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:

饶浩聪. 闫明远 and 谢家峻

Student ID:

201530612644 201530613320 201530613191

Supervisor:

Qingyao Wu

Grade:

Undergraduate

December 10, 2017

Face Classification Based on AdaBoost Algorithm

Abstract—

I. INTRODUCTION

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

the principle of Adaboost is briefly described.

Given a training set of data T={(X1, Y1), (X2, Y2)... (XN, yN)}, Yi belongs to the tag set {-1, +1},

(Xi is the NPD characteristic one-dimensional matrix of the pixels of the processed 24*24 grayscale map, and the size is (24*24) * (24*24-1) /2 =165600)

: NPD feature is to detect the difference between the 2 pixels of (somewhat similar to Viola Johns), the difference is defined as a function of F (x, y). In this, X and y are pixel values of any two pixels. And specify f (0,0) =0.

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

The purpose of Adaboost is to learn from the training data of a series of weak classifiers or basic classifier,

then these weak Adaboost algorithm process is as follows:

Step 1: first, the weight distribution of the training data initialization. Each training sample is given the same weight at the beginning: 1/N.

$$D_1=(w_{11},w_{12}\cdots w_{1i}\cdots,w_{1N}),\ w_{1i}=\frac{1}{N},\ i=1,2,\cdots,N$$

Step 2:. conducted several rounds of iteration, with M = 1,2,..., M said the number of rounds of iteration

A: using training data set with weight distribution Dm learning, get the basic classifier (select let the lowest error rate threshold to design basic classifier):

$$G_m(x):\ \chi\rightarrow\{-1,+1\}$$

B.: calculation of Gm (x) classification error on the training data set on the rate

$$e_m = P\left(G_m\left(x_i\right) \neq y_i\right) = \sum_{i=1}^{N} w_{mi} I\left(G_m\left(x_i\right) \neq y_i\right)$$

From the above formula shows that Gm (x) error in the training data set on the rate of EM is Gm (x) and the weight of the sample classification error.

C. calculates the coefficients of Gm (x), and am represents the importance of Gm (x) in the final classifier (purpose: get the weight of the basic classifier in the final classifier):

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

From the above formula shows that the EM of 1/2, am and am with EM > = 0, decreases with the increase means that the basic classification error rate is smaller in the final classifier in the greater.

D. updates the weight distribution of the training dataset (Objective: to get the new weight distribution of the sample) for the next round of iterations

$$D_{m+1} = \left(w_{m+1,1}, w_{m+1,2} \cdots w_{m+1,i} \cdots, w_{m+1,N}\right),$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \exp\left(-\alpha_m y_i G_m\left(x_i\right)\right), \quad i = 1,2,\cdots,N$$

The weight value of the miss classified sample by the basic classifier Gm (x) is increased, and the weight of the correct classification sample is reduced. So, in this way, the Adaboost method can "focus" or "focusing on" those with other samples.

Among them, Zm is a standardized factor, make Dm+1 a probability distribution:

$$Z_m = \sum_{i=1}^{N} w_{mi} \exp\left(-\alpha_m y_i G_m\left(x_i\right)\right)$$

Step 3: combination of each weak classifier:

$$f\left(x\right) = \sum_{m=1}^{M} \alpha_m G_m\left(x\right)$$

The final classifier is obtained, as follows:

$$G\left(x\right) = sign\left(f\left(x\right)\right) = sign\left(\sum_{m=1}^{M} \alpha_m G_m\left(x\right)\right)$$

II. METHODS AND THEORY

Data sets and data analysis:

This experiment provides 1000 pictures, of which 500 are RGB pictures containing human faces, which are stored in./datasets/original/face, and the other 500 are RGB maps without human faces, which are stored in./datasets/original/nonface.

Experimental steps:

1.read data set. Read the picture, turn the data into a grayscale map of 24*24, the number and proportion of the positive and negative samples of the data set are not limited, and the form of the data set is not limited.

