華南理工大學

South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Lizhao Liu

Supervisor:
Mingkui Tan

Student ID:
201730683109

Grade:
Undergraduate

November 14, 2019

Adaboost for face image classification

***Abstract*—In this report, we solve the face/non-face classification problem by adaboost with Decision Tree Classifier (DTC) as weak learner.**
We perform experiments on three aspects:
1. Weak learner's performance effect.
2. Number of weak learner's exploration.

I. Introduction

ADABOOST short for "Adaptive Boosting", is an ensemble method that combine a bunch of weak learners into a strong learner which outperforms all the weak learners. It's wildly used in academic and industry community, which make it very important for us to fully explore it. We will explore the the effect of weak learner's performance and the performance with different number of weak learner.

II. Methods and Theory

In this part, we define Adaboost algorithm.

A. Adaboost

Given dataset $D = \{x_i, y_i\}_{i=1}^m$, where $x_i \in \mathbf{R}^n$. $y_i \in \{-1, 1\}$ and weak Learner $L(x) \in \{-1, 1\}$, Adaboost algorithm can be found in Algorithm 1.

Algorithm 1: Adaboost

Input: Training Dataset $D = \{x_i, y_i\}_{i=1}^m$, num weak learner N, weak learner L
Output: Final learner $H(x)$

$w_1(i) \leftarrow \frac{1}{m}$ where i = 1, 2, ..., m
for $idx = 1, 2, ..., N$ **do**
  $h_{idx}(x) = L(D, w_{idx})$
  $\epsilon_{idx} = \Sigma_{i=1}^m w_{idx}(i)\mathbb{1}(h_{idx}(x_i) \neq y_i)$
  $\alpha_{idx} = \frac{1}{2} \log \frac{1-\epsilon_{idx}}{\epsilon_{idx}}$
  $w_{idx+1}(i) = w_{idx}(i) * exp(-1 * \alpha_{idx} * y_i * h_{idx}(x_i))$ where i = 1, 2, .., m
  $w_{idx+1} = \frac{w_{idx+1}}{\Sigma_{i=1}^m w_{idx}(i)}$
return $H(x) = \Sigma_{idx=1}^N \alpha_{idx} h_{idx}(x)$

We can see that, in each iteration, the examples that correctly classified by previous weak learner, will get a smaller weight, so that current learner can pay more attention to the misclassified examples.

III. Experiments

A. Dataset

We conduct all the experiments on self constructed face/non-face dataset, which has 500 256x256 face images and 500 32x32 non-face images. All images have 3 channels. We preprocess all the images into 24x24 gray scale images. We then convert them by using NPD features. Finally, each example's form is (x, y), where $x \in \mathbf{R}^{165600}$ and $y \in \{-1, +1\}$, where $+1$ means face and -1 means non-face. We randomly split the dataset into three part: 700 training examples, 100 validation examples and 200 test examples. $X_{train}, X_{val}\ and\ X_{test}$ are divided by 256 for scale purporse.

B. Implementation

We implement the Adaboost using python and mainly rely on the numpy package.

C. Weak learner's performance's effect.

In this section, we explore the influence of weak learner's performance in Adaboost.

We first explore the performance of DTC under different depths setting. From Fig 1 we can see that, as the depth of DTC goes depper, its performance on training set become better but in validation set, the performance of it first gets better and then goes down which indicating the overfitting problem. The best performance in validation set is 89 occurs when $depth = 4$.

Then we explore Adaboost's performance with different weaker learner. We set number of weak learner to five due to computation limitation. We can see from Fig 2 that, both Adaboost and DTC suffer from overfitting prolem. But Adaboost outperform DTC under all setting except in validation set when depth equals to 5. We can see that, as the performance of weak learner goes up, the performance of Adaboost goes down, maybe due to the overfitting problem. The intuition is that, if a weak learner's performance is good enough in training set, then Adaboost with more weak learner is prone to overfit the dataset.

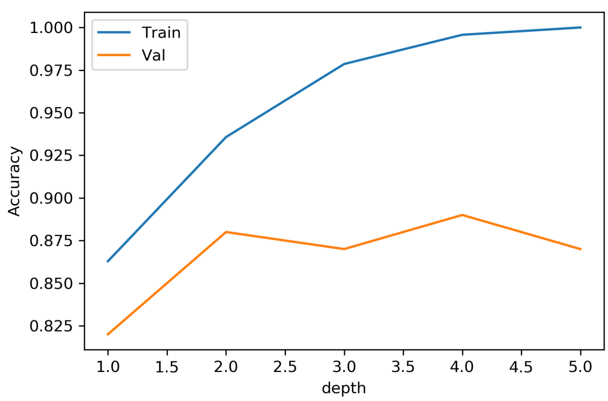

Fig. 1. DTC accuracy under different $depth$.

