# **Face Detection Using Ensemble Methods**

# **– Random Forest and Mixture of Experts**

|  |  |
| --- | --- |
| Wasim Khan  Boston University  Boston, MA, 02215  wasimk95@bu.edu | Lingshan Yang  Boston University  Boston, MA, 02215  yls@bu.edu |

**Abstract**

*Face detection is one kind of biometric detection technique based on facial features. It has many practical applications such as in security systems, electronic passports or identity cards, and tracking fugitives. We plan to explore, understand and implement face detection system using Random Forest and Mixture of Experts, two implementations of ensemble methods. In addition, we will implement K-Nearest Neighbor as a baseline and then compare the performance based on several metrics and different datasets.*

# Introduction

Face detection is a problem that involves extracting features from an image or a video, and identifying human faces. Research of face detection began in 1960s and quickly improved due to the development of computer techniques and optical imaging techniques in 1980s. There are several challenges in face detection. First is the internal changes of faces such as different skin colors, face shapes, facial expressions and facial shelters (like hair, beard, glasses and masks). Second is the external changes such as different filming angles, light conditions and imaging methods. [1]

An ensemble method itself is a supervised learning algorithm. It combines several learning algorithms to gain a better predictive performance. Although there are appropriate learning algorithms for a particular problem, ensemble methods are designed to find the best solution when facing a new problem. In other words, the ensemble methods have more flexibility to fit the training data. Due to the diversity among the algorithms, it can address over-fitting problems in some cases. [2]

# Literature Review

## Random Forest

Random forest is one kind of classification method to combine many tree classifiers. There are two characteristics of random forest, one is randomly selecting training data and the other is randomly selecting features. Using ensemble of trees and letting these trees vote for the most popular class can significantly increase the accuracy of the tree classifiers. Random vector is created for each tree independently and each tree grows according to the values of the random vector. Each node of a tree will also randomly select features. Compared to the Adaboost, random forests’ error rate are more robust regarding the noise. [3]

## Mixture of Experts

Typical ensemble methods train all of the base learners using the same algorithm, and then combine results to make a prediction. Instead, Mixture of Experts trains each weak learner to specialize in a specific part of the dataset. A gating function is used to assign probabilities to each expert, which determines the likelihood of picking that expert given the specific input, and is modeled by a softmax function [4]. Since the gating network outputs higher probabilities when the input belongs to the region that the learner specializes in, the parameters for each weak learner will only be affected by data that the weak classifier is an expert in. And just like other ensemble methods, the predictions of each weak classifier can be combined, although weighted by the gating network manager, to predict the final outcome.

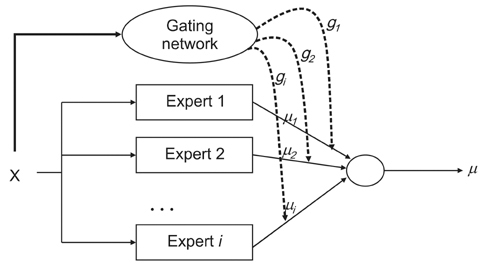


Figure 1. Mixture of Methods Model.

## Baseline - K-Nearest Neighbor Classifier

One of the most basic and popular classifiers, K-Nearest Neighbor (K-NN) has good performance in many cases. We choose K-NN as the baseline and intend to compare the Correct Classification Rate, Precision, Recall and F-score between the baseline and our system. The essential meaning of K-NN is to put the training data into a vector space, then count the k nearest points’ classes of one sample in the vector space and give the class with the most nearest points to that sample. In case of face detection, there are only two classes, one positive class that has faces and one negative class that has not. It assumes that all the data with faces will cluster together. [5] A test sample inside the cluster will be classified as a positive sample while any test samples outside the cluster will be treated as a negative one.

