us about which expert is good at discriminating which classes.

A classifier may be accurate for some classes but not all, and it is more desirable to use it only for classes it is good at. Each classifier has assumptions over the data, which may hold for some classes but not all. Take the case of a linear discriminant: In a multi-class case, one of the classes may be linearly separable from the others but another classes may not be. The output of the linear classifier may therefore be included in the ensemble of discriminants for that class, but for some other class, a more complex discriminant should be included in the ensemble. Certain classes may be handled by a single discriminant, i.e., one classifier can be very successful at separating it from the others, but more discriminants may be necessary for more complex classes.

In case we know that a classifier will not be used for some classes, the overall complexity can be decreased, both in terms of memory and time. There is no need to

store the parameters for the unused discriminants and the unused discriminants need not be calculated. Let us consider the case of support vector machines. We do not need to store the support vectors of a class nor do the costly kernel calculations if it is not used for that class.

For the proposed discriminant selection and combination methods in this thesis, the base discriminants do not need to be posteriors nor normalized in any manner. Both the linear model and the decision tree can use inputs in any scale and range, and no transformation or scaling is necessary, avoiding any risk of distorting values (Jain *et al.*, 2005).

The ensemble of discriminants also provides knowledge extraction throughout feature selection. It informs us which base classifiers can be trusted for which classes. There may be certain base classifiers which are not selected for any class, there may be ones which are selected for one or few, and there may be ones which are trusted for many. Certain classes may be easy to separate with one discriminant; more difficult classes might need to combine a number of discriminants.

2.3. Simulation Results

2.3.1. Datasets

We use 38 datasets from the UCI machine learning repository (Newman *et al.*, 1998), Delve (Rasmussen *et al.*, 1996). Table 2.1 shows the properties of datasets.

2.3.2. Training, Validation and Test Set Divisions

Any given dataset is divided firstly into two parts: 1/3 of it is test set, *test*, and the remaining 2/3 is training set, *trainall*. The training is resampled using 5×2 cross-validation with stratification. Two equal size random splits of training sets shift roles as training set and validation set for five times, which gives ten folds of training and validation sets, X^i_{tra} and X^i_{val}, $i = 1, \ldots, 10$. The validation set X^i_{val} is also divided

Table 2.1. Properties of the Datasets

name	instances	inputs	classes	source
zoo	101	16	7	uci
iris	150	4	3	uci
tae	151	5	3	uci
hepatitis	155	19	2	uci
wine	178	13	3	uci
flags	194	26	8	uci
glass	214	9	6	uci
heart	270	13	2	uci
haberman	306	3	2	uci
flare	323	10	3	uci
ecoli	336	7	8	uci
bupa	345	6	2	uci
ionosphere	351	34	2	uci
dermatology	366	34	6	uci
horse	368	26	2	uci
monks	432	6	2	uci
vote	435	16	2	uci
cylinder	540	45	2	uci
balance	625	4	3	uci
australian	690	14	2	uci
credit	690	15	2	uci
breast	699	9	2	uci
pima	768	8	2	uci

Table 2.1. Properties of the datasets (continued)

name	instances	inputs	classes	source
tictactoe	958	9	2	uci
cmc	1473	9	3	uci
yeast	1484	8	10	uci
car	1728	6	4	uci
titanic	2201	3	2	delve
segment	2310	19	7	uci
thyroid	2800	27	4	uci
optdigits	3823	64	10	uci
spambase	4601	57	2	uci
pageblock	5473	10	5	uci
ringnorm	7400	20	2	delve
twonorm	7400	20	2	delve
pendigits	7494	16	10	uci
mushroom	8124	22	2	uci
nursery	12960	8	4	uci

randomly into two as X^i_{valA} and X^i_{valB}, where X^i_{valA} is used for training the combiner and X^i_{valB} is used for model validation of ensemble in each fold. The *test* results evaluated at each fold are used in 5×2 cross-validation (cv) F tests (Alpaydın, 1999) for comparison of statistically significant difference between ensemble accuracies.

2.3.3. Base Classifiers

Ten base classifiers are chosen from a possible 19 to decrease the computational complexity as follows: Whenever there is a hyperparameter (as k of k-nearest neighbor), the version that has the highest accuracy on X_{valB} are selected and used. This is because we notice that when the hyperparameter is varied, the classifiers are very correlated (Ulaş, 2007) and another one almost never is chosen and keeping them unnecessarily increases computation.

- *c4* (C4.5): The standard C4.5 decision tree algorithm.
- *ga* (gaussian classifier): This is the parametric classifier where each class is represented by a Gaussian and a common covariance matrix is shared by all classes. This is a linear model.
- *kn* (k-nearest neighbour): Of $k = 1, 3, 5, 7$, we choose the most accurate.
- *lo* (logistic classifier): This is the linear logistic discriminator trained to minimize cross-entropy by gradient-descent. Discrete features are converted to numeric features by 1-of-n encoding.
- *ml* (multilayer perceptron): There is a single hidden layer where, with d inputs and K classes, the number of hidden units can be d, K, $d+K$, $(d+K)/2$, $2(d+K)$. Of these five, the most accurate one (on the validation set) is taken.
- *mu* (multinomial tree): This is the multivariate tree where each decision node is linear (Yıldız and Alpaydın, 2000) and uses all inputs as opposed to a univariate decision tree, such as C4.5.
- *se* (selectmax): This simple classifier assigns the class with the highest prior for any given sample without any evaluation.
- *sv* (support vector machine): LIBSVM 2.82 (Chan and Lin, 2001) with a linear kernel (*sv1*), radial (Gaussian) kernel (*svr*), and the most accurate of three

polynomial kernels of degree 2, 3, 4. (*sv2*, *sv3*, *sv4*) are used in the experiments.

2.3.4. Compared Ensembles

In this thesis, the following ensemble techniques are compared:

- BEST: We order the base classifiers in terms of accuracy and use the first best 1, 3, 5, 7, 9 of them. This has two variants; BEST.SUM uses the fixed sum rule and BEST.LIN uses the trained linear combiner.
- RND: We randomly choose 1, 3, 5, 7, 9 base classifiers, with SUM and LIN options.
- ALL: All the available base classifiers are combined without selection, with SUM and LIN options.
- OPT: We try all possible subsets (there are 2^{10}) and choose the best. It has SUM and LIN options.
- ICON: The classifier ensembles generated by Icon variants, ACC, CV, QSTAT, and CORR, are used. They all have SUM and LIN options (Ulaş, 2007).
- FSS: This is the proposed discriminant ensemble technique using Forward Subset Selection and a linear combiner.
- DT: This is the proposed discriminant ensemble technique which uses decision tree.
- DT.LIN: This is the proposed discriminant ensemble technique which uses decision tree for feature selection and a linear combiner.

2.3.5. Case Studies

In this section, two datasets, optdigits and nursery, are investigated deeply as case studies.

2.3.5.1. Optdigits Dataset. Table 2.2 shows the accuracies of base classifiers and the ensembles on *test* (X_{test}), together with the number of classifiers (# cla), the number of discriminants (# disc), and the chosen ensembles. Figure 2.6 is the plot of accuracies

vs the number of base classifiers on *test*. On this dataset (as many others), using a linear combiner .LIN does not contribute to accuracy significantly, and therefore only .SUM results are given to keep the table and figures simpler.

The optimal subset OPT chooses only three out of ten and it is more accurate than ALL that uses all ten. Choosing the best m base classifier, BEST, with respect to accuracy and combining them seems to work rather well. The reason is that the base classifiers are from different families and already complement each other. When too many base classifiers are added, the accuracy of BEST decreases; this is because the increase in bias surpasses the decrease in variance. Choosing a random subset, RND, works well but more base classifiers are needed. The ICON variants that use diversity measures, QSTAT and CORR, use too many base classifiers, showing that it is best to use accuracy as the ensemble evaluation criterion rather than an intermediate diversity measure.

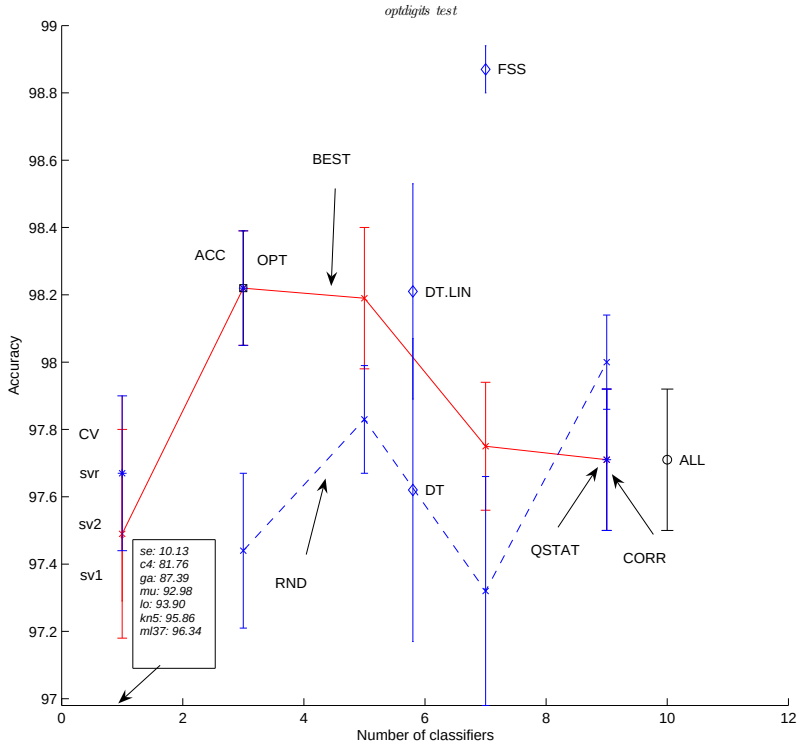


Figure 2.6. Accuracy vs the number of classifiers on Optdigits *test*.

FSS uses 21 discriminants from seven base classifiers. It is interesting that it

Table 2.2. Results on Optdigits. Number of classifiers and discriminants of DT are the average of ten folds, and the chosen discriminants are one of ten folds.

Alg	*test*	# cla	# disc	Chosen
se	10.13±0.0	1	10	*se*
c4	81.76±1.3	1	10	*c4*
ga	87.39±0.3	1	10	*ga*
mu	92.98±1.0	1	10	*mu*
lo	93.90±0.4	1	10	*lo*
kn5	95.86±0.3	1	10	*kn5*
ml37	96.34±0.3	1	10	*ml37*
sv1	97.48±0.2	1	10	*sv1*
svr	97.67±0.2	1	10	*svr*
sv2	97.49±0.3	1	10	*sv2*
BEST.3.SUM	98.22±0.2	3	30	*sv2 svr sv1*
BEST.5.SUM	98.19±0.2	5	50	*sv2 svr sv1 ml37 kn5*
BEST.7.SUM	97.75±0.2	7	70	*sv2 svr sv1 ml37 kn5 lo mu*
BEST.9.SUM	97.71±0.2	9	90	*sv2 svr sv1 ml37 kn5 lo mu ga c4*
RND.3.SUM	97.44±0.2	3	30	
RND.5.SUM	97.83±0.2	5	50	
RND.7.SUM	97.32±0.3	7	70	
RND.9.SUM	98.00±0.1	9	90	
ALL.SUM	97.71±0.2	10	100	
OPT.SUM	98.22±0.2	3	30	*svr sv2 sv1*
ACC.SUM	98.22±0.2	3	30	*sv2 svr sv1*
CV.SUM	97.67±0.2	1	10	*svr*
CORR.SUM	97.71±0.2	9	90	*c4 svr ga mu sv2 kn5 ml37 lo sv1*
QSTAT.SUM	97.71±0.2	9	90	*c4 ga mu lo kn5 ml37 sv2 sv1 svr*
FSS	98.87±0.1	7	21	*kn5*(0,3) *ga*(0,3) *ml37*(9) *mu*(4) *sv1*(1,2,5,9) *sv2*(2,4,5,7,8,9) *svr*(4,5,6,7,8)
DT	97.62±0.4	5.8	11.9	*kn5*(5,6) *sv1*(0,7) *sv2*(1,2,3,4) *svr*(8,9)
DT.LIN	98.21±0.3	5.8	11.9	*kn5*(5,6) *sv1*(0,7) *sv2*(1,2,3,4) *svr*(8,9)

chooses more than one discriminants for some classes. For example, it uses three discriminants for class "5", while DT uses mostly one discriminant for it. A reasonable cause for this is that, it uses discriminants of class "5" not only for separating it from other classes but also for discriminating the other classes from each other. It has more base classifiers than ACC but uses less discriminants. It outperforms even OPT which is the best classifier ensemble that can be achieved.

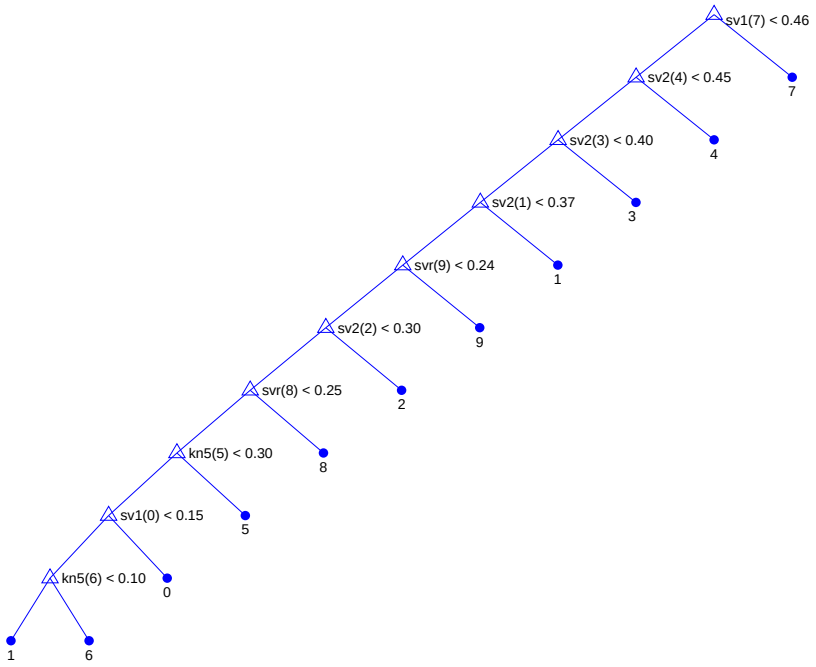


Figure 2.7. A decision tree learned by DT on Optdigits

DT chooses 11.9 discriminants (where the smallest possible set with ten classes can be nine discriminants) from 5.8 classifiers on average. It is even more explanatory than FSS, since it gives the threshold values used and for which classes they are used. Figure 2.7 shows an example DT obtained in one of the ten folds. It starts by looking at the output of *sv1* for class "7" and chooses "7" if this value is bigger than 0.46; there is no more discriminant evaluation for class "7". Note that we only need to evaluate the discriminants in our path; for example we see here that only *kn5* is evaluated to distinguish "5" from others. Most classes can be separated from the rest by looking at the value of a single discriminant; only "1" requires two, (*sv2* and *kn5*). DT.LIN improves over DT, but not significantly. DT.LIN using the discriminants chosen by DT, seems to be a little more stable (smaller standard deviation) and more robust to

the noise. Because unlike the case for DT, not a certain discriminant gives the decision but all discriminants contribute, which means more information is used while making the final decision.

2.3.5.2. Nursery Dataset. Similar results are obtained for Nursery dataset, except that .LIN significantly outperforms .SUM, as shown in Figure 2.8. The overall classifier and ensemble results are given in Table 2.3.

On this dataset, the base classifiers have accuracies ranging from 33.33 to 99.41 per cent and we see the benefit of subset selection. We see that .LIN should be preferred when the ensemble is made of classifiers with different accuracy rates. The reason why .LIN outperforms .SUM is that when the ensemble size increases, .SUM becomes influenced by the inaccurate base classifiers and overall accuracy degrades, but .LIN adapts itself and ignores the inaccurate classifiers.

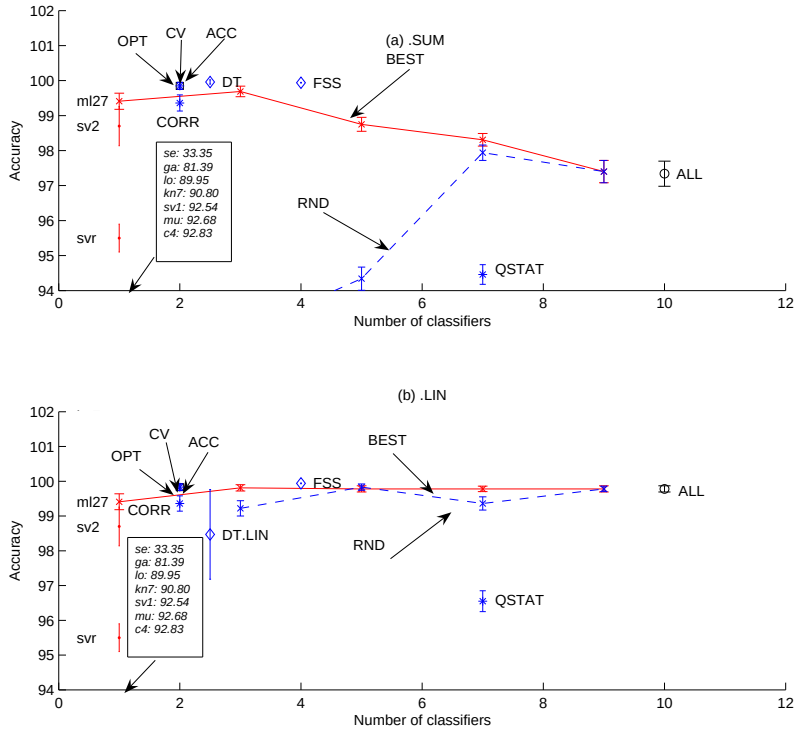


Figure 2.8. Comparing the two combination methods, fixed .SUM and trained .LIN, in terms of accuracy vs the number of classifiers on Nursery *test*.

