ECEN 649 PATTERN RECOGNITION FINAL PROJECT

A REPORT

BY

XIAOMU DONG

Submitted to Dr. Tie Liu

NOV 20, 2020

METHODOLOGY

The methodology of this project is based on Viola Jones paper on AdaBoost and the Chapter 10.1.1 of the textbook. The implementation of the algorithm can be divided into three parts which are extracting Haar features, implementing ERM for decision stumps, and implementing AdaBoost predictor.

(1) Extracting Harr features

Harr features are some rectangle features with different size and location in the sub-images. The values of them are computed by subtracting the sum of pixel values in the dark area from the sum of the pixel values in the white area. In this project, four types of Harr features are used which are shown underneath. The size of the sub-image is 19x19 and within each sub-image, over 50000 Harr features need to be computed which can bring huge problems in computational efficiency and memory. Therefore, a proper algorithm needs to be implemented to reduce the computation.

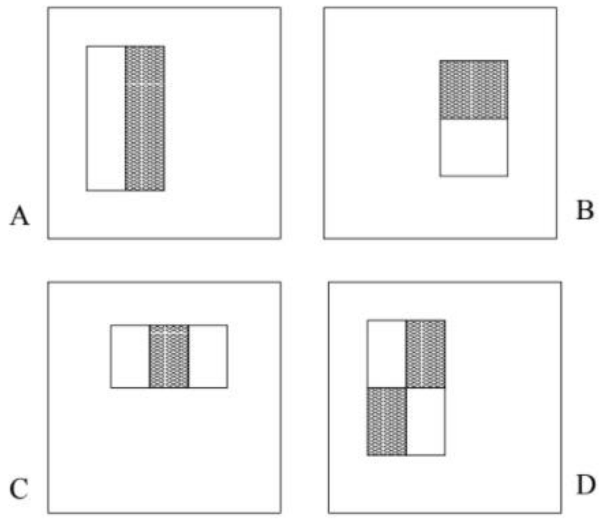


Figure 1

Integral image is an efficient method introduced in the Viola Jones paper to reduce the computation of the sum of pixel values in rectangle areas. Firstly, for each chosen pixel (x, y) in the sub-image, the sum of the pixels on the left and top of p can be computed using the equation underneath where ii(x, y) is the integral image and i(x, y) is the original image. In this method, the complexity is in the order of O(n^2). However, we can optimize this method and reduce the complexity to O(n) by using the two equations (2) and (3) where s(x, y) is the cumulative row sum.

$$ii(x,y) = \sum_{x' \leq x,\ y' \leq y} i(x',y') \qquad (1)$$

$$s(x,y) = s(x,y-1) + i(x,y) \qquad (2)$$

$$ii(x,y) = ii(x-1,y) + s(x,y) \qquad (3)$$

With the values of the integral images, the sum of pixel values within any specific rectangle area can be computed easily by 4 + 1 - (2 + 3).

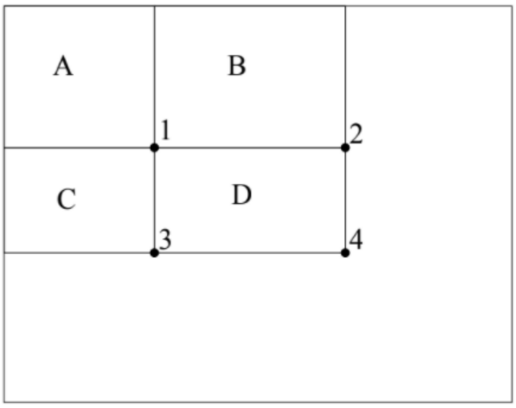


Figure 2

(2) Implementing ERM for decision stumps

In this project, for each AdaBoost round, an efficient algorithm of ERM for decision stumps is implemented. It is, for each round, we select the best feature and the threshold value which generate the smallest error based on weight. The detailed algorithm is shown in the Figure 3 where the j* and $\theta*$ denote the Harr feature number and the threshold value for each round.

ERM for Decision Stumps

input:
training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
distribution vector $\mathbf{D}$
goal: Find $j^\star, \theta^\star$ that solve Equation (10.1)
initialize: $F^\star = \infty$
for $j = 1, \ldots, d$
sort S using the j'th coordinate, and denote
$x_{1,j} \leq x_{2,j} \leq \cdots \leq x_{m,j} \leq x_{m+1,j} \stackrel{\text{def}}{=} x_{m,j} + 1$
$F = \sum_{i:y_i=1} D_i$
if $F < F^\star$
$F^\star = F, \theta^\star = x_{1,j} - 1, j^\star = j$
for $i = 1, \ldots, m$
$F = F - y_i D_i$
if $F < F^\star$ and $x_{i,j} \neq x_{i+1,j}$
$F^\star = F, \theta^\star = \frac{1}{2}(x_{i,j} + x_{i+1,j}), j^\star = j$
output $j^\star, \theta^\star$

Figure 3

(3) Implementing AdaBoost predictor

At the first round, uniform weights are used for faces and non faces samples. For each round, we pick out the best classifier using the algorithm mentioned above and renew the weights. For those samples which are labeled incorrectly, the weights are increased automatically which means those samples are emphasized in the next training round. Also, the higher the accuracy of the round t, the higher the value of round weight $\alpha^{(t)}$. At the end of T round, the final classifier is computed by combining weak learner obtained from each training round and a final threshold value is chosen. The detailed algorithm is shown in the Figure 4.

