UNIVERSITY OF CALIFORNIA
Los Angeles

Probability-Based Classifier Combination

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Fan Zhang

2017

ABSTRACT OF THE THESIS

Probability-Based Classifier Combination

by

Fan Zhang
Master of Science in Statistics
University of California, Los Angeles, 2017
Professor Qing Zhou, Chair

Classifier combination is an effective and popular method to improve the predictive performance of classification models. It has been employed in various fields, including pattern recognition and biometrics. This thesis proposes a novel classifier combination method based on the uniformness, a statistical measurement of the predicted probabilities of base classifiers. By choosing different measurement functions, three combination schemes are explored. The new method is designed to achieve improved accuracy and efficiency on the classification. It is tested on a real multi-class classification problem of plant species using leaf image features, which proves the advantage and robustness of this combination method.

The thesis of Fan Zhang is approved.

Arash Ali Amini

Yingnian Wu

Qing Zhou, Committee Chair

University of California, Los Angeles

2017

To my mother and father

who have always encouraged and supported me

to explore the unknown

TABLE OF CONTENTS

1 Introduction **1**

2 Background **4**

2.1 Dataset 4

2.2 Problem Statement 5

2.3 Individual Classifiers 6

3 Classifier Combination **9**

3.1 Combination Scheme 9

3.2 Probability-based Combination 10

3.3 Uniformness Measurement 11

4 Experiment Result **13**

4.1 Plant Leaf Classification 13

4.2 Modified Version of Classification Setting 15

5 Conclusion and Future Work **26**

References **28**

LIST OF FIGURES

4.1 Uniformness Measurement Histogram 192 Features 16

4.1 Uniformness Measurement Histogram 192 Features (Continued) 17

4.2 Feature Relative Importance 19

4.3 Uniformness Measurement Histogram 42 Features 20

4.3 Uniformness Measurement Histogram 42 Features (Continued) 21

4.3 Uniformness Measurement Histogram 42 Features (Continued 2) 22

LIST OF TABLES

4.1 Cross Validation of Base Classifiers with All 192 Features 14

4.2 Test Predictive Performance with All 192 Features 14

4.3 Cross Validation of Base Classifiers with 42 Features 23

4.4 Test Predictive Performance with 42 Features 24

4.5 Test Predictive Performance with 42 Features and Weaker Classifiers 25

CHAPTER 1

Introduction

The goal of training a classifier is to accurately predict the class label of novel input patterns, which means the classifier generalizes [2]. People usually train multiple classifiers for a real problem and select the model which generalizes best according to specific test data. However, this neglects the information provided by other classifiers at the risk of choosing a poor model for the beforehand unseen inputs, since each classifier has its own assumptions and different decision boundaries which may not be completely covered in the given test data. A set of different classifiers with a good performance on a validation/testing set may, however, have a different generalization behavior. Consequently, it is difficult to know which one is better for the real prediction situation. Instead of relying on a single model, combining classifiers enables one to utilize information from various models to improve the classification performance. Such combined classification model tends to be robust and often outperforms the individual classifiers which are combined as the base models. Classifier combination has been widely applied to the fields of optical character recognition and biometrics.

Instead of ensemble algorithms, this thesis focuses on combining heterogeneous classifiers. The advantage is that one can tackle the problem from various perspectives, rather than stick to a single approach. Ideally, the expertises of the specialized classifiers do not overlap. Complementary classifiers make it possible to model functions that a single algorithm alone cannot. For example, linear discriminant classifiers cannot model curves, but using various kernels in SVMs can model nonlinear boundaries that are closer to the optimal one. On the other hand, each classification model has its own assumptions which may not be perfectly consistent with the dataset. By combining a range of models, the bias resulting from a

specific unrealistic assumption can be diluted.

The output of a classifier usually is a vector of dimension C, where C is the number of classes. The classifier combination is to use such C-dim vectors from M individual classifiers to generate the final result in the form of a C-dim vector [1]. Therefore, the classifier combination scheme is *de facto* a secondary classification model, using the outputs of various classifiers as the input. Based on the elements of primary individual classifiers' output vectors, the classifier combinations are usually categorized into three types [1]:

1. Abstract Level: Each individual classifier only provides the most probable class label. The C-dim vectors are one-hot, with only a single value 1 for the corresponding predicted class, and 0 for the other classes.

2. Rank Level: Individual classifiers sort the class labels according to their probability values. The elements of the output vector is the relative rank of likeliness of each class. Other variant forms of output, such as n-best list, are also operated on this level.

3. Measurement Level: The elements of the C-dim vectors are the scores of each class. The probability of class labels can be a monotonic function of the score. This is the highest level of combination, since the base classifiers provide the most information about each class label. However, the base classifiers may have different interpretations of the scores, which require normalization or further transformation before combining.

Various methods have been developed on different levels of classifier combination. Majority voting, including weighted majority voting, can be applied on Abstract Level, since it requires no information about the confidence or probability of each class. Borda count, a variant of voting methods, is popular for Rank Level [6]. Various rules or pre-defined functions can be used on Measurement Level, including sum-rule, product-rule, and max-rule [4]. Such rules need no further training of the combination scheme. Some generic methods, such as neural network and logistic regression, can also combine on this level, i.e. training a real classifier on the outputs of primary base classifiers. This is a special application of

those well-developed machine learning models, where the input and the output have the same format.

This project will focus on the multi-class classifier combination on Measurement Level. Each classifier will output the predicted probability for all classes. The elements of each output vector have the same meaning and consequently are ready for combining. The following part of this thesis proposes a new design for the classifier combination rule, instead of merely applying a generic secondary classifier to the output vectors. A good combination scheme should generalize well without the heavy computation cost of training a generic classifier. The proposed combination schemes will be applied to a plant leaf classification dataset for the experiment, which is a multi-class classification problem.

CHAPTER 2

Background

The number of plant species is estimated to be nearly half a million. Plant species classification is extremely useful for botany research, food industry and pharmaceutical industry. Although DNA analysis technology can provide feasible and precise answers to species identification, people still want easier and cheaper methods based on other biological features. Taxonomy suggests that leaves can indicate the plant species. Since leaves are usually more available than other organs, such as flower, fruit and seed, it is useful to classify the species based on leaves. Charles Mallah, James Cope, James Orwell introduced a systematic method based on three categories of leaf features: margin, shape and texture [3]. They estimated three individual k-NN models for these types of features and used a linear combination as the final result, which generated a 96.81% classification accuracy. Inspired by their exploratory research, this project continues to analyze that leaf dataset using other statistical methods.

2.1 Dataset

The original leaf dataset from [3] is hosted by the Center for Machine Learning and Intelligent Systems at UCI [1]. This project uses a slightly modified version of the dataset [2], which contains 1584 leaf specimens (16 samples in each of 99 plant species). The training dataset contains 10 observations for each species, and the test set contains the remaining 6 observations in each class. This training/test split maintains equal sample size among species to avoid the negative effects of unbalanced classification.