FSS uses seven discriminants from six base classifiers, while DT uses seven discriminants from four classifiers on average. It is simple and has an acceptable accuracy range. Figure 2.9 shows one of the trees learned on the dataset and has only three decision nodes. It is very simple and interpretable and has 99.96 per cent *test* accuracy, which is higher than what we get when all the outputs of all base classifiers for that fold are used (99.78).

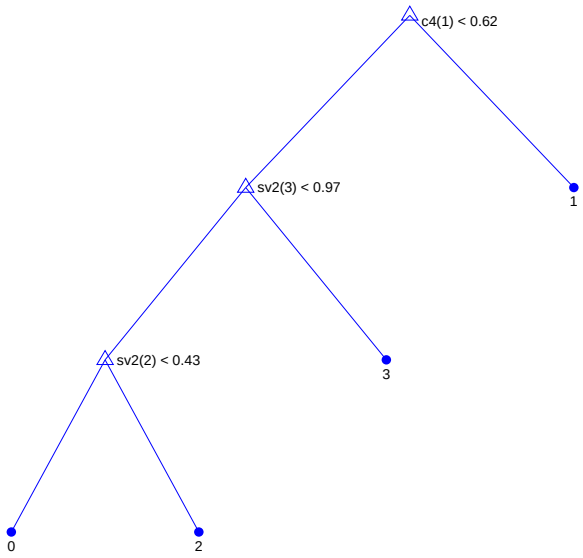


Figure 2.9. A decision tree learned by DT on Nursery

2.4. Overall Results

Table 2.4 gives the comparison of the all ensemble methods in a pairwise manner on *test*. The numbers are wins and losses of method in the row over the method in the column. The number of ties can be found when the sum of wins and losses is subtracted from 38. A bold entry means that the number of wins/losses over 38 is statistically significant using the Sign test.

The average number of base classifiers(/discriminants) contained in the ensembles constructed is given in Table 2.5. The discriminant values are normalized by the class number to have a common range.

Table 2.3. Results on Nursery dataset. Number of classifiers and discriminants of DT are the average of ten folds, and the chosen discriminants are one of ten folds.

Alg	*test*	# cla	# disc	Chosen
se	33.33±0.0	1	4	*se*
ga	81.36±0.3	1	4	*ga*
lo	89.94±1.4	1	4	*lo*
kn7	90.43±0.3	1	4	*kn7*
sv1	92.43±0.3	1	4	*sv1*
mu	92.74±0.4	1	4	*mu*
c4	92.72±0.5	1	4	*c4*
svr	95.50±0.4	1	4	*svr*
sv2	98.70±0.6	1	4	*sv2*
ml27	99.41±0.2	1	4	*ml27*
BEST.3.LIN	99.81±0.1	3	12	*ml27 sv2 svr*
BEST.5.LIN	99.78±0.1	5	20	*ml27 sv2 svr c4 mu*
BEST.7.LIN	99.78±0.1	7	28	*ml27 sv2 svr c4 mu kn7 sv1*
BEST.9.LIN	99.78±0.1	9	36	*ml27 sv2 svr c4 mu kn7 sv1 lo ga*
RND.3.LIN	99.22±0.2	3	12	
RND.5.LIN	99.83±0.1	5	20	
RND.7.LIN	99.36±0.2	7	28	
RND.9.LIN	99.78±0.1	9	36	
ALL.LIN	99.78±0.1	10	40	
OPT.LIN	99.83±0.1	2	8	*ml27 sv2*
ACC.LIN	99.83±0.1	2	8	*ml27 sv2*
CV.LIN	99.83±0.1	2	8	*ml27 sv2*
CORR.LIN	99.36±0.2	2	8	*se ml27*
QSTAT.LIN	96.55±0.3	7	28	*ga svr c4 kn7 lo mu sv1*
FSS	99.94±0.0	4	7	*kn7*(3) *ml27*(0,3) *mu*(0) *sv2*(0,2,3)
DT	99.96±0.1	2.5	3	*c4*(1) *sv2*(2,3)
DT.LIN	98.47±1.3	2.5	3	*c4*(1) *sv2*(2,3)

Table 2.4. Pairwise comparison of accuracies (wins/losses over 38) of all methods using 5×2 cv F-Test.

	Best	Rnd	All	Opt	Acc	Cv	Qstat	Corr	Fss	Dt	Dt.Lin
Best	0/0	3/0	**8/0**	0/1	0/0	0/0	**16/0**	**13/2**	9/5	7/2	4/4
Rnd	0/3	0/0	3/0	**0/6**	0/5	0/2	**14/0**	8/2	6/4	6/3	6/4
All	**0/8**	0/3	0/0	**0/10**	**0/8**	**0/6**	**13/0**	7/4	7/8	5/6	4/7
Opt	1/0	**6/0**	**10/0**	0/0	0/0	2/0	**17/0**	**14/1**	11/3	**10/1**	6/1
Acc	0/0	5/0	**8/0**	0/0	0/0	1/0	**16/0**	**13/1**	10/3	**8/1**	7/3
Cv	0/0	2/0	**6/0**	0/2	0/1	0/0	**15/0**	**11/2**	8/5	7/1	5/4
Qstat	**0/16**	**0/14**	**0/13**	**0/17**	**0/16**	**0/15**	0/0	2/6	**5/15**	4/9	**3/12**
Corr	**2/13**	2/8	4/7	**1/14**	**1/13**	**2/11**	6/2	0/0	**4/13**	2/5	2/9
Fss	5/9	4/6	8/7	3/11	3/10	5/8	**15/5**	**13/4**	0/0	6/2	3/6
Dt	2/7	3/6	6/5	**1/10**	**1/8**	1/7	9/4	5/2	2/6	0/0	1/5
Dt.Lin	4/4	4/6	7/4	1/6	3/7	4/5	**12/3**	9/2	6/3	5/1	0/0

Table 2.5. Average number of base classifiers (/discriminants) contained in different ensembles.

	Best	Rnd	All	Opt	Acc	Cv	Qstat	Corr	Fss	Dt
Sum	3.74	5.26	10.00	3.32	1.89	1.08	4.39	3.55	3.71/1.59	4.69/2.55
Lin	6.05	7.63	10.00	4.39	2.87	1.18	4.39	3.55		4.69/2.55

Besides their accuracies and complexities, the similarity between the ensembles found by different methods can show us a relationship between methods used. Given two ensembles E_i and E_j, we define the similarity between them as the number of shared base classifiers (or discriminants):

$$\text{Sim}(E_i, E_j) = \frac{|E_i \cap E_j|}{|E_i \cup E_j|} \tag{2.3}$$

A similarity score of 1 shows that the two ensembles are exactly the same (they use the same base classifiers), while a similarity score of 0 means that the two ensembles have no base classifiers in common.The average similarity between ensembles found by different methods can be found in Table 2.6.

Analyzing the all the tables contaning test results, similarity scores and average number of classifiers/discriminants, the overall results can be summarized in the paragraphs below:

1. With respect to the average number of base classifiers selected during the construction of the ensembles, the following ordering is obtained: Cv < Acc < Opt < Corr < Fss < Best < Qstat < Dt < Rnd < All.
2. Comparing the three discriminant ensembles, we see that even though there does not seem to be significant difference between them, on *test*, Dt.Lin is more accurate than Fss. On the average, Fss selects fewer discriminants than Dt variants and they have a low discriminant similarity score of 0.17. A reasonable explanation for this is that Fss focuses on choosing a feature that decreases the error over the whole training set, while Dt aims to minimize error over the data arriving to the current node (subtree) by selecting a convenient feature, and when we take a union over the features we get a larger set.
3. On the average, Fss results in a smaller ensemble than Dt. It mostly contains fewer base classifiers and discriminants. As explained above, the reason is that Fss stops increasing the ensemble when no more improvement is observed, but Dt can create deeper leaf levels if the current node (subtree) can reduce its error

Table 2.6. Average Similarity of Base Classifiers (Discriminants) Between Ensembles Found by Different Methods.

	Best	All	Opt	Acc	Cv	Qstat	Corr	Fss	Dt
Best	1.00	0.37	0.49	0.55	0.33	0.18	0.19	0.40	0.34
								0.23	0.22
All	0.37	1.00	0.33	0.19	0.11	0.44	0.36	0.37	0.47
								0.16	0.26
Opt	0.49	0.33	1.00	0.65	0.29	0.21	0.23	0.40	0.34
								0.23	0.21
Acc	0.55	0.19	0.65	1.00	0.36	0.12	0.15	0.30	0.25
								0.20	0.18
Cv	0.33	0.11	0.29	0.36	1.00	0.08	0.10	0.20	0.15
								0.15	0.10
Qstat	0.18	0.44	0.21	0.12	0.08	1.00	0.68	0.23	0.26
								0.13	0.15
Corr	0.19	0.36	0.23	0.15	0.10	0.68	1.00	0.21	0.21
								0.13	0.13
Fss	0.40	0.37	0.40	0.30	0.20	0.23	0.21	1.00	0.34
	0.23	0.16	0.23	0.20	0.15	0.13	0.13	1.00	0.17
Dt	0.34	0.47	0.34	0.25	0.15	0.26	0.21	0.34	1.00
	0.22	0.26	0.21	0.18	0.10	0.15	0.13	0.17	1.00

rate with a different discriminant. Consequently, DT results in bigger and more variant (distinct classifiers) ensembles.

4. Comparing the FSS and DT variants, FSS takes much more time to train, which is an disadvantage since shorter training time would be preferable.
5. DT.LIN seems to be more accurate than DT, but using an additional linear combiner has some obvious disadvantages, such as the need for extra validation data and training time. We can also use the advantages of DT being a tree: It is interpretable and easier to evaluate since only the nodes (discriminants) in our path must be calculated. We do not need to evaluate the extra linear combiner, implying faster test performance.
6. Comparing the ensembles of classifiers and the ensembles of discriminants, ACC is significantly more accurate than DT. ACC uses fewer classifiers but FSS uses fewer discriminants. ACC uses all the discriminants of the chosen classifiers which makes it costly but robust to noise due to redundancy; FSS and DT use a subset of the discriminants, are less noise-tolerant, but are simpler and faster. Particularly, DT will be more influenced by noise than DT.LIN, since the linear model, if trained enough, can overcome some noise.
7. Although the classifier and discriminant ensembles seem to contain a comparable number of base classifiers, in terms of the discriminants, the discriminant ensembles tend to need around half as many base discriminants, cutting by half the space and time complexity of training/testing. On some datasets, this can be as low as 20 per cent.

2.5. Conclusions and Future Work

In this study, we compare mainly two construction methods. In an ensemble of classifiers, we choose a subset from a larger set of base classifiers. In an ensemble of discriminants, we choose a subset of base discriminants, where a discriminant output of a base classifier by itself is assessed for inclusion in the ensemble.

A greedy, forward, incremental classifier/discriminant ensemble finds ensembles that are small, as accurate as the optimal ensemble, and does this in polynomial time.

Instead of using all the classifiers in an ensemble, using a subset of them can provide us an easier and more accurate ensemble. A collection of discriminants, either by an incremental method or a selection method, can generate us a simpler ensemble.

The discriminant ensemble has the advantage that not all discriminants of chosen classifiers are used. Just as not all base classifiers may not be needed for a given dataset and it is better to weed out those not needed to keep complexity in check, it may not be necessary to use a base classifier for all classes. Considering a classifier not as a single entity but as a collection of discriminants, one for each class, we can choose a subset of the discriminants thereby using a base classifier for some classes but not all. The corresponding discriminant is included, when the inductive bias of that classifier matches for a class, otherwise it is not used. This makes the whole ensemble much simpler, faster, and interpretable.

In this thesis, two major discriminant ensemble construction methods are proposed; one is an incremental construction method (FSS), the other is a construction method by discriminant selection (DT). The incremental method results in smaller ensembles than the selection method, but the selection method seems more interpretable. The test results show that although there is no significant difference, DT.LIN is the most accurate among them.

As a future work, other discriminant selection algorithms, such as genetic algorithms, can be implemented. Besides incremental methods, decremantal methods such as *Backward Subset Selection* or hybrid methods such as *Floating Search* can be used. Also cost–conscious construction methods can be proposed to get more simpler ensembles (Demir and Alpaydın, 2005). Although no transformation or normalization preprocessing is needed for discriminant ensembles, it might be interesting to investigate the influence of any normalization or other preprocessing on the accuracy of ensemble.

3. ERROR ANALYSIS OF CLASSIFIER FUSION RULES

This chapter is dedicated to error analysis of fusion rules. Fusion rules are used for improving the accuracy. But they err, too. 0/1 loss is usually used to evaluate the performance of classification algorithm. The concepts of bias, variance and noise in squared loss are also extended to 0/1 loss. This chapter focuses on bias-variance-noise decomposition of error of fusion rules. We see that such a decomposition is insufficient and we introduce the concept of the area of intersection as a measure of performance evaluation for fusion rules.

3.1. Introduction

Combining the decisions from multiple classifiers has recently become very popular (Kuncheva, 2004). The combination can be done using a fixed fusion rule (Kittler *et al.*, 1998) or by a trained combiner (Wolpert, 1992). There are many methods to train the base classifiers and a trainable combiner if it is used. How to train a combiner and how to make use of the available dataset for training depends on the conditions and the assumptions made (Duin, 2002). A trained combiner is more flexible but requires training, data put aside for its training, storage for its parameters, and more computation during test. In this study, we investigate fixed fusion rules and the ones we focus on are the average, minimum, median, and maximum rules on two-class problems.

We assume that there is an ensemble of L independent classifiers which in parallel produce their posterior probability estimates for given instance $\mathbf{x}$:

$$\hat{P}_1^j(\mathbf{x}) + \hat{P}_2^j(\mathbf{x}) = 1, j = 1, \ldots, L \tag{3.1}$$

where $\hat{P}_i^j(\mathbf{x})$ denotes the estimate by classifier j for the true posterior probability for class ω_i: $P(\omega_i|\mathbf{x})$. If the classifiers do not generate posterior probabilities, their outputs are suitably normalized and transformed to the same range (Jain *et al.*, 2005). The fusion rules, $F()$, given in Table 3.1 are used to combine the classifier estimates to

Table 3.1. Classifier Fusion Rules Used

Rule	Fusion Scores	Assuming $P_1 > 0.5$ Misclassification If
Average	$\hat{P}_i = \frac{1}{L}\sum_{j=1}^{L} \hat{P}_i^j$	$\hat{P}_1 \leq 0.5$
Minimum	$\hat{P}_i = \min_j \hat{P}_i^j$	$\hat{P}_1 \leq \hat{P}_2$
Median	$\hat{P}_i = \text{median}_j \hat{P}_i^j$	$\hat{P}_1 \leq 0.5$
Maximum	$\hat{P}_i = \max_j \hat{P}_i^j$	$\hat{P}_1 \leq \hat{P}_2$

calculate the overall estimate $\hat{P}_i(\mathbf{x}) \equiv \hat{P}(\omega_i|\mathbf{x})$:

$$\hat{P}_i(\mathbf{x}) = F(\hat{P}_i^1(\mathbf{x}), \hat{P}_i^2(\mathbf{x}), \ldots, \hat{P}_i^L(\mathbf{x})) \tag{3.2}$$

For an instance $\mathbf{x}$, if the true posterior $P_1(\mathbf{x}) > 0.5$, then the Bayes optimal class is ω_1 and it will be misclassified if $\hat{P}_1(\mathbf{x}) < 0.5$.

Some fusion rules work on hard labels (majority voting), and some work on soft continuous output scores (average, product, maximum, median, minimum). For majority voting, the scores of the classifiers must be converted into hard labels: $P_1^j(\mathbf{x}) = 1, P_2^j(\mathbf{x}) = 0$ if $\hat{P}_1^j(\mathbf{x}) > 0.5$, and $P_1^j(\mathbf{x}) = 0, P_2^j(\mathbf{x}) = 1$, otherwise. Then, voting chooses the class which receives the maximum number of votes. The average and product rules take the sum and product of the scores as their names indicate and label the instance with the class which has the maximum average or product value. The other three rules, minimum, median, and maximum, order the scores for each class and choose the score using order statistics, assigning the instance to the class with the maximum score. In this study, we do not analyze majority voting and product rules because their fused outputs are not probabilities and do not permit the type of analysis we do. Table 3.1 lists the rules used in this study and when a misclassification error occurs.

3.2. Previous Work of Analysis of Fusion Rules

The accuracy of fusion rules have previously been investigated: Chen and Cheng (2001) analyzed the asymptotic performance behavior of the ensemble as the number of classifiers increases. Alkoot and Kittler (1999), Kuncheva (2002), and Kittler and Alkoot (2003) investigated the effect of noise in classifier outputs on the fusion rules. In those studies, as we do here in this study, it is assumed that the noise added to a classifier comes from a uniform distribution or Gaussian (normal) distribution, and the performance of the fusion rule is evaluated either theoretically or empirically while parameters of the distributions and the size of the ensemble is varied.

