South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:

Ziwei Zhang, Haixu Liu and Jintao Xuan

Supervisor:

Mingkui Tan

Student ID：

201530613719, 201530612347 and 201530031377

Grade:

Undergraduate

December 17, 2017

Experiments with AdaBoost Algorithm

Abstract— We implement a simple AdaBoost algorithm used in classification problem. It can be used to reduce the error of a vanilla classification algorithm. In this report, we record experiments we carried out to implement and access the performance of the AdaBoost algorithm.

I. INTRODUCTION

AdaBoost is a popular ensemble method for improving the performance of many learning algorithm. The core principle of AdaBoost is to fit a sequence of weak models by continuously reweighted data samples. The predictions from the sequence of weak models are weighted summed up to produce final prediction.

Face detection problem here is to tell a given image contains face or not. To tackle the problem using AdaBoost, a feature extraction step is needed to transform the raw pixel information into some more useful feature. The computed image feature can then be fed into the AdaBoost classifier to learn some rules to fit the data.

In our experiments, we completed three sub-tasks. First, we implemented a simple AdaBoost classifier in a general form which can solve a binary classification problem. The AdaBoost classifier can used any classifier model which implement the specified interfaces. Second, we used Decision Tree Classifier as weak model to ensemble to solve face detection problem. Third, we study the behavior of the AdaBoost classifier with different settings.

II. METHODS AND THEORY

A. The AdaBoost algorithm

The AdaBoost classifier is assembled by a sequence of weak models in an iterative way. The progress to learn and assemble a new weak model is called a boost. At each boost step, we reweight the training data, putting a heavier weight on the samples which were incorrectly predicted by the boosted model while decreasing the weight of those correctly predicted by the boosted model. As boosting continuing, the examples which are difficult to handle with receive ever-increased weight. Each subsequent weak model is thereby forced to concentrate on the examples that were missed by the previous weak model.

In our experiments, we used Decision Tree Classifier in sklearn.tree modul as weak model $h_m(x)$, where m indicated the m-th weak model. Then the initialized sample weight is set to give each sample the same weight:

$$w_i = \frac{1}{n}, \qquad i = 1, 2, \dots, n$$

where i indicates the weight of the i-th sample. Then, use the samples and sample weight to perform a boosted step. Feed the weighted data into a weak model to learn some decision rules and compute the error of the weak model in a weighted form:

$$\varepsilon_m = \Sigma_{i=1}^{n} \omega_i \mathbb{I}(h_m(X_i) \neq y_i)$$

Then compute the model importance coefficient α_m and the new sample weight $\omega_{m+1}(i)$ as a function of ϵ_m:

$$\alpha_m = \frac{1}{2} \, log\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)$$

$$\omega_{m+1}(i) = \frac{\omega_{m(i)}}{z_m} e^{-\alpha_m y_i h_m(X_i)}, \mathrm{i} = 1, 2, \dots, \mathrm{n}$$

Where z_m is a normalization term:

$$z_m = \Sigma_{i=1}^{n} \omega_m(i) e^{-\alpha_m y_i h_m(X_i)}$$

And enter the next boosted step. After some number of iterations, a boosted model is built. When predicate a new sample, we simply perform a weighted sum of all the weak models to output a score:

$$\mathrm{H}(x) = \Sigma_{i=1}^{n} \alpha_m h_m(x)$$

Then given a threshold we get a predicted label for the sample. In the experiment, we set the threshold to zero. If the score is positive the model predicates it as positive class, otherwise predicates as negative class.

B. Feature extraction

In our experiments, we used NPD feature of an image to do classification. The extracted NPD feature is then directedly fed into the AdaBoost model.

III. EXPERIMENT

A. Dataset

We carried out the experiment on a dataset provided by the TA. The dataset contains 1000 images (250*250). 500 images of the dataset are human face RGB images and other 500 images are non-face RGB images. We respectively took 300 images from face set and non-face set to form the training set. The remained 400 images form the test set.