[1] https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set

[2] One species is excluded. https://www.kaggle.com/c/leaf-classification

Leaf features extracted from grey-scale images have three main categories: margin, shape, and texture. Each category is associated with 64 attributes: $margin1$ to $margin64$, $shape1$ to $shape64$, and $texture1$ to $texture64$. Thus, each data point has 192 features, which are all continuous numeric variables. There exist no missing data in the training set. Both the training set and the test set are standardized before analysis.

2.2 Problem Statement

The task of this project is to predict the species label of each data point in the test set, in the form of a probability vector $P(X_i) = (p_{i1}, p_{i2}, \ldots, p_{ij}, \ldots, p_{iC})$ with probability p_{ij} that the i^{th} leaf belongs to the j^{th} species ($\sum_{j=1}^{C} p_{ij} = 1$, $C = 99$ is the number of species). The quality of classification is evaluated using logarithmic loss and accuracy, defined in Equation (2.1) and Equation (2.3), respectively.

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}) \tag{2.1}$$

$$y_{ij} = \mathbb{1}(\text{observation } i \text{ is from class } j) \tag{2.2}$$

$$accuracy = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\mathbb{1}(p_{ij} > p_{ik}, \forall k \neq j) \tag{2.3}$$

Here, N is the number of observations in the test set. $C = 99$ is the number of class labels. The indicator function y_{ij} is 1 if observation i is in class j and 0 otherwise. p_{ij} is the predicted probability that observation i belongs to class j. $logloss$ measures the negative average natural logarithmic value of the predicted probability for the true class label. The perfect prediction of assigning probability of 1 to the true species would give a zero $logloss$[3].

[3]In this case, other class labels contribute $0\log(0)$, which is set as 0 to avoid undefined undesirable effects.

A worse prediction would generate a larger *logloss*. *accuracy* is the percentage ratio of how often the true class label receives the largest predicted probability.

2.3 Individual Classifiers

This project uses several classifiers as the base models for combination, including Support Vector Machine (SVM), Random Forest, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Logistic Regression, and k-Nearest Neighbors (k-NN).

SVM constructs a hyper-plane in a high-dimensional space. Intuitively, a good separation is achieved by the hyper-plane that has the largest margin. SVM is effective in high dimensional spaces, where the number of dimensions is greater than the number of samples. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. And the model complexity depends on the number of support vectors. Different Kernel functions can be specified for the decision function. $\nu-$SVM [5], a variant of the original version of SVM, introduces a new parameter ν which controls the number of support vectors and training errors. The parameter $\nu \in (0, 1]$ is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors.

LDA is a method to find a linear combination of features that characterizes or separates two or more classes of objects or events. It assumes the conditional probability density function given the class label $P(X|y = k)$ as a multivariate Gaussian distribution and each class has the same vector μ and variance-covariance matrix Σ. A variant model, QDA does not have such strong assumptions on the variance-covariance matrices Σ_k of the Gaussian distribution, leading to quadratic decision surfaces. It estimates a different Σ_k for each class.

Logistic regression is used to predict the odds of being a case based on the values of the independent variables (predictors). The odds are defined as the probability that a particular outcome is a case divided by the probability that it is a noncase.

Random forest is an ensemble method for classification. Given a training dataset, it constructs multiple decision trees using bootstrapped sub-samples. Every decision tree is estimated using a random subset of the original predictors. A random forest classifier can control overfitting, compared to a single decision tree.

k-NN is the original model in the paper [3]. It is an instance-based learning algorithm using simple majority vote of the nearest neighbors of each data point. k-NN predicts the probability of each class using the normalized frequency of samples that have that class label within the neighbor.

Some classifiers are inherently designed as a binary model to distinguish two class labels. In order to solve the multi-class classification problem, there are two common strategies: One-vs-Rest and One-vs-One.

One-vs-Rest (also known as One-vs-All) trains a single classifier per class, with the samples of that class labelled as $+1$ and all the other samples as -1. This method has C binary classifiers in total and applies all the classifiers to a new data point. It predicts the class label for which the corresponding classifier reports the highest score. Here, the score can be the distance from the boundary hyperplane in SVMs or the regression score in Logistic Regressions.

One-vs-One trains a binary classifier for each pair of classes from the original set to distinguish these two classes. This method has $\frac{C(C-1)}{2}$ classifiers in total and applies all of them to a new data point. The class with the highest number of $+1$ predictions is the final output of the multi-class classifier.

In this project, SVM and $\nu-$SVM [5] use the One-vs-One method and Logistic Regression uses the One-vs-Rest method. The One-vs-One method guarantees the classifiers have

equal number of data points from the two classes each time. It's free from the negative effects of unbalanced classes. And each individual SVM has a small number of data points which makes computation much easier. The One-vs-Rest method helps to estimate the regression coefficients of all the features, since the number of features is much greater than the sample size of each class. It also reduces the total number of underlying binary models from $O(C^2)$ to $O(C)$.

CHAPTER 3

Classifier Combination

3.1 Combination Scheme

The most common combination models are usually constructed by simple average or weighted average. In the case of classification, the final output is the majority voting or weighted voting result (for Abstract Level). It can be expressed by Equation (3.1).

$$P(X_i) = \eta_i \sum_{m=1}^{M} weight_m \cdot P_m(X_i) \tag{3.1}$$

Here, the prediction $P(X_i)$ for a specific data point i with feature vector X_i is the weighted average of M classifiers' predictions: $P_m(X_i)$, with the model weight of $weight_m$. And η_i is the normalizing constant. The cross-validation prediction accuracy of each individual classifier can be used as the averaging weight $weight_m$. For this leaf classification problem, the prediction output of Equation (3.1) is a multinomial distribution probability vector $P(X_i) = (p_{i1}, p_{i2}, \ldots, p_{ij}, \ldots, p_{iC})$, $\sum_{j=1}^{C} p_{ij} = 1$ with probability p_{ij} that this leaf i belongs to the j^{th} species.

This combination scheme is straightforward and easy to compute. However, it is a model-level combination using a set of fixed parameters for all data points, which can be too rigid. For instance, a subset of data points with different class labels may be linearly separable, while another subset of classes may have non-linear boundaries. Therefore, LDA should be assigned a higher averaging weight in the previous case.

3.2 Probability-based Combination

For each data point, classifiers may have very different predictive output with different probability distributions on the set of class labels. Consider the two extreme cases which might appear in the predictions. The best case has all the probability mass on one class and zero probability mass on the rest: $p_{ij} = 1, p_{ik} = 0, \forall k \neq j$ by an individual model. It suggests that the classifier is completely confident that the observation belongs to a specific class, which is the most valuable information for the multi-class classification. Thus, this prediction should be emphasized. However, the worst case is a uniform distribution with equal probability mass on each species: $p_{ij} = \frac{1}{C}, \forall j$, which means the classifier cannot distinguish the class label given the input features. It is completely uncertain of the ground truth. Outputs of this type provide no useful information for the final decision and should be penalized by a smaller averaging weight or even be excluded from combination.