Alkoot and Kittler (1999) experimentally investigate the performance of the fusion rules in case the posterior estimates are from normal or uniform distributions. They realize the experiments by adding independent and identically distributed (iid) noise to a fixed posterior value, p. They perform experiments by simulating base classifiers such that the base classifiers' decisions come from either a normal distribution, $N(p, \sigma^2)$, or a uniform distribution, $U[p-b, p+b]$. In other words, they add either normal noise with a predetermined σ and 0 mean, or uniform noise with a predetermined b and 0 mean to the fixed p value. If the obtained scores exceed either 0 or 1, they are clipped. They observe how the performance of the fusion rules change as the size of the ensemble (L), the base posterior (p) or the distribution spread parameter (σ or b) changes. They conclude that the performance of the ensemble improves with increasing L and there are borders where the success of the strategies changes. They order the fusion strategies with respect to their performance, and note that the order changes when the distribution spread parameter exceeds a boundary value. For uniform distribution this boundary value depends on the base posterior , p, we add the noise to. For normal noise, there is a constant boundary value, C, whose value depends on the number of classes and ensemble size.

The study of Chen and Cheng (2001) focuses on asymptotic behavior of average, median and majority rules. They assume that the estimates can come from any distributions and they theoretically demonstrate how the success of the rules changes when

the size of ensemble, L, goes to infinity. The median and maximum rule show the same performance while the average rule might behave differently depending on whether the distribution is symmetric or not.

There is also a case study of Kittler and Alkoot (2003) which compares sum with vote fusion in ensembles. They conclude that as long as the estimates follow a normal distribution, the sum rule will outperform the majority vote. But if they have heavy tail distributions, which means a significant mass at the tails, it is possible that vote gives better accuracy. They also note that the vote can outperform the sum rule, if either the margin between the two posterior probabilities are small or if the ensemble size is small even in case of large margins. They also draw attention to the fact that there can be much difference between real and theoretical results, since most assumptions such as identical and independent classifiers do not hold in practice.

Table 3.2. Theoretical Misclassification Error Rates of Fusion Rules

Method	$P_e \sim N(0,\sigma^2)$	$P_e \sim U[-b,+b]$ with $(p-b<0.5)$
Single classifier	$\Phi(\frac{0.5-p}{\sigma})$	$\frac{0.5-p+b}{2b}$
Minimum/Maximum		$\frac{1}{2}(\frac{1-2p}{2b}+1)^L$
Average	$\Phi(\frac{\sqrt{L}(0.5-p)}{\sigma})$	$\Phi(\frac{\sqrt{3L}(0.5-p)}{b})$
Median/Vote	$\sum_{j-\frac{L+1}{2}}^{L}\binom{L}{j}\Phi(\frac{0.5-p}{\sigma})^j[1-\Phi(\frac{0.5-p}{\sigma})]^{L-j}$	$\sum_{j-\frac{L+1}{2}}^{L}\binom{L}{j}(\frac{0.5-p+b}{2b})^j[1-(\frac{0.5-p+b}{2b})]^{L-j}$
Oracle	$\Phi(\frac{0.5-p}{\sigma})^L$	$(\frac{0.5-p+b}{2b})^L$

In this work, we start from the study by Kuncheva (2002) which shows how the fixed rules behave under noise for two–class case. It is assumed that each classifier output is generated by adding iid 0 mean noise to the true posterior probability, where the noise is uniform or Gaussian, where $p \equiv P_1(\mathbf{x})$ and $\hat{P}_1^j \equiv \hat{P}_1^j(\mathbf{x})$:

$$\hat{P}_1^j = p + \epsilon_j, j = 1, 2, .., L \tag{3.3}$$

where ϵ_j is either uniform, $U[-b,+b]$, where b denotes the spread, or Gaussian $N(0,\sigma^2)$ with variance σ^2. The estimate for ω_2 is then taken as $1-\hat{P}_1^j$.

These noisy estimates are combined using the fusion rule to calculate the overall

estimate for the two classes:

$$\begin{aligned} \hat{P}_1 &= F(\hat{P}_1^1, \hat{P}_1^2, \ldots, \hat{P}_1^L) \\ \hat{P}_2 &= F(1-\hat{P}_1^1, 1-\hat{P}_1^2, \ldots, 1-\hat{P}_1^L) \end{aligned} \tag{3.4}$$

Although she follows a similar experimental setup with them, unlike the study of Alkoot and Kittler (1999), she does not clip the observed scores, $\hat{P}_i^j$, when they exceed 0 or 1. If they exceed 0 or 1, she claims that they still give us their "support" for classes. Kuncheva theoretically calculates what the misclassification rate is and demonstrates how the misclassification error, P_e, is influenced when the size of the ensemble and the parameters of the noise distributions are changed. How a classification error occurs and the overall formulas of theoretical errors are given in Tables 3.1 and 3.2, respectively. The instance will be misclassified if the true probability $p \geq 0.5$, but $\hat{P}_1 < 0.5$ (or equivalently, if $\hat{P}_1 \leq \hat{P}_2$).

Since the classifiers are assumed to be independent from each other, for the oracle and median (which gives the same results with vote for two–class case) rules, she evaluates the misclassification error by means of binomial distribution: An individual classifier classifies either accurately or inaccurately. She calculates the misclassification by considering all possible cases of misclassification. For the average rule, she uses the fact that the average score performs a normal distribution for both normal and uniform noises, when L is big enough. For the minimum and maximum cases, no evaluation can be performed for the normal noise case. But for the uniform noise, she addresses an evaluation from a previous example (Mood *et al.*, 1974). Kuncheva concludes that the maximum and minimum rules outperform the other rules in uniform distribution case.

In this study, we go one step further in analyzing the performance of fusion rules by using a loss function and splitting it into bias, variance, and noise. We use both squared loss and the more appropriate 0/1 loss functions and the bias/variance decomposition for these loss functions. We give the reason why maximum and minimum

rules perform more accurately than the other fusion rules in the uniform case. We show that in th case of squared loss, the classification error of any instance increases and in the case of 0/1 loss, it seems to behave differently (increases, decreases and remains unchanged) depending on the decomposition and rule used, whereas in reality the misclassification rate, which is our real criterion, decreases. We give its causes. We therefore conclude that squared and 0/1 loss are not the appropriate loss functions to analyze the beaviour of fusion rules. We propose another measure, namely the area of intersection, which explains the behavior of fusion rules.

3.3. Bias, Variance and Noise Decompositions

In this section, we review the bias-variance-noise decompositions in the literature.

3.3.1. Squared Loss

The bias-variance-noise decomposition is originally for the squared loss, defined for regression. Let us assume that the true function is given as $y = P(\mathbf{x}) + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$. We calculate the estimate $\hat{P}(\mathbf{x})$ for y, using a set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, and the expected error is $\Sigma_i (y_i - \hat{P}(\mathbf{x}_i))^2$.

For a given unseen instance $\{(\mathbf{x}', y')\}$, the estimate $\hat{P}(\mathbf{x}')$, the expected square loss (mean square error) can be decomposed as:

$$\begin{aligned} E[(\hat{P}(\mathbf{x}') - y')^2] &= (E[\hat{P}(\mathbf{x}')] - P(\mathbf{x}'))^2 + E[(\hat{P}(\mathbf{x}') - \overline{\hat{P}(\mathbf{x}')})^2] + \\ & \quad E[(y' - P(\mathbf{x}'))^2] \end{aligned} \tag{3.5}$$

The first term in the righthand side of Equation 3.5 is the *squared bias* and shows the average goodness of fit of the model to the given data; for example, a linear model may have high bias if the underlying problem is not linear. The second term is *variance* and accounts for the sensitivity of the algorithm to the instances in the training set. We expect nonparametric models such as trees to have high variance,

whereas simple models such as linear models to have small variance. The last term is *noise* and represents the variance of the noise added to the data and is unpreventable even if the optimal model is used.

If the problem is regression, then using squared loss is reasonable. But for minimum or maximum rules, squared loss is not a meaningful criterion.

3.3.2. 0/1 Loss

For classification problems, more appropriate than the squared loss is the 0/1 loss where the loss of misclassification is 1 if there is a prediction error, i.e., $label(\hat{P}(\mathbf{x}')) \neq label(y')$, and is 0 if the prediction is correct. The expected 0/1 loss then corresponds to the probability of making a prediction error.

The bias-variance-noise decomposition for squared loss has similarly been proposed for 0/1 loss by several authors, as we review below.

3.3.2.1. Kohavi-Wolpert's Decomposition. The decomposition proposed by Kohavi and Wolpert (1996) is reminiscent of the case for squared loss:

$$\begin{aligned}
\text{bias} &= \frac{1}{2}\sum_i (P_i(\mathbf{x}) - \overline{\hat{P}_i(\mathbf{x})})^2 \\
\text{variance} &= \frac{1}{2}\left(1 - \sum_i (\overline{\hat{P}_i(\mathbf{x})})^2\right) \\
&= \frac{1}{2}\sum_i \overline{\hat{P}_i(\mathbf{x})}(1 - \overline{\hat{P}_i(\mathbf{x})}) \\
\text{noise} &= \frac{1}{2}\left(1 - \sum_i (P_i)^2\right) \\
&= \frac{1}{2}\sum_i P_i(\mathbf{x})(1 - P_i(\mathbf{x}))
\end{aligned} \tag{3.6}$$

Bias, as in the squared loss, compares the estimated posterior with the optimal posterior. Variance is high if the estimated posteriors are close to 0.5: It is maximum for the two-class case, when $\hat{P}_1(\mathbf{x}) = \hat{P}_2(\mathbf{x}) = 0.5$. Noise similarly is high if the true posteriors

are close to 0.5.

3.3.2.2. Breiman's Decomposition. Let us say α denotes the actual class and a denotes the estimated class:

$$\alpha = \arg\max_i P_i(\mathbf{x}) \quad a = \arg\max_i \overline{\hat{P}_i(\mathbf{x})} \tag{3.7}$$

The decomposition proposed by Breiman (1999) focuses on comparing the posteriors of α and a:

$$\begin{aligned} \text{bias} &= (P_\alpha(\mathbf{x}) - P_a(\mathbf{x}))\overline{\hat{P}_a(\mathbf{x})} \\ \text{variance} &= \sum_{i \neq a}(P_\alpha(\mathbf{x}) - P_i(\mathbf{x}))\overline{\hat{P}_i(\mathbf{x})} \\ \text{noise} &= 1 - P_\alpha(\mathbf{x}) \end{aligned} \tag{3.8}$$

Bias occurs only when the optimal classifier and the trained classifier disagree: $a \neq \alpha$. Variance is a measure of how much the estimated posteriors vary over the other classes. We note that if $\alpha = a$ and $\overline{\hat{P}_a(\mathbf{x})} = 1$, there will be no bias and variance, although the trained classifier is not same as Bayes optimal classifier. Noise measures how much the optimal class posterior deviates from 1, that is, the uncertainty of the Bayes classifier.

We note here that the misclassification error and the 0/1 loss classification error (bias/variance decomposition) are two closely related but different measures. Although in 0/1 loss, the expected error probability of instances in the dataset almost always is non-zero because of the irreducible noise component, it is possible to have no misclassification error.

3.3.2.3. Domingos' Decomposition. Domingos (2000) proposed a general decomposition for any kind of loss function. If the loss function $L(t, \hat{y})$ is the measure of the cost of predicting the common choice $\hat{y}$ instead of t, where y^* is the optimal value y is the

estimate of any random classifier, then the general decomposition can be defined as;

$$\begin{aligned} \text{bias} &= L(y^*, \hat{y}) \\ \text{variance} &= E(L(\hat{y}, y)) \\ \text{noise} &= E(L(t, y^*)) \end{aligned} \tag{3.9}$$

Using 0/1 loss, we get:

$$\begin{aligned} \text{bias} &= 1(\alpha \neq a) \\ \text{variance} &= \sum_{i \neq a} \hat{P}_i(\mathbf{x}) \\ \text{noise} &= 1 - P_\alpha(\mathbf{x}) \end{aligned} \tag{3.10}$$

But unlike the previous decompositions, Domingos' error definition is not a simple sum.

$$\text{error} = c_1 * \text{noise} + \text{bias} + c_2 * \text{variance} \tag{3.11}$$

where c_1 and c_2 takes values depending on the used loss functions. For two-class and 0/1 loss case, Eq. 3.11 becomes:

$$\text{error} = P_1 \overline{\hat{P}_2} + P_2 \overline{\hat{P}_1} \tag{3.12}$$

3.4. Bias/Variance Analysis of Fusion Rules

3.4.1. Squared Loss

We first investigate the bias and variance decomposition of fusion rules for squared loss. We have L iid classifiers, all trying to estimate p value for ω_1 sampled as $\hat{P}_j = p + \epsilon_j$, $j = 1, 2, ..., L$. ϵ_j is either uniform, $U[-b, b]$, or Gaussian, $N(0, \sigma^2)$.

3.4.1.1. Average Rule. For the average rule, have:

$$\begin{aligned} E[(1-\hat{P})^2] &= E[(\hat{P}-\overline{\hat{P}})^2] + (\overline{\hat{P}}-p)^2 + E[(1-p)^2] \\ &= E[(p+\overline{\epsilon}-\overline{p-\overline{\epsilon}})^2] + (\overline{p+\overline{\epsilon}}-p)^2 \\ &\quad +E[(1-p)^2] \\ &= E[(\overline{\epsilon}-\overline{\overline{\epsilon}})^2] + (\overline{\overline{\epsilon}})^2 + E[(1-p)^2] \end{aligned} \tag{3.13}$$

where $\overline{\epsilon}$ is the average of the noise in the fold, $\overline{\overline{\epsilon}}$ is the average of the average noises for all folds. We can easily see that $E[\overline{\overline{\epsilon}}]$ is 0, since we assume that noise comes from a symmetric (uniform or normal) distribution with mean 0. Also we have;

$$\sigma_{\overline{\epsilon}} = \frac{\sigma_\epsilon}{L} \tag{3.14}$$

3.4.1.2. Order Rules. For the minimum, maximum and median rules, we use order statistics (see Appendix for a review). Let us say that $\hat{P}_{(j)}$ are ordered as $\hat{P}_{(1)}, \hat{P}_{(2)}, \ldots, \hat{P}_{(L)}$ in ascending order. Then the rule chooses $\hat{P}_{(k)}$ where $k = 1, \frac{L+1}{2}, L$, for minimum, median and maximum rules respectively (assuming that L is odd). Then the expected square loss is:

$$\begin{aligned} E[(1-\hat{p}_{(k)})^2] &= E[(\hat{p}_{(k)}-\overline{\hat{p}}_{(k)})^2] + (\overline{\hat{p}}_{(k)}-p_{(k)})^2 + E[(1-p)^2] \\ &= E[(p+\epsilon_{(k)}-p-\overline{\epsilon}_{(k)})^2] + (p+\overline{\epsilon}_{(k)}-p)^2 + E[(1-p)^2] \\ &= E[(\epsilon_{(k)}-\overline{\epsilon}_{(k)})^2] + (\overline{\epsilon}_{(k)})^2 + E[(1-p)^2] \end{aligned} \tag{3.15}$$

which decomposes as variance + squared bias + noise. We see that except for noise, the loss depends on $\epsilon_{(k)}$. If we can calculate $\sigma^2_{\epsilon_{(k)}}$ and $\overline{\epsilon}_{(k)}$, we can calculate the expected square loss. Towards this, we need to be able to evaluate the first and second moments. For the first moment (expected value), we have

$$E[\epsilon_{(k)}] = \int_{-\infty}^{\infty} f_{\epsilon_{(k)}}(\epsilon)\epsilon d\epsilon \tag{3.16}$$

$$= \int_{-\infty}^{\infty} \frac{L!}{(k-1)!(L-k)!} F(\epsilon)^{k-1}(1-F(\epsilon))^{L-k} f(\epsilon)\epsilon d\epsilon$$

which, after some manipulation, reduces to:

$$\begin{aligned} E[\epsilon_{(k)}] &= \frac{L!}{(k-1)!(L-k)!} \sum_{i=0}^{L-k} \binom{L-k}{i} (-1)^i \\ &\quad \int_{-\infty}^{\infty} (F(\epsilon)^{i+k-1}) f(\epsilon)\epsilon d\epsilon \\ &= \frac{L!}{(k-1)!(L-k)!} \sum_{i=0}^{L-k} \binom{L-k}{i} (-1)^i \\ &\quad \left(\frac{F(\epsilon)^{k+1}}{k+1}\epsilon \Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \left(\frac{F(\epsilon)^{i+k}}{i+k} \right) d\epsilon \right) \end{aligned} \tag{3.17}$$

To calculate the variance, we need the second moment as well, because $\mathrm{Var}(\epsilon_{(k)}) = E[\epsilon_{(k)}^2] - E[\epsilon_{(k)}]^2$.

$$\begin{aligned} E[\epsilon_{(k)}^2] &= \int_{-\infty}^{\infty} f_{\epsilon_{(k)}}(\epsilon)\epsilon^2 d\epsilon \\ &= \frac{L!}{(k-1)!(L-k)!} \\ &\quad \int_{-\infty}^{\infty} F(\epsilon)^{k-1}(1-F(\epsilon))^{L-k} f(\epsilon)\epsilon^2 d\epsilon \end{aligned} \tag{3.18}$$

which, again, after some manipulation, can be written as:

$$\begin{aligned} E[\epsilon_{(k)}^2] &= \frac{L!}{(k-1)!(L-k)!} \sum_{i=0}^{L-k} \binom{L-k}{i} (-1)^i \\ &\quad \left(\frac{F(\epsilon)^{k+1}}{k+1}\epsilon^2 \Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \left(\frac{F(\epsilon)^{i+k}}{i+k} \right) 2\epsilon d\epsilon \right) \end{aligned} \tag{3.19}$$

To be able to calculate these, we need to be able to integrate $F(\epsilon)$, the cumulative distribution of the noise, which in our case can be uniform or Gaussian.

