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:
黄启琛 陈星宇 张华奎

Supervisor:
Mingkui Tan

Student ID：
201530611777 201530611289
201530613610

Grade:
Undergraduate

December 14, 2017

Face Classification Based on AdaBoost Algorithm

Abstract— Face Classification is a popular application of machine learning and deep learning. In this paper, we try to explore the detail of face classification. Based on the power of Adaboost algorithm, our trials are based on Adaboost algorithm. On the other hand, we also try to discover the performance of Adaboost algorithm.

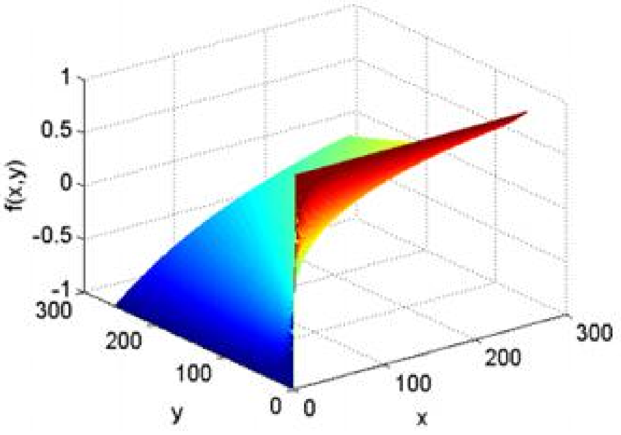
1. INTRODUCTION

When applying machine learning algorithms into image classification, an crucial problem need to be resolved is the feature extraction. As we known, before put into the model, the features of those images need to be extracted, whose quality usually influence the final result.

In our face classification, we apply a new image feature called Normalized Pixel Difference (NPD), proposed in [1]. NPD feature is computed as the difference to sum ratio between two pixel values, inspired by the Weber Fraction in experimental psychology. The new feature is scale invariant, bounded, and is able to reconstruct the original image.

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

NPD function



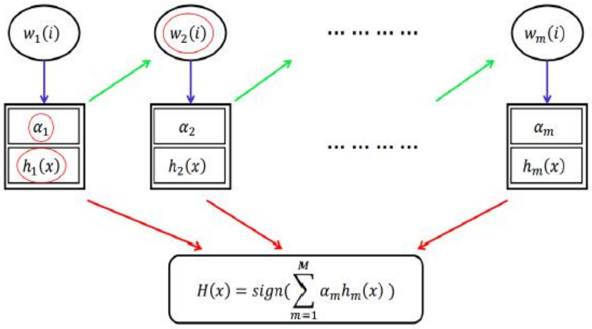
plot of NPD function

Furthermore, in our experiment ,we show that the NPD features can be efficiently obtained from a look up table, in which $V_{i,j} = \frac{i-j}{i+j}$, given i and j.

Adaboost is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire. It can be 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.

2. METHODS AND THEORY

Adaboost is a kind of ensemble learning, which combine numerous weak learners to a strong learner. The essence of adaboost is to make wrong predictive samples in this iteration more important when handling in next iteration. Every iteration generates a new base learner $h_m(x)$ and its importance score α_m , and updates the sample distribution $w_m(i)$.



Input: Datasets (x_i, y_i),..., (x_n, y_n) where $x_i \in X$, $y_i \in \{-1,1\}$

Initialize: Sample weight distribution $w_i = \frac{1}{n}$,where i = 1,2,..., n

error rate: $\boldsymbol{\epsilon_m = p(h_m(x_i) \neq y_i) = \sum_{i=1}^{n} w_m(i) I(h_m(x_i) \neq y_i)}$

importance score:

$$\boldsymbol{\alpha_m = \frac{1}{2}\ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)}$$

Weight updating formula (sample distribution):

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

where $\boldsymbol{z_m = \sum_{i=1}^{m} w_m(i) e^{-\alpha_m y_i h_m(x_i)}}$

Final learner:

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

3. EXPERIMENT

The dataset of this experiment is 1000 pictures, of which 500 are human face RGB images, the other 500 is a non-face RGB images.

In this experiment, the objective of adaboost algorithm is to classify whether the picture contains a human face or not, using NPD feature of images as mentioned above.

experiment step

1. Read data set data. Split dataset into training set,with 400 face pictures and 400 non-face pictures,and validation set,with 100 face pictures and 100 non-face pictures. The images are converted into a size of 24 * 24 grayscale
2. Processing data set data to extract NPD features. Extracted features can be prehandle with pickle function library dump () save the data in the cache, then can be used load () function reads the characteristic data from cache, efficiently.

3. Train the AdaboostClassifier with training set and validation set
4. 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 .

In our experiment code, the most important part is the class AdaboostClassifier. Class AdaboostClassifier represent the adaboost model, which can take image features as input and output the final result of classification.

The fit function in the AdaboostClassifier class acts as the training function. Here is the procedures in fit function.

1.Initialize training set weights $\mathbf{w_0(i)}$, each training sample is given the same weight $\frac{1}{n}$.

2.Training a base classifier . we use sklearn.tree library DecisionTreeClassifier as the “weak learner”

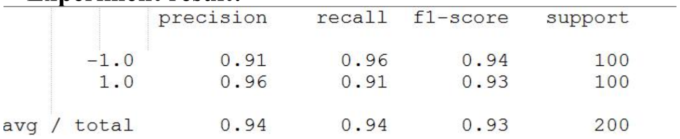
3. Calculate the classification error rate $\boldsymbol{\epsilon_m}$ of the base classifier on the training set.

4. Calculate the importance scores $\boldsymbol{\alpha_m}$ according to the classification error rate $\boldsymbol{\epsilon_m}$.

5. Update training set weights .

6. Repeat steps 2-5 above for iteration, the number of iterations is based on the number of classifiers.

Experiment result:



```
             precision    recall  f1-score   support

       -1.0       0.91      0.96      0.94       100
        1.0       0.96      0.91      0.93       100

avg / total       0.94      0.94      0.93       200
```

result output in report.txt

The final adaboost model get a high accuracy in validation set up to 94%.

4. CONCLUSION

Adaboost is a powerful model. By combining “weak learner”, it can easily obtain a high accuracy with weighted sum of the output of those “weak learner”. However, before computed by adaboost, the images need to be extracted totally, which takes a long time because of such huge amount of data. Also, training is another time-consume process, even we have resize those pictures into smaller pictures.