Therefore, the uniformness of the output probability vector reflects the value of a classifier's prediction for a data point. A less uniform prediction provides more valuable information and should receive a higher weight when combined to generate the final output. The classifier combination scheme is further developed to incorporate the uniformness of probability distribution, which can be expressed by Equation (3.2).

$$P(X_i) = \eta_i \sum_{m=1}^{M} P_m(X_i) \cdot f(P_m(X_i)) \cdot accuracy_m \tag{3.2}$$

Here, $f(\cdot)$ is the uniformness measurement and $accuracy_m$ is the cross-validation accuracy of the m^{th} individual model. A high $f(\cdot)$ value corresponds to low uniformness. η_i is the normalizing constant to maintain the sum of $P(X_i)$ elements equal to 1. The details of uniformness measurement are discussed in the following section.

This new combination scheme is performed on the individual data point level, instead of the overall model level. The combination weights of classifiers are calculated according to the classifier's overall accuracy and also the predictive probability of each model for the specific input data point. This is dynamically calculated after the leaf feature input is given,

instead of using only the fixed parameters of classifier accuracies. It utilizes the properties of input data to improve the predictive performance. Compared with other existing generic combination schemes, including Logistic Regression or Neural Network on the outputs of base classifiers, this new scheme does not require further training, to achieve the balance between computation efficiency and accuracy.

3.3 Uniformness Measurement

Three types of probability vector uniformness measure scores, based on entropy, Gini index and the coefficient of variation respectively, are explored in this novel and flexible combination scheme. Entropy is defined in Equation (3.3).

$$\text{Entropy}(P_m(X_i)) = -\sum_{j=1}^{C} p_{mij} \cdot \log p_{mij} \tag{3.3}$$

Here, $P_m(X_i) = (p_{mij} : j = 1, \ldots, C)$ is the predicted probability vector of the m^{th} classifier for the i^{th} data point, and p_{mij} is the predicted probability that the i^{th} data point belongs to the j^{th} class label by the m^{th} base classifier. A larger value of entropy reflects the distribution is more uniform. The extreme case of $p_{mij} = 1, p_{mik} = 0, \forall k \neq j$, has entropy=0 [1]. Therefore, the Entropy-uniformness measurement function is defined by Equation (3.4)

$$f(P_m(X_i)) = \frac{1}{a + \text{Entropy}(P_m(X_i))} \tag{3.4}$$

Here, a is a pre-specified positive constant.

Gini method for this multi-class classification uses Relative Mean Absolute Difference (RMAD) defined in Equation (3.5), which is equal to twice the Gini coefficient defined in terms of the Lorenz curve. It is directly used as the second uniformness measurement function $f(\cdot)$.

$$G(P_m(X_i)) = \frac{\sum_{s=1}^{C}\sum_{t=1}^{C} |p_{mis} - p_{mit}|}{2C\sum_{j=1}^{C} p_{mij}} = \frac{\sum_{s=1}^{C}\sum_{t=1}^{C} |p_{mis} - p_{mit}|}{2C} \tag{3.5}$$

[1] Define $0 \log(0) = 0$.

The coefficient of variation defined in Equation (3.6) is directly used as the third uniformness measurement function.

$$\text{Coefficient of Variation}(P_m(X_i)) = \frac{\text{Standard Deviation}(\mathrm{P_m(X_i)})}{\text{Mean}(\mathrm{P_m(X_i)})} \tag{3.6}$$

Since $P_m(X_i)$ represents the probability mass, with the element sum of 1. The denominator of the coefficient of variation is always $\frac{1}{C}$. Thus, Equation (3.6) can be simplified to be Equation (3.7).

$$\text{Coefficient of Variation}(P_m(X_i)) = C \cdot \text{Standard Deviation}(P_m(X_i)) \tag{3.7}$$

Both the methods based on Gini and the coefficient of variation assign zero weight to a uniform probability vector. Thus, the complete uncertainty of such cases would be excluded from the final result.

CHAPTER 4

Experiment Result

4.1 Plant Leaf Classification

The base classifiers are estimated individually. The tuning of model parameters is based on a grid search in the parameter grids. This process utilizes a 10-fold stratified cross validation on the training set. Since the dataset contains an equal number of leaves for each class, each fold is selected randomly with the equal number of data points in each class. This guarantees a balanced sampling scheme. After the best model parameters have been selected, each classifier is trained using a 10-fold stratified cross validation to be evaluated based on *accuracy* and *logloss*. The average cross validation accuracy is used as the overall accuracy parameter in the combination scheme of Equation (3.2) for each individual classifier. The constant a in Equation (3.4) is set as 10^{-6}, to avoid the undesired effects of zero entropy in the denomenator.

As listed in Table 4.1, all these models achieve high predictive accuracy in the cross validation, except for QDA, which also has the worst *logloss*. QDA assumes the conditional probability density function given the class label $P(X|y = k)$ as a multivariate Gaussian distribution and each class has its own mean vector μ and variance-covariance matrix Σ. However, LDA assumes an equal Σ for all classes. Recall that the training set has only 10 data points for each class, which makes it very difficult to the estimate a different Σ for each class. QDA is very unstable and has poor predictive performance on this dataset. Consequently, it is excluded in the combination. All three combination schemes are applied to the test set. The covered base classifiers are also individually tested using the same dataset. The

Classifier	**CV Accuracy**	**CV logloss**
k-NN(3)	0.968687	0.383901
LDA	0.985354	1.292611
Logistic Regression	0.988889	0.109696
QDA	0.027273	33.596810
Random Forest	0.977273	0.752276
SVM	0.988889	2.328283
$\nu-$SVM	0.990404	2.326457

Table 4.1: Cross Validation of Base Classifiers with All 192 Features

results of the simple average method and the weighted average method of Equation (3.1) are introduced as the benchmarks. The test *accuracy* and *logloss* are listed in Table 4.2.