B. Face detection task

First, we preprocessed the data. We convert the original image into greyscale and resize it from 250*250 to 24*24. Then extracted the NPD features and got a feature dimension of 165600 for each image.

Then, we setup an AdaBoost model using decision tree classifier of depth 2 and used it to build an ensemble model containing 50 weak classifiers to fit the training data.

Finally, we evaluated the boosted model on both training set and test set. The result is show in Table 1 and Table 2.

Table 1 Training result

	precision	recall	f1-score	support
non-face	1.00	1.00	1.00	300
face	1.00	1.00	1.00	300
total	1.00	1.00	1.00	600

Table 2 Test result

	precision	recall	f1-score	support
non-face	0.98	0.97	0.98	200
face	0.97	0.98	0.98	200
total	0.98	0.98	0.98	400

We also monitored the performance of the AdaBoost classifier. We computed the prediction accuracy after each boosted step and as expected the model's performance improve as the number of weak classifiers increases (see Figure 1.).

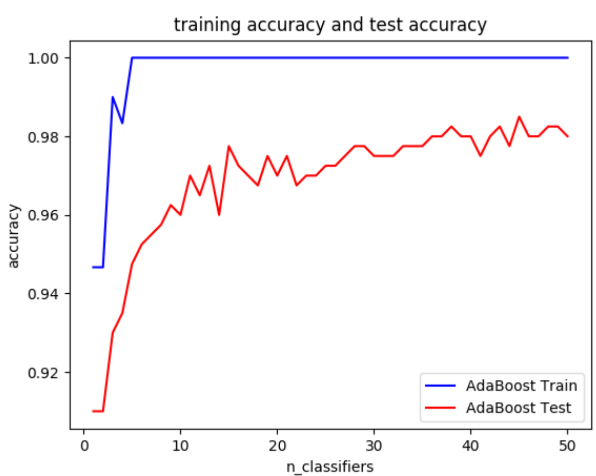

Figure 1. Prediction accuracy with increasing number of weak classifiers

C. Study of AdaBoost

In this part, study the behavior of AdaBoost model.

First, we compared the performance of the AdaBoost classifier with a raw decision tree classifier without depth limitation. As expected, the AdaBoost classifier can greatly improve the perform of a raw learning algorithm. In the face detection problem, AdaBoost classifiers achieve an accuracy about 98.0% on the test set, which is much higher than an ordinary Decision Tree Classifier (See Figure 2.).

Then, we changed the setting of weak classifier by changing the depth of the tree we used and make a comparison. We found out that the difference between different AdaBoost with different classifier weak classifier setting is little when they converge. But when the weak classifier become strong enough, it perform bad on the task. This maybe due to the strong fitting ability of the weak classifier itself, which lead to the zero error on the training data and break the boosting iteration.

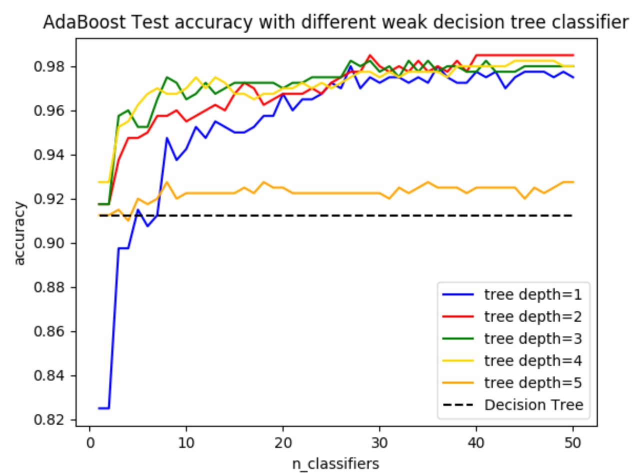

Figure 2. Comparison of different settings of AdaBoost

IV. CONCLUSION

We have implemented an AdaBoost classifier and demonstrated that the AdaBoost performs well on a face detection problem. By combining a sequence of decision trees, the AdaBoost classifier get a greatly improvement on prediction accuracy.