2. processing data set and extracting NPD features. Use t

he method of NPDFeature class in feature.py to extract features. (hint: because the time of preprocessing data set is relatively long, we can use the dump () function in pickle library to save the preprocessed feature data to the cache, then we can read the feature data with load () function.

3.the dataset is divided into training set and validation set, and the experiment is not divided into test sets.

4. write all the functions of AdaboostClassifier according to the reserved interface in ensemble.py. The following is the idea of the fit () method in the AdaboostClassifier class:

4.1 initializes the weights of the training set, and each training sample is given the same weight value.

4.2, training a base classifier. The base classifier can use DecisionTreeClassifier in the sklearn.tree library.

4.3 the classification error rate of the base classifier on the training set.

4.4 according to the classification error rate, the parameters are calculated.

4.5 update the weight of the training set.

4.6 iterate the steps of the above 4.2-4.6, and the number of iterations is the number of the base classifier.

5., we use AdaboostClassifier method to predict and calculate the accuracy rate on the validation set, and use the classification_report () function of the sklearn.metrics library to write the prediction results into report.txt.

III. EXPERIMENT

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

Weak classifier: 2-10, the decision tree depth is ultimately determined to use 2 and 4, for the following reasons

Question:

During the experiment, found that sklearn decision tree classification when not in depth is very good (perhaps the data set is too simple), up to 100%, resulting in the error iteration process in AdaBoost value of 0 to the sample weight update and update the classifier of the importance.

Measures：

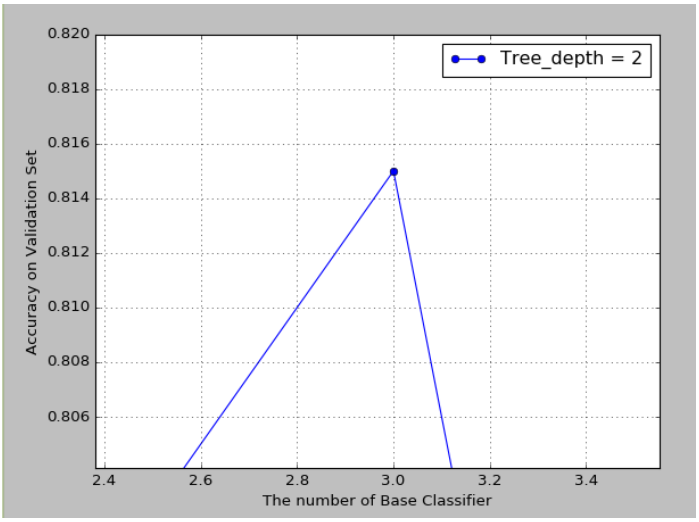
Through continuous testing, max_depth (decision tree depth) is 6-7, the classification effect can reach 100% without error, is not conducive to the development of the experiment, therefore, in order to deepen the understanding of the iterative process, make the combination of multiple classifiers has the possibility of the experiment process, decision tree depth is set to 2 or 4, one is to let the decision tree (i.e. based classifier) was the most close to the best, but also take this choice in the practical application, on the other hand can be fully applied into the AdaBoost experiment.

