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:
Anni Chen and Tianxin Cai and Shuyi Zhen

Supervisor:
Qingyao Wu

Student ID：
201530611142 and 201530611067 and
201530613771

Grade:
Undergraduate or Graduate

December 9, 2017

Face Classification Based on AdaBoost Algorithm

Abstract—
In this experiment, we read 1000 pictures, of which 500 are human face RGB images, the other 500 is a non-face RGB images, and change all of them into grayscale with the size 24 * 24 and label them, then extract features. Using the DecisionTreeClassifier as the base classifier to train a series of new weakly classifier, and the weak classifier is combined into a strong classifier, which is named AdaBoost Algorithm. Then, we use AdaBoost algorithm to solve the problem of face classification and get the good classification accuracy.

I. INTRODUCTION

A. Motivation of Experiment

1.Understand Adaboost further.
2.Get familiar with the basic method of face detection.
3.Learn to use Adaboost to solve the face classification problem, and combine the theory with the actual project.
4.Experience the complete process of machine learning.

B. Dataset

This experiment provides 1000 pictures, of which 500 are human face RGB images, stored in datasets/original/face; the other 500 is a non-face RGB images, stored in datasets/original/nonface. Use open() function from Image to read the image and change all of them into grayscale with the size 24 * 24. The face image were labeled as positive samples (+ 1) and the non-face image were labeled as negative samples (-1), then extract features using the NPD Feature class in feature.py. Because the time of the pretreatment is relatively long, it can be pretreated with pickle function library dump () save the data in the cache, then may be used load () function reads the characteristic data from cache. The data sets were randomly divided into 67% of training set and 33% of the validation set by train_test_split() function.

C. Experiment Step

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 NPDFeature class in feature.py. (Tip: Because the time of the pretreatment is relatively long, it can be pretreated with pickle function library dump () save the data in the cache, then may be used load () function reads the characteristic data from cache.)
3.The data set is divisded into training set and calidation set, 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 ω, each training sample is given the same weight.
4.2Training a base classifier , which can be sklearn.tree library DecisionTreeClassifier (note that the training time you need to pass the weight ω as a parameter).
4.3 Calculate the classification error rate ε of the base classifier on the training set.
4.4 Calculate the parameter α according to the classification error rate ε.
4.5 Update training set weights ω.
4.6 Repeat steps 4.2-4.6 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 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).

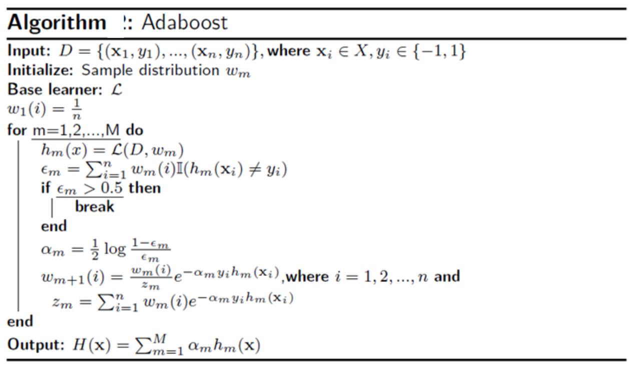
II. METHODS AND THEORY

A. *Description of the Adaboost*

AdaBoost, "the Adaptive Boosting" (Adaptive enhancement), was put forward by Yoav Freund and Robert Schapire in 1995. Its self-adaptation is that the sample of the previous basic classifier is strengthened and the weighted group of samples is used again to train the next basic classifier. At the same time, a new weakly classifier is added to each round until it reaches a predetermined error rate or the maximum number of iterations specified.

Specifically, the entire Adaboost iteration algorithm is 3 steps:
1. The weight distribution of the initial training data. If there are N samples, each of the training samples will initially be given the same weight: 1/ N.
2. Training weak classifier. In the specific training process, if a sample point has been accurately classified, then the weight of the next training set will be reduced. Conversely, if a sample point is not classified correctly, its weight is improved. The updated sample set is then used to train the next classifier, and the whole training process goes on so iteratively.
3. Combine the weak classifier from training to strong classifier. Each weak classifier after the training process, increase the weight of weak classifier classification error rate is small, in the final classification function plays a decisive role, and reduce classification error rate of the weak classifier weights, in the final classification function plays a smaller decision role. In other words, the weak classifier with low error rate occupies a large weight in the final classifier, otherwise it is small.