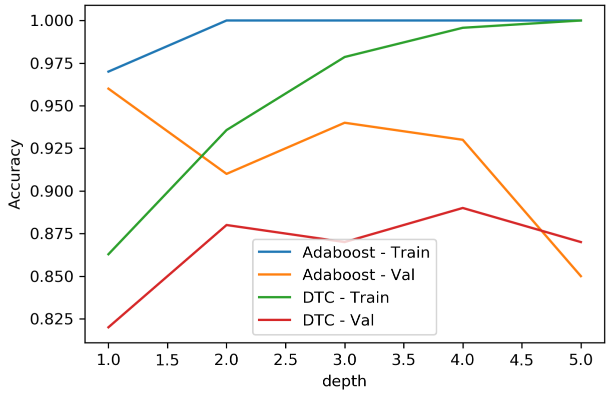

Fig. 2. Adaboost and DTC accuracy under different *depth*. Note that we set number of weak learner to 5 in Adaboost.

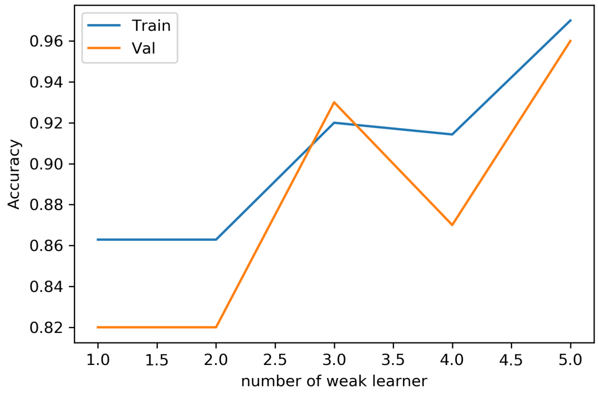

Fig. 3. Adaboost accuracy with different number of DTC ($depth = 1$).

TABLE I
DTC AND ADABOOST ACCURACY ON VALIDATION SET. BEST RESULT ARE IN BOLD.

depth	1	2	3	4	5
DTC	0.82	0.88	0.87	**0.89**	0.87
Adaboost	**0.96**	0.91	0.94	0.93	0.85

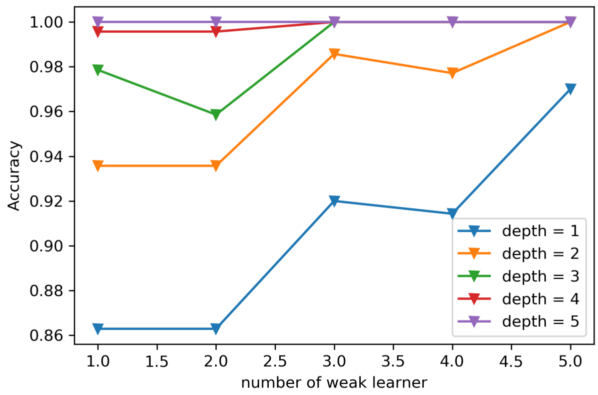

Fig. 4. Adaboost accuracy with different number of DTC ($depth = 1, 2, 3, 4, 5$) in training set.

D. Performance

Now we report the final performance of Adaboost under the best settings e.g. DTC'depth = 3 and number of weak learner = 5. The final result is shown in Fig II.

TABLE II
ADABOOST'S BEST ACCURACY.

train	*validation*	*test*
1.00	0.98	0.97

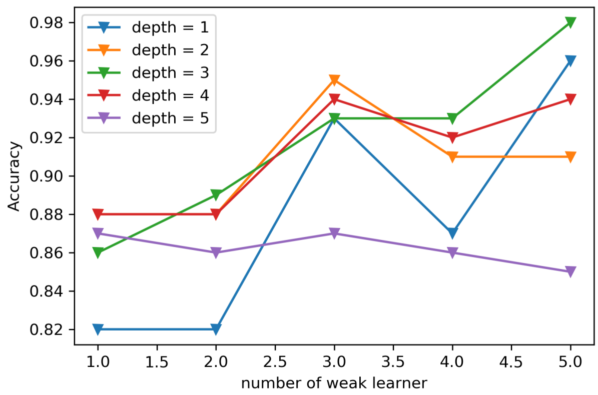

Fig. 5. Adaboost accuracy with different number of DTC ($depth = 1, 2, 3, 4, 5$) in validation set.

E. Number of weak learner's exploration

In this section, we conduct experiment on number of weak learner's exploration.

We first set the depth to 1 due to computation limitation. Under this setting, More weak Learner benefits the Adaboost, as shown in Fig 3.

Then we evalutate the performance of Adaboost under settings that DTC' depth ranges from 1 to 5 and number of weak learner ranges from 1 to 5. As illustrated in Fig 4, in training set, Adaboost benefits from both the depth and the number of weak learners. Note that, as the depth of weak learner goes up, the performance gained by adaboost goes down. As shown in Fig 5, when DTC's depth equals to 3, as the number of weak learner goes up, Adaboost has better performance. When DTC's depth equals to 5, Adaboost's performance is the worst.

Now, we cann draw the following conclusions:
1. Weaker learner gets larger performance gain in Adaboost.
2. Very Strong weak learner tends to have overfitting problem in Adaboost.
3. More weak learner boost the performance in Adaboost when weak learner is not very strong.

IV. CONCLUSION

In this report, we explore Adaboost algorithm. Specifically, we explore the weak learner's performance and number of weak learner in Adaboost. We draw the conclusions that weaker learner can get more performance gain in Adaboost and stronger learner tends to have overfitting problem in Adaboost. And finally we report the performance on test set under the best validation setting.