Contrast：

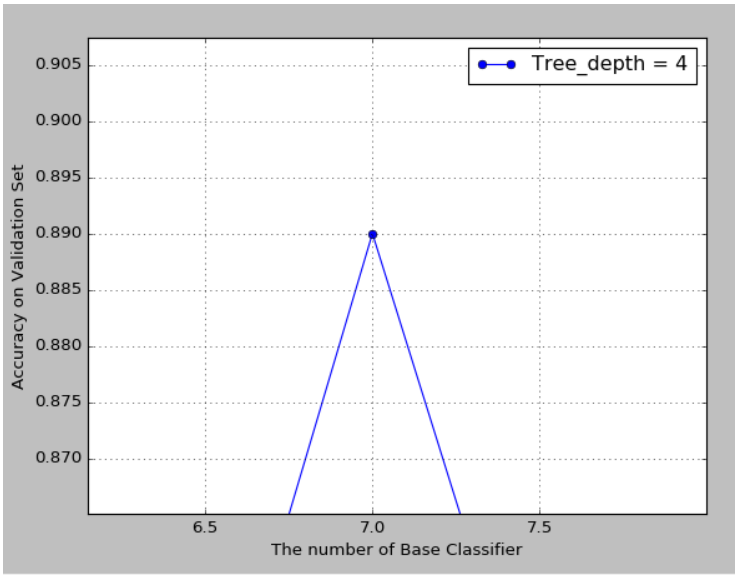
When the decision tree depth is 2 and 4, we use the same base classifier trained by 5, 6, 7, 8, 9 and 10 numbers to compare the effect of the last model and the accuracy of prediction.

Prediction results (best results): (the accuracy of the prediction on a test set of 200 samples separated from the training set)

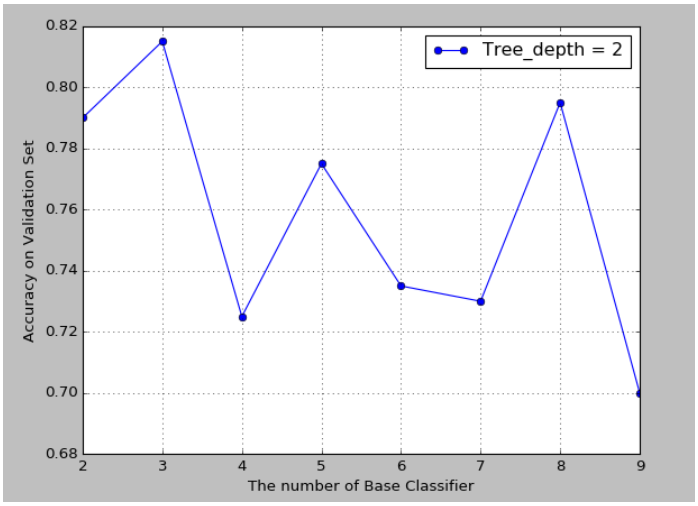
The decision tree with a depth of 2 has the maximum accuracy of 81.5% when the number is 7:

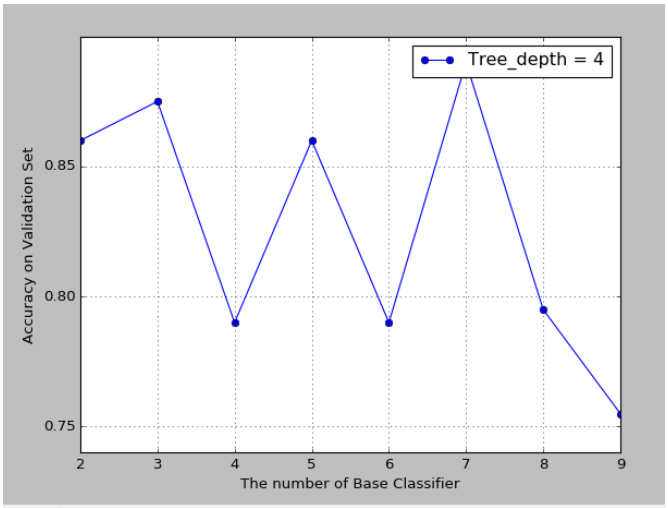


The decision tree with a depth of 4 has the maximum accuracy of 89% when the number is 7.



The accuracy curve is: (below is the prediction accuracy of the 200 samples separated from the training set), and 2-9 numbers of decision trees with 2 or 4 depth are compared





IV. CONCLUSION

Code content:

feature.py ensemble.py train.py

Analysis of the results of the experiment:

As can be seen from the above two diagrams, the depth of decision tree has a certain effect on the classification performance of the model. When using the same number of base classifiers (self), the decision tree with a depth of 4 has better classification effect (picture recognition effect) than the depth 2. The longer the depth is, the longer the classification branch is, the stronger the judgement ability is, the better the classification effect is.