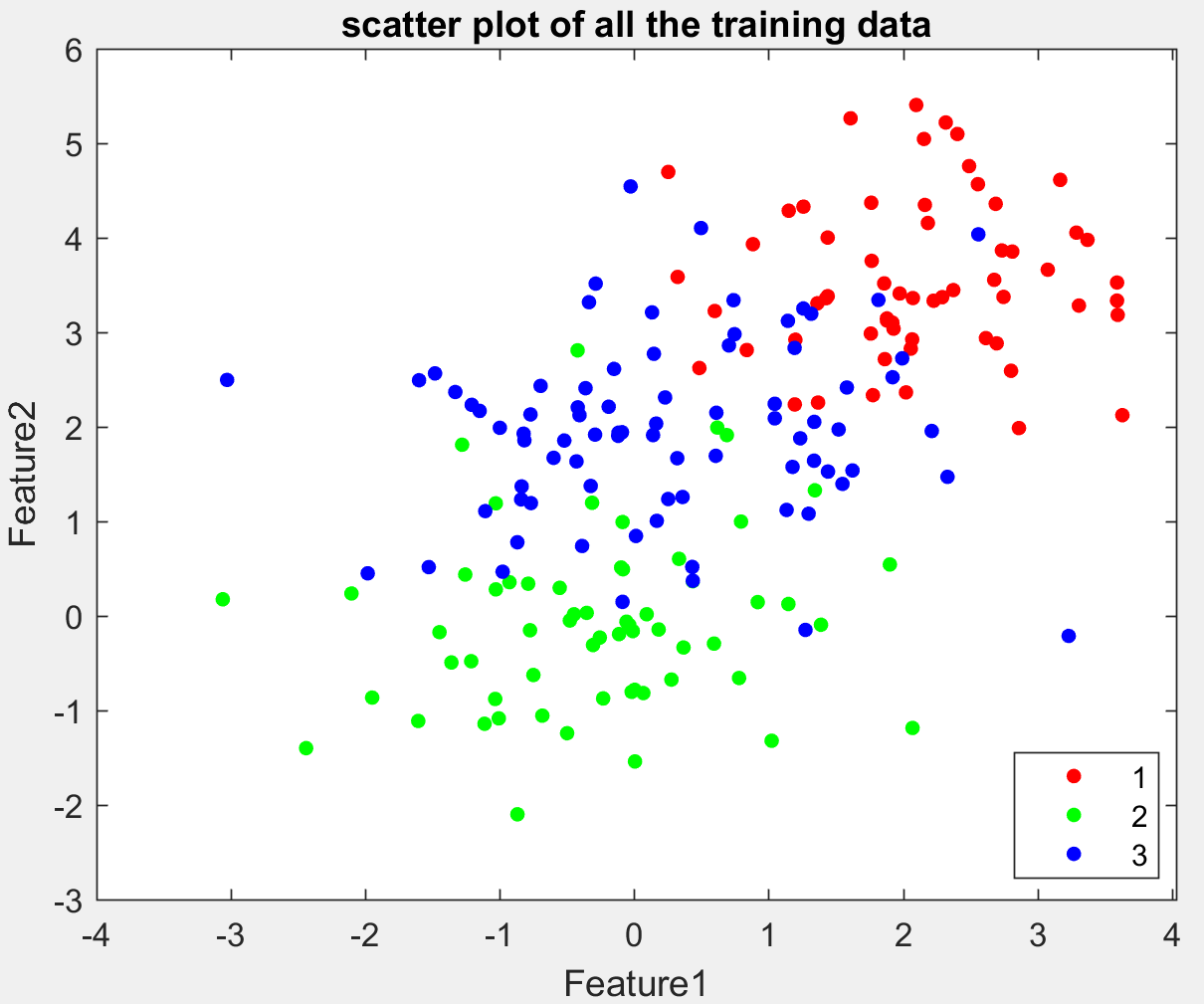


Figure 2. Scatter plot of 3-class K-NN data.

