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:
Qiang Yuan, Yihui Zhu, Junpeng Su

Supervisor:
Mingkui Tan

Student ID：201530613511,
201530613955, 201530612743

Grade:
Undergraduate

December 18, 2017

Face Classification Based on AdaBoost Algorithm

Abstract— It is an experiment of machine learning about Face Classification Based on AdaBoost Algorithm. In this paper, by introducing a dataset which is made up of 1000 pictures, we propose that implementing face detection by using Adaboost algorithm can effectively solve the face classification problem, and we try to do some expansion to verify the knowledge, which is using hierarchy classification method based on AdaBoost algorithm to deal with multi-class face classification, so we can combine the theory with the actual project and experience the complete process of machine learning. Experimental results on the dataset show that Adaboost is an effective way to solve classification problems, and it has good results in face detection.

I. INTRODUCTION

In the experiment, we try to use Adaboost algorithm to realize face detection by solving face classification problem and adjust parameters to get the right results.

The experiment is to help us understand Adaboost further, get familiar with the basic method of face detection, learn to use Adaboost to solve the face classification problem, and we try to do some expansion to verify our knowledge so that we combine the theory with the actual project and experience the complete process of machine learning.

We use python3 as the environment of experiment, including python package: sklearn, numpy, matplotlib, pickle, PIL. And we are going to accomplish the experiment by extracting Normalized Pixel Difference(NPD) features, training the model by learning NPD features and using validation set to test the effect of the model.

We wish to see that these methods have higher accuracy, rapider convergence, smoother loss curve, and better learning and diagnostic properties to complex data, and the results have lower losses finally.

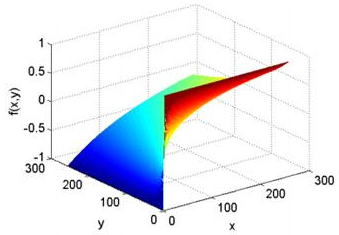
II. METHODS AND THEORY

We are going to extract the NPD features from pictures, complete the model of Adaboost classifier to fit the data, and update the weights by training base learner iteratively so that the results could be more accurate. Then we will talk about those methods and theory.

To begin with, the NPD features is to detect the divergence between 2 pixels, which is defined as a function f(x, y). x and y are values of any two pixels. And it specifies f(0, 0)=0.

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

The curve of this function is as follows:



Let's introduce the theory and methods of Adaboost algorithm.

The function of Adaboost algorithm is:

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

$h_m(x)$ is the base learner, and α_m is its importance score.

Adaboost sample weight updating fomula:

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

$z_m = \sum_{i=1}^{n} w_m(i) e^{-\alpha_m y_i h_m(\mathbf{x}_i)}$ is normalization term, makes $w_m(i)$ become probability distributions:

$$w_{m+1}(i) = \begin{cases} \frac{w_m(i)}{z_m} e^{-\alpha_m} & \text{for right predictive sample} \\ \frac{w_m(i)}{z_m} e^{\alpha_m} & \text{for wrong predictive sample} \end{cases}$$

so in the next round, $\frac{w_{wrong}(i)}{w_{right}(i)} = e^{2\alpha_m} = \frac{1-\epsilon_m}{\epsilon_m}$

and $\varepsilon_m < 0.5$, wrong samples will be more important.

Base learner:

$$h_m(X): x \to \{-1, 1\}$$

Error rate:

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

$\varepsilon_m < 0.5$, or the performance of Adaboost is weaker than random classification.

Make the base learner with lower ε_m more important:

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

Final learner:

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

Note: $h_m(X) = \text{sign}(w^T X)$ is a nonlinear function, so the Adaboost can deal with nonlinear problem.

III. EXPERIMENT

In this section, we analyze the performance of Adaboost for face detection, and we also investigate how to update weights to improve our algorithm.

3.1 Data sets and data analysis:

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.