B. *AdaBoost Algorithm*



```
Algorithm  : Adaboost
Input: $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in X, y_i \in \{-1, 1\}$
Initialize: Sample distribution $w_m$
Base learner: $\mathcal{L}$
$w_1(i) = \frac{1}{n}$
for m=1,2,...,M do
    $h_m(x) = \mathcal{L}(D, w_m)$
    $\epsilon_m = \sum_{i=1}^{n} w_m(i)\mathbb{I}(h_m(\mathbf{x}_i) \neq y_i)$
    if $\epsilon_m > 0.5$ then
        break
    end
    $\alpha_m = \frac{1}{2}\log\frac{1-\epsilon_m}{\epsilon_m}$
    $w_{m+1}(i) = \frac{w_m(i)}{z_m} e^{-\alpha_m y_i h_m(\mathbf{x}_i)}$, where $i = 1, 2, ..., n$ and
      $z_m = \sum_{i=1}^{n} w_m(i) e^{-\alpha_m y_i h_m(\mathbf{x}_i)}$
end
Output: $H(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m h_m(\mathbf{x})$
```

Initialize training set weights ω :

$$w_1(i) = \frac{1}{n}$$

Weights ω :

$$w_{m+1}(i) = \frac{w_m(i)}{z_m} e^{-\alpha_m y_i h_m(\mathbf{x}_i)}$$

Normalization Z:

$$z_m = \sum_{i=1}^{n} w_m(i) e^{-\alpha_m y_i h_m(\mathbf{x}_i)}$$

Base learner:

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

Error rate:

$$\epsilon_m = p(h_m(\mathbf{x}_i) \neq y_i) = \sum_{i=1}^{n} w_m(i)\mathbb{I}(h_m(\mathbf{x}_i) \neq y_i)$$

ϵm<0.5, or the performance of Adaboost is weaker than random classfication.

Parameter α :

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

Final learner:

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

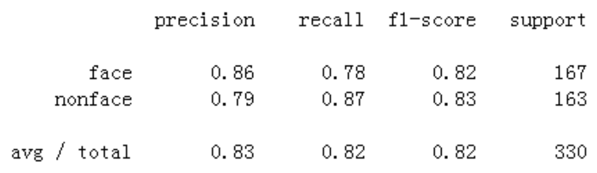
s.t. $h_m(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x})$

III. EXPERIMENT

A. *Super parameter selection (weak classifier number, decision tree depth, etc.) :*

n_weakers_limit=10，
max_depth=3，
test_size=0.33,
random_state=0

B. *Prediction results (best results) :*

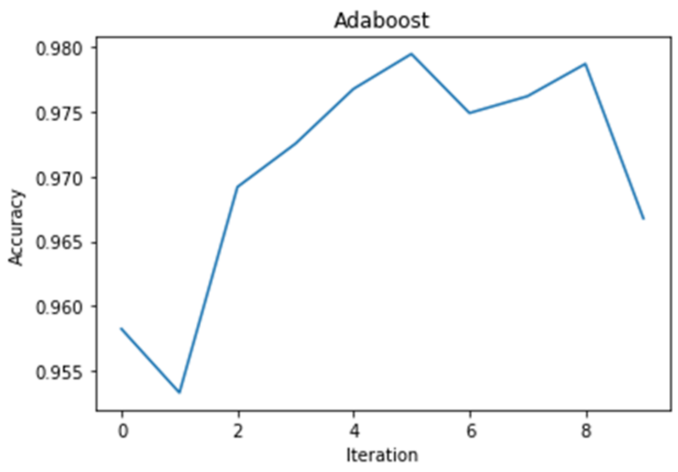


```
             precision    recall  f1-score   support

       face       0.86      0.78      0.82       167
    nonface       0.79      0.87      0.83       163

avg / total       0.83      0.82      0.82       330
```

C. *Accuracy curve:*



IV. CONCLUSION

The results of the experiment show that:
1. The larger the training set and the more the training samples, the better the classification effect of the strong classifier and the higher the classification accuracy.
2. In the data concentration of this experiment, the number of the optimal weak classifiers is 10, the decision tree depth is 3. The number of weak classifiers is too low, the accuracy of the classifier is low, and the number of classifiers is too large, which can easily result in overfitting, and the accuracy of classifier will decrease. Similarly, the depth of the decision tree is most suitable for 3.
3. Since the classification of training set and verification set is random, the results of the classifier's test results are also uncertain, but the test results are relatively concentrated and not volatile.