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.

1. 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]

2. Literature Review

2.1 Random Forests

Random forest is one kind of classification method to combine many tree classifiers. There are two characteristics of random forests, 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 is more robust regarding the noise. [3]

2.2 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]. A simple cost function of this model is the multiplication of this probability output by the gating function and the squared error of the prediction of that weak classifier. 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 as determined by the gating network manager.

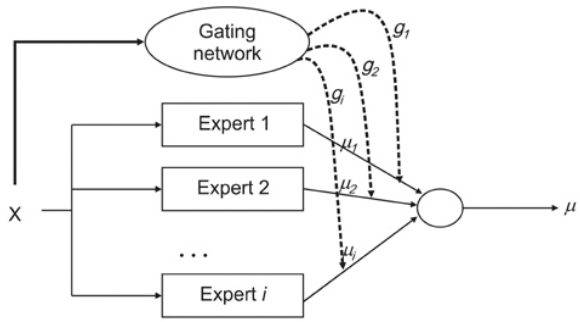

Figure 1. Mixture of Methods Model

2.3 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 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.

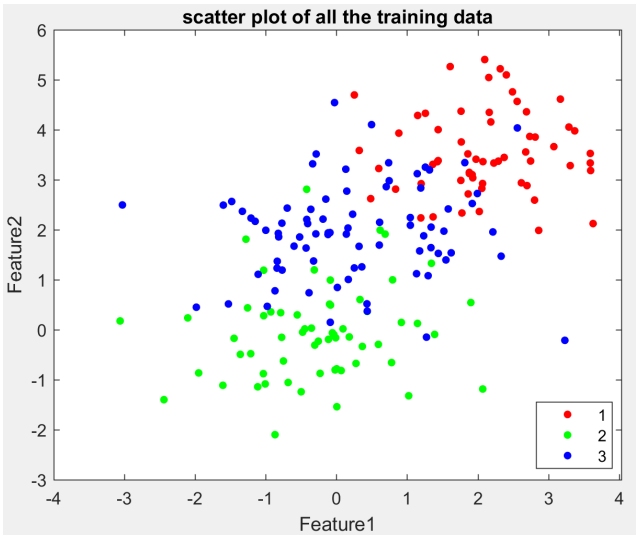

Figure 2. Scatter plot of 3-class K-nn data

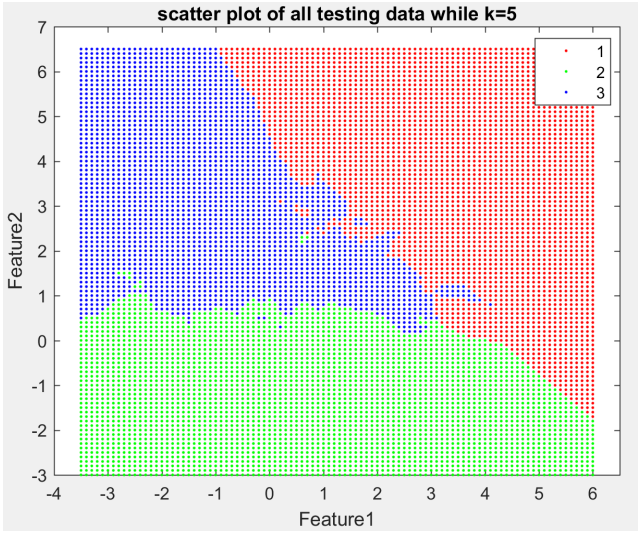

Figure 3. 5-nn prediction of points on graph

3. Proposed Work

We will be evaluating the accuracy and performance of ensemble methods when applied to detecting faces. We will be implementing a random forest algorithm that is an extension of bagging. In addition, we will be investigating combining methods by implementing mixture of experts, and comparing the results to random forests and a baseline K-nn algorithm. We will compare metrics such as the Correct Classification Rate, Precision, Recall and F-score based on several datasets. We will be reporting our understanding of why one algorithm

performs better than another on certain datasets.

4. Datasets

We will be using datasets with raw images where each image has zero to many faces. There are many facial detection datasets out there from raw images of faces to faces that have already been annotated. Some examples include Wider Face, which is a face detection benchmark, and the Faces in the Wild dataset. For our case, we will use the raw images to detect if the image has a face.

5. Division of Labor

Although we intend to collaborate on every part of our project, the main responsibility of each person will be the following:
Lingshan - Random Forests implementation and editor of the reports
Wasim - Mixture of experts and K-nn implementation and presentation of final reports

References

[1] http://facedetection.com/
[2] R. Polikar. *Ensemble based systems in Decision Making. IEEE Circuis and Systems Magazine.* 6 (3): 21-45, 2006.
[3] Leo Breiman. *Random Forests. Machine Learning.* 45 (1): 5–32, 2001.
[4] Zhou, Zhi-Hua. *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.