But at the same depth, the number of basic classifier model classification effect is not very significant (there may be the strong classification model, using the decision tree may use some weak models such as linear classification may be different), so in practical application, the classification model of the effect is not simply raise the classifier quantity, but rather through the test to determine the optimal number of classifiers, thus obtains the optimal model.

Summary of the experiment:

1. this experiment through the most basic data processing work, including gray-scale image processing, pixel matrix, deepen the understanding of the basic image processing process, but also learned to principle of NPD features, simple and efficient to show the difference between an image of each pixel, from another point of view is the complete show all the characteristics of a picture, clever.

2. this experiment tried to put all the data into an array of numpy format, a good method is not done before was found in the exploration process such as: before processing large-scale text using pickle memory, IO operating speed is very slow, now use numpy.save to read the characteristics of efficient storage and converted to NPY format file (although the requirement is pickle, but I think there is a need to numpy storage, especially for large-scale data, this is actually quite large =1000*165600)

3. in the feature.py file, we found __calculate_NPD_table (), a function similar to the hash table, which can be used for reference to the efficiency of the conversion table idea.

4., by adding various ways to build classifiers in AdaBoostClassifier class, I can deepen the understanding of object-oriented programming. Later, I will use this efficient and standardized method to classify myself.

5.Adaboost is a simple combination, but through this experiment, with the combination of the thought of as an opportunity to further the theory and practice together, to make a graphical interface with the adjustable parameters of visual images for training & prediction program (based on the +Tkinter project)

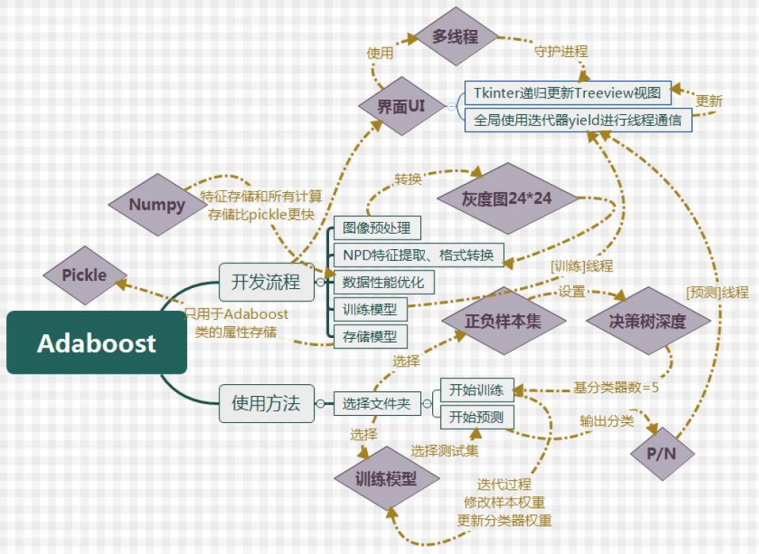
My project address is: (separate from the experimental address)

Https://github.com/Kali-Hac/Adaboost_V/tree/dev_v1.x

The experimental address is:

Https://github.com/Kali-Hac/ML_experiments/tree/Lab3

Development flow chart (thinking guide map of your own):



Experimental reference materials:

https://zh.wikipedia.org/wiki/AdaBoost

http://blog.csdn.net/qq_14845119/article/details/52576902

http://blog.csdn.net/cv_family_z/article/details/50037323

http://blog.csdn.net/frog_in_a_well/article/details/30725341

http://cwiki.apachecn.org/pages/viewpage.action?pageId=10814387

http://www.jianshu.com/p/479e92cf4c2c