South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Jiachen Wang and Xiongshuai Zhang
and Xuanyi Zhang

Supervisor:
Qingyao Wu

Student ID：
201530612859 and 201530613641
and 201530613658

Grade:
Undergraduate

December 19, 2017

Face Classification Based on AdaBoost Algorithm

Abstract—In this experiment we understand Adaboost further and get familiar with the basic method of the face detection through classifying faces. We also experience the complete process of machine learning.

I. Introduction

AdaBoost, short for Adaptive Boosting. 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.

AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.

NPD is computed as the ratio of the difference between any two pixel intensity values to the sum of their values, in the same form as the Weber Fraction in experimental psychology. The NPD feature has several desirable properties, such as scale invariance, boundedness, and ability to reconstruct the original image. we further show that NPD features can be obtained from a look up table, and the resulting face detection template can be easily scaled for multiscale face detection.

II. Methods and Theory

A.Adaboost

1） Training

AdaBoost refers to a particular method of training a boosted classifier. A boost classifier is a classifier in the form：

$$F_T(x) = \sum_{t=1}^{T} f_t(x)$$

where each f_t is a weak learner that takes an object $\boldsymbol{x}$ as input and returns a value indicating the class of the object. For example, in the two class problem, the sign of the weak learner output identifies the predicted object class and the absolute value gives the confidence in that classification. Similarly, the $\boldsymbol{T}$th classifier is positive if the sample is in the positive class and negative otherwise.

Each weak learner produces an output hypothesis, $h(x_i)$, for each sample in the training set. At each iteration $\boldsymbol{t}$, a weak learner is selected and assigned a coefficient α_t such that the sum training error E_t of the resulting $\boldsymbol{t}$-stage boost classifier is minimized.

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)]$$

Here $F_{t-1}(x)$ is the boosted classifier that has been built up to the previous stage of training, ***E(F)*** is some error function and $f_t(x) = \alpha_t h(x)$ is the weak learner that is being considered for addition to the final classifier.

2） Weighting

At each iteration of the training process, a weight w_t is assigned to each sample in the training set equal to the current error $E(F_{t-1}(x_i))$ on that sample. These weights can be used to inform the training of the weak learner, for instance, decision trees can be grown that favor splitting sets of samples with high weights.

B. Normalized Pixel Difference (NPD)

The Normalized Pixel Difference (NPD) feature between two pixels in an image is defined as

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

where x, y ⩾ 0 are intensity values of the two pixels1, and f(0, 0) is defined as 0 when x = y = 0.
The NPD feature measures the relative difference between two pixel values. The sign of f(x, y) indicates the ordinal relationship between the two pixels x and y , and the magnitude of f(x, y) measures the relative difference (as a percentage of the joint intensity x+y) between x and y. Note that the definition f(0, 0) _ 0 is reasonable because, in this case, there is no difference between the two pixels x and y. Compared

to the absolute difference |x − y|, NPD is invariant to scale change of the pixel intensities.

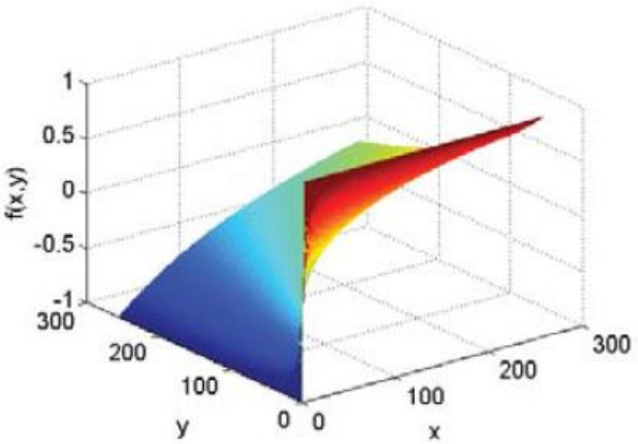

Fig. A plot of the NPD function f(x, y).

III. Experiment

In the experiment, we use the Adaboost Algorithm to classify the images by they include human faces or not.

A.Experient Steps:

(1) Initialize the distribution weight of training data

D1 = {w11, w12, ..., w1i, ..., w1N}, i = 1,2,3, ..., N

w1i = 1 / N

(2), To generate M base learners, then for m = 1, 2, ..., M:

(a), using the training data with weight distribution Dm to generate a base learner:

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

(b) Calculate the classification error rate of Gm (x) on the training dataset:

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

(c) Calculate the parameters of the base learner Gm (x):

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

(d) Update the weight distribution of the training data set:

$$D_{m+1} = \{w_{m+1,1}, \dots, w_{m+1,i}, \dots, w_{m+1,N}\}$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \exp(-\alpha_m y_i G_m(x)) = \begin{cases} \frac{w_{mi}}{Z_m} \exp(-\alpha_m), G_m(x_i) = y_i \\ \frac{w_{mi}}{Z_m} \exp(\alpha_m), G_m(x_i) \neq y_i \end{cases}$$

In this formula:

$$Z_m = \sum_{i=1}^{N} \alpha_m \exp(-\alpha_m y_i G_m(x))$$

(3) Construct a linear combination of basic classifiers\

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

Then we get the final classifier:

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

The sum of αm is not equal to 1, which indicates the importance of each classifier. The symbol of f (x) indicates the category of x. | f (x) | indicates the confidence of classification. The larger the classification result, the more reliable it is.

B.Experiment Result:

	precision	recall	f1-score	support
-1.0	0.96	0.94	0.95	162
1.0	0.94	0.95	0.94	138
avg / total	0.95	0.95	0.95	300

IV. Conclusion

The NPD feature evaluation is extremely fast, requiring a single memory access using a look up table.

The face detector is able to handle illumination variations, pose variations, occlusions, out-of- focus blur, and low resolution face images in unconstrained scenarios.

through the experiment result we can draw a conclusion that the smaller the classification error rate of the classifier in the final classifier role is more and more, but also to ensure the final classifier performance reasons. As a result, a single cascade AdaBoost classifier is able to achieve promising results for face detection
with large pose variations and occlusions.