For the uniform case, if $\epsilon \in U[a,b]$, we can calculate both $E[\epsilon_{(k)}]$ and $E[\epsilon_{(k)}^2]$, since

Table 3.3. Estimates Chosen by Fusion Rules

Rule	Chosen Scores
Average	$\hat{P}_1 = p + \overline{\epsilon}$ $\hat{P}_2 = 1 - p - \overline{\epsilon}$
Minimum	$\hat{P}_1 = p + \epsilon_{(1)}$ $\hat{P}_2 = 1 - p - \epsilon_{(L)}$
Median	$\hat{P}_1 = p + \epsilon_{(\frac{L+1}{2})}$ $\hat{P}_2 = 1 - p - \epsilon_{(\frac{L+1}{2})}$
Maximum	$\hat{P}_1 = p + \epsilon_{(L)}$ $\hat{P}_2 = 1 - p - \epsilon_{(1)}$

we know $F(\epsilon)$.

$$F(\epsilon) = \left\{ \begin{array}{ll} 0 & \epsilon < a \\ \frac{\epsilon - a}{b - a} & a \leq \epsilon < b \\ 1 & \epsilon \geq b \end{array} \right\} \tag{3.20}$$

$$\begin{aligned} E[\epsilon_{(k)}] &= \frac{L!}{(k-1)!(L-k)!} \sum_{i=0}^{L-k} \binom{L-k}{i} (-1)^i \\ &\quad \left(\frac{b}{k+i} - \frac{b-a}{(k+i)(k+i+1)} \right) \end{aligned} \tag{3.21}$$

$$\begin{aligned} E[\epsilon_{(k)}^2] &= \frac{L!}{(k-1)!(L-k)!} \sum_{i=0}^{L-k} \binom{L-k}{i} (-1)^i \\ &\quad (\frac{b^2}{k+i} - \frac{2b(b-a)}{(k+i)(k+i+1)} + \\ &\quad \frac{2(b-a)^2}{(k+i)(k+i+1)(k+i+2)}) \end{aligned} \tag{3.22}$$

Similar calculation can not be performed analytically for $\epsilon \sim N(0, \sigma^2)$, and we use numerical integration to evaluate the integral of the cumulative distribution function, $\Phi(\epsilon)$.

3.4.2. 0/1 Loss

Now, we can do the same decomposition for 0/1 loss. First we take a quick glimpse on the rules to use in decomposition in Table 3.3.

Without any loss, we can generalize the order rules as below, for any k;

$$\begin{aligned} \hat{P}_1 &= p + \epsilon_{(k)} \\ \hat{P}_2 &= 1 - p - \epsilon_{(L-k+1)} \end{aligned} \tag{3.23}$$

It must be kept in mind that, for the minimum and maximum rules, the sum of estimated values for posteriors for the classes mostly will not sum up to 1. Taking these into account, we can rearrange Kohavi-Wolpert and Breiman definitions for two-class case for 0/1 loss.

3.4.2.1. Kohavi-Wolpert's Decomposition. For the average rule, we have;

$$\begin{aligned} \text{bias} &= \frac{1}{2}\left((p - p - \overline{\overline{\epsilon}})^2 + (1 - p - 1 + p + \overline{\overline{\epsilon}})^2\right) \\ &= \frac{1}{2}\left((-\overline{\overline{\epsilon}})^2 + (\overline{\overline{\epsilon}})^2\right) \\ \text{variance} &= \frac{1}{2}\left(1 - (p + \overline{\overline{\epsilon}})^2 - (1 - p - \overline{\overline{\epsilon}})^2\right) \end{aligned} \tag{3.24}$$

It must be noted that for the average rule, the bias is 0 and variance does not depend on ϵ since its average is 0, which is one of our assumptions.

For the order rules (minimum, median and maximum), we have;

$$\begin{aligned} \text{bias} &= \frac{1}{2}\left((p - p - \overline{\epsilon}_{(k)})^2 + (1 - p - 1 + p + \overline{\epsilon}_{(L-k+1)})^2\right) \\ &= \frac{1}{2}\left((-\overline{\epsilon}_{(k)})^2 + (\overline{\epsilon}_{(L-k+1)})^2\right) \\ \text{variance} &= \frac{1}{2}\left(1 - (p + \overline{\epsilon}_{(k)})^2 - (1 - p - \overline{\epsilon}_{(L-k+1)})^2\right) \end{aligned} \tag{3.25}$$

3.4.2.2. Breiman's Decomposition. For Breiman's decomposition, our ensemble will have bias only when the majority of the experiments misclassify the sample ($\alpha = 1, a = 2$). However we can have variance. When we have bias, then we do not have variance. For the average rule, we have;

$$\begin{aligned} \text{bias} &= (p-(1-p))(1-p-\overline{\epsilon}) \\ \text{variance} &= (p-(1-p))(1-p-\overline{\epsilon}) \end{aligned} \tag{3.26}$$

and for the order rules, we have;

$$\begin{aligned} \text{bias} &= (p-(1-p))(1-p-\overline{\epsilon}_{(L-k+1)}) \\ \text{variance} &= (p-(1-p))(1-p-\overline{\epsilon}_{(L-k+1)}) \end{aligned} \tag{3.27}$$

3.4.2.3. Domingos' Decomposition. There will not be any bias as long as the majority of the classifiers classify accurately. But the variance and error must be calculated. For the average rule, we have;

$$\begin{aligned} \text{variance} &= (1-p-\overline{\epsilon}) \\ \text{error} &= p(1-p-\overline{\epsilon}) + (1-p)(p+\overline{\epsilon}) \end{aligned} \tag{3.28}$$

For the order rules, we have;

$$\begin{aligned} \text{variance} &= (1-p-\overline{\epsilon}_{(L-k+1)}) \\ \text{error} &= p(1-p-\overline{\epsilon}_{(L-k+1)}) + (1-p)(p+\overline{\epsilon_{(k)}}) \end{aligned} \tag{3.29}$$

3.5. Experimental Results

The classification error of the order rules (maximum, median and minimum rules) and average rule can be investigated in terms of bias, variance and noise decomposition

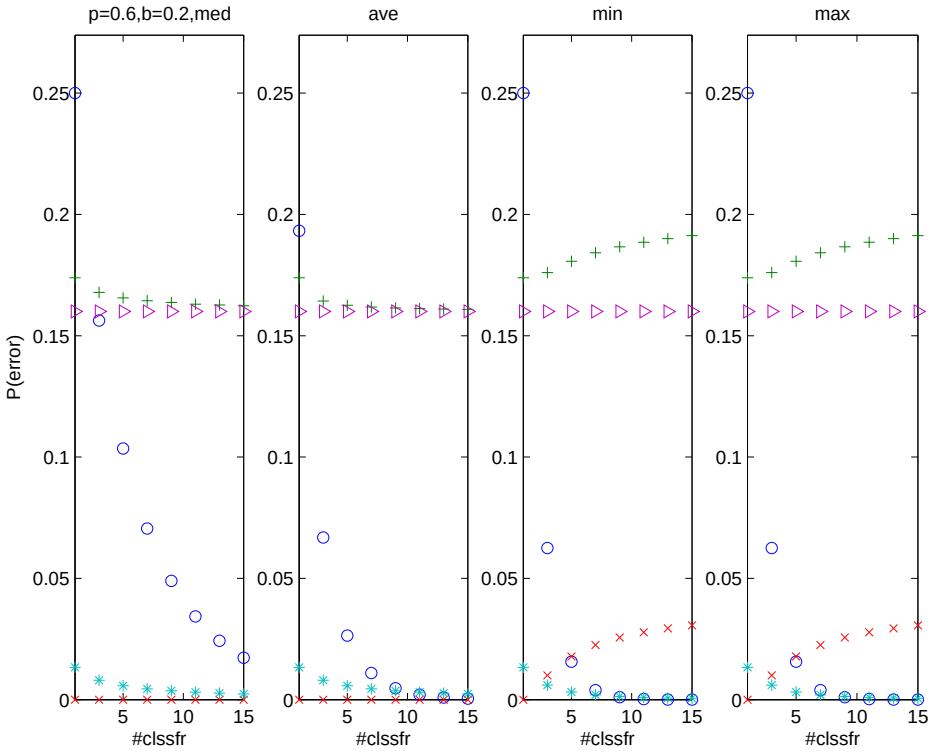
Table 3.4. Notation Used in the Loss Figures

Notation	Symbol	Expression
ave		Average Rule
min		Minimum Rule
med		Median Rule
max		Maximum Rule
B	$\times$	Theoretical Bias
V	$*$	Theoretical Variance
N	$\triangleright$	Noise
B+V+N	$+$	Total Loss
k	$\circ$	Kuncheva's Misclassification Error
CE	-	Empirical Misclassification Error

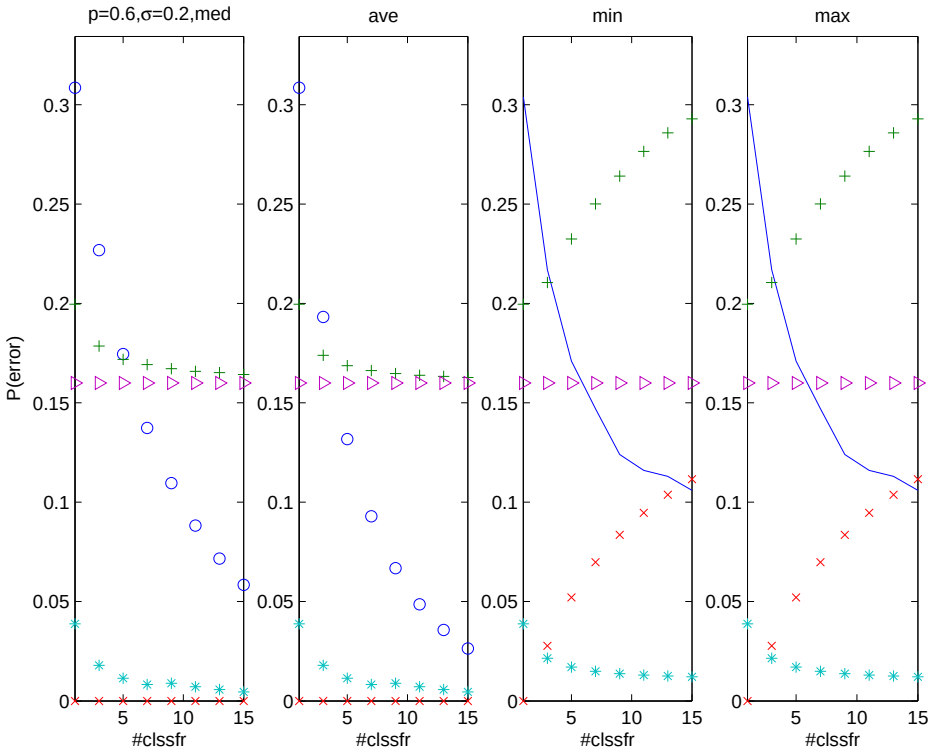
for both squared loss and 0/1 loss. We assume that for a two–class dataset, the true posterior probability (or score) of $p \geq 0.5$ for class ω_1, and we have L classifiers, $L = 3, 5, 7, ..., 15$. Each classifier returns an estimate of $\hat{p} = p + \epsilon$, where $\epsilon \sim U[-b, b]$, or $\epsilon \sim N(0, \sigma^2)$. We observe how misclassification error, bias and variance change if we change L, p, or the distribution spread parameters (σ and b). We evaluate the bias, variance and error of the added noise theoretically. When we empirically evaluate the bias, variance and noise, we see that they match the theoretical ones. We, therefore, only plot the theoretical results. For the case of normal noise, we numerically evaluate the variance and bias, since closed-form evaluation is not possible.

3.5.1. Squared Loss

We see the theoretical values for the misclassification error, bias, variance and noise, for uniform and Gaussian noise in Figure 3.1. Kuncheva's formulas hold. For both, increasing L decreases the variance but increases the bias with the minimum and maximum rules. The empirical results hold almost exactly for both error distributions. The calculated bias and empirical bias show a little difference for larger L in the normal



(a) Uniform



(b) Gaussian

Figure 3.1. Squared Loss, (a) Uniform with $b = 0.2$ and (b) Gaussian $\sigma = 0.2$, $p = 0.6$, L increases from 1 to 15.

distribution, possibly due to approximation by numerical integration.

In Figure 3.2, the impact of the spread of noise on misclassification error, bias and variance is exhibited for uniform and Gaussian error. The empirical and theoretical misclassification rate seem to match. The median and average rules do not have any bias as expected. But the minimum and maximum rules show higher biases with increasing spread.

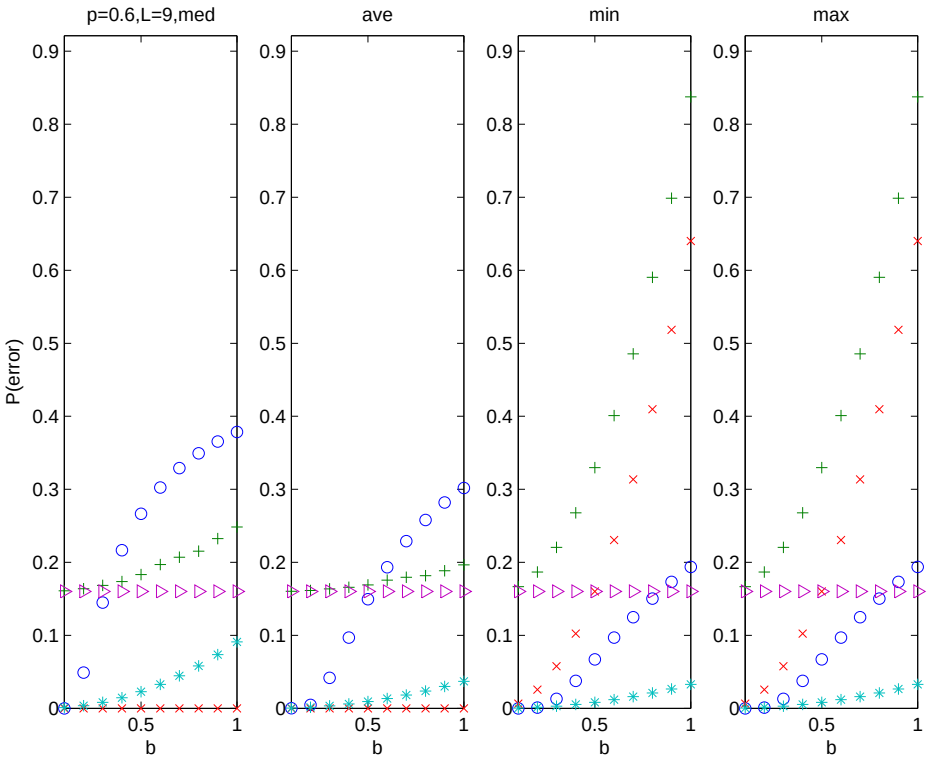
As Figure 3.3 demonstrates, increasing the base posterior value does not have any effect on bias and variance since they do not depend on posterior value but on the spread of the distribution. Inevitably, the misclassification rate drops down with increase in posterior value (as it moves away from the boundary of 0.5).

3.5.2. 0/1 Loss

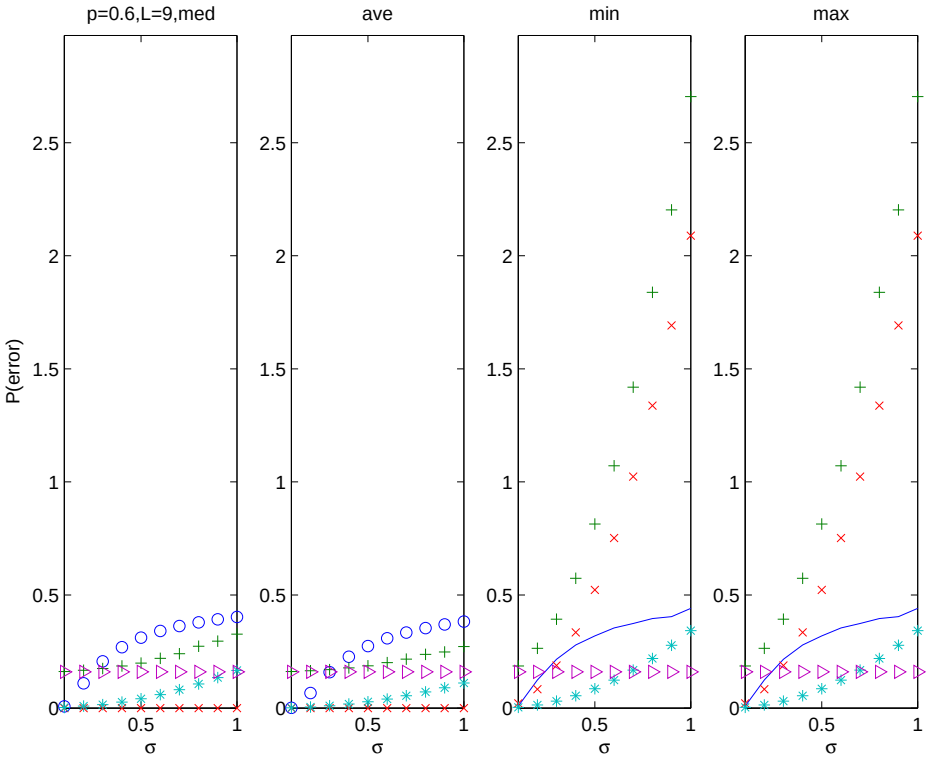
We can, in the same manner, investigate 0/1 loss bias-variance-noise decompositions for fusion rules. Here we consider only Kohavi-Wolpert's decomposition; similar analysis can also be done for the other two decompositions.