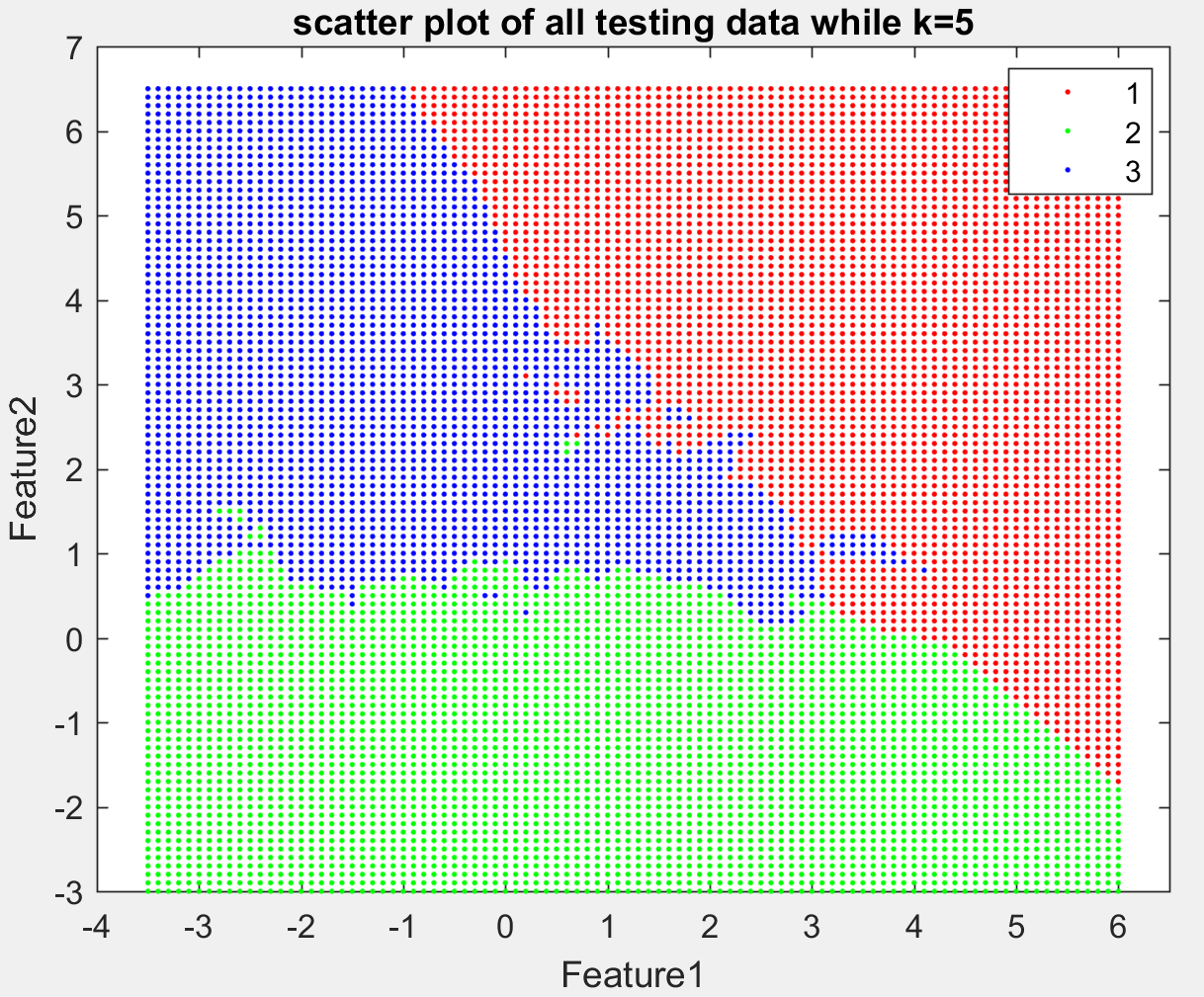


Figure 3. 5-NN prediction of points on graph.

# Problem Formulation and Solution Approaches

Mixture of experts is a discriminative mixture model, and so it can be solved using the expectation maximization (EM) algorithm. By introducing a hidden variable Z, and taking the expectation of the log likelihood, the problem can be solved in two separate optimization steps. The likelihood can be written as the following equation

(1)

Taking the expectation of the log of the likelihood allows us to divide the likelihood into two separate models since the log can be distributed to both. In addition, they are weighted by some terms determined by the weights from the previous iteration, which are used to calculate weights for the current iteration. The E step consists of finding these weights. The following equation is the expected log likelihood and shows this simplification

(2)

In this case, is found in the E step. In the M step we maximize, the above equation. The models we chose for the face detection application were logistic regression for both the gating function and the experts. This is because it is a classification problem and using a softmax seems like a logical decision to make. The following equations represent the models

Gating Function: =

Experts: =

where .

The two equations for the gating function and experts can be optimized through gradient descent at each iteration of the EM algorithm.

# Implementation

The implementation consists of first creating the algorithm and testing it on synthetic data, and then running it on the face detection dataset and evaluating the performance using experiments. The first part of both of these steps was defining the parameters and then executing the algorithm to predict the labels. Through trial and error, we chose parameters that took into account both runtime and correctness of results so that we may run it in a timely fashion and get descent results.

**4.1 Gradient Descent**

For the gradient descent part of the algorithm, the parameters that did best, which we determined through trial and error were a fixed step size of 10^-5 and 500 maximum iterations. The gradient descent algorithm was used for both the experts and the gating function. The only difference was the number of parameters. For the gating function, there were k number of d-vector weights,

**4.2 Expectation Maximization**

For the EM algorithm, we chose 2000 max iterations, which we evaluated against the time it took to run and concluded it worked best for the synthetic dataset since the number of samples were minimal. For the face detection application, since the number of dimensions was large, even after running PCA for dimensionality reduction, the we chose 100 max number of iterations for both gradient descent and EM.

**4.3 Regression**

. This was the initial step towards implementing a working mixture of experts implementation. The difference between regression and classification was that for the maximum likelihood estimate, we used a Gaussian model instead of logistic regression to model the experts. This resulted in a higher runtime since the optimization for the Gaussian model could be done in one step. This is almost the same as MMSE, except that it was a weighted sum of least squares. Furthermore, the decisions made after estimating the parameters were soft decisions, and a weighted sum of the decisions from the individual experts is made with the weights determined by the gating function.