Model	**Accuracy**	**Logloss**
Combination-Entropy	0.989899	0.10732
Combination-Gini	0.991582	0.70907
Combination-Variation	0.986532	2.11106
Simple Average	0.989899	0.87522
Weighted Average	0.989899	0.87609
k-NN(3)	0.984848	0.10093
LDA	0.983165	1.36711
Logistic Regression	0.993266	0.10027
Random Forest	0.981481	0.66238
SVM	0.991582	2.08398
$\nu-$SVM	0.991582	2.10675

Table 4.2: Test Predictive Performance with All 192 Features

The individual classifiers have similar test *accuracy* and *logloss*, compared with the training cross validation performance. Therefore, overfitting is not a concern here. In terms of *accuracy*, the three combination schemes are close to the individual classifiers: better than the worst classifier and worse than the best classifier. This result is reasonable, since the schemes combine information from multiple classifiers, instead of trusting only the best classifier of the test set, which is unknown beforehand. The combination schemes avoid the risk of selecting a single bad classifier, at the cost of the performance of the unknown best classifier being diluted. The probability-based combination schemes' performance is also close to the benchmarks of the simple average and the weighted average. As for *logloss*, the Entropy scheme and the Gini scheme are between the best individual classifier and the worst individual classifier, and they are better than the simple average as well as the weighted average. However, the Variation Coefficient scheme is worse than any base classifier and the two benchmarks.

Figure 4.1 shows the histograms of base classifiers' three uniformness measures for all data points in the test set. The classifiers have different behaviors in predicting the probabilities of class labels, especially in terms of the variation coefficient. SVM and $\nu-$SVM both have higher values in Entropy and lower values in Gini than other classifiers, which is consistent with their high *logloss*. Though SVM and $\nu-$SVM can correctly predict the class label in most cases by assigning the largest probability, the difference between the most probable one and the remaining classes is smaller than other classifiers. Therefore, these two models are not that confident of the predictive results. It partially explains the discrepancy between *accuracy* and *logloss* performances.

4.2 Modified Version of Classification Setting

The base classifiers have satisfactory individual classification performance and the similar predictive *accuracy* performance, within the range from 0.981481 to 0.983165. The dataset with all 192 features, therefore, is easy for the plant species identification problem. The

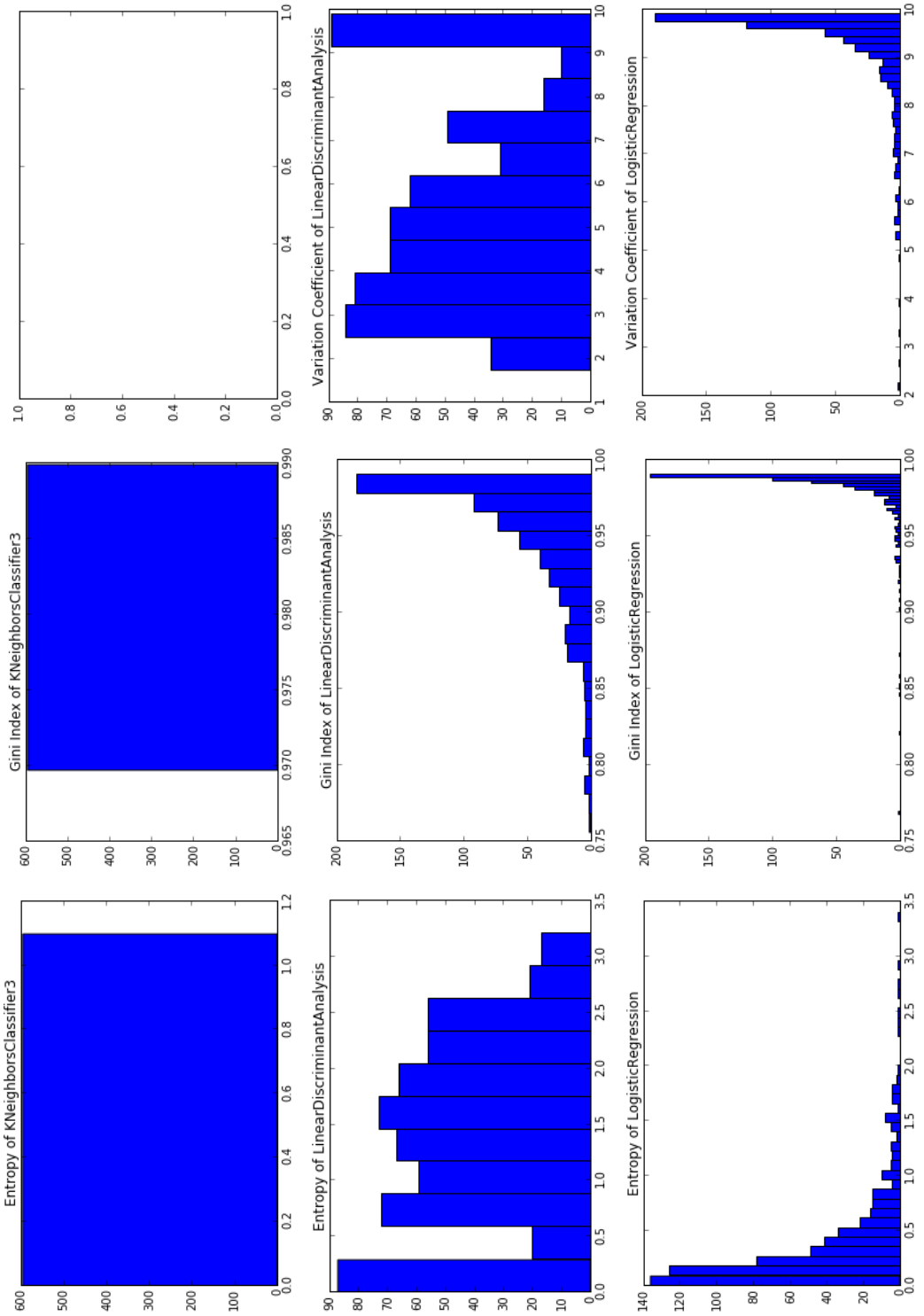

Figure 4.1: Histograms of the uniformness measurement functions with all 192 features.

