The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Yingting Huang and Doudou Yang
and Fangyuan Ding

Supervisor:
Qingyao Wu

Student ID：
201720145136 and 201720145129
201720144979

Grade:
Graduate

December 21, 2017

Face Classification Based on AdaBoost Algorithm

Abstract—To further understand of Adaboost and get familiar with the basic method of face detection, We conduct Adaboost to solve the face classification problem, and used 1000 pictures, of which 500 are human face RGB images to train and validate this Adaboost model. Additionally, we experienced the complete process of machine learning.

I. INTRODUCTION

The objective of face detection is to find and locate faces in an image.

II. METHODS AND THEORY

A. AdaBoost Algorithm

AdaBoost, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire. It is based on the idea of Boosting. The idea of Boosting is to make different assignments to the samples and set the weights of the samples for error learning to be larger. In this way, the difficult samples are concentratedly processed in subsequent learning and finally obtained a series of predictions, where each forecast has a weight, and a larger weight indicates that the forecast is better. AdaBoost is an iterative algorithm whose core idea is to train different weak learning algorithms for the same training set, and then combine these weak learning algorithms to construct a stronger final learning algorithm. And in some problems it can be less susceptible to the overfitting problem than other learning algorithms. 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.

Suppose we have a data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ where each item x_i has an associated class $y_i \in \{-1,1\}$ and a set of weak classifiers $g_1, \dots, g_m$,each of which outputs a classification $h_m(x_i) \in \{-1,1\}$ for each item. After the $T-1$th iteration our boosted classifier is a linear combination of the weak classifiers of the form:

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

The weighted error function is :

$$\varepsilon = \sum_{i=1}^{N} \omega_i I(h(x_i) \neq f(x_i))$$

The weight of weak classification α can be defined as :

$$\alpha = \frac{1}{2}\log\left(\frac{1-\varepsilon}{\varepsilon}\right)$$

and use normalization factor to normalize ω:

$$z_m = \sum_{i=1}^{N} \omega(x) e^{-\alpha f(x) h(x)}$$

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

The procedure of AbaBoost Algorithm is shown here:

Algorithm 1: Adaboost

Input: $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, **where** $x_i \in X, y_i \in \{-1, 1\}$
Initialize: Sample distribution w_m
Base learner: $\mathcal{L}$

$w_1(i) = \frac{1}{n}$
for m=1,2,...,M **do**
  $h_m(x) = \mathcal{L}(D, w_m)$
  $\epsilon_m = \sum_{i=1}^{n} w_m(i)\mathbb{I}(h_m(x_i) \neq y_i)$
  if $\epsilon_m > 0.5$ **then**
    break
  end
  $\alpha_m = \frac{1}{2}\log\frac{1-\epsilon_m}{\epsilon_m}$
  $w_{m+1}(i) = \frac{w_m(i)}{z_m} e^{-\alpha_m y_i h_m(x_i)}$, **where** $i = 1, 2, \dots, n$ **and** $z_m = \sum_{i=1}^{n} w_m(i) e^{-\alpha_m y_i h_m(x_i)}$
end
Output: $H(x) = \sum_{m=1}^{M} \alpha_m h_m(x)$

III. EXPERIMENT

A. Data sets and data analysis

This experiment provides 1000 pictures, of which 500 are human face RGB images and the rests are non-face RGB images. In the experiment, we label the face images as 1 and the non-face as -1 respectively.

B. Experimental steps

1. Main Frame:

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 NPD Feature class in feature.py.

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.

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* .

6)Organize the experiment results and complete the lab report (the lab report template will be included in the example repository).

2.The following is the guide of fit function in the Adaboost-Classifier class:

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

2)Training a base classifier , which can be sklearn.tree library DecisionTreeClassifier.

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

4) Calculate the parameter according to the classification error rate .

5) Update training set weights .

6) Repeat steps 2-6 above for iteration, the number of iterations is based on the number of classifiers.

C. Experimental results and curve:

1) Hyper-parameter selection

The number of weak classifiers is 40,and the depth of the decision trees is 2. Besides, the maximum number of iterations is set to 100000.

2) Loss Curve

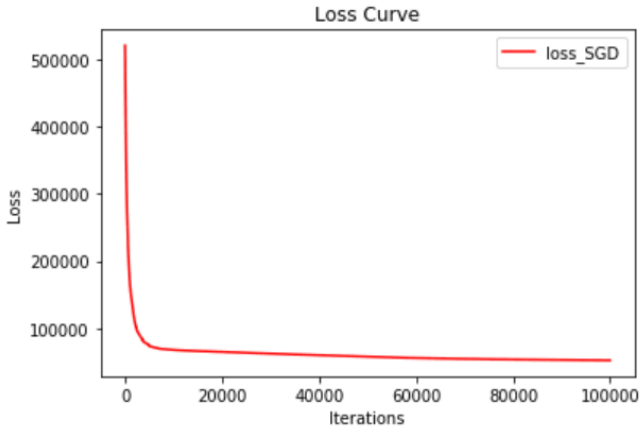

Fig 1

3) The accuracy

	precision	recall	f1-score	support
nonface	0.92	0.97	0.94	104
face	0.97	0.91	0.94	96
avg / total	0.94	0.94	0.94	200

Fig 2

F. Result analysis

From the loss curve (Fig 1), after 10000 times of iterations the loss converge descend, which means the models we have trained are robust.

From the accuracy on the validation set in Fig 2, we can see that the accuracies of face classification are high.

IV. CONCLUSION

In this experiment, we combined the machine learning theory with the actual project ,and implemented Adaboost algorithm to solve the face classification problem. Besides, we have noticed the important of learning from the former experience, which enables the Adaboost to boost the performance of the decision tree. Moreover we got a further understand of Adaboost. Finally, the experimental result shows that Adaboost do well in the face classification problem.