华南理工大学

South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Zhiquan Wen and Zijia Zhang

Supervisor:
Qingyao Wu

Student ID：
201530612996 or 201530613726

Grade:
Undergraduate

December 22, 2017

Face Classification Based On AdaBoost Algorithm

Abstract—The short abstract is intended to give the reader an overview of the experiment. It should be brief and to the point.

I. INTRODUCTION

(1) Motivation of Experiment

1. Understand Adaboost further
2. Get familiar with the basic method of face detection
3. Learn to use Adaboost to solve the face classification problem, and combine the theory with the actual project
4. Experience the complete proces

$$w_{m+1}(i) = \begin{cases} \frac{w_m(i)}{z_m} e^{-\alpha_m} & \text{for right predictive sample} \\ \frac{w_m(i)}{z_m} e^{\alpha_m} & \text{for wrong predictive sample} \end{cases}$$

- Error rate

$$\epsilon_m = p(h_m(\mathbf{x}_i) \neq y_i) = \sum_{i=1}^{n} w_m(i) \mathbb{I}(h_m(\mathbf{x}_i) \neq y_i)$$

$$\alpha_m = \frac{1}{2} \log \frac{1 - \epsilon_m}{\epsilon_m}$$

- Final learner

$$H(\mathbf{x}) = \text{sign}(\sum_{m=1}^{M} \alpha_m h_m(\mathbf{x}))$$

II. METHODS AND THEORY

We use AdaBoost Algorithm to complete this experiment.

And we use NPDFeature class's method to get the feature of the picture

We use sklearn.tree.DecisionTree to build the base-classifier

$$w_{m+1}(i) = \frac{w_m(i)}{z_m} e^{-\alpha_m y_i h_m(\mathbf{x}_i)}$$

III. EXPERIMENT

(1) Experiment Step

1. Read data set data. The images are supposed to converted into a size of 24 * 24 grayscale, the number and the proportion of the positive and negative samples is not limited, the data set label is not limited.
2. Processing data set data to extract NPD features. Extract features using the NPDFeature class in feature.py. (Tip: Because the time of the pretreatment is relatively

long, it can be pretreated with pickle function librarydump () save the data in the cache, then may be used load () function reads the characteristic data from cache.)

3. The data set is divisded into training set and calidation set, this experiment does not divide the test set.
4. Write all *AdaboostClassifier* functi ons based on the reserved interface in *ensemble.py*. The following is the guide of *fit* function in the *AdaboostClassifier* class:

   4.1 Initialize training set weights, each training sample is given the same weight.

   4.2Training a base classifier, which can be sklearn.tree library DecisionTreeClassifie r (note that the training time you need to pass the weight as a parameter).

   4.3 Calculate the classification error rate of the base classifier on the training set.

   4.4 Calculate the parameter according to the classification error rate.

   4.5 Update training set weights.

   4.6 Repeat steps 4.2-4.6 above for iteration, the number of iterations is based on the number of classifiers.

5. Predict and verify the accuracy on the validation set using the method in AdaboostClassifier and useclassification_report () of the sklearn.metrics library function writes predicted result to *report.txt*.
6. Organize the experiment results and complete the lab report (the lab report template will be included in the example repository)

IV. CONCLUSION

(1) Dataset

1. This experiment provides 1000 pictures, of which 500 are human face RGB images, stored in*datasets/original/face*; the other 500 is a non-face RGB images, stored in *datasets/original/nonface*.
2. The dataset is included in the example repository. Please download it and divide it into training set and validation set.

After running the code, we get the picture such as :

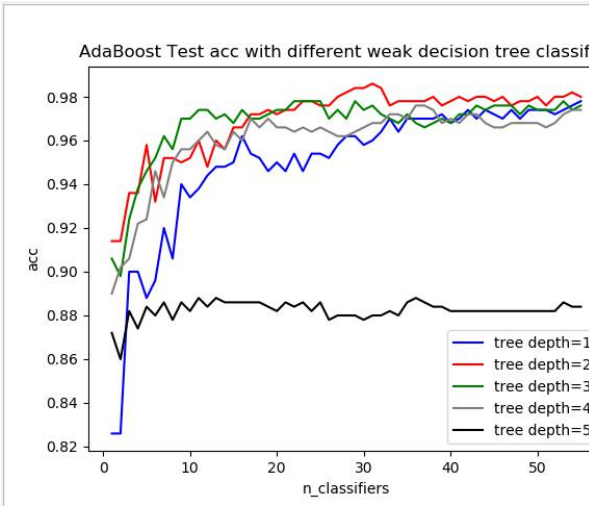

Fig.1

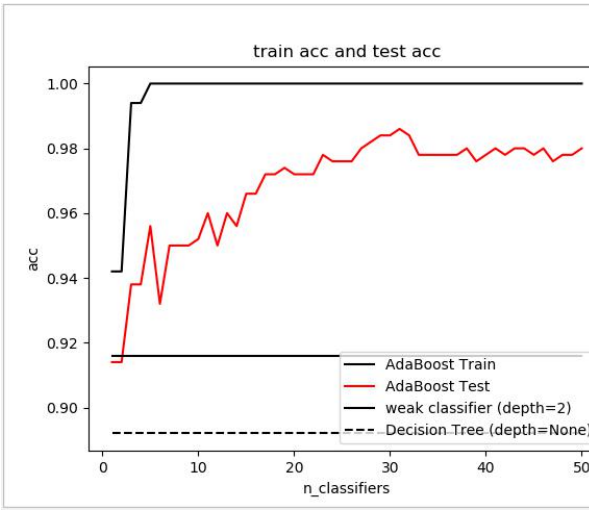

Fig.2

From Fig. 2, after adaboost, the accuracy of the training set is 1, the accuracy of the test set is 0.98, the accuracy of the base learner is about 0.91, and the accuracy rate of adaboost is obviously improved.

From Fig. 1, the depth of the tree has an effect on the accuracy of the adaboost, when the depth of the tree reaches 5, the accuracy is reduced to 0.88, so the depth of the tree can not be too deep, that is, the complexity can not be too high, otherwise it will fit