華南理工大學

South China University of Technology

Face Classification Based on *AdaBoost* Algorithm

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Renxin Zhuang, Wei Wen, ZiWen Zhao

Supervisor:
Mingkui Tan

Student ID:
201530613979, 201530613009,
201530613764

Grade:
Undergraduate

December 22, 2017

***Abstract*—This is the third experiment of the Machine Learning course. Instead of completing the experiment indisivually, this time we are asked to accomplish the experiment by group. We worked together to solve the Face Classification problem.**

I. Introduction

THE motivation for this experiment has four folds. First, it helps us to understand Adaboost further. Second, through this experiment we can get familiar with the basic method of face detection. Third, we get to learn to use Adaboost to solve the face classification problem,and combine the theory with the actual projectLast, it gives us a better way to experience the complete process of machine learning.

II. Methods and Theory

We use Adaboost with sklearn.tree.DecisionTreeClassifier as base learner to recognize face from a picture. Suppose we have a base learner

$$h_m(x) : x- > \{-1, +1\}$$

Then we calculate the error rate ϵ

$$\epsilon_m = p(h_m(x_i) \neq y_i) = \sum_{i=1}^{n} w_m(i) I(h_m(x_i) \neq y_i)$$

According to the error rate we compute the score α of this base learner

$$\alpha_m = \frac{1}{2} log \frac{(1-\epsilon)}{\epsilon}$$

Next we need to recompute the weight of different samples to make the wrong predicted samples more important to the next classiffier.

$$z_m = \sum_{i=1}^{n} w_m(i) e^{-\alpha_m y_i h_m(x_i)}$$

$$w_{m+1}(i) = \begin{cases} \frac{w_m(i)}{z_m} e^{-\alpha_m} & forrightpredictivesample \\ \frac{w_m(i)}{z_m} e^{\alpha_m} & forwrongpreditivesample \end{cases}$$

Repeat this process until the performance is good enough. So we have the final boosted classifier

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

III. Experiments

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

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 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 divisded into training set and calidation 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 w 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 .

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.

C. Result

At first, we use 5 base learners and 100 samples as training set. When the max_depth of sklearn.tree.DecisionTreeClassifier is set to be 1,the precision score and recall score are all 0.65. It's relatively small. Next we set the max_depth to be 2, this time the precision score grows to be 0.75 and the recall score is 0.74. The performance is getting better but it's not enough. So we still promote the max_depth and now it became 3, this time we found that in the process of boosting the error rate becomes 0 and the final precision score and the recall score are all 0.80. This seems pretty good. Finally, we decide to use $\frac{2}{3}$ samples as training set so we can set the max_depth to be 6 and the number of base learners to be 10, the precision score is and the recall score are all 0.83.

IV. Conclusion

In conclusion, this paper demonstrates the process of applying AdaBoost Algorithm to solve face recognization problem.

In this experiment, we learned about how to use AdaBoost and work together to solve a problem. The parameter adjusting process is quite tough and sometimes exhausting but we're glad that we made it.