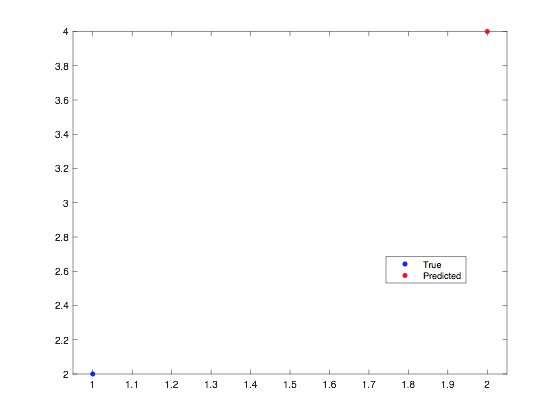
**4.4 Classification**

The classification implementation was used for both synthetic data and the face detection application. The difference between this and regression was that the model for the experts was logistic regression, and a hard decision was made based on the posterior calculated for the whole model.

# Experimental Results

**5.1 Regression – Synthetic**

We generated n one dimensional samples that were clearly separated in space. This allows us to show the correctness of the algorithm without introducing complexity. As can be seen from the figure, the data is modeled fairly accurately for 5 experts with 5 clearly separated regions. The prediction is the weighted sum of all of the experts, as a result, the borders between regions are not well defined.



**5.2 Classification – Synthetic**

****

Using the toy dataset in figure ….. with 500 samples, the CCR increases with increasing experts as can be seen from the figure below.

Using the same toy dataset,

**5.3 Face Detection**

The CCR for the face detection algorithm using number of experts from 1-10 resulted in a consistent CCR of about .5. In addition we ran tests for large number of experts equal to 2^k for k=5-11, and that resulted resulted in a consistent CCR around .5 as well as shown in figure….





# Conclusion

## Random Forest

* For each decision tree inside the random forest, the number of nodes of it should be some appropriate value. This is because the classification will not be precise enough if the nodes are too less and it will arise overfitting problems on training set if the nodes of tree are too more.
* As the number of trees increasing, the performance of the random forest tends to converge. This is because if the size of the forest is big enough, those randomly selected features, which are redundant and less informative, will counteract each other and make some offset in the final prediction and those which are most decisive and powerful will be strengthened and play significant roles in the final result.

## Mixture of Experts

### General Results

The performance of mixture of experts was highly dependent on the separation of data in space. This was influenced by two points.

1. The number of samples in the dataset
2. Number of experts

In general, mixture of experts does well when the data is separated. This means that each expert, which focuses on a specific part of the dataset, has a large number of samples to model that sub region. In addition, the data in that sub region should be linear in the case of regression or a single class in the case of classification. As the number of experts increases, the CCR increases, up to a specific point determined by the number of samples in the training set. Any increase in experts after that point produces problems of over-fitting. In general, how well each expert can model the sub region will determine the accuracy of the model.

On the other hand, one also has to take into account the complexity of the algorithm. As, the number of experts increases, so does the runtime. In addition, since we are using two gradient descent algorithms in each EM iteration, the number of samples and experts is limited. This produces a big problem for large datasets with several classes.

### Face Detection

Mixture of experts does not perform well for face detection using Pascal VOC dataset. This is due to the fact that the number of samples in the dataset is small. Even though the application is binary, the number of dimensions is large and not only is a large amount of samples needed to accurately model the application. This is a problem in this implementation of mixture of experts, since we are bound by

# Description of Individual Effort

Although we intend to collaborate on every part of our project, the main responsibility of each person will be the following:

Lingshan Yang – Literature of survey of random forest, implementation of code of random forest and dimension reduction (PCA), analysis of the experimental results and all the stuff correlated to random forest in the final report.

Wasim Khan - Literature of survey of mixture of experts, implementation of code of mixture of experts and edge detection, analysis of the experimental results and all the stuff correlated to mixture of experts in the final report.

# References

1. <http://facedetection.com/>
2. R. Polikar. Ensemble based systems in Decision Making. IEEE Circuits and Systems Magazine. 6(3):21-45.2006.
3. Leo Breiman. Random Forests. Machine Learning. 45(1):5-32, 2001.
4. Zhihua Zhou. Ensemble Methods Foundations and Algorithms, Taylor & Francis Group, 2012.
5. Y. Liao and V. R. Vemuri. Use of K-Nearest Neighbor Classifier for Instruction Detection. Computers & Security, 21(5):439-448, 2002.