



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:

Chaoxian Zhang and Yuchen Zhou and
Wende Zhu

Supervisor:

Mingkui Tan

Student ID: 201530613597 and

201530613887 and 201530613948

Grade:

Undergraduate

December 21, 2017

Face Classification Based on AdaBoost Algorithm

Abstract—Based on the Adaboost algorithm, the weights w are updated by training the base classifier. Then predict and verify the accuracy on the validation set using the method in AdaboostClassifier.

I. INTRODUCTION

Most efficient algorithms can only be implemented by extracting high efficiency and benefiting recognition and processing features, and the Adaboost algorithm is the same. Adaboost, the algorithm idea is: Based on a large number of positive and negative samples of learning, through learning feedback, weak classifier in the premise of training do not know a priori error under adaptive adjustment error the rate and the corresponding weights, until the strong classifier to achieve the desired performance.

II. METHODS AND THEORY

1. Read data set data. The images are supposed to be 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 library dump() save the data in the cache, then may be used load() function reads the characteristic data from cache.)

3. The data set is divided into training set and validation set, this experiment does not divide the test set.

4. Write all AdaboostClassifier functions 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 w , each training sample is given the same weight.

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

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

4.4 Calculate the parameter α according to the classification error rate ϵ .

4.5 Update training set weights w .

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 use classification_report() of the sklearn.metrics library function writes predicted result to report.txt.

III. EXPERIMENT

A. Dataset

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.

B. Implementation

Firstly, we converted these images into a size of 24×24 grayscale, such as in Fig. 1.



Fig. 1. Face images' grayscale

In order to extract NPD features, we use the NPDFeature class in feature.py, and then divided them into train set and test set.

We initialize w to $1/n$ (n is the number of train set). For $t = 1, 2, 3, \dots$ (t is epoch times) w updates as Fig. 2. By the way, α_m

is $\alpha_m = \frac{1}{2} \log \frac{1 - \epsilon_m}{\epsilon_m}$. The max depth = 1.

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

Fig. 2. Update w

For each feature, we use weak learning algorithm to train the weak classifier, and calculation error rate. Select the weak classifier which has the minimum error rate, then update w . Repeat it and gain the strong classifier.

Now, we gain the strong classifier, we can test the accuracy of it through test set. The result is showed in Fig. 3.

```
0 the total accuracy rate is 0.775757575758
1 the total accuracy rate is 0.775757575758
2 the total accuracy rate is 0.878787878788
3 the total accuracy rate is 0.878787878788
4 the total accuracy rate is 0.927272727273
5 the total accuracy rate is 0.930303030303
6 the total accuracy rate is 0.933333333333
7 the total accuracy rate is 0.927272727273
8 the total accuracy rate is 0.939393939394
9 the total accuracy rate is 0.939393939394
10 the total accuracy rate is 0.939393939394
11 the total accuracy rate is 0.936363636364
12 the total accuracy rate is 0.933333333333
13 the total accuracy rate is 0.936363636364
14 the total accuracy rate is 0.936363636364
```

Fig. 3. The total accuracy rate update with the number of base classifiers

IV. CONCLUSION

Through the experiment, we have achieved high efficiency face classification. The result shows that when the number of base classifiers more than 8, the total accuracy rate will be around 93.6%. Adaboost Algorithm is based on huge train set data. How to reduce training cycle time is still inadequate. But in this experiment, we use a txt to store the data. So we needn't to preprocessing images again. All in all, we successfully completed the experiment under the experimental guidance.