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:
Anran Lin

Supervisor:
Qingyao Wu

Student ID:
201630665045

Grade:
Undergraduate

November 17, 2018

Face Detection Based on AdaBoost Algorithm

***Abstract*—The experiment is a approach of face detection based in AdaBoost Algorithm.The purpose is to have further understand of the algorithm as well as show the result of a practice of the algorithm.**

I. Introduction

ENSEMBLE learning is a machine learning paradigm where multiple learners are trained to solve the same problem.The combination of multiple weak learner can be a strong learner.One of a main methods of ensemble learning is Boosting,where AdaBoost is the first practical boosting algorithm.The full name of AdaBoost is Adaptive Boosting algorithm,which focuses on classification problems and aims to convert a set of weak classifiers into a strong one.

II. Methods and Theory

A. Train the base learners

The additive model:

$$F_m(x) = F_{m-1}(x) + \alpha_m h_m(x) \tag{1}$$

Exponential loss function:

$$L(y, F_m(x)) = e^{-yF_m(x)}$$

The objective is to minimize the exponential loss on training data.Introduce the additive model into the loss:

$$\begin{aligned}
&\alpha_m, h_m(x) = arg\min_{\alpha,h} \sum_{i=1}^{N} e^{-y_i(F_{m-1}(x_i)+\alpha h(x_i))} \\
&\simeq arg\min_{\alpha,h} \sum_{i=1}^{N} e^{-y_iF_{m-1}(x_i)}(1 - y_i\alpha h(x_i) + \frac{y_i^2\alpha^2 h(x_i)^2}{2}) \\
&= arg\min_{\alpha,h} \sum_{i=1}^{N} e^{-y_iF_{m-1}(x_i)}(1 - y_i\alpha h(x_i) + \frac{\alpha^2}{2})
\end{aligned} \tag{2}$$

since $y_i^2 = h(x)^2 = 1$

Let $\omega_{(m,i)} = e^{-y_iF_{m-1}(x_i)}$ denote the distribution weight of samples, which is irrelevant with either α and $h(x)$
Optimize the latter term to train base learners:

$$h_m(x) = arg\min_{h} \sum_{i=1}^{N} [1 - y_i\alpha h(x_i) + \frac{\alpha^2}{2}] \tag{3}$$

Fixing the α,this is equivalent to optimizing:

$$h_m(x) = arg\min_{h} \sum_{i=1}^{N} [y_i h(x_i)] \tag{4}$$

where $y_ih(x_i) = -1$when it is wrong,or $y_ih(x_i) = 1$.
So the solution is to train the base learner by maximizing the margin $y_ih(x_i)$

B. Combine the base learners

Based on equation(1),it is seeking for α_m the only task to obtain the new stronger learner $h_m(x)$
The weighted exponential loss at current round:

$$\begin{aligned}
&L(y, h_m(x)) = \sum_{i=1}^{N} [\omega_{(m,i)} e^{-y_i\alpha_m h_m(x_i)}] \\
&= \sum_{i=1}^{N} [\omega_{(m,i)} e^{-\alpha_m} * \mathbb{I}(y_i = h_m(x)) + \omega_{(m,i)} e^{\alpha_m} * \mathbb{I}(y \neq h_m(x))] \\
&= e^{\alpha_m}(1 - \epsilon_m) + e^{\alpha_m}\epsilon_m
\end{aligned} \tag{5}$$

where $\epsilon_m = \sum_{i=1}^{N} \omega_{(m,i)} \mathbb{I}(y_i \neq h_m(x_i)$
The derivation of this loss:

$$\frac{\partial L(y, h_m(x))}{\partial \alpha_m} = -e^{\alpha_m}(1 - \epsilon_m) + e^{\alpha_m}\epsilon_m$$

Setting the derivation as zero will give:

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

So the solution is to combine the learner $h_m(x)$ by the importance weight α_m

C. update data distribution

The distribution weight of sample:

$$\omega_{(m,i)} = e^{-y_iF_{m-1}(x_i)} \tag{6}$$

Based on equation(1) and equation(6):

$$\begin{aligned}
\omega_{(m+1,i)} &= e^{-y_iF_m(x_i)} \\
&= e^{-y_i(F_{m-1}(x_i)+\alpha_m h_m(x_i))} \\
&= e^{-y_iF_{m-1}(x_i)} * e^{-y_i\alpha_m h_m(x_i)} \\
&= \omega_{(m,i)} e^{-y_i\alpha_m h_m(x_i)}
\end{aligned} \tag{7}$$

Final update equation:

$$\omega_{(m+1,i)} = \frac{\omega_{(m,i)} e^{-y_i\alpha_m h_m(x_i)}}{z_m} \tag{8}$$

where $z_m = \sum_{i=1}^{N} \omega_{(m,i)} e^{-y_i\alpha_m h_m(x_i)}$ aims to renormalization

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 are non-face RGB images, stored in datasets/original/nonface.