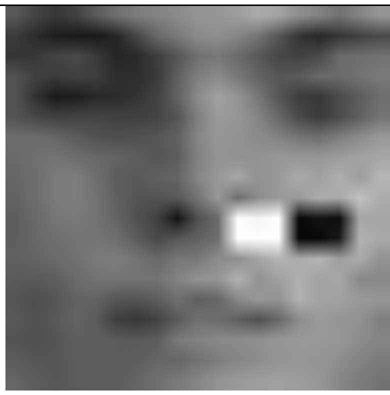
- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

Figure 4

RESULTS AND DISCUSSION

(i) (10 Points) The top 10 features selected by AdaBoost. Your description of these features needs to be self-explanatory.

The coordinates and the location in the sub-image of the top 10 Harr features are listed in the Table 1. From the top 10 Harr features, we can see that the Harr features are trying to capture the characteristics of the face organs. The first one locates exactly on the nose of the face which is the most obvious feature that a human face has. The second Harr feature locates on the boundary of nostril and cheek. The third one locates on top of the eyebrows. The fourth one locates on the left shade of the nose. The other ones are less obvious to be coincided with face organs because the sample weights have been modified after several round of AdaBoost training.

Harr feature #: 1 Harr feature type: A Top left coordinates: (1, 8) Right bottom coordinates:(12, 9)	
Harr feature #: 2 Harr feature type: A Top left coordinates: (10, 11) Right bottom coordinates:(11, 16)	
Harr feature #: 3 Harr feature type: A Top left coordinates: (0, 0) Right bottom coordinates:(0, 9)	

Harr feature #: 4 Harr feature type: C Top left coordinates: (5, 6) Right bottom coordinates:(7, 7)	
Harr feature #: 5 Harr feature type: B Top left coordinates: (0, 6) Right bottom coordinates:(1, 14)	
Harr feature #: 6 Harr feature type: B Top left coordinates: (17, 15) Right bottom coordinates:(18, 18)	
Harr feature #: 7 Harr feature type: A Top left coordinates: (12, 7) Right bottom coordinates:(16, 8)	

Harr feature #: 8 Harr feature type: D Top left coordinates: (3, 0) Right bottom coordinates:(10, 1)	
Harr feature #: 9 Harr feature type: A Top left coordinates: (3, 11) Right bottom coordinates:(11, 14)	
Harr feature #: 10 Harr feature type: B Top left coordinates: (10, 8) Right bottom coordinates:(11, 11)	

Table 1

(ii) (10 Points) The combined classifiers after running 1, 3, 5, and 10 boosting rounds. Again, your description of the combined classifiers needs to be self-explanatory.

The round weight and round threshold for the 10 first rounds are provided in Table 2. The combined classifiers after running 1, 3, 5, 10 boosting rounds are shown underneath where j is the value of the selected Haar feature. The round weight has a decreased trend. This can be explained by the weak learner obtained by later boosting rounds can generate worse accuracy on themselves compared with the weak learner selected earlier. This is one of the properties of AdaBoost.

Round 1:

$$H^{(1)} = sign(\alpha^{(1)} h^{(1)} - \theta_{final})$$

$$h^{(t)} = sign(\theta^{(t)} - j^{(t)})$$

Round 3:

$$H^{(3)} = \sum_{t=1}^{3} \alpha^{(t)} h^{(t)} - \theta_{final}$$

$$h^{(t)} = sign(\theta^{(t)} - j^{(t)})$$

Round 5:

$$H^{(5)} = \sum_{t=1}^{5} \alpha^{(t)} h^{(t)} - \theta_{final}$$

$$h^{(t)} = sign(\theta^{(t)} - j^{(t)})$$

Round 10:

$$H^{(10)} = \sum_{t=1}^{10} \alpha^{(t)} h^{(t)} - \theta_{final}$$

$$h^{(t)} = sign(\theta^{(t)} - j^{(t)})$$

Round	Round weight $\alpha^{(t)}$	Round threshold $\theta^{(t)}$
1	0.96653147	-125.5
2	0.89368438	-133.5
3	0.7514285	-132.5
4	0.73667449	-28.5
5	0.66674196	-137.5
6	0.70322097	-5.5
7	0.64568161	-16.5
8	0.60366237	-43.5

9	0.5854151	-294.5
10	0.57270683	-77.5

Table 2

(iii) (20 Points) The ROC curve of the combined classifiers after running 1, 3, 5, and 10 boosting rounds when applied to the test set. To obtain such curves, you need to manually adjust the thresholds in the combined classifiers to obtain various tradeoffs between the detection rate and the false positive rate. A general description of the ROC curve can be found in the following Wikipedia page: https://en.wikipedia. org/wiki/Receiver_operating_characteristic. Please carefully label your ROC curves in your report. How do the ROC curves change with the number of boosting rounds?