Figure 3.4 demonstrates that the average and median rules have 0 bias and variance components and the size of the ensemble do not have any effect on them. Misclassification rate drops down by the ensemble size as expected. With minimum and maximum rules, bias increases with the size of the ensemble but the variance changes in different directions. Although they both have the same misclassification rates, we can not propose the same behavior path for variance-bias decomposition.

Similarly, using median and average rules, the bias and variance are not influenced by the noise distribution interval, while misclassification increases with larger range, as shown in Figure 3.5. For the other two rules, bias and variance changes with increasing range. Bias increases in both of them. With the minimum rule, the variance performs a parabolic behavior while it continuously decreases with the maximum rule.

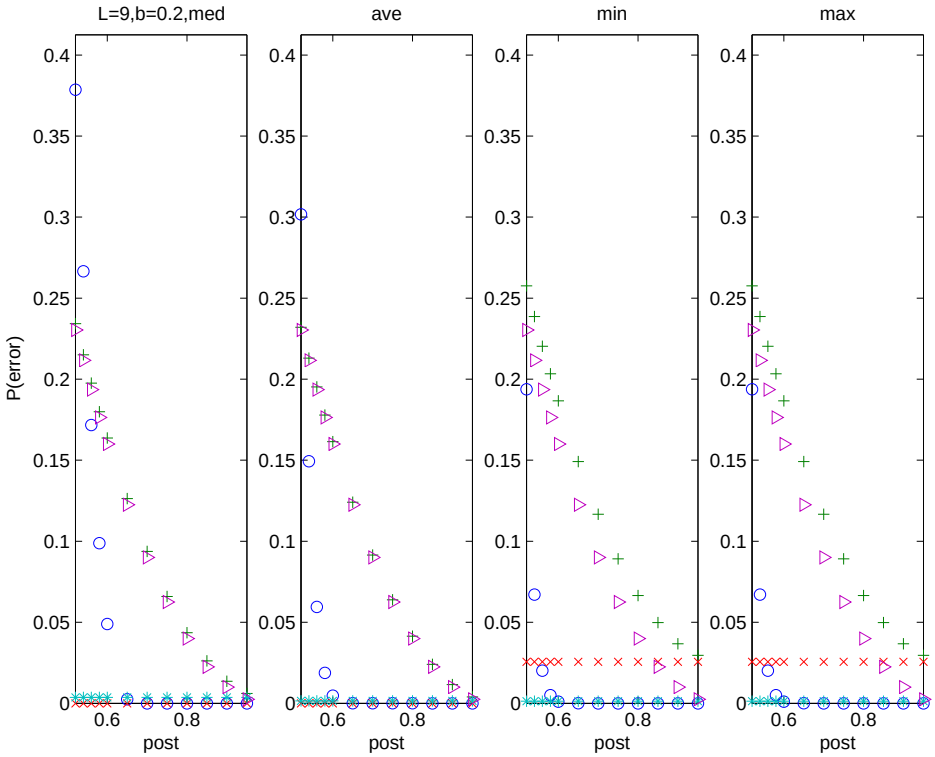


(a) Uniform

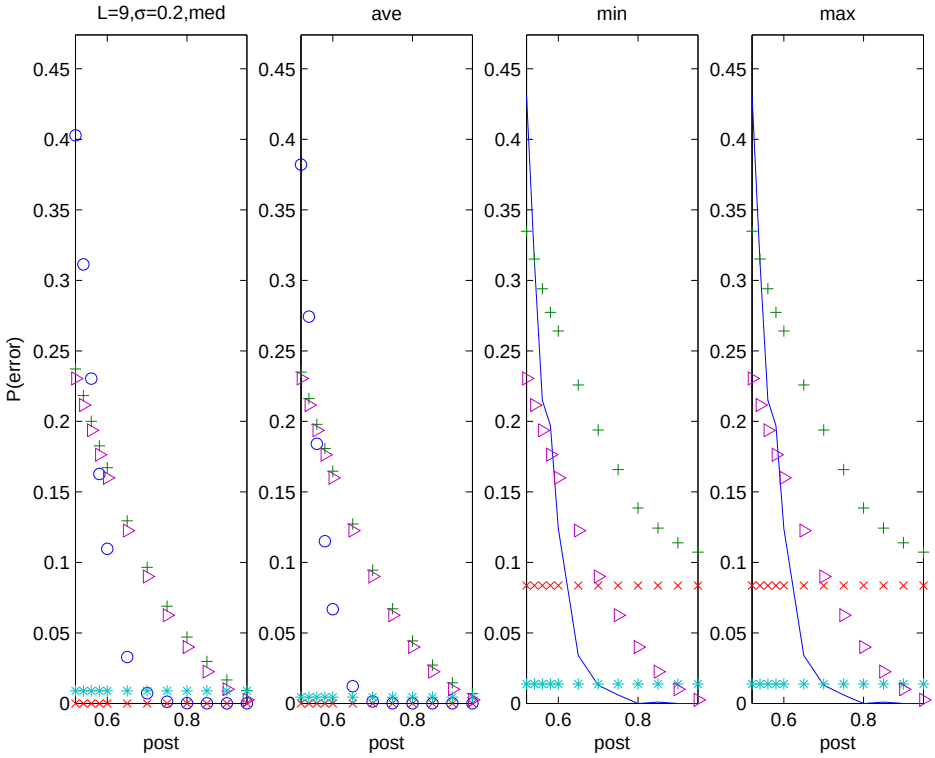


(b) Gaussian

Figure 3.2. Squared Loss $p = 0.6$ and $L = 9$, and (a) Uniform with $b = 0.1, \ldots, 1$ and (b) Gaussian with $\sigma = 0.1, \ldots, 1$.

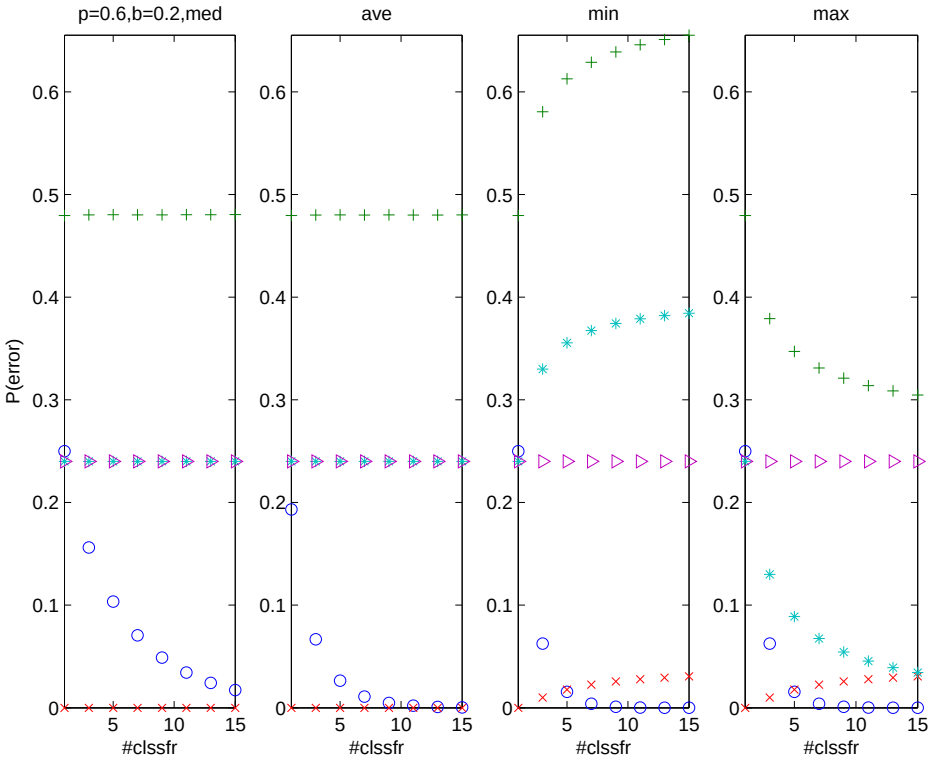


(a) Uniform

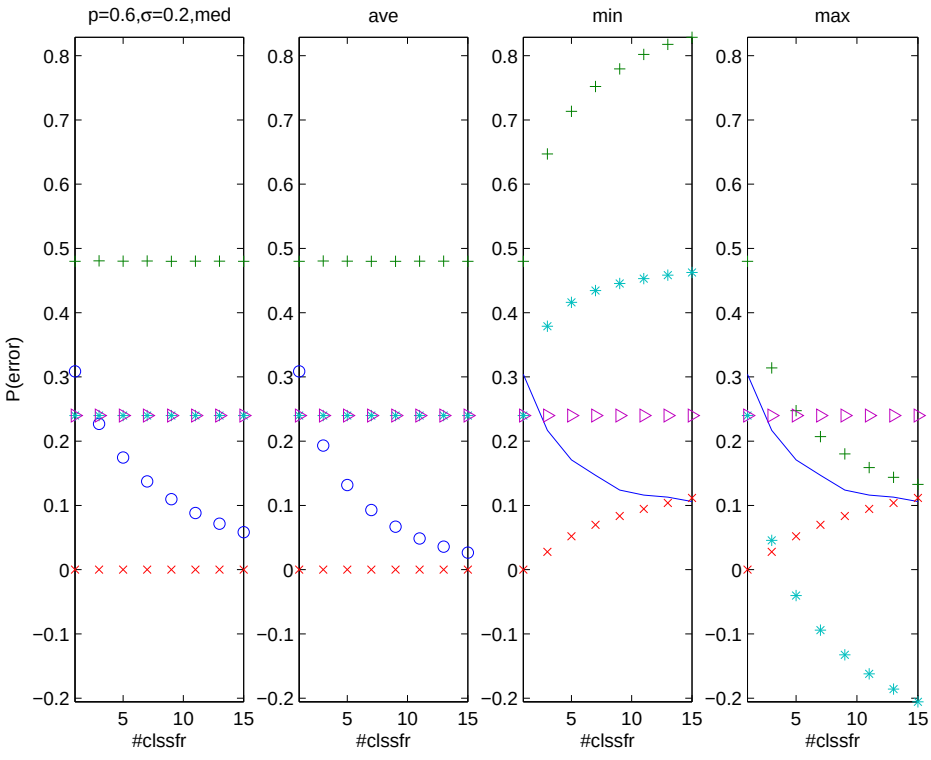


(b) Gaussian

Figure 3.3. Squared Loss, $L = 9$, (a) Uniform with $b = 0.2$ and (b) Gaussian with $\sigma = 0.2$, p increases from 0.5 to 1.

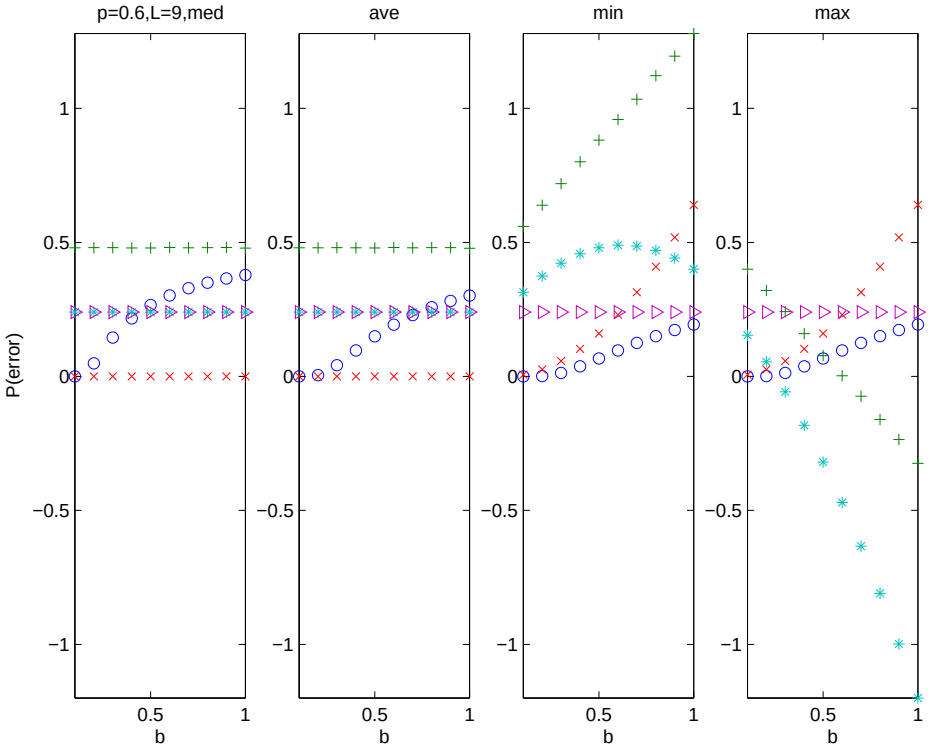


(a) Uniform

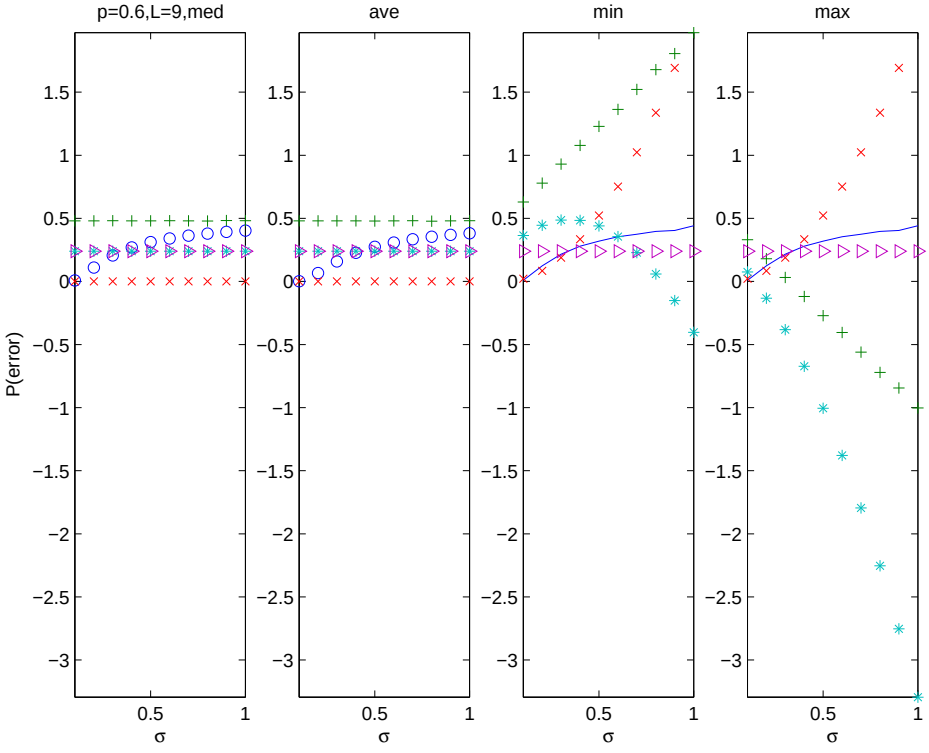


(b) Gaussian

Figure 3.4. 0/1 Loss, Kohavi-Wolpert, (a) Uniform with $b = 0.2$ and (b) Gaussian with $\sigma = 0.2$, $p = 0.6$, L increases from 1 to 15.



(a) Uniform



(b) Gaussian

Figure 3.5. 0/1 Loss, Kohavi-Wolpert, $p = 0.6$, $L = 9$, and (a) Uniform with $b = 0.1, \ldots, 1$ and (b) Gaussian with $\sigma = 0.1, \ldots, 1$

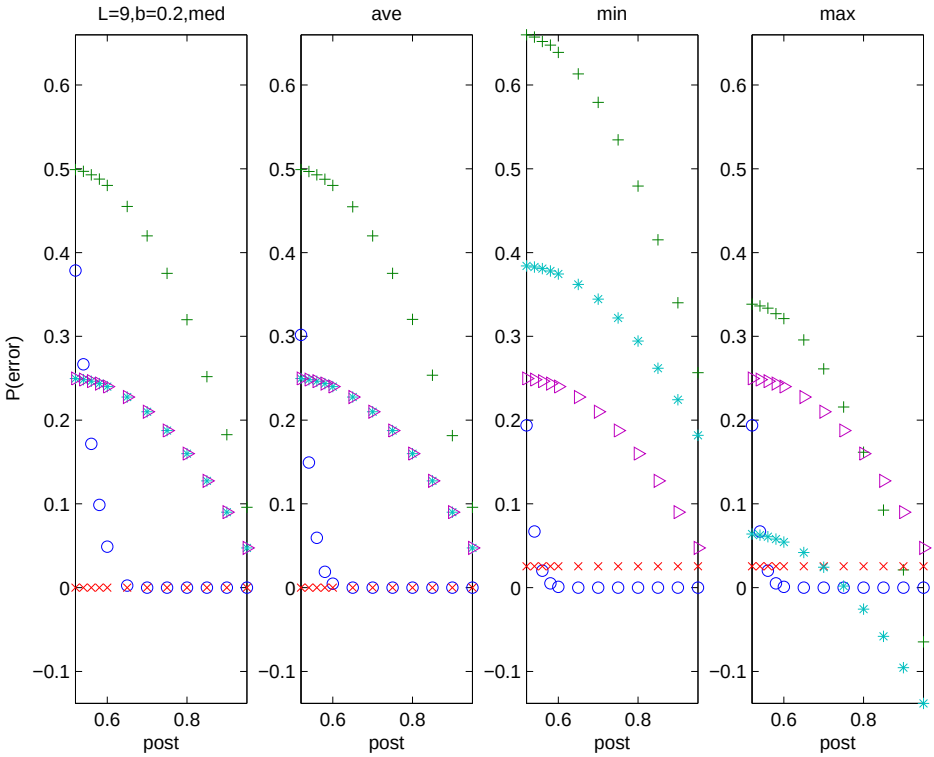
When we examine Figure 3.6, we notice that the bias is fixed and the posterior value does not have any influence with any fixed rules, but the variance decreases in all rules if we increase the base posterior.

