華南理工大學

South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:	*Supervisor:*
Peng Cheng	Mingkui Tan
Student ID:	*Grade:*
201830020255	Undergraduate

2020-11-22

Face Detection Based on Adaboost

Abstract—This paper uses Adaboost to solve the face detection problem. We first extract the features of the face and then build the base classifier, then iteratively adapt to the errors made by base learners in the previous iterations using Adaboost.

I. INTRODUCTION

This paper is intended to realize face classification and face detection. For face classification, aiming to distinguish human face and non-face pictures, we use a boosting algorithm to train a classifier which is capable of processing images rapidly while having high detection rates: Adaboost, which is an effective and classic method this paper applied. And for face detection, meaning that for any given image, a certain strategy is used to search it to determine whether it contains a face, and if it is, the position, size and posture of the face are returned, OpenCV's built-in method of face detection using Haar Feature-based Cascade Classifiers is adopted in this paper. And we can see that the experiment is highly effective and gain a high degree of accuracy

II. METHODS AND THEORY

In this section, we shall introduce the main idea and method we used in the experiment, include the background knowledge for Adaboost.

2.1 Ensemble learning

In the supervised learning algorithm of machine learning, our goal is to learn a stable model that performs well in all aspects, but the actual situation is often not so ideal. Sometimes we can only get multiple models with preferences (The weakly supervised model performs better in some aspects). Ensemble learning is to combine multiple weakly-supervised models here in order to obtain a better and more comprehensive strong-supervised model. The underlying idea of ensemble learning is that even if a certain weak classifier gets a wrong prediction, other weak classifiers can also be correct it back.

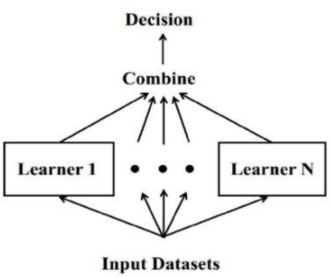

Figure. 1. Ensemble learning.

And there are mainly two types of ensemble learning algorithms for homogeneous learners: Boosting and Bagging. In our experiment, boosting is adopted, whose main idea is to assemble the weak classifier into a strong classifier. It trains a series of classifiers iteratively, and the sample selection method used by each classifier is related to the results of the previous round of learning.

AdaBoost was the first successful boosting algorithm developed for binary classification. Modern boosting methods build on AdaBoost, most notably stochastic gradient boosting machines. However, in this experiment we use the basic form of AdaBoost to solve the face classification problem.

2.2 Adaboost (Adaptive Boosting)

It is a learning model proposed by Freund and Schapiro on the basis of the PAC (Probably Approximately Correct) model. Its algorithmic idea is: through learning a large number of positive and negative samples, through learning feedback, weak classifiers without knowing the prior training error, the error rate and the corresponding weight are adaptively adjusted until the strong classifier reaches the predetermined performance.

AdaBoost solves the following two problems: First, how to choose a group of weak learners with different advantages and disadvantages so that they can make up for each other's deficiencies. Secondly, how to combine the output of weak learners to obtain better overall decision-making performance. For the first problem, re-weighting scheme is adopted, which assigns higher weight to incorrectly classified data point and lower to the correctly classified one. For the second problem, intuitively, intuitively, when we want to combine the judgments of each weak learner into the final prediction result, if the weak learner performs well in the previous task, we will believe it more, on the contrary, if the weak learner In the previous task, the performance is poor, we believe it less. In other words, we will combine weak learners in a weighted manner, and assign each weak learner a value indicating the degree of credibility α_t, which depends on its performance in the assigned task, the better the performance is, the bigger α_t will be.

And the detailed algorithm for Adaboost is as follows:

Algorithm: Adaboost

Input: Training set$D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$;weaking learning algorithm

Output: A strong classifier

Initialize the weight vector: $w_1(i) = \frac{1}{n}, i = 1,2,\dots,n$

for t=1,…,T **do**

fit base learner $h_t(x) = \{-1, +1\}$

Calculate the classification error rate e_t of $h_t(x)$

$$e_t = p(h_t(x) \neq y_i) = \sum_{i=1}^{N} w_t(i)\mathbb{I}(h_t(x) \neq y_i)$$

where $\mathbb{I}\{\cdot\}$ is the indicator function: $\mathbb{I}(X = x_i) = \begin{cases} 1, h_t(x) \neq y_i \\ 0, h_t(x) = y_i \end{cases}$

