The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERIN
G

Author:
Feiqin Huang and Yuebo Guo
and Wanting Su

Supervisor:
Qingyao Wu

Student ID：
201530611678and 201530611517
and 201530612767

Grade:
Undergraduate

December 9, 2017

Face Classification Based on AdaBoost Algorithm

Abstract—In this experiment we use Adaboost to solve the face classification problem and combine the theory with the actual project. From this experiment we understand adaboost further. We also get familiar with the basic method of face detection and experience the complete process of machine learning.

I. INTRODUCTION

AdaBoost(Adaptive Boosting) is a machine learning meta-algorithm 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.

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. It can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class.

NPD(Normalized Pixel Difference) is an unconstrained face detection method.It can execute parallel processing based on cpu multi-threaded OpenMP, pthread as well as based on GPU CUDA parallel processing. Obtained through a look up table, NPD feature avoiding each recalculation and speeding up program preprocessing.

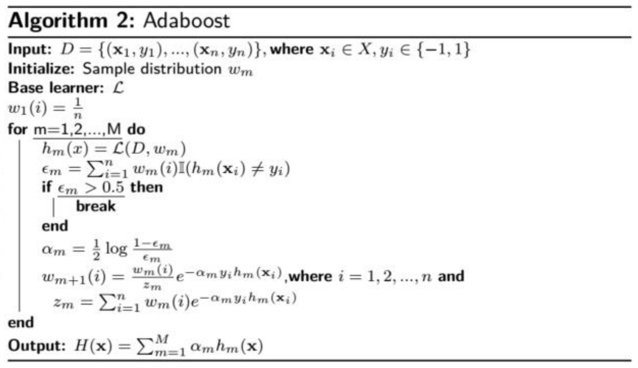
In this experiment we implement face classification by using Adaboost.

II. METHODS AND THEORY

A. Adaboost

In this section, we implement an AdaBoost algorithm,which is the most representational algorithm in boosting method.

To illustrate the process of Adaboost, we list the whole process as figure 1.



```
Algorithm 2: 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: $H(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m h_m(\mathbf{x})$
```

Figure 1: AdaBoost Algorithm

In this method, we increase the weights of sub-samples by reducing the weights of the sample pairs in each round, so that the classifier improves gradually in the iterative process.

Finally, all the classifiers are linearly combined to obtain the final classifier

B. NPD(Normalized Pixel Difference)

In this section, we extract NPD feature from processing data set data. NPD feature is to detect the difference between two pixels.We define the difference as function f(x,y):

$$f(x, y) = \frac{x - y}{x + y}$$

where x and y denote the pixel value of any two pixels.

Since that the function f(x,y) owning the anti-symmetry properties on x and y (only symbol will change in the function if we change the position of x and y), it is enough for us to calculate once.

Then we can draw a conclusion that, for a window that has p pixels, we have: $\frac{p(p-1)}{2}$ features.

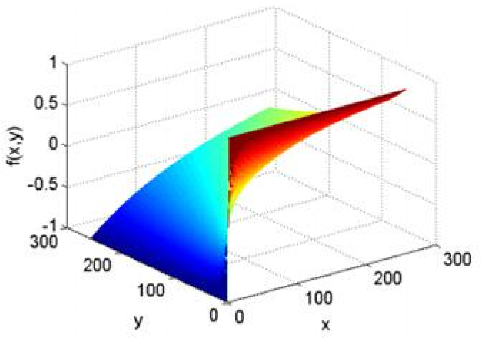


Figure 2: Function f(x,y) figure

III. EXPERIMENT

A. Dataset

This experiment provides 1000 pictures, of which 500 are human face RGB images, the other 500 is a non-face RGB images. We use 800 pictures as training set and 200 pictures as verification set. And we label the human face images with label '+1', and the non-face images with label '-1'.

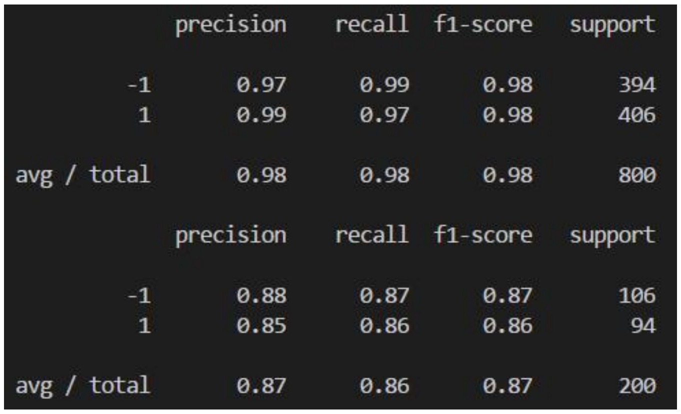
B.Implementation

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 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. 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.
   4.2Training a base classifier, which can be sklearn.tree library DecisionTreeClassifier (note that the training time you need to pass the weight as a parameter).
   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.
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 .