We decide to extend the utility of the Adaboost classifier so that it can deal with muti-class classification problem, we make new labels for the original data as we divide face into 'man' or 'woman' and divide nonface into 'animal' and 'vehicle'.

3.2 Experimental steps:

1) Read data set data. The images are supposed to converted into a size of 24 * 24 grayscale. The face images are labeled as 1 while the nonface images are labeled as -1.
2) Processing data set data to extract NPD features. Extract features using the NPDFeature class in feature.py. And the extracted features are saved in pickle files.
3) The data set is divided 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.
   4.2)Training 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.

3.3 Experimental results and curve:

The below are our experimental results and the accuracy curves of models.

	precision	recall	F1-score	support
Face	0.95	0.97	0.96	149
Non-face	0.97	0.95	0.96	151
Ave/total	0.96	0.96	0.96	300

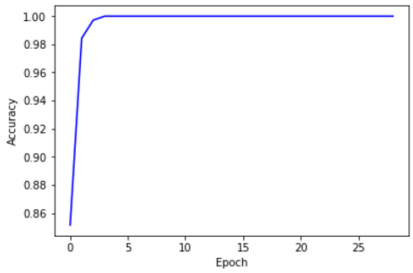
Accuracy of AdaBoost Classifier

	precision	recall	F1-score	support
Face	0.90	0.92	0.91	149
Non-face	0.92	0.90	0.91	151
Ave/total	0.91	0.91	0.91	300

Accuracy of Decision Tree Classifier

To prevent ε from being 0, the formula ε=max{ error_rate, 0.001 }. And end the fit process when a weaker classifier makes 100% correct classification.

The comparison of the accuracy of AdaBoost Classifier and Decision Tree Classifier shows that AdaBoost Classifier performs much better than Decision Tree Classifier. It proves the effectiveness of Adaboost Algorithm.



Accuracy Curve of AdaBoost Classifier

From the accuracy curve we learn that AdaBoost Classifier make 100% correct prediction under train dataset after 4th epoch. However, the accuracy under validation dataset is only 96% which means AdaBoost Classifier may encounter a bottleneck.

The result we use hierarchy classification method is as below

	precision	recall	f1-score	support
class 1	0.72	0.83	0.77	109
class 2	0.25	0.12	0.17	40
class 3	0.67	0.72	0.69	92
class 3	0.53	0.51	0.52	59
avg / total	0.60	0.64	0.62	300

The result we use Decision Tree method is as below

	precision	recall	f1-score	support
class 1	0.73	0.63	0.68	109
class 2	0.34	0.42	0.38	40
class 3	0.56	0.48	0.51	92
class 3	0.33	0.42	0.37	59
avg / total	0.54	0.52	0.53	300

We can see that our method is much better, which means it works.

3.3 Expansion:

To extend the experiment, we use hierarchy classification based on Adaboost algorithm to tackle multi-class classification problem based on the original data. Make new labels and train the model.

We find that in the face pictures, we can still divide them into two classes, male and female. And in non-face pictures, we divide them into animal and object. So we get four classes, we denote their labels as 1, 2, 3, 4, respectively. When to recognize a picture, the first thing is to judge whether it is a face picture by the AdaBoost in the first layer. If it is a face picture, then we judge whether it is a male picture or female picture by the AdaBoost in the second layer. Otherwise, judge whether it is an animal picture or object picture. So we can do multi-class classifier.

IV. CONCLUSION

In this experiment, we use Adaboost algorithm to learn a sparse feature subset to solve face classification problem so that it can do face detection, and improve the performance by updating the weights. There are some gains in the middle of our tests. Primarily, we learn the theory of Adaboost and practice the method in the experiment, we are beginning to have some insight into the algorithm. Next, we get familiar with the NPD feature and Adaboost Classifier and use them solve the face classification problem during the period, we combine the theory with the actual project, experience the process of machine learning. Moreover, we try to extend the experiment by using Adaboost to settle a muti-class classification problem and make some results and the result shows that our model performs better than Decision Tree Classifier in solving multi-class face classification problem.