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:
Yuehao Xiao

Supervisor:
Mingkui Tan

Student ID:
201630719984

Grade:
Undergraduate

October 31, 2018

Face Detection Based on AdaBoost Algorithm

Abstract—A face detection method based on AdaBoost Algorithm, using DecisionTree Classifier as a base classifier.

I. Introduction

NOWADAYS, artificial intelligence is very popular. In rencent years, it has developed into so many branches. In this report, we will talk about a face detection method based on AdaBoost Algorithm.

II. Methods and Theory

For the face classification, we load images with size 24*24. One half are images of face, the rest are images not containing a face. First, processing data set to extract NPD features. Then we can training some base classifiers which can be Decision Tree. After we calculate the classification error rate of the base classifier on the training set, we can calculate the weights of classifiers according to the classification error rate. Then we update training set weights and repeat steps for iteration. Finally, we can predict and verify the accuracy on the validation set. As a comparation, we try using opencv library to face detection, as it is known to be mature and stable.

III. Experiments

A. Dataset

This experiment provides 1000 pictures, of which 500 are human face RGB images and the other 500 are non-face RGB images.

B. AdaBoost Algorithm

Recall the additive model:

$$F_m(x) = F_{m-1}(x) + \alpha_m h_m(x)$$

Exponential loss function:

$$L(y, F_m(x)) = e^{-yF_m(x)}$$

Introduce the additive model into the loss:

$$\{\alpha_m, h_m(x)\} = \arg\min \sum_{i=1}^{N} e^{-y_i(F_{m-1}(x_i)+\alpha h(x_i))}$$

$$= \arg\min \sum_{i=1}^{N} e^{-y_i F_{m-1}(x_i)}(1 - y_i \alpha h(x_i) + \frac{\alpha^2}{2})$$

since $y_i^2 = h(x)^2 = 1$

We can directly optimize the latter term to train base learners:

$$h_m(x) = \arg\min_h \sum 6N_{i=1}[1 - y_i \alpha h(x_i) + \frac{\alpha^2}{2}]$$

Fixing the α, this is equivalent to optimizing:

$$h_m(x) = \arg\max_h \sum_{i=1}^{N} [y_i h(x_i)]$$

where $y_i h(x_i) = -1$ when we make mistakes, or $y_i h(x_i) = 1$.

Based on $F_m(x) = F_{m-1}(x) + \alpha_m h_m(x)$, we have obtained the new base learner $h_m(x)$, so the only task is to seek for α_m. Let us consider the weighted exponential loss at current round:

$$L(y, h_m(x)) = \sum_{i=1}^{N} [\omega_{(m,i)} e^{-y_i \alpha_m h_m(x_i)}] = e^{-\alpha_m}(1-\epsilon_m) + e^{\alpha_m}\epsilon_m$$

The derivation of this loss:

$$\frac{\partial L(y, h_m(x))}{\partial \alpha_m} = -e^{-\alpha_m}(1 - \epsilon_m) + e^{\alpha_m}\epsilon_m$$

Setting the derivation as zero will give:

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

Recall the distribution weight of samples: $\omega_{(m,i)} = e^{-y_i F_{m-1}(x_i)}$, based on $F_m(x) = F_{m-1}(x) + \alpha_m h_m(x)$, we have:

$$\begin{aligned} \omega_{(m+1,i)} &= e^{-y_i F_m(x_i)} \\ &= e^{-y_i(F_{m-1}(x)+\alpha_m h_m(x))} \\ &= e^{-y_i F_{m-1}(x_i)} * e^{-y_i \alpha_m h_m(x_i)} \\ &= \omega_{(m,i)} e^{-y_i \alpha_m h_m(x_i)} \end{aligned}$$

Final update equation:

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

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

Output:

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

Results:

TABLE I
Classifier Report

	precision	recall	f1-score	support
face	0.87	0.78	0.82	50
nonface	0.80	0.88	0.84	50
avg / total	0.83	0.83	0.83	100

C. Face detection using opencv

Opencv is a well known package for image detection. As a comparation to above algorithm, we choose opencv to do face detection.

As expected, opencv is very easy to use, and very fast to do detection. The program can using a square to show faces.
Result:

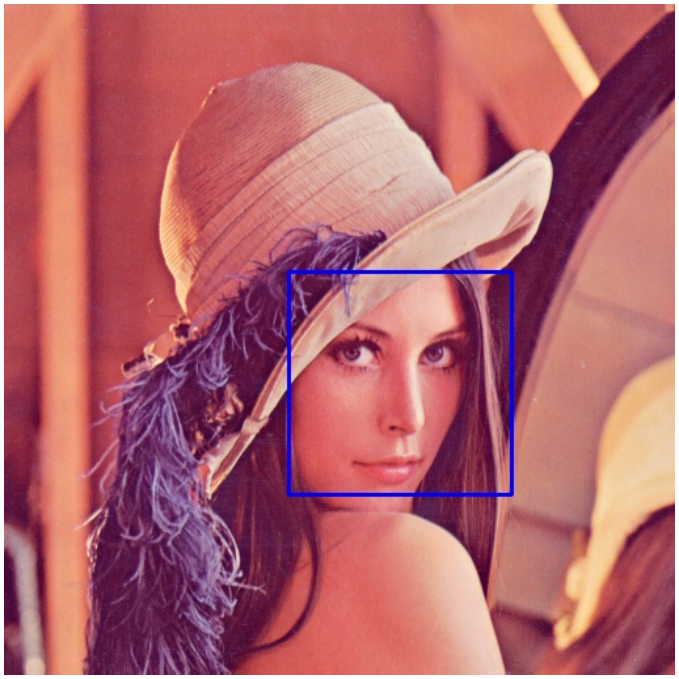

Fig. 1. Result of face detection

IV. Conclusion

After this experiment, we understood AdaBoost further and got familiar with the basic method of face detection. Which is more important, we experienced the complete process of machine learning.