C.Experiment result

In the experiment, we use DecisionTreeClassifier which from sklearn.tree library as the base classifier. And we set max_depth = 1. We try to increase the parameter but the effect is not significant.

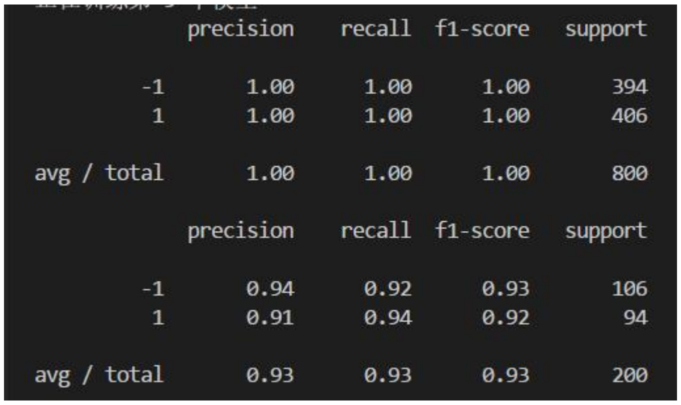
From the very beginning, we try to use only one base classifier with an accuracy of about 0.87.



```
             precision    recall  f1-score   support

         -1       0.97      0.99      0.98       394
          1       0.99      0.97      0.98       406

avg / total       0.98      0.98      0.98       800
             precision    recall  f1-score   support

         -1       0.88      0.87      0.87       106
          1       0.85      0.86      0.86        94

avg / total       0.87      0.86      0.87       200
```

Figure 3: Result of Using One Base Classifier

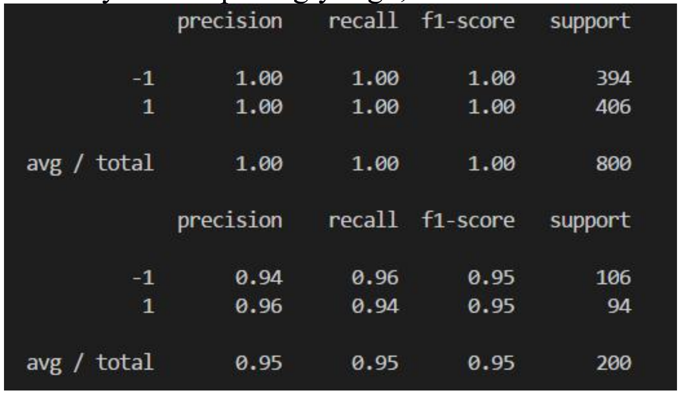
Then, we set the number of iterations to 10 (that is, the number of basis classifiers) to achieve a prediction accuracy of 0.93



```
             precision    recall  f1-score   support

         -1       1.00      1.00      1.00       394
          1       1.00      1.00      1.00       406

avg / total       1.00      1.00      1.00       800

             precision    recall  f1-score   support

         -1       0.94      0.92      0.93       106
          1       0.91      0.94      0.92        94

avg / total       0.93      0.93      0.93       200
```

Figure 4: Result of Iterating ten Times

After the iteration times adjusted to 10, the accuracy rate surprisingly high, about 0.95.



```
             precision    recall  f1-score   support

         -1       1.00      1.00      1.00       394
          1       1.00      1.00      1.00       406

avg / total       1.00      1.00      1.00       800

             precision    recall  f1-score   support

         -1       0.94      0.96      0.95       106
          1       0.96      0.94      0.95        94

avg / total       0.95      0.95      0.95       200
```

Figure 5: Result of Adjusting Iteration Times

As the number of iterations (that is, the number of base classifiers) increases, the accuracy of the model trained by Adaboost algorithm is also getting higher and higher. For each additional classifier, the loss function will be reduced, that is, the deviation is getting smaller and smaller, the variance is getting bigger and bigger. Adaboost is not easy to overfit.

IV. CONCLUSION

In this experiment we implement AdaBoost algorithm and extracting NPD feature, solving face detection problem by using Adaboost method. Through the experiment we find that with the increase of times of iteration,the accuracy will be improved apparently. We now can have a clear understanding of Adaboost and methods of face detection, combine the theory with actual project.The detail of face detection is worth further investigating and we will study deeply in the coming research .