When we observe the bias/variance decompositions for 0/1 loss, we conclude that there is no clear relation between them and misclassification error, and that the bias/variance decomposition does not help us in understanding the source of the error.

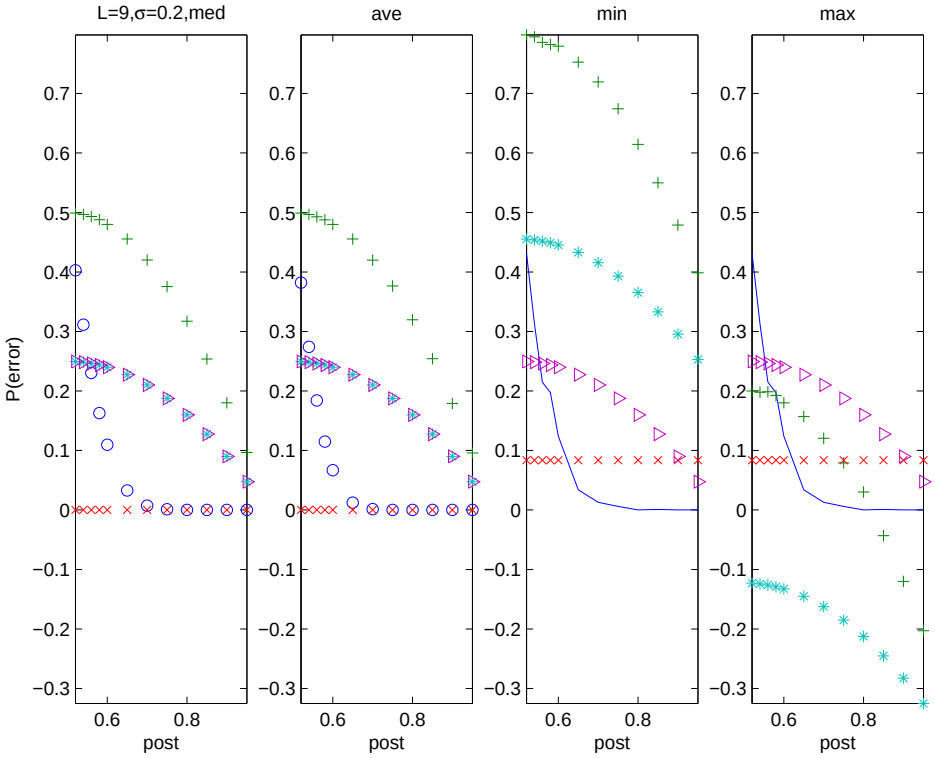
3.6. Intersection Area

In the previous section, we have seen that although conceptually bias-variance decompositions account for the cause of regression error, the proposed decomposition do not actually seem to be a good measure of relation between the classification error and 0/1 loss. They perform distinct patterns. For example, for the two-class cases, the minimum and maximum rules the give same misclassification error (they are same), but when we extend the decompositions for fusion rules, we see they perform different characteristics. The proposed decompositions are not adequate to explain the error of fusion rules.

Friedman (1997) explains that an additive decomposition of bias and variance on error is not possible for misclassification error; the two multiplicatively influence the error rate. He proposes the concept of "boundary bias," which corresponds to checking whether the bias component and true posterior are on the same side, and suggests is the determinant of the classification error rate. If they are on the same side, the variance inversely influences the classification accuracy. If the bias and true posterior are on different sides of 0.5, then variance helps reduce the error. This explanation holds exactly for the fusion rules for which $\hat{P}_1 + \hat{P}_2 = 1$, but this is not mostly true for minimum and maximum rules (which we will discuss next). It does not shed light on why they are so successful in the case of uniform error but not in the case of Gaussian error despite the fact that the bias is much higher. The median and average rules do not have any bias, but they have smaller variances, and therefore have less error.

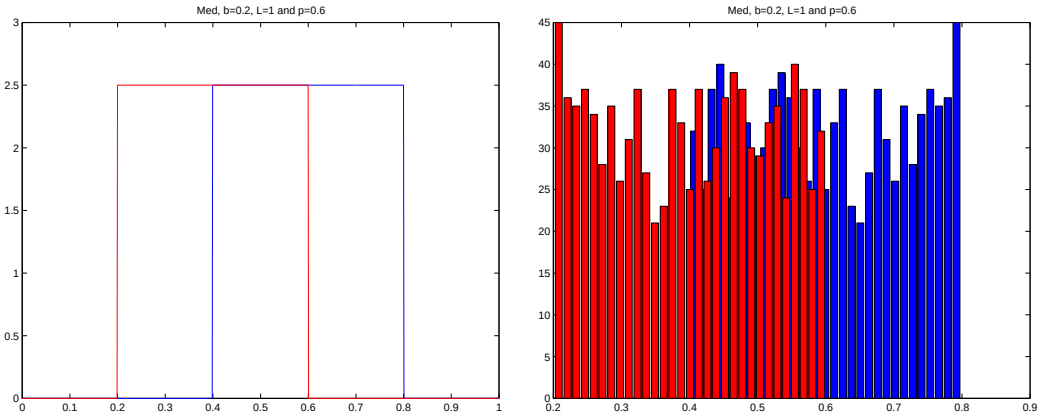


(a) Uniform



(b) Gaussian

Figure 3.6. 0/1 Loss, Kohavi-Wolpert, $L = 9$, (a) Uniform with $b = 0.2$ and (b) Gaussian with $\sigma = 0.2$, p increases from 0.5 to 1.



(a) Theoretical (b) Histogram

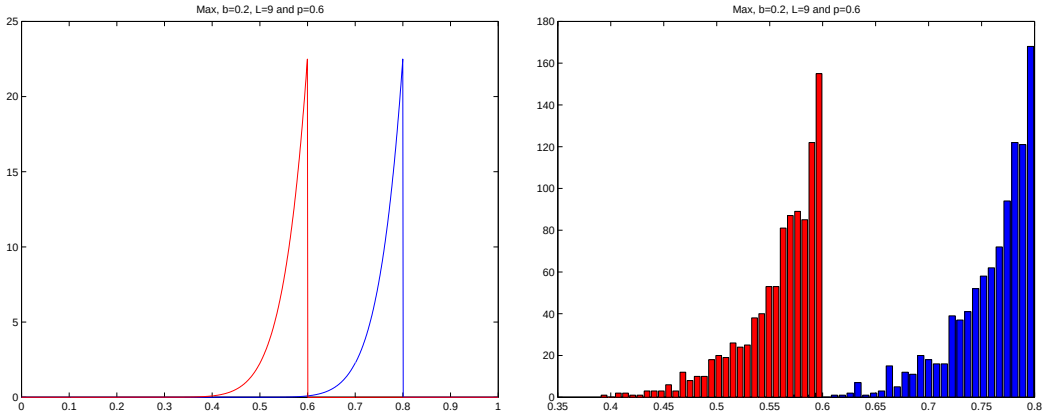
Figure 3.7. Uniform Distribution $b = 0.2$, $p_1 = 0.6$ (blue) and $p_2 = 0.4$ (red)

We propose to explain the success of fixed rules by means of a measure based on the intersection area between the distributions of chosen scores by the rules. It perfectly reflects the effects of bias and variances, and shows why and how they work depending on distribution the estimates show. The fusion scores show skewed distributions that depend on the size of the ensemble and the spread parameters. Half of the area of intersection between the scores of distributions of the two classes, is the misclassification error that occurs.

Figure 3.7 shows a case where posterior of the first class is 0.6 and its spread parameter, b, is 0.2. In such a case, we would expect a total misclassification rate of of 0.25, half of the area, which is the probability value Kuncheva calculates as error.

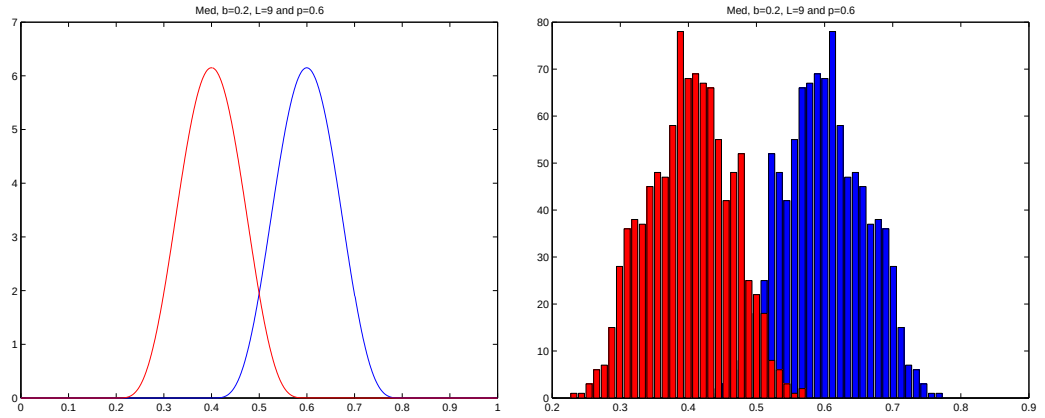
Figures 3.8 to 3.11 show the fixed rule score distributions when the ensemble is made of $L = 9$ classifiers. We see that as L increases, the concentration begins to shrink, whose shape and location of mass center depends on L, p and b. But for fixed posterior and spread values, as in the examples, while the ensemble size increases the variance of the score distribution (as a measure of how much the concantration shrinks), continuously decreases and is the main reason for the decrease of the misclassification error. The area of the intersection gets smaller as the classifiers in the ensemble increases.

As figures show, if we increase the value of the spread parameter, more misclas-



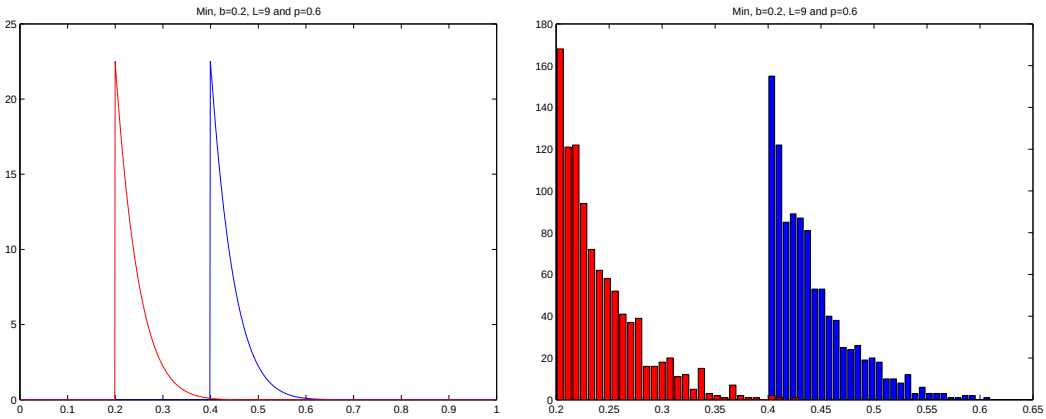
(a) Theoretical (b) Histogram

Figure 3.8. Uniform Distribution, Max Rule, $L = 9$, $b = 0.2$ and $p_1 = 0.6$ (blue)



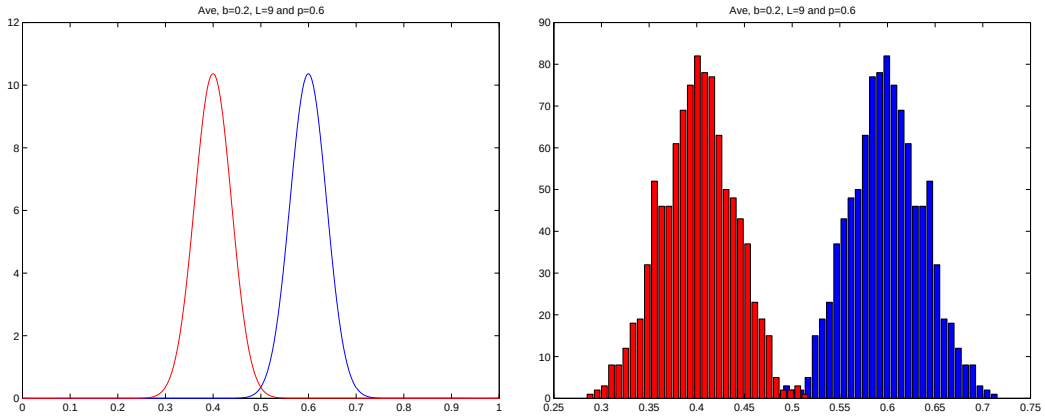
(a) Theoretical (b) Histogram

Figure 3.9. Uniform Distribution, Med Rule, $L = 9$, $b = 0.2$ and $p_1 = 0.6$ (blue)



(a) Theoretical (b) Histogram

Figure 3.10. Uniform Distribution, Min Rule, $L = 9$, $b = 0.2$ and $p_1 = 0.6$ (blue)



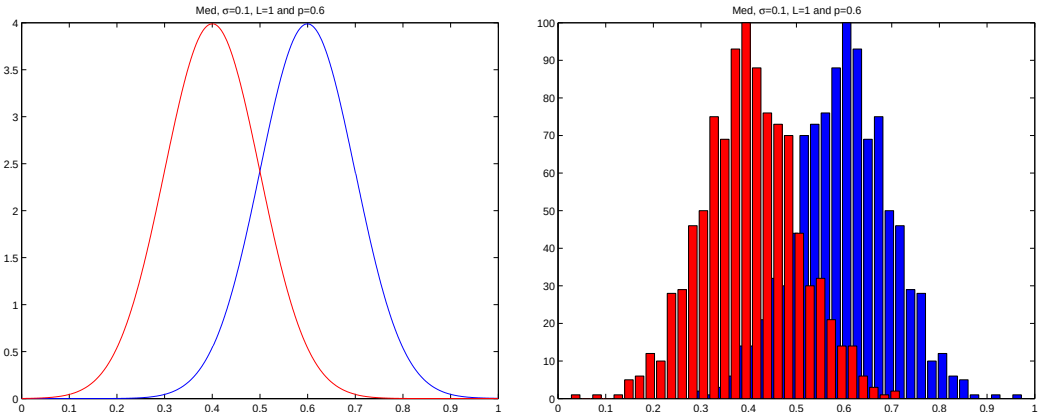
(a) Theoretical (b) Histogram

Figure 3.11. Uniform Distribution, Ave Rule, $L = 9$, $b = 0.2$ and $p_1 = 0.6$ (blue)

sification occurs since the area of intersection gets bigger (variance in terms of square error loss). The posterior value p is significant because the larger is the difference between p and $1 - p$, the less the areas overlap. Increasing the ensemble size contributes to the accuracy.

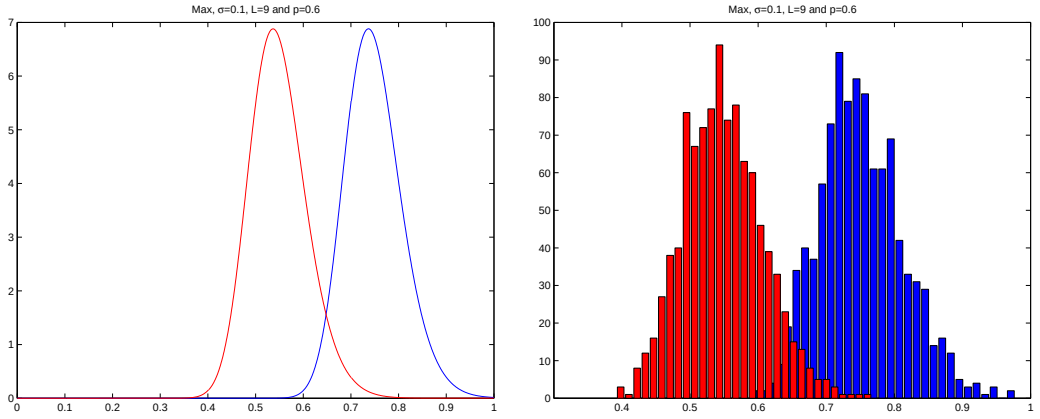
With the median and average rules, we see that the distributions are still symmetric with respect to their means. They do not move, but shrink around their means (variance decreases). On the contrary, when we use minimum or maximum rule, the center of mass moves to the left or right, depending on the rule. They show skewness. The figures also manifest that the minimum and maximum rules are more accurate (almost no area of intersection). Next best is the average rule, and the median rule performs poorly, since it has the biggest area of intersection.