The ROC curve is shown in Figure 5. Theoretically speaking, the more boosting rounds are run, the higher the ROC curve. This is because more boosting rounds means more features are used in the final combined classifier, and a more complex model can capture more characteristics in the sub-image which can generate a higher true positive rate. However, in my ROC curve, this trend is not perfectly shown. We can see that the curve of 10 rounds has the highest value for most of the FPR value. However, for the other three curves, they do not show the trend of more rounds, higher TPR. This is probably because at the first several rounds, the model is not stable. When more boosting rounds are used, this trend should be more obvious.

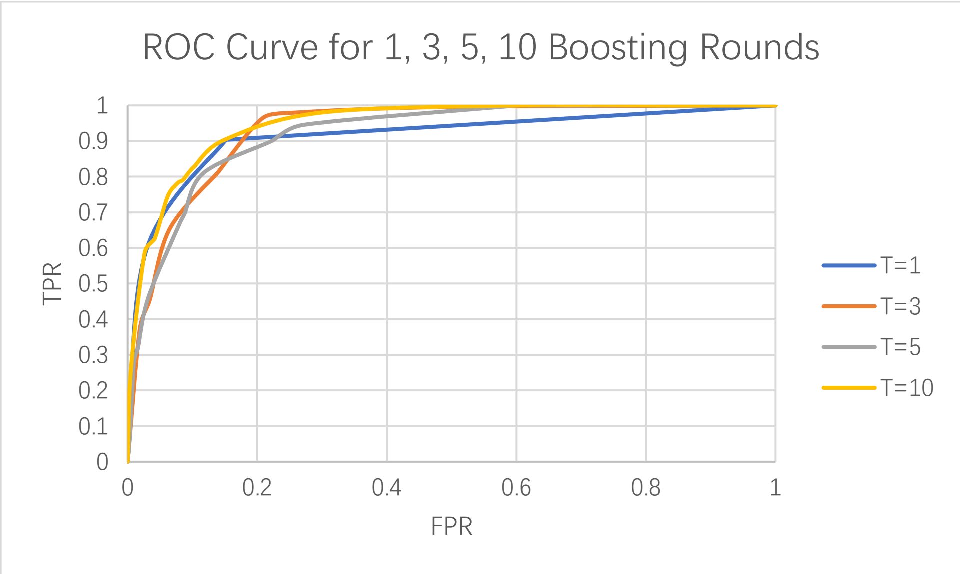


Figure 5

(iv) (Bonus points) Suggest how you may be able to improve the performance of the AdaBoost predictor in terms of accuracy and/or efficiency. Verify your suggestion using the given dataset.

One way to increase the efficiency of the algorithm is to reduce the size of possible number of Harr features. By observation, we can find out that the most important facial features are around the center of the sub-image which means we can only pick Harr features around the center of the sub-image within certain boundary. The range I pick is from pixel (0, 5) to pixel (15, 15) which can significantly reduce the possible number of Harr features from about 53000 to about 17000. As a result, the training time reduced from 6841 second to 1129 seconds. The accuracy and face recognition rate for two cases are shown in Table 3. We can see that the accuracy and recognition rate of two cases are almost the same and in some of the rounds, the recognition rate of the second case is higher than the first case. This is probably because in the second the case, we limit the range of the Harr feature to where the actual faces are instead of the whole sub-image.

	Full Sub-image		Selected Range	
Round	Accuracy	Recognition Rate	Accuracy	Recognition Rate
1	0.8591	0.9036	0.8591	0.9036
2	0.8591	0.9036	0.8591	0.9036
3	0.8513	0.8112	0.8505	0.755
4	0.8654	0.8232	0.8575	0.7831
5	0.8708	0.6666	0.8302	0.6024
6	0.8849	0.7831	0.863	0.7951
7	0.8818	0.7831	0.8466	0.7469
8	0.8904	0.8032	0.8693	0.771
9	0.8888	0.6907	0.8755	0.7469
10	0.9006	0.7469	0.8881	0.7791

Table 3

REFERENCES

1. Shalev-Shwartz S., Ben-David S. - Understanding Machine Learning_ From Theory to Algorithms-CUP (2014).

2. Paul Viola, Michael Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, - Conference on Computer Vision and Pattern Recognition 2001.