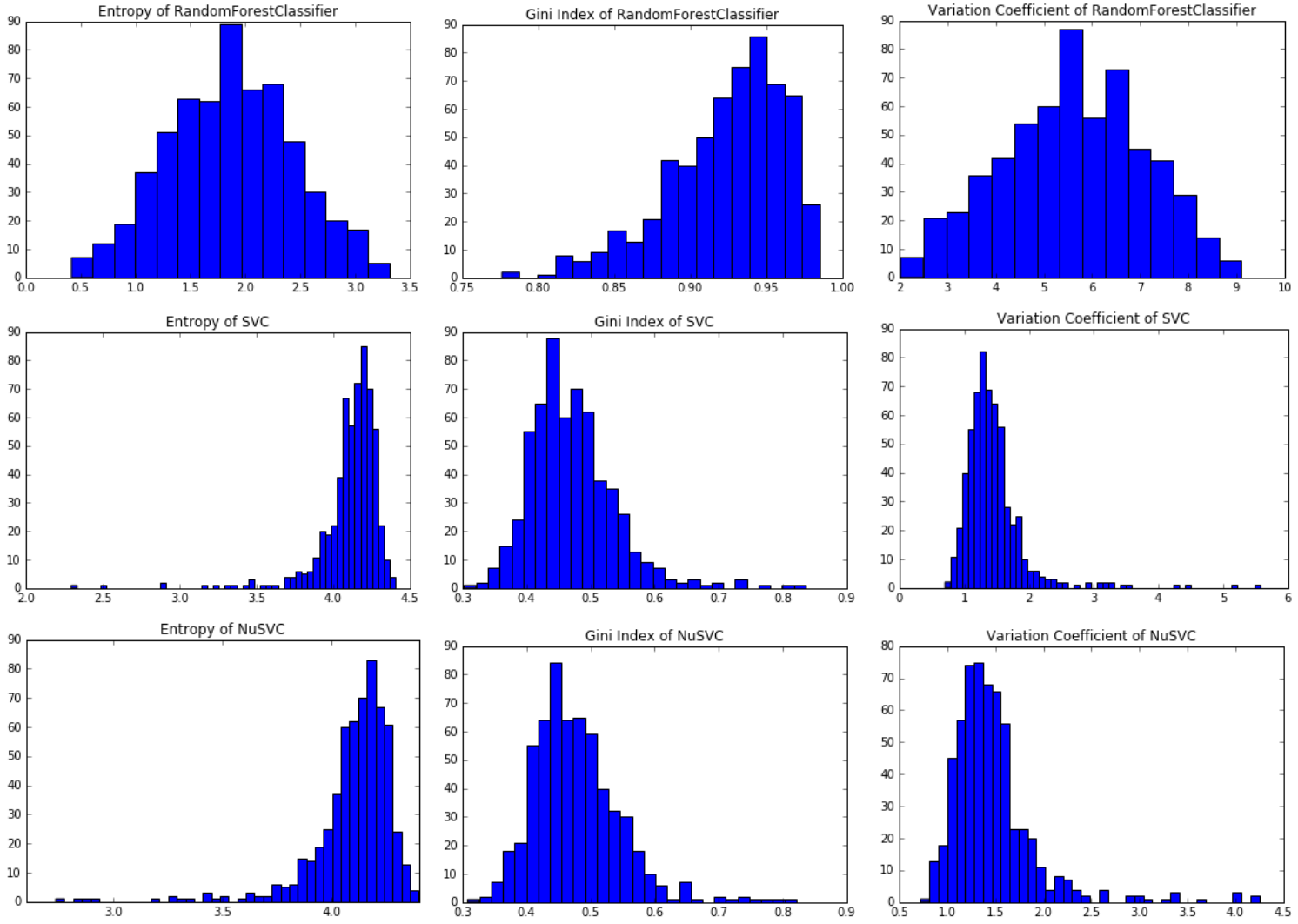

Figure 4.1: Histograms of the uniformness measurement functions with all 192 features.

advantage of classifier combination is not very obvious in the previous experiment. In order to test on a more difficult classification scenario, noises are added in two aspects. Firstly, some strong features in the original dataset are removed. Secondly, several weaker classifiers are introduced into the combination.

The original features' importance is assessed by Random Forest. In each decision tree, features used as the split criterion at the top nodes (closer to the root) exert larger effects on the final class label of the data points, since a larger fraction of samples are split according to these features. Thus, the depth of node in the classification tree can be used to estimate the expected fraction of the samples it contributes to, which reflects the importance of this split feature.[1] Averaging over all the decision trees, Random Forest can provide the relative importance of predictor features, as shown in Figure 4.2. The top 150 features with high importance are removed from the original dataset to increase the classification difficulty.

Three extra k-NN models are introduced. With larger k parameters than the best value selected by the stratified cross validation, they have worse predictive accuracy. This also expands the range of individual classifiers' accuracy, to differentiate the predictive strengths of base models. The aim is to test the combination schemes on a more general and realistic problem setting.

Each classifier is again trained using the 10-fold stratified cross validation. The result is listed in Table 4.3. The average cross validation accuracy is used as the overall accuracy parameter in the combination scheme of Equation (3.2) for each individual classifier. After removing the strong features, the previous classifiers have worse cross validation performance, with *accuracy* between 0.788889 and 0.874747, based on the remaining 42 features. The three new k-NNs have even lower *accuracy*. The *logloss* cross validation performance is also worse under the new problem setting.

[1] http://scikit-learn.org/stable/modules/ensemble.html#forest

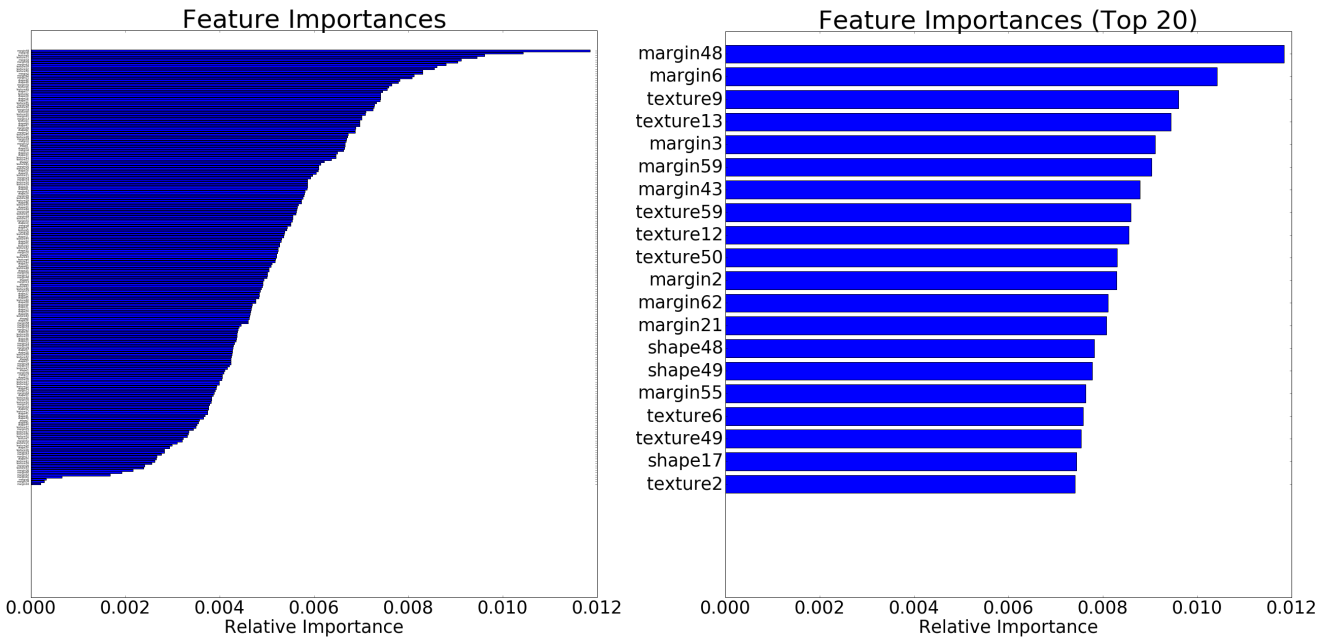

Figure 4.2: Random Forest Feature Relative Importance, the left figure contains all 192 features, while the right figure plots the Top 20 features.