Figures 3.12 to 3.16 show similar behavior for normal error. We see that the average performs the best because the final score distributions seem almost isolated from each other. The median rule seems to follow the average rule and both perform a symmetric distributions. Unlike the uniform distribution case, the minimum and maximum rules perform the worst. The reason is that for maximum rule, the distribution becomes positively skewed and for minimum rule, it becomes negatively skewed. Although the distributions shrink, they have longer tails and the tails constitute a considerable area of intersection and therefore error. After a certain size, adding more classifiers to the ensemble does not improve maximum or minimum rule considerably,



(a) Theoretical (b) Histogram

Figure 3.12. Gaussian Distribution $\sigma = 0.1$, $p_1 = 0.6$ (blue) and $p_2 = 0.4$ (red)



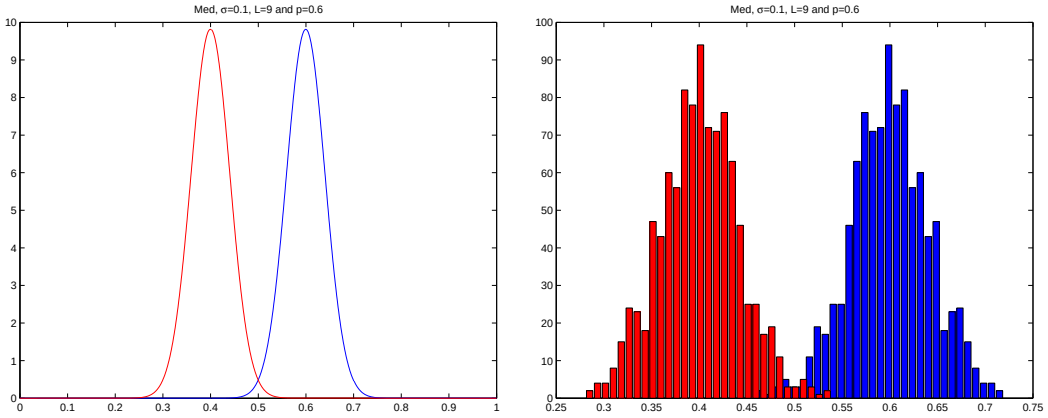
(a) Theoretical (b) Histogram

Figure 3.13. Gaussian Distribution, Max Rule, $L = 9$, $\sigma = 0.1$ and $p_1 = 0.6$ (blue)

because whereas the distributions shrink, the tails get longer.

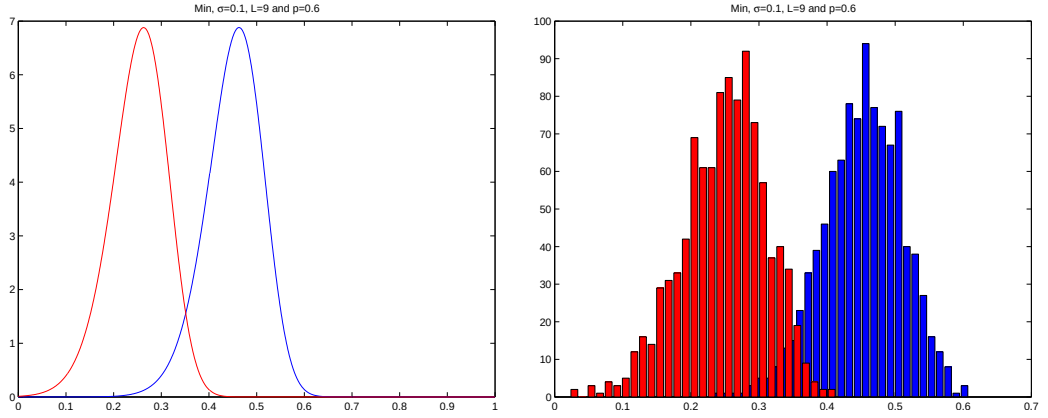
Let us see how the area of intersection, Kuncheva's misclassification error and the empirical misclassifications change when we only change one parameter at a time (posterior, spread parameter or ensemble size).

Figure 3.17 shows that adding a new classifier to the ensemble in uniform error, improves accuracy very quickly. But with Gaussian error, and maximum and minimum rules, after a certain size ($L = 7$ for our example), adding a new classifier does not contribute much to the accuracy, because the tails still have a considerable area of intersection. Here, we see that for normal case, the intersection area and empirical misclassification error do not seem to match perfectly. The most probable reason is



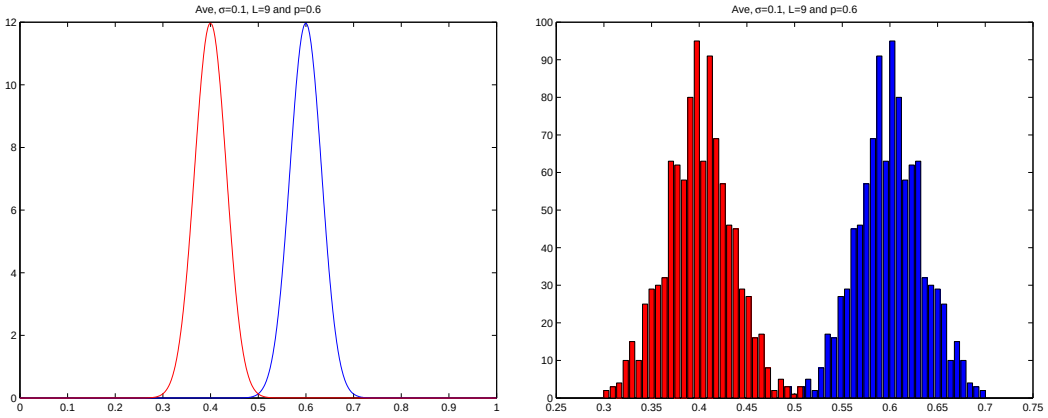
(a) Theoretical (b) Histogram

Figure 3.14. Gaussian Distribution, Med Rule, $L = 9$, $\sigma = 0.1$ and $p_1 = 0.6$ (blue)



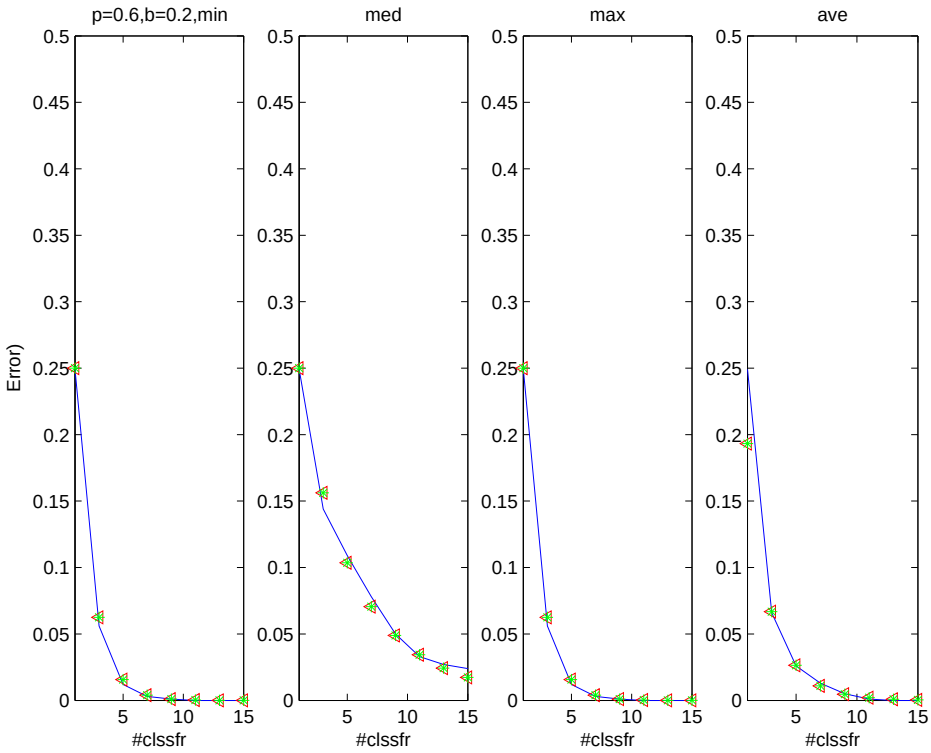
(a) Theoretical (b) Histogram

Figure 3.15. Gaussian Distribution, Min Rule, $L = 9$, $\sigma = 0.1$ and $p_1 = 0.6$ (blue)

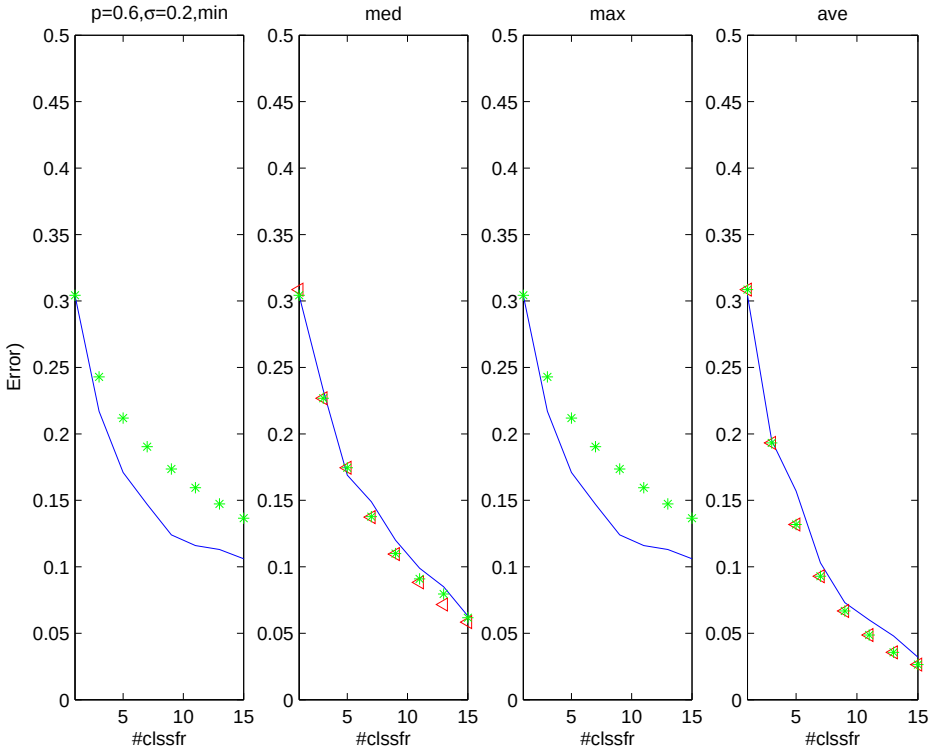


(a) Theoretical (b) Histogram

Figure 3.16. Gaussian Distribution, Ave Rule, $L = 9$, $\sigma = 0.1$ and $p_1 = 0.6$ (blue)



(a) Uniform Dist.



(b) Normal Dist.

Figure 3.17. The misclassification error as L changes, $p = 0.6$, $\sigma = 0.2$ and $b = 0.2$

Table 3.5. Notation Used in the Error Figures

Notation	Symbol	Expression
CE	-	Empirical Misclassification Error
area	$*$	0.5·Area of Intersection
k	$\triangleleft$	Kuncheva's Misclassification Error

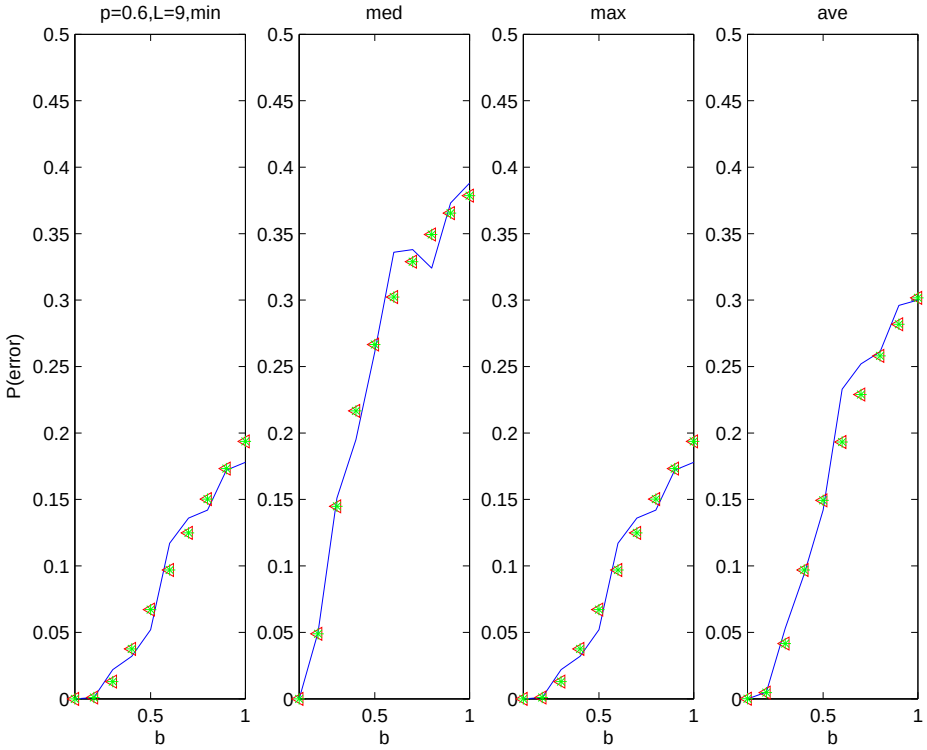
that for normal case we numerically evaluate the area of intersection.

When we increase the spread parameter (b or σ), we see that the misclassification error increases as expected. Also when we increase the true posterior, the misclassification error decreases drastically since the distance between the center of masses of the score distributions get further and the area of intersection gets smaller.

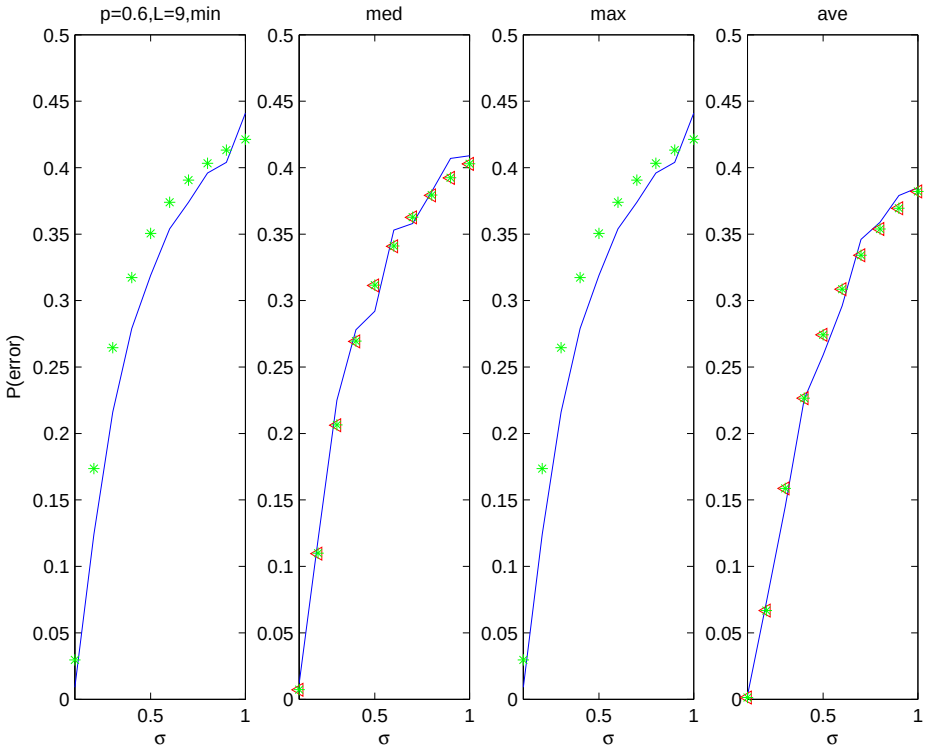
3.7. Conclusions

In this study, we have investigated how to decompose the bias and variance components of some fusion rules. We extend squared and 0/1 loss for the fusion rules. Neither squared nor 0/1 loss seem to explain the error behavior of fusion rules exactly. They do not show the pattern that describes the classification error of the fusion rules.

A convenient measure of the misclassification can be the area of intersection between the distributions of the class fusion scores. The variance and bias (center of mass) of the distribution accounts for the total area of the intersection (long variance means longer tails and bias, when in case asymmetric distributions, can cause larger intersection area by shifting the center of mass). We see that it explains the error of fusion rules.

