The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Deng Huang
Haoming Xu
Ruiqiu cao

Student ID：
201530211588
201530781357
201330610031

Supervisor:
Qingyao Wu

Grade:
Undergraduate

December 15, 2017

Face classification based on AdaBoost Algorithm

***Abstract*—Adaboost is an iterative algorithm. Its core idea is to train different classifiers (weak classifiers) on the same training set, and then combine these weak classifiers to form a stronger final classifier (strong classifier).**

In this paper, we implement it , test it for the purpose of understanding it and the complete process of the machine lear--ing further.

I. INTRODUCTION

We do this experiment for the purpose of:

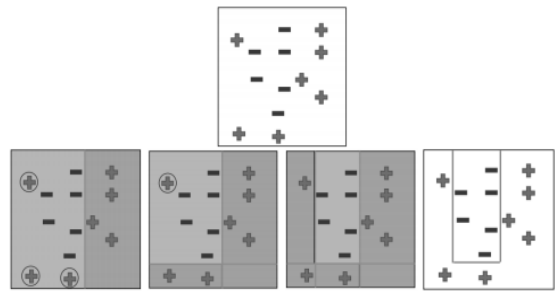
1. Understanding Adaboost further
2. Getting familiar with the basic method of face detection
3. Learning to use Adaboost to solve the face classification problem,and combine the theory with the actual project
4. Experiencing the complete process of machine learning

II. METHODS AND THEORY

The main idea and formulas that use in the paper & the pseudocode of algorithm:

A. The main idea of the Adaboost:

Make the wrong predictive samples more important, and handle it in next round:



B. The main formulas:

Base learner

$$h_m(\mathbf{x}) : \mathbf{x} \mapsto \{-1, 1\}$$

Error rate

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

Make the base learner with lower ϵ_m more important

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

Final learner

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

Note: $h_m(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^\top \mathbf{x})$ is a nonlinear function, so the Adaboost can deal with nonlinear problem

C. The pseudocode of algorithm:

Algorithm 2: Adaboost

Input: $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in X, y_i \in \{$
Initialize: Sample distribution w_m
Base learner: $\mathcal{L}$

$w_1(i) = \frac{1}{n}$
for m=1,2,...,M **do**
  $h_m(x) = \mathcal{L}(D, w_m)$
  $\epsilon_m = \sum_{i=1}^{n} w_m(i)\mathbb{I}(h_m(\mathbf{x}_i) \neq y_i)$
  if $\epsilon_m > 0.5$ **then**
    break
  end
  $\alpha_m = \frac{1}{2}\log\frac{1-\epsilon_m}{\epsilon_m}$
  $w_{m+1}(i) = \frac{w_m(i)}{z_m} e^{-\alpha_m y_i h_m(\mathbf{x}_i)}$, where $i = 1, 2, ...,$
  $z_m = \sum_{i=1}^{n} w_m(i) e^{-\alpha_m y_i h_m(\mathbf{x}_i)}$
end
Output: $H(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m h_m(\mathbf{x})$

III. Experiment

A. Dataset

Experiment use 500 human face and 500 nonface which collected from internet as the dataset. After shuffle them, we take 600s as the training set and 400s as the validation set.

B. Implementation

The main process of Adaboost implemented using python as shown below:

```
for iboost in range(self.n_weakers_limit):

    clf = self.weak_classifier(max_depth=1)
    clf.fit(X, y, sample_weight=self.w)
    y_pred = clf.predict(X)

    incorrect = y_pred != y

    error = np.mean(
        np.average(incorrect, weights=self.w,

    if error <= 0:
        print(error)
        alpha = 1
    else:

        alpha = self.learning_rate * (
            np.log((1. - error) / (error)) +
            np.log(self.n_classes - 1.))

        self.w *= np.exp(alpha * incorrect *
                         ((self.w > 0) |
                          (alpha < 0)))

        self.models.append(clf)
        self.alphas.append(alpha)

    if error == 0:
        break

    self.w /= self.w.sum()
```

C. Testing

1. We train the Adaboost classifier using different amount of the weak classifiers, and also different depth of the tree(decision tree in this experiment) to see the effect:

a. Adaboost classifier with 1 decision tree with 1 depth:

	precision	recall	f1-score	support
face	0.76	0.95	0.85	203
nonface	0.93	0.70	0.80	197
avg / total	0.84	0.82	0.82	400

b. 20 decision trees with 1 depth:

	precision	recall	f1-score	support
face	0.975	0.975	0.975	203
nonface	0.975	0.975	0.975	197
avg / total	0.975	0.975	0.975	400

c. 50 decision trees with 1 depth:

	precision	recall	f1-score	support
face	0.980	0.966	0.973	203
nonface	0.965	0.980	0.972	197
avg / total	0.973	0.973	0.973	400

d. 1 decision tree with 2 depth:

	precision	recall	f1-score	support
face	0.93	0.91	0.92	203
nonface	0.91	0.93	0.92	197
avg / total	0.92	0.92	0.92	400

e. 20 decision trees with 2 depth:

	precision	recall	f1-score	support
face	0.976	0.990	0.983	203
nonface	0.990	0.975	0.982	197
avg / total	0.983	0.983	0.982	400

f. 50 decision trees with 2 depth:

	precision	recall	f1-score	support
face	0.980	0.966	0.973	203
nonface	0.965	0.980	0.972	197
avg / total	0.973	0.973	0.973	400

2. Analysis to the result of the experiment:

a. We see that the Adaboost classifier behaves better obviously the weak classifier.

b. When amount of the classifier reach a level, the Adaboost can not become if increment the amount.

I think it is restricted by the base weak classifers.

c. Just a satisfied amount of the 1 or 2 depth decision tree(it is a very weak classifier)can behave well enough in the experiment, the algorithm is so effective.

IV. CONCLUSION

Through the experiment we experiment the relatively complete process of machine learning. We experience the process of changing the things of reality to data, and then use the algorithm to train the computer with these data, instead of using the ready-made ones as the two experiments former. This experiment helps us to combine the theory with the actual project better.

Of course, we get more familiar with the Adaboost algorithm because we implement it by ourselves and test it with data instead of just studying it on the book.