The test result of three combination schemes are listed in Table 4.4. All base classifiers' performances, both *accuracy* and *logloss*, are similar to the stratified cross validation, which excludes the problem of overfitting. Gini-based combination scheme outperforms all the individual classifiers, in term of *accuracy*. Entropy-based combination scheme achieves the same *accuracy* level as the best individual classifier, SVM. These two schemes also have lower *logloss* than all base classifiers. They both outperform the benchmarks of the simple average and the weighted average by *accuracy* and *logloss*. However, Variation-based combination scheme's predictive performance is still among the individual classifiers. It's worse than the two Average-based combination methods.

Figure 4.3 shows the histograms of all base classifiers' three uniformness measures for the test set observations. The classifiers have similar behaviors, compared with the original dataset. SVM and $\nu-$SVM still have higher values in Entropy and lower values in Gini than other

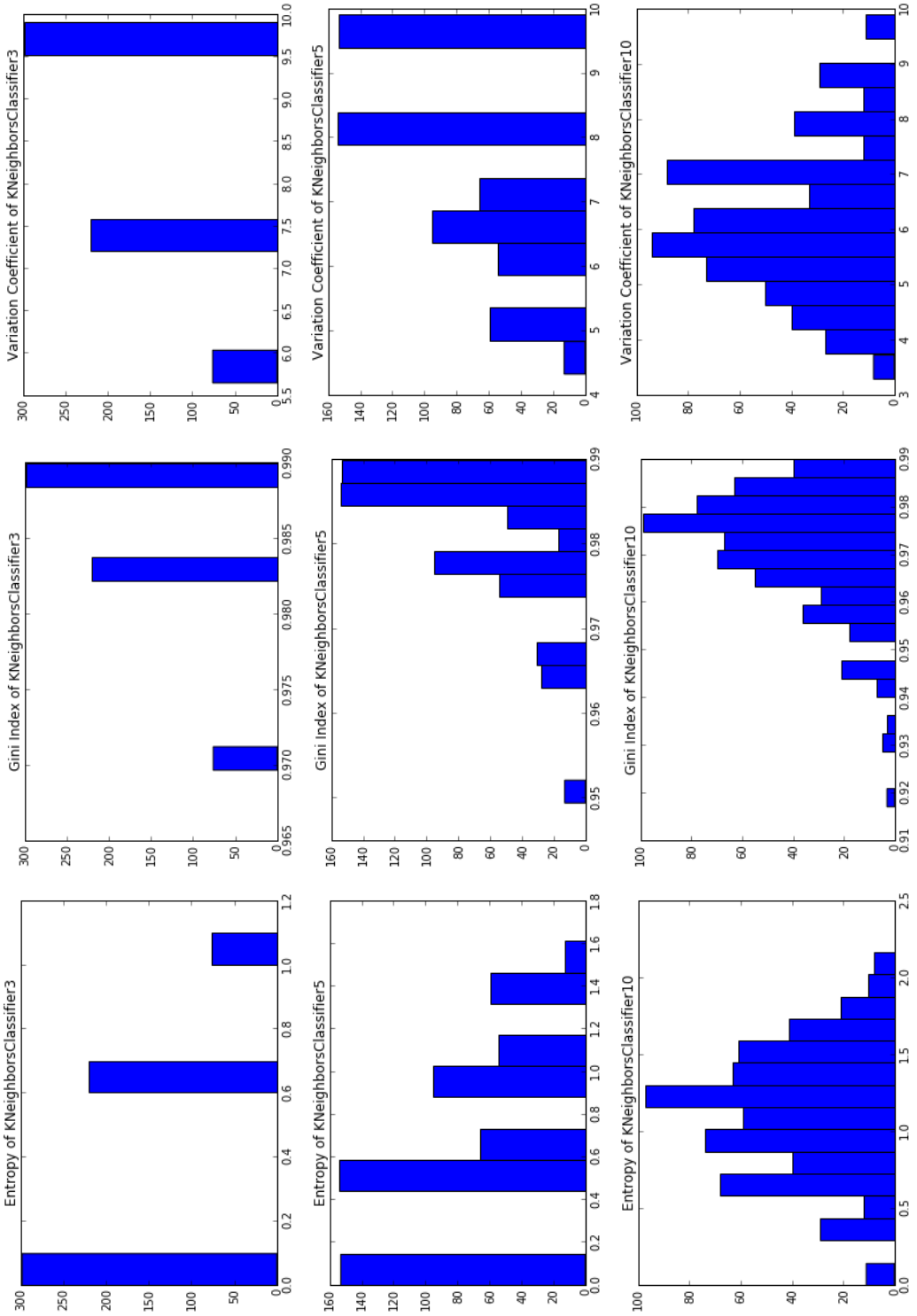

Figure 4.3: Histograms of the uniformness measurement functions with all 192 features.