Calculate the weight α_t of $h_t(x)$

$$\alpha_t = \frac{1}{2} ln \frac{1 - e_t}{e_t}$$

Update the weights of each data point

$$w_{t+1}(i) = \frac{w_t(i)}{z_t} e^{-\alpha_t y_i h_t(x_i)}$$

$$where\ z_t = \sum_{i=1}^{n} w_t(i)\, e_t^{-\alpha_t y_i h_t(x_i)}$$

end

Output the final hypothesis: $H(x) = sign(\sum_{i=1}^{n} \alpha_t h_t(x))$

However, in this experiment, to simplify,

$$w_{t+1}(i) = \begin{cases} \frac{w_t(i)}{z_t} e^{-\alpha_t}, for\ the\ right\ predictive\ sample \\ \frac{w_t(i)}{z_t} e^{\alpha_t}, for\ the\ wrong\ predictive\ sample \end{cases}$$

III. EXPERIMENT

A. Dataset

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

B. Implementation

The implementation process of face classification is as follows:

1. Load data set. The images are converted into grayscale images with size of 24 *24, 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.
3. The data set is divided into training set and validation set, which comprises 75% and 25% of the raw data, respectively. And 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 , each training sample is given the same weight $\frac{1}{N}$, while N is the number of the samples.

4.2 Training a base classifier , which we use sklearn.tree library DecisionTreeClassifier.

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

4.4 Calculate the parameter according to the classification error rate .

4.5 Update training set weights .

4.6 Repeat steps 4.2-4.6 above for iteration, the number of iterations is based on the number of classifiers we set.

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 *classifier_report.txt*.

The parameters we choose in this experiment is in Table I.

Then results we get are shown in Fig.2 and Table Ⅱ.

TABLE I

MODEL PARAMETERS

Iteration number	maxIteration = 20
Sample number	sampleSize = 1000

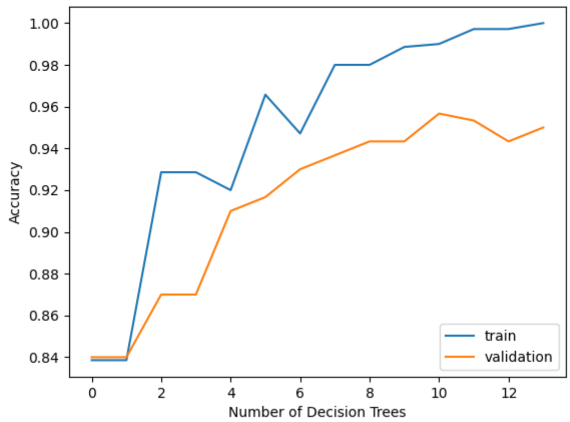

Figure. 2.The accuracy with number of decision trees

TABLE Ⅱ

CLASSIFIVATION REPORT

	precision	recall	f1-score	support
-1	0.96	0.94	0.95	150
1	0.94	0.96	0.95	150
accuracy			0.95	300
macro avg	0.95	0.95	0.95	300
weighted avg	0.95	0.95	0.95	300

The implementation process of face detection is as follows:

1. Run the *face_detection.py* file. Use the OpenCV's built-in method of face detection using Haar Feature-based Cascade Classifiers. The result will be save

as *detect_result.jpg*.

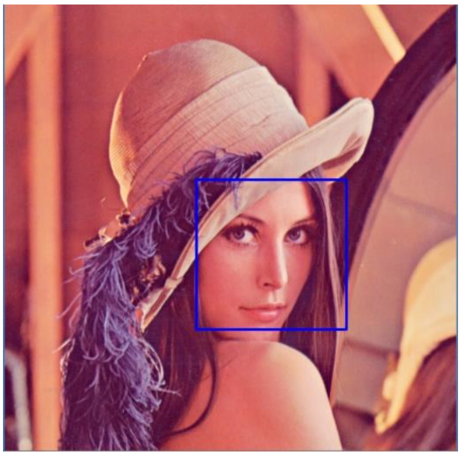

Figure. 3.The result of face detection

IV. CONCLUSION

4.1 Summary

As we can see from our experiment, Adaboost face detection algorithm has good recognition effect, fast detection speed, simple and efficient, and has high practicability. After a few iterations, the accuracy of the validation set can reach 0.95. Also, it requires little parameter tuning and it provides frame, which means we can use various classifiers under the frame.

4.2 Gains and Inspirations

As we increase the number of iterations, the accuracy of the face classification does not decrease, which means that there is not overfitting problem.