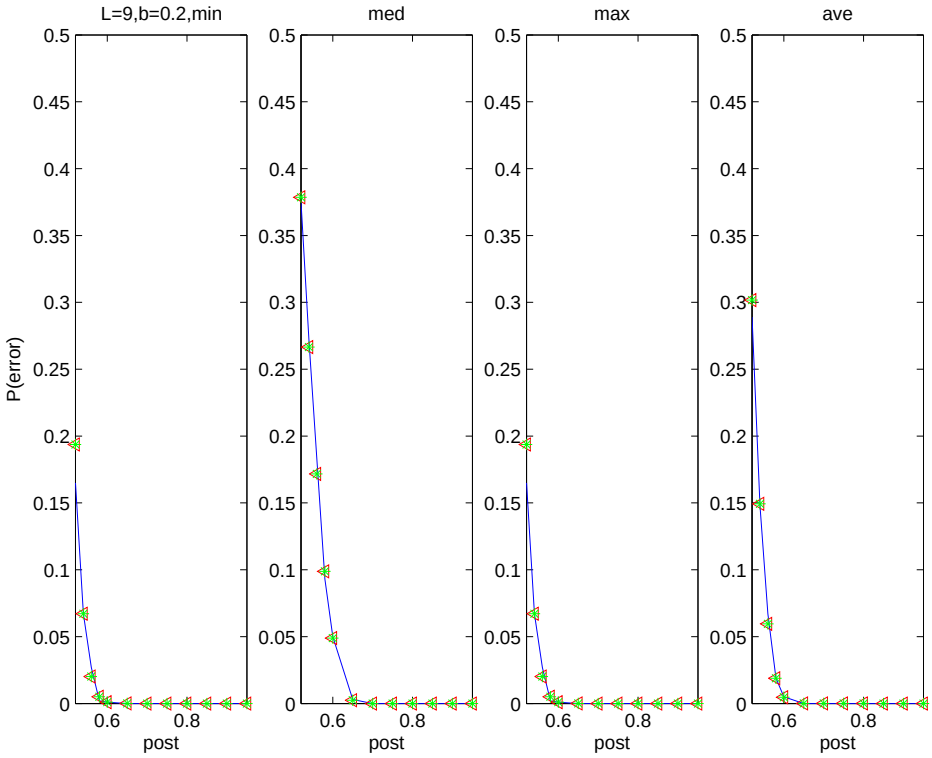


(a) Uniform Dist.

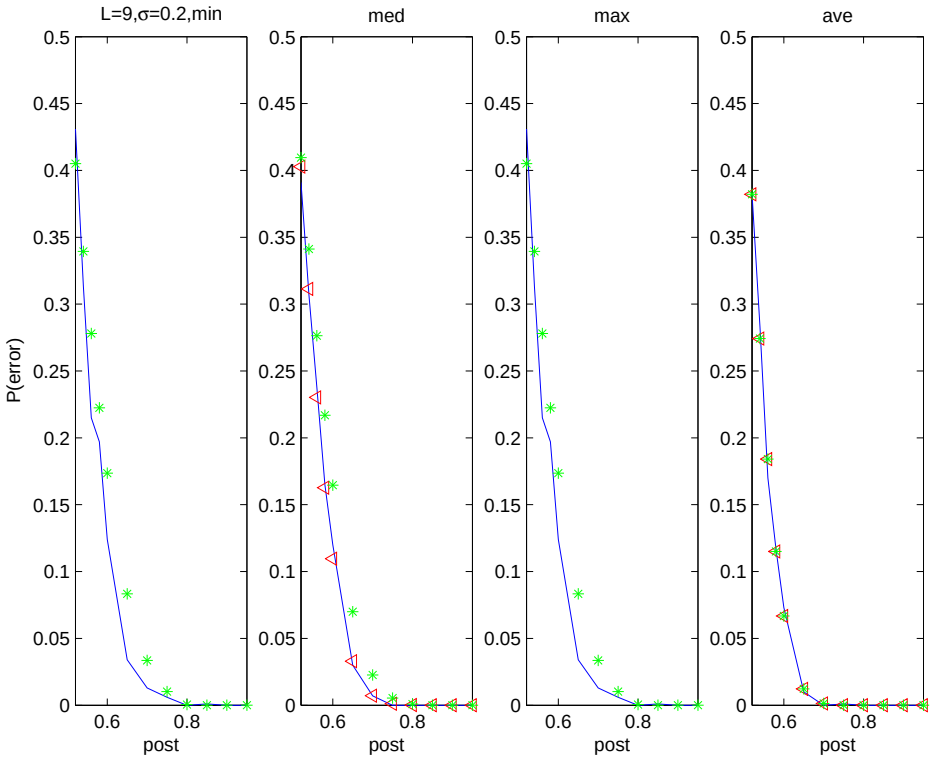


(b) Normal Dist.

Figure 3.18. The misclassification error as spread parameter(b or σ) changes, $p = 0.6$, $L = 9$



(a) Uniform Dist.



(b) Normal Dist.

Figure 3.19. The misclassification error as true posterior p changes, $L = 9$, $\sigma = 0.2$ and $b = 0.2$

4. CONCLUSIONS AND FUTURE WORK

This study consists of two main parts. In the first part, the idea of discriminant ensembles is proposed and investigated. Three different discriminant ensemble construction methods are proposed. Forward Subset Selection (Fss) is an incremental ensemble construction algorithm. It iteratively adds a discriminant to the ensemble to greedily improve accuracy. It stops when no further improvement is obtained. The other two are decision tree (Dt) and decision tree with a linear model (Dt.Lin). In the first one, the decision tree selects the discriminants and also is the combiner. The other one uses the discriminants chosen by the decision tree and a linear combiner is trained on these features.

There are certain differences between those construction methods. Fss takes longer time to train, while Dt can be trained in a shorter time. With respect to the ensemble constructed, Fss mostly results in smaller ensembles than Dt.

Compared with the classifier ensembles, discriminant ensembles are simpler and interpretable. They use just the needed discriminants to improve accuracy. They give us more information about the classifiers and the classes they are able to classify efficiently. But discriminant ensembles are less tolerant to noise since there is less redundancy.

The second part deals with the error analysis of fixed fusion rules. It extends bias-variance and noise decomposition of the squared and 0/1 loss for fusion rules. We conclude that these loss functions and their decompositions are unable to explain the decrease in error with those fusion rule, especially with minimum and maximum rules. They do not show a pattern which might help us understand the error behavior of such rules.

We propose the area of intersection and as we show, a new measure based on it matches with theoretical error values proposed by Kuncheva (2002).

As future work, for the discriminant ensembles, one can investigate other search methods, such as backward or floating subset selection, or genetic algorithms. Also cost–conscious discriminant ensembles, in terms of training time, memory requirement, vs., can be built. The influence of normalization or transformation of the base classifier scores on the ensemble is also an open subject. For the part on error analysis, the expression of suitable new loss functions that account for the behavior of fusion rules is also yet to be found.

APPENDIX A: Order Statistics

Let $x_1, x_2, ..., x_n$ be independent, identically, continuously and absolutely distributed random variables and $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$ be the corresponding order of these variables (David, 1970). Assume $f(x)$ denotes the probability density function (pdf) and $F(x)$ denotes the cumulative distribution function of x. Then, pdf of the k^{th} statistics can found as:

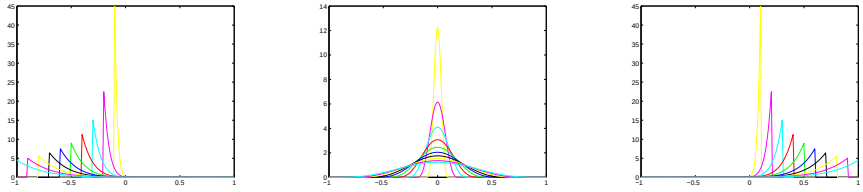
$$f_{x_{(k)}}(x) = \frac{n!}{(k-1)!(n-k)!} F(x)^{k-1}(1-F(x))^{n-k} f(x) \tag{A.1}$$

Using this, the pdfs of the smallest ($k = 1$), the middle ($k = (n+1)/2$), assume n is odd) and the biggest ($k = n$) members can be calculated:

$$\begin{aligned} f_{x_{(1)}}(x) &= n(1-F(x))^{n-1} f(x) \\ f_{x_{(\frac{n+1}{2})}}(x) &= \frac{n!}{(\frac{n-1}{2})!(\frac{n-1}{2})!} F(x)^{\frac{n-1}{2}}(1-F(x))^{\frac{n-1}{2}} f(x) \\ f_{x_{(n)}}(x) &= nF(x)^{n-1} f(x) \end{aligned} \tag{A.2}$$

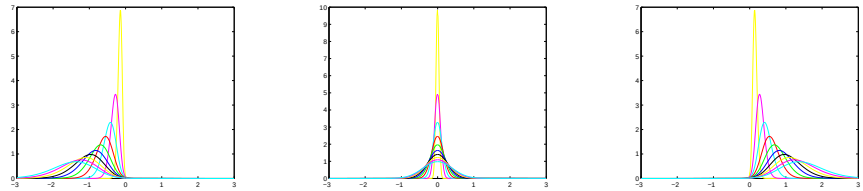
In this study, we are concerned with the cases where $f(x)$ is either uniform $U[-b, b]$, or Gaussian $N(0, \sigma^2)$. The k^{th} order pdf depends on k, n and the spread parameters, b for uniform distribution (Fig. A.1) and σ for Gaussian (Fig. A.2). We see that for symmetric distributions, 1^{st} (minimum) and n^{th} (maximum) order probability distribution functions are symmetric to each other with respect to the mean, which is 0 in our case. As long as $f(x)$ is symmetric, so is the pdf of the $(\frac{n+1}{2})^{th}$ order (median). We also see that as b or σ increases, the pdf spreads and cover a larger range. For the minimum, the mass is concentrated on the left (pdf is right-skewed) and for the maximum, the mass is concentrated on the right (it is left-skewed). The pdf of the median is symmetric around the mean.

We see in Figures A.3 and A.4 that as L increases, pdf shrinks. When L increases, in uniform distribution, for the minimum, the pdf shrinks and mass is concentrated



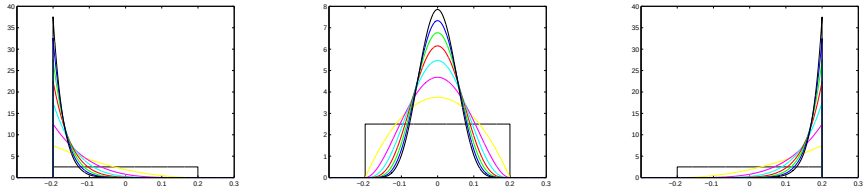
(a) Minimum (b) Median (c) Maximum

Figure A.1. For the uniform distribution, $f_{x_{(k)}}, k = 1, (n+1)/2, n$ when $n = 9$ and b increases from 0.1 to 1 by 0.1 (from the highest to the lowest peak)



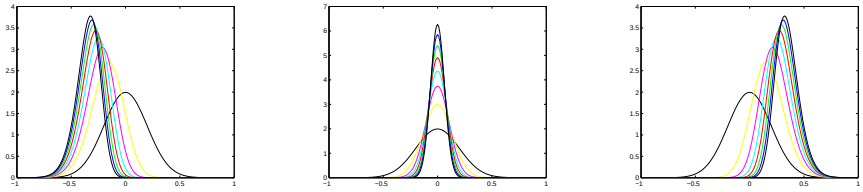
(a) Minimum (b) Median (c) Maximum

Figure A.2. For the Gaussian distribution, $f_{x_{(k)}}, k = 1, (n+1)/2, n$ when $n = 9$ and σ increases from 0.1 to 1 by 0.1 (from the highest to the lowest peak)



(a) Minimum (b) Median (c) Maximum

Figure A.3. For the uniform distribution, $f_{x_{(k)}}, k = 1, (n+1)/2, n$ when $b = 0.2$ and n increases from 1 to 15 by 2 (from the highest to the lowest peak)



(a) Minimum (b) Median (c) Maximum

Figure A.4. For the Gaussian distribution, $f_{x_{(k)}}, k = 1, (n+1)/2, n$ when $\sigma = 0.2$,and n increases from 1 to 15 by 2 (from the lowest to the highest peak)

on the left; the maximum similarly shrinks and concentrates on the right. The median shrinks and concentrates around the mean. Similar behavior is observed for the Gaussian distribution as well.

REFERENCES

Alkoot, F. M. and J. Kittler, 1999, "Experimental Evaluation of Expert Fusion Strategies", *Pattern Recognition Letters*, Vol. 20, No. 11-13, pp. 1361–1369.

Alpaydın, E., 1999, "Combined 5×2 cv F Test for Comparing Supervised Classification Learning Algorithms", *Neural Computation*, Vol. 11, No. 8, pp. 1885–1892.

Alpaydın, E., 2004, *Introduction to Machine Learning*, The MIT Press.

Breiman, L., 1996, "Bagging Predictors", *Machine Learning*, Vol. 24, No. 2, pp. 123–140.

Breiman, L., 1999, "Combining Predictors", *Combining Artificial Neural Nets: Emsemble and Modular Multi-Net Systems*, Edit. A. Sharkey, Springer-Verlag, pp. 31–50.

Chang, C.-C. and C.-J. Lin, *"LIBSVM A Library for Support Vector Machines"*, http://www.csie.ntu.edu.tw/∼cjlin/libsvm/, 2001.

Chen, D. and X. Cheng, 2001, "An Asymptotic Analysis of Some Expert Fusion Methods", *Pattern Recognition Letters*, Vol. 22, No. 8, pp. 901–904.

David, H. A., 1970, *Order Statistics*, John Wiley & Sons, Inc..

Demir, Ç. and E. Alpaydın, 2005,"Cost–Conscious Classifier Ensembles", *Pattern Recognition Letters*, Vol. 26, Issue 14, pp. 2206–2214.

Domingos, P., 2000, "A Unified Bias-Variance Decomposition for Zero-One and Squared Loss", *Proceedings of National Conference on Artificial Intelligence*, AAAI'00, pp .564–569.

Duda, R. O., P. E. Hart, and D. G. Stork, 2001, *Pattern Classification*, John Wiley & Sons, Inc..

Duin, R. P. W., 2002, "The Combining Classifier: To Train or Not To Train?", *Proceedings of International Conference on Pattern Recognition*, Vol. 2, pp. 765–770.

Friedman, J. H., 1997, "On Bias, Variance, 0/1–Loss, and the Curse-of-Dimensionality", *Data Mining and Knowledge Discovery*, Vol. 1, No. 1, pp. 55–77.

Freund, Y., and R. E. Schapire, 1996, "Experiments With a New Boosting Algorithm", *Proceedings of the International Conference on Machine Learning*, ICML'96, pp. 148–156.

Jacobs, R. A., M. I. Jordan, S. J. Nowlan, and G. E. Hinton, 1991, "Adaptive Mixtures of Local–Experts", *Neural Computation*, Vol. 3, pp. 79–87.

Jain, A., K. Nandakumar, and A. Ross, 2005, "Score Normalization in Multimodal Biometric Systems",*Pattern Recognition*, Vol. 38, Issue 12, pp. 2270–2285.

Kılıç, E., 2007, *Selecting From an Ensemble of Experts for Machine Learning*, M.S. Thesis, Boğaziçi University.

Kittler, J., M. Hatef, R. P. Duin, and J. Matas, 1998, " On Combining Classifiers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 226–239.

Kittler, J. and F. M. Alkoot , 2003, "Sum Versus Vote Fusion in Multiple Classifier Systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 1, pp. 110–115.

Kohavi R. and D. H. Wolpert, 1996, "Bias Plus Variance Decomposition for Zero-One Loss Functions", *Proceedings of the International Conference on Machine Learning*, ICML'96, pp. 275–283.

Kuncheva, L. I., 2002, "A Theoretical Study on Six Classifier Fusion Strategies", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 2, pp.281–286.

Kuncheva, L. I., 2004, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience.

Kuncheva, L. I., and C. J. Whitaker, 2003, "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy", *Machine Learning*, Vol. 51, No. 2, pp. 181–207.

Liu, C. L., 2005, "Classifier Combination Based on Confidence Transformation", *Pattern Recognition*, Vol. 38, Issue 1, pp. 11–28.

Mood A., F. Graybill, and D. Boes, 1974, *Introduction to the Theory of Statistics*, McGraw-Hill, Ed. 3.

Newman, D. J., S. Hettich, C. L. Blake, and C. J. Merz, *"UCI Repository of Machine Learning Databases"*, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

Rahman, A. F. R., and M. C. Fairhust, 2003, "Multiple Classifier Decision Combination Strategies for Character Recognition: A Review", *International Journal on Document Analysis and Recognition*, Vol. 5, pp. 166–194.

Rasmussen, C. E., R. M. Neal, G. E. Hinton, D. van Camp, M. Revow, Z. Ghahramani, R. Kustra, and R. Tibshirani, *"Delve Data for Evaluating Learning in Valid Experiments"*, http://www.cs.utoronto.ca/∼delve/, 1996.

Raudys, S., 2006a, "Trainable Fusion Rules. I. Large Sample Size Case", *Neural Networks*, Vol. 19, pp. 1506–1516.

Raudys, S., 2006b, "Trainable Fusion Rules. II. Small Sample–Size Effects", *Neural Networks*, Vol. 19, pp. 1517–1527.

Tumer, K. and J. Ghosh, 1999, "Linear and Order Statistics Combiners for Pattern Classification", *Combining Artificial Neural Nets: Emsemble and Modular Multi-Net Systems*, Edit. A. Sharkey, Springer-Verlag, pp. 127–162.

Ulaş, M. A., 2007, *Incremental Construction of Cost–Conscious Ensembles Using Multiple Learners and Representations in Machine Learning*, Ph.D. Thesis, Boğaziçi University.

Ulaş, M. A., M. Semerci, O. T. Yıldız, and E. Alpaydın, 2007, "Incremental Construction of Classifier and Discriminant Ensembles", *Submitted*.

Wolpert, D. H., 1992, "Stacked Generalization", *Neural Networks*, Vol. 5, pp. 241–259.

Yıldız, O. T. and E. Alpaydın, 2000, "Linear Discriminant Trees", *International Conference on Machine Learning*, Stanford University, USA.