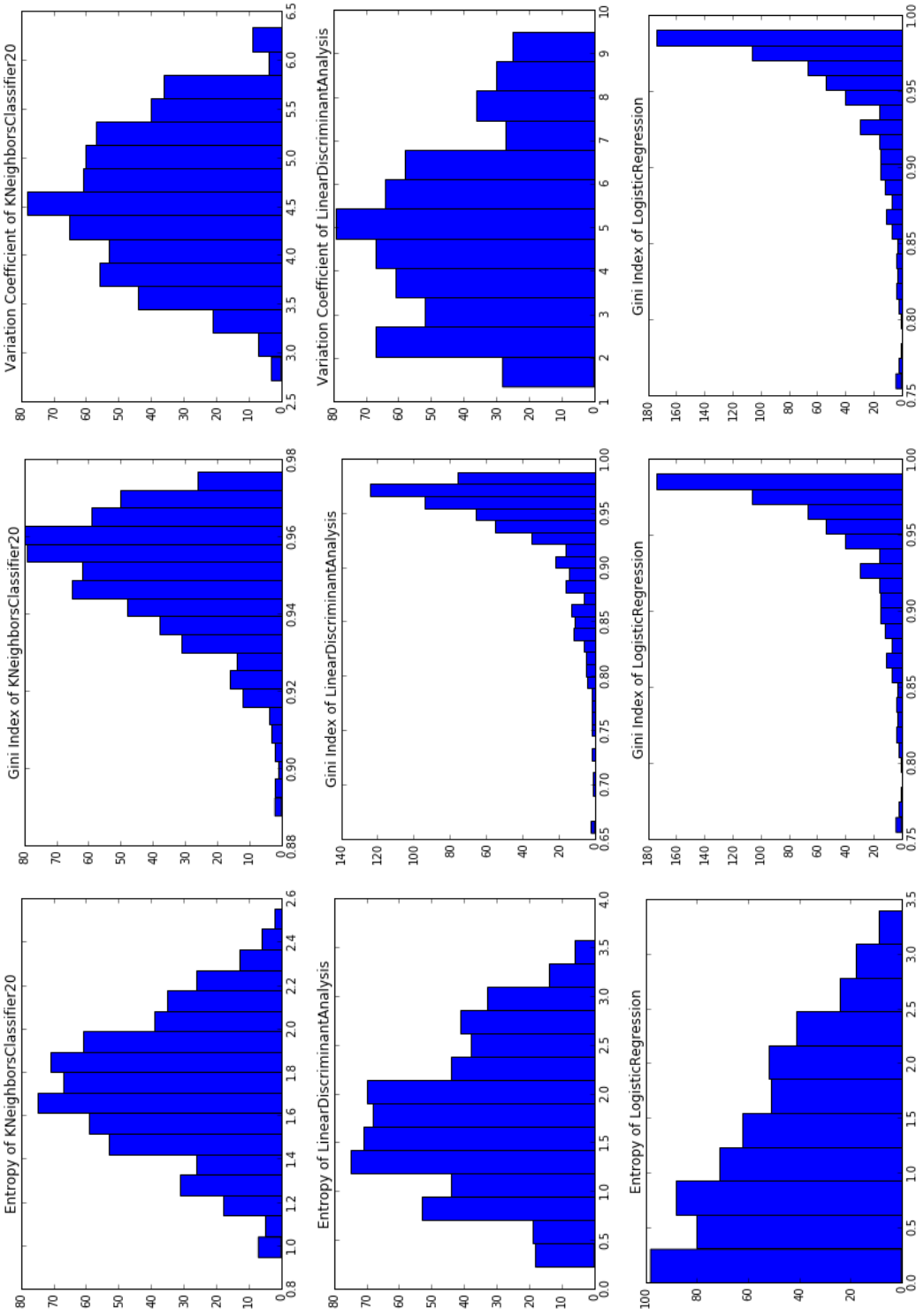

Figure 4.3: Histograms of the uniformness measurement functions with all 192 features.

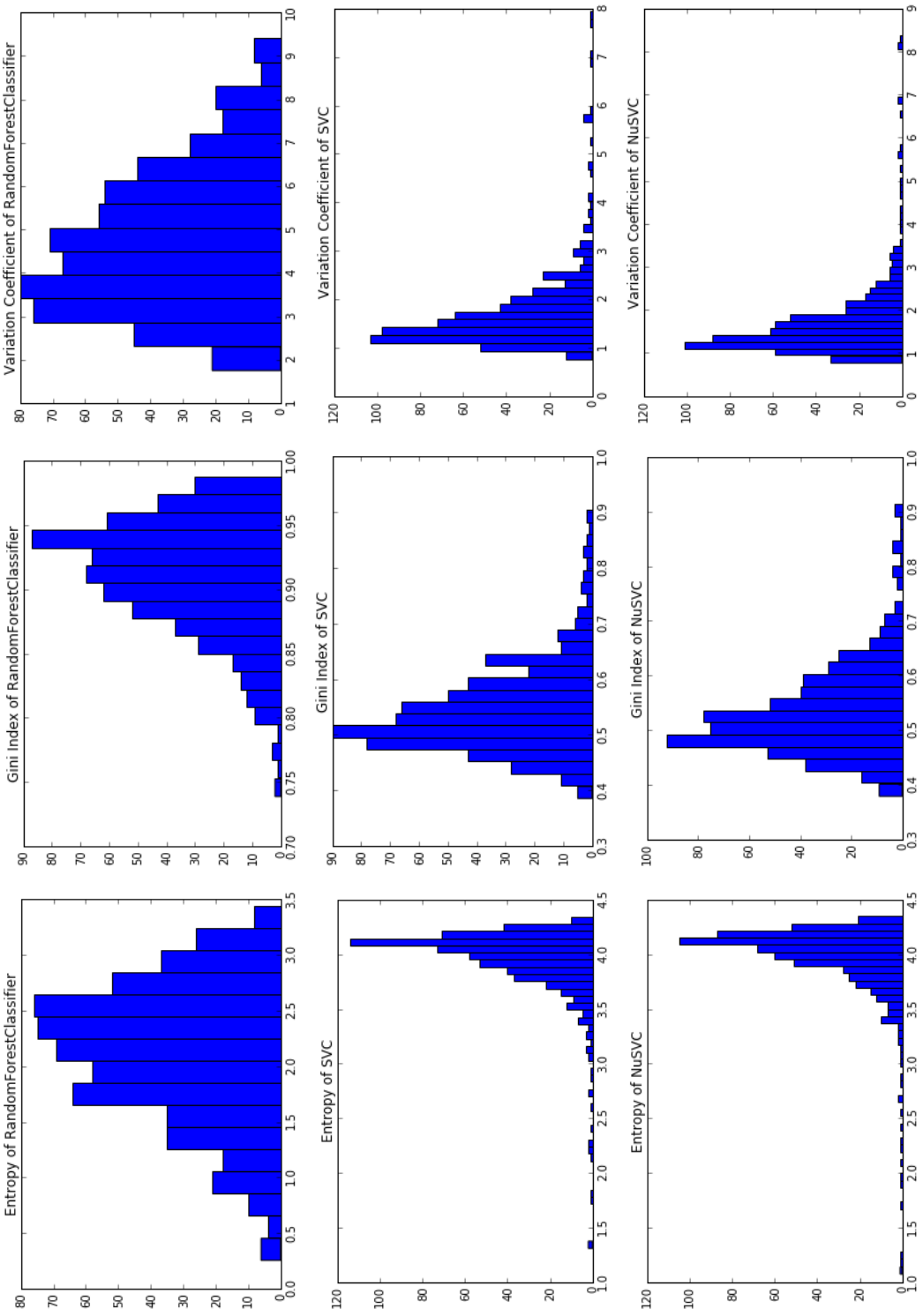

Figure 4.3: Histograms of the uniformness measurement functions with all 192 features.

Classifier	**CV Accuracy**	**CV logloss**
k-NN(3)	0.788889	2.655676
k-NN(5)	0.779293	1.831460
k-NN(10)	0.743434	1.647518
k-NN(20)	0.659091	1.696260
LDA	0.847980	1.361570
Logistic Regression	0.874747	0.618047
Random Forest	0.853030	1.095315
SVM	0.830303	2.400027
$\nu-$SVM	0.865657	2.363698

Table 4.3: Cross Validation of Base Classifiers with 42 Features

classifiers, which is consistent with their high *logloss*. The newly introduced three kNN models have outputs with higher uniformness, as the parameter k increases. The shifts of the peaks in their Entropy and Gini histograms are obvious. This reflects the noise and the lower level of model confidence.