B. Implementation

1) Face Classification:NPDFeature:

a) initialization and parameters: Load data set data. The images are supposed to converted into grayscale images with size of 24 * 24.Set the label of face image of 1 and -1 otherwise.

b) process: Processing data set data to extract NPD features. Extract features using the NPDFeature class in feature.py.Obtain the feature set of all images.Split the training set and validation set,choosing the validation set of 0.3 size of the original set.
initialize a AdaBoostClassifier using the given file ensemble.py and use the fit() function. The steps of fit function:
first,Initialize training set weights ω, each training sample is given the same weight.Second, Training a base classifier , which can be sklearn.tree library DecisionTreeClassifier.Tired,Calculate the classification error rate ϵ of the base classifier on the training set. Forth,Calculate the parameter α according to the classification error rate ϵ . Fifth,update training set weights ω.And then repeat the second to the fifth steps above for iteration,the number of iterations is set to 20,which is the number of classifier.

c) result: Predict and verify the accuracy on the validation set.The accuracy is 0.9929 of the training set and 0.96 of the validation set,see Fig.1.
And use classification_report () of the sklearn.metrics library function writes predicted result to classifier_report.txt ,see TABLE 1.

TABLE I
PREDICTED RESULT

Label	Precision	Recall	f1-score	Support
-1	0.97	0.95	0.96	146
1	0.95	0.97	0.96	154
avg/total	0.96	0.96	0.96	300

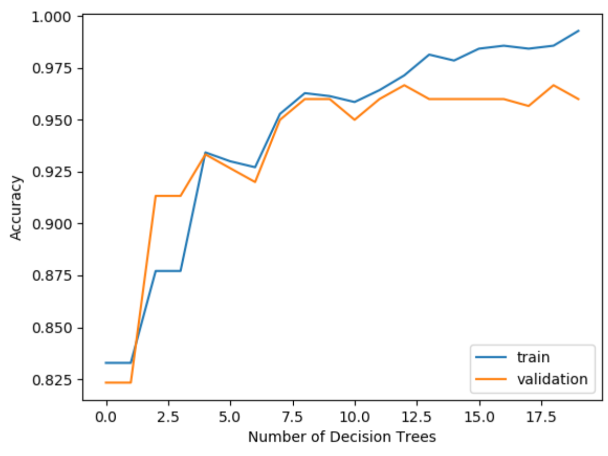

Fig. 1. The Accuracy of training set and validation set

2) Face Detection:OpenCV: Run the face_detection.py file. Experience the OpenCV's built-in method of face detection using Haar Feature-based Cascade Classifiers.The result will be save as detect_result.jpg ,as shown in Fig.2.

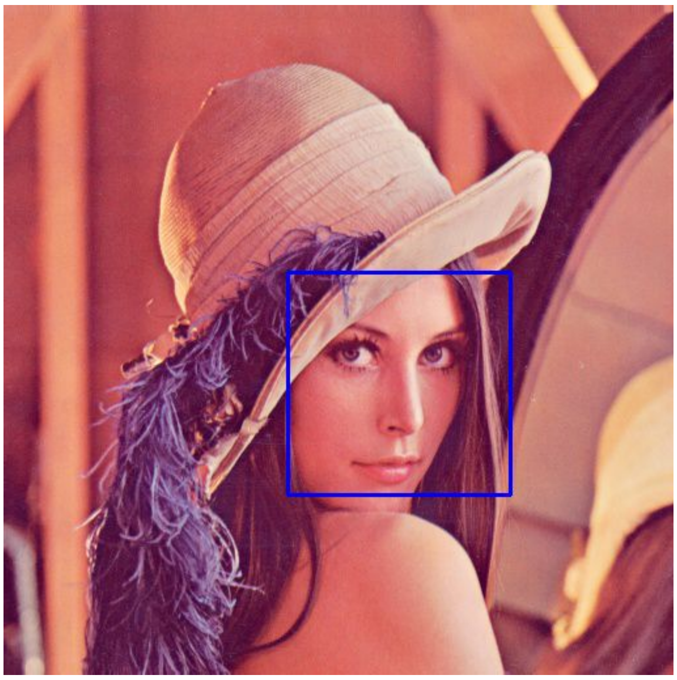

Fig. 2. The result of face_detection.py

IV. Conclusion

Through this experiment,I got a further understand of Adaboost Algorithm as well as experience the complete process of machine learning.The experiment result is also satisfied.