South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Jining He and Hui Han

Supervisor:
Qingyao Wu

Student ID:
201721045565 and 201721045572

Grade:
Graduate

December 22, 2017

Face Classification Based on AdaBoost Algorithm

***Abstract*—In this experiment, we use AdaBoost algorithm to classify whether a image is a face image. We use 1000 images including 500 human face images and 500 non-face images to train a AdaBoost model. The result shows that AdaBoost algorithm performs well in face classification problem.**

I. Introduction

With the development of artificial intelligence, face detection technology is more and more applied to life. In this experiment, we use Adaboost algorithm to implement a face detection model. We can use the model to determine whether the image is a face image. In the training process, we first extract the NPD features of the image, and then use the adaboost algorithm for training. After the training is completed, we use the validation set to evaluate the model. We hope that the model has a good performance on face image classification problem.

II. Methods and Theory

A. AdaBoost

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

Base learner

$$h_m(x){:}x \mapsto \{-1,1\} \tag{1}$$

Error rate

$$\epsilon_m = p(h_m(x_i) \neq y_i) = \sum_{i=1}^{n} w_m(i) \| (h_m|(x_i) \neq y_i) \tag{2}$$

$\epsilon_m > 0.5$, or the performance of Adaboost is weaker than random classfication.

Make the base learner with lower ϵ_m more important

$$\alpha_m = \frac{1}{2} \log \frac{1-\epsilon_m}{\epsilon_m} \tag{3}$$

Final learner

$$H(x) = sign(\sum_{m=1}^{M} \alpha_m h_m(x)) \tag{4}$$

$h_m(x) = sign(w^T x)$ is a nonlinear function, so the Adaboost can deal with nonlinear problem

The whole algorithm is described below.

Algorithm 1 AdaBoost

Input: $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, where $x_i \in X, y_i \in \{-1, 1\}$

Initizlize: Sample distribution ω_m

Base learner: Sample distribution Ł

$\omega_1(i) = \frac{1}{n}$

for $m = 1, 2, ..., M$ **do**

  $h_m(x) = L(D, \omega_m)$

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

  if $\epsilon_m > 0.5$ **then**

    break

  end if

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

  $\omega_{m+1}(i) = \frac{\omega_m(i)}{z_m} e^{-\alpha_m y_i h_m(x_i)}$,where $i = 1, 2, ..., n$ and $z_m = \sum_{i=1}^{n} \omega_m(i) e^{-\alpha_m y_i h_m(x_i)}$

end for

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

III. Experiments

A. Dataset

This experiment provides 1000 pictures, of which 500 are human face RGB images and the other 500 is a non-face RGB images.

B. Main Steps

1. Load data and converte images into a size of 24 * 24 grayscale. We label positive and negative classes as 1 and -1 respectively. And we also make the number of positive and negative samples equal.
2. Process datat to extract NPD features. Extract features using the NPDFeature class in feature.py.
3. Divide the data set into training set and validation set. The test_size we used is 0.25. We also make the number of number of positive and negative samples in training set and validation set equal.
4. Write all AdaboostClassifier functions based on the reserved interface in ensemble.py. The fit function in AdaboostClassifier class:
   (1) Initialize training set weights ω, each training sample is given the same weight.
   (2) Training a base classifier using DecisionTreeClassifier in sklearn.tree library. We pass the weight ω as a parameter and we set max_depth equals 1.
   (3) Calculate the classification error rate ϵ of the base classifier on the training set.
   (4) Calculate the parameter α according to the classification error rate ϵ.

   (5) Update training set weights ϵ.
   (6) Repeat steps from (2) to (5) 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 pickup the best model. Then use classification_report() of the sklearn.metrics library function writes predicted result to report.txt.
6. Organize the experiment results and complete the lab report.

C. Result

We save all the models and calculate accuracy in every iterations, plot as Fig. 1. We can find that when the number of weak classifiers reaches 14, the training set accuracy rate reaches 100%. And when the number of weak classifiers is 15, the verification set reaches the local maximum. So we pick the model when the number of weak classifiers is 15. And the classification report show as in Table I.

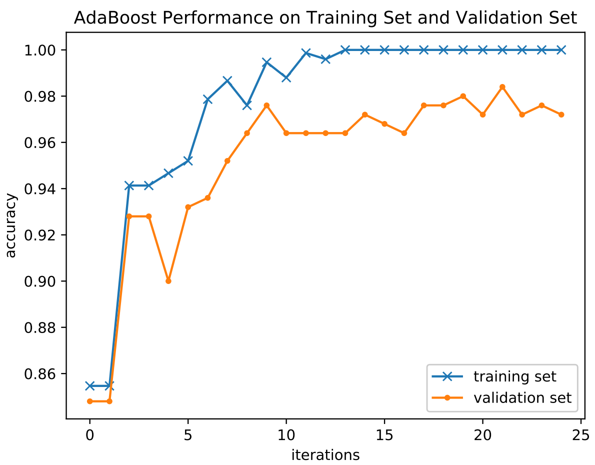

Fig. 1. AdaBoost performance on both training set and validation set.

TABLE I
FACE CLASSIFICATION REPORT

	precision	recall	f1-score	support
nonface	0.98	0.97	0.97	125
face	0.97	0.98	0.97	125
avg / total	0.97	0.97	0.97	250

IV. CONCLUSION

In this experiment, we implement a AdaBoost model to solve face classification problem. We train the model on training set and evaluate it on validation set. The result show that AdaBoost performs well in this problem. This process gives us a better understanding of the AdaBoost algorithm and how to combine the theory with the actual project. That let us experience the complete process of machine learning.