In order to test the robustness of the probability-based combination schemes, a further modified version excludes the top 3 classifiers with the highest predictive *accuracy*: Logistic Regression, Random Forest, and $\nu-$SVM, while others remain the same. This new test setting adds to the difficulty of the classification, since the combination contains fewer and weaker base classifiers. The result is listed in Table 4.5. All the three combination schemes' performances deteriorate because of the exclusion of the strong classifiers. However, they still have advantage over the remaining base classifiers. Gini-based combination scheme outperforms all the individual classifiers, in term of *accuracy*. Entropy-based combination scheme achieves the same *accuracy* level as the best individual classifier, Logistic Regression. These two schemes also have lower *logloss* than all base classifiers, except for Logistic Regression. However, Variation-based combination scheme's predictive performance is still among the

Model	**Accuracy**	**Logloss**
Combination-Entropy	0.892256	0.74693
Combination-Gini	0.922559	0.80351
Combination-Variation	0.856902	2.22167
Simple Average	0.878788	1.01403
Weighted Average	0.890572	0.92351
k-NN(3)	0.797980	3.23597
k-NN(5)	0.774411	2.30688
k-NN(10)	0.767677	1.57074
k-NN(20)	0.703704	1.65144
LDA	0.855219	1.31816
Logistic Regression	0.892256	0.55779
Random Forest	0.877104	1.00408
SVM	0.853535	2.12663
$\nu-$SVM	0.872054	2.13151

Table 4.4: Test Predictive Performance with 42 Features

individual classifiers. All the three combination schemes outperforms the benchmarks of the simple average method and the weighted average method by *accuracy*.

Model	**Accuracy**	**Logloss**
Combination-Entropy	0.855219	0.88710
Combination-Gini	0.867003	1.23904
Combination-Variation	0.831650	2.26138
Simple Average	0.823232	0.91618
Weighted Average	0.823232	0.91730
k-NN(3)	0.797980	3.23597
k-NN(5)	0.774411	2.30688
k-NN(10)	0.767677	1.57074
k-NN(20)	0.703704	1.65144
LDA	0.855219	1.31816
SVM	0.853535	2.12663

Table 4.5: Test Predictive Performance with 42 Features and Weaker Classifiers

CHAPTER 5

Conclusion and Future Work

This thesis proposes a novel approach for the multi-class classifier combination. The combination incorporates the confidence attached to each class by the base classifiers in the form of uniformness of the output probability vector. This is a special case of Measurement Level combination, where the scores of base classifiers have a consistent format and interpretation. The approach combines the classifiers on an individualized basis, dynamically calculating the weights of classes for each individual data points. Therefore, it considers the specific characteristics of each input feature vector, instead of using a set of fixed parameters for the combination scheme, like the traditional methods including majority voting and Borda count. Such flexibility improves the classification performance while keeping the computation cheap.

In the real problem of plant leaf classification, the probability-based classifier combination schemes achieve good classification performance. In the two more difficult situations where the base classifiers are relatively weaker, the Entropy-based combination scheme and the Gini-based combination scheme outperform all the individual base classifiers. These results reflect the advantage and robustness of the probability-based classifier combination method. The new design provides a reliable approach even when the classification problem is quite difficult for many classifiers. It also provides better results than traditional methods, such as the simple average combination and the weighted average combination. However, the combination scheme of variation coefficient does not have better predictive performance than the individual classifiers, either in terms of *accuracy* or *logloss*, which suggests the importance of selecting the uniformness measurement function.

Future development of the probability-based classifier combination method can assign a different formula for each class. From the Bayesian perspective, each class needs a set of base classifiers to predict the posterior probability and needs a specific combination scheme. This is not covered in the current project since the sample size of each class in the plant leaf dataset is much smaller than the number of features. Also, the priori distribution of the class labels, which is a known uniform distribution in this dataset, should be estimated. Other uniformness measurement functions can also be explored. In the experiment, the two evaluation indices *accuracy* and *logloss* are not alway consistent. The discrepancy also requires further research.

REFERENCES

[1] Tulyakov, Sergey, et al. Review of classifier combination methods. *Machine Learning in Document Analysis and Recognition*. Springer Berlin Heidelberg, 2008. 361-386.

[2] Tax, David MJ, Robert PW Duin, and Martijn Van Breukelen. Comparison between product and mean classifier combination rules. *Proc. Workshop on Statistical Pattern Recognition, Prague, Czech.* 1997.

[3] Mallah, Charles, James Cope, and James Orwell. Plant leaf classification using probabilistic integration of shape, texture and margin features. *Signal Processing, Pattern Recognition and Applications* 5 (2013): 1.

[4] Kittler, Josef, et al. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence* 20.3 (1998): 226-239.

[5] Schölkopf, Bernhard, et al. New support vector algorithms. *Neural computation* 12.5 (2000): 1207-1245.

[6] Van Erp, Merijn, Louis Vuurpijl, and Lambert Schomaker. An overview and comparison of voting methods for pattern recognition. *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on.* IEEE, 2002.

[7] Rahman, Ahmad Fuad Rezaur, Hassan Alam, and Michael C. Fairhurst. Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations. *International Workshop on Document Analysis Systems.* Springer Berlin Heidelberg, 2002.

[8] Suen, Ching, and Louisa Lam. Multiple classifier combination methodologies for different output levels. *Multiple Classifier Systems* (2000): 52-66.

[9] Florian, Radu, et al. Named entity recognition through classifier combination. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4.* Association for Computational Linguistics, 2003.

[10] Shipp, Catherine A., and Ludmila I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information fusion* 3.2 (2002): 135-148.

[11] Rohlfing, Torsten, Daniel B. Russakoff, and Calvin R. Maurer. Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation. *IEEE transactions on medical imaging* 23.8 (2004): 983-994.

[12] Kim, Eunju, Wooju Kim, and Yillbyung Lee. Combination of multiple classifiers for the customer's purchase behavior prediction. *Decision Support Systems* 34.2 (2003): 167-175.

[13] Brill, Eric, and Jun Wu. Classifier combination for improved lexical disambiguation. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1.* Association for Computational Linguistics, 1998.

[14] Ho, Tin Kam, Jonathan J. Hull, and Sargur N. Srihari. Decision combination in multiple classifier systems. *IEEE transactions on pattern analysis and machine intelligence* 16.1 (1994): 66-75.

[15] Sohn, S. Y., and H. W. Shin. Experimental study for the comparison of classifier combination methods. *Pattern Recognition* 40.1 (2007): 33-40.

[16] Kirchhoff, Katrin, and Jeff A. Bilmes. Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values. *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on.* Vol. 2. IEEE, 1999.