华南理工大学

South China University of Technology

The Experiment Report of Machine Learning

School: School of Software Engineering

Subject: Software Engineering

Author:
ShiWangHua

Supervisor:
Mingkui Tan or Qingyao Wu

Student ID:
201630676843

Grade:
Undergraduate

December 22, 2018

Face Detection Based on AdaBoost Algorithm

Abstract—In this experiment, there are two parts about face detection based on AdaBoost Algorithm. Our main task is among the first step which require to achieve the adaboost algorithm with the help of sklearn.tree library called DecisionTreeClassifier. Besides, preprocesssing of the data including extracting features from 1000 pictures is no small or negligible task.

I. Introduction

A. Problems

The main problems to be solved within my reserarch are shown as below:

1) Extract NPD features correctly
2) Achieve Adaboost by aid of DecisionTreeClassifier
3) Adjust model, update parameters and improve the accuracy of the prediction

B. Motivations

The motivations of this experiment are shown below:

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

C. Expectation

My expectation of this experiment have three points:

1) Extract the features and save them to the local file
2) Use adaboost algorithm and get an accuracy rate more than 0.9 on the given data set.
3) Use OpenCV's build-in method based on Haar feature to achieve face detection

II. Methods and Theory

A. Extract NPD features

We can use this expression to create a NPD table(256*256):

$$\boldsymbol{table}[i][j] = (i-j)/(i+j) \tag{1}$$

The table is initialized to zero before calculating the formula above.

Then we can extract NPD features by this formula:

$$\boldsymbol{features}[k] = \boldsymbol{table}[\boldsymbol{image}[i]][\boldsymbol{image}[j]] \tag{2}$$

In this formula, the $\boldsymbol{features}$ is a one dimension array(1*165600) in which the 165600 menans 576 plus 575 then divide by 2 and the 576 means the number of image's pixels. The $\boldsymbol{image}$ is a one dimension array storing the value of pixels.

B. Calculate the error rate $\boldsymbol{\epsilon}$

Error rate $\boldsymbol{\epsilon}$ is used to evaluate the accuracy of the output of weak classifiers. For any $\boldsymbol{y}$ predicted by weak classifier which is not equal to the actual $\boldsymbol{y}$ value, we can get a cumulative sum of all the $\boldsymbol{w}$ (a w value means the weight of a sample) in corresponding $\boldsymbol{y}$'s position. Then we use the cumulative sum as error rate $\boldsymbol{\epsilon}$. This calculation can be expressed in mathmatical formula as follows:

$$\boldsymbol{\epsilon}_m = \sum_{i=1}^{N} \boldsymbol{w}_{(\mathrm{m,i})} II(y_i \neq h_m(x_i)) \tag{3}$$

In this formula, the m means the number of weak classifier, N means the number of samples whcih equal to 1000, and $h_m(x_i)$ means the predict value $\boldsymbol{y}$. If the prediction is wrong, $II(y_i \neq h_m(x_i))$ is equal to 1.

C. Update weak classifier's weight $\boldsymbol{\alpha}$

To calculate α, we need to calculate the derivation of the weighted exponential loss and set the derivation as zero. The final loss we work out is:

$$\boldsymbol{L}(\boldsymbol{y}, \boldsymbol{h}_m(\boldsymbol{x})) = e^{-\alpha_m}(1-\epsilon_m) + e_m^{\alpha}\epsilon_m \tag{4}$$

Calculate the derivation of (4):

$$\frac{\partial \boldsymbol{L}(\boldsymbol{y}, \boldsymbol{h}_m(\boldsymbol{x}))}{\partial \alpha_m} = -e^{-\alpha_m}(1-\epsilon_m) + e^{\alpha_m}\epsilon_m \tag{5}$$

Set (5) as zero, we finally get the expression of α_m:

$$\alpha_m = \frac{1}{2} log_2 \frac{1-\epsilon_m}{\epsilon_m} \tag{6}$$

D. Update sample weight $\boldsymbol{w}$

Based on $\boldsymbol{F}_m(x) = \boldsymbol{F}_{m-1}(x) + \alpha_m \boldsymbol{h}_m(x)$, we can finally get the update equationof w:

$$w_{(m+1,i)} = \frac{w_{(m,i)} e^{-y_i \alpha_m h_m(x_i)}}{z_m} \tag{7}$$

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

III. Experiments

A. Dataset

1) 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.
2) The dataset is included in the example repository. We need to divide it into training set and validation set.

B. Implementation

- Face Classification
  1) Load data set and read them into memory. Use ***Image.open.convert*** to convert the images to grayscale images with size of 24 * 24.
  2) Process data set: extract NPD features using the NPDFeature class in feature.py.
  3) Use ***sklearn.model_selection.train_test_split*** to divide the data set into training set and validation set.
  4) Write all AdaBoostClassifier functions based on the reserved interface in ensemble.py according to part II.
  5) Predict and verify the accuracy on the validation set using the method in AdaBoostClassifier. Use classification_report() of the ***sklearn.metrics*** library function to write predicted result to ***classifier_report.txt***.
- Face Detection
  1) 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.
  2) Use other images to replace the default test image.

IV. Conclusion

The ouput is shown in Fig. 1. We can see the accuracy is about 90%.

```
accuracy on the validation: 0.895
             precision    recall  f1-score   support

    class 1       0.88      0.89      0.89        91
   class -1       0.91      0.90      0.90       109

avg / total       0.90      0.90      0.90       200
```

Fig. 1. Classification Report of Face Detection

By this expriment, I learned a lot about machine learning and experience an complete process of machine learning. Besides, this is my first time to write an experiment report using latex, which makes me gain another useful skill for my personal dedvelopment. Thanks to teacher Mr.Tan and assistants a lot !