ECE763 Computer Vision: Models, Learning and Inference (including Deep Learning)

HW1: out 01/17, due 01/31, [DONE]

HW2: out 02/06, due 02/19 [DONE]

- HW3: out 02/27, due 03/16
- HW4: out 03/17, due 03/31
- HW5: out 04/01, due 04/15
- Project1: out 02/04, due 02/27 [DONE]
- Project2: out 02/27, due 03/16 [This One]
- Project3: out 03/18, due 04/30 (no late days allowed)

Instructor: Tianfu (Matt) Wu

NC STATE UNIVERSITY Electrical and Computer Engineering

Instructions and Notes

- *How to submit your solutions*: put your report (word or pdf or powerpoint) and results images (.png) if had in a folder named [your_unityid]_proj02 (e.g., twu19_proj02), and then compress it as a zip file (e.g., twu19_proj02.zip). Submit the zip file through **moodle**.
- *If you miss the deadline and still have unused late days, please send your zip file to TAs and me (0.5-day counted if later <= 6 hours based on email timestamp).*
- **Important Note**: We will **NOT** accept any replacement of submission after deadline ([+late days you use]), even if you can show the time stamp of the replacement is earlier than the deadline. So, **please double-check whether you submit correct files**.

Adaptive Boosting (AdaBoost) for Face Detection

- HW3 and Project 2 are coupled together. The former focuses on theoretical aspects, while the latter focuses on implementation.
- 2-Class AdaBoost
  - Input
    - Data: A set of labeled training data from 2 classes (e.g., the positives, face and the negatives, non-face image patches used in Project 1)

$$D_{train} = \{ (x_i, y_i): x_i \in R^d, y_i \in \{+1, -1\}, i = 1, 2, \cdots, N \},$$

where x_i represents the feature vector extracted for an image patch I_i . E.g., we may use 20×20 gray images for face and non-face, and use Haar features (see Project 2) to form x_i.

    - Weak Classifiers: A set of weak classifiers,

$$\Delta = \{h_t(x): t = 1, 2, \cdots, T\},$$

where $h_t(x): \mathrm{R}^{\mathrm{d}} \rightarrow \{+1, -1\}$, which is said to be a weak classifier because its error rate on D_{train} is slightly better than random guess (0.5 for a 2-class task),

$\epsilon(h_t) = \frac{1}{N}\sum_{(x_i,y_i)\in D_{train}} 1_{h_t(x_i)\neq y_i} < \frac{1}{2}$, where the indicator function $1_z = \begin{cases} 1, if\ z\ is\ true \\ 0, otherwise \end{cases}$

Adaptive Boosting (AdaBoost) for Face Detection

- 2-Class AdaBoost
  - Output: a strong classifier,

$$H(x) = sign\left(\sum_{i=1}^{M} \alpha_i h_i(x)\right)$$

Let $h = (h_1, h_2, \cdots, h_M),\ h_i \in \Delta,\ and\ \alpha = (\alpha_1, \alpha_2, \cdots, \alpha_M)$. We can rewrite it as,

$$H(x) = sign(F(x)), \qquad F(x) =< h, (x)\ \alpha >$$

  - Learning Objective:

$$h^*, \alpha^* = \arg\min_{(h,\alpha)} Err(H), \qquad Err(H) = \frac{1}{N} \sum_{(x_i, y_i) \in D_{train}} 1_{H(x_i) \neq y_i}$$

Adaptive Boosting (AdaBoost) for Face Detection

- 2-Class AdaBoost
  - Algorithm
    - Step 0: Initialize the training data with uniform weight,

$$w_1(x_i) = \frac{1}{N}, \forall x_i \in D_{train}$$

Or applying uniform weight within the positive and negative subset individually when the two classes are imbalanced while retaining the total weight being 1.

- Step 1: At iteration t, we have $H_{t-1}(x) = sign\left(\sum_{i=1}^{t-1} \alpha_i h_i(x)\right)$. $\forall h \in \Delta$, we compute the weighted error,

$$\epsilon_t(h) = \sum_{(x_i,y_i)\in D_{train}} w_t(x_i) \cdot 1_{h(x_i)\neq y_i}$$

- Step 2: Add a new weak classifier at iteration t with weight α_t

$$h_t = \arg\min_{h\in\Delta} \epsilon_t(h), \qquad \alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t(h_t)}{\epsilon_t(h_t)}$$

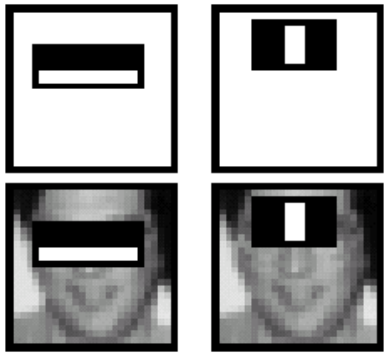
- Step 3: Update the data weight,

$$w_{t+1}(x_i) = \frac{1}{Z_t} w_t(x_i) e^{-y_i \alpha_t h_t(x_i)}, \qquad Z_t = \sum_{(x_i,y_i)\in D_{train}} w_t(x_i) e^{-y_i \alpha_t h_t(x_i)}$$

- Step 4: $t = t + 1$, go back to Step 0 if $t < maxT$ otherwise stop

Adaptive Boosting (AdaBoost) for Face Detection

- Project 3:
  - Data: Reuse the face /non-face data in Project 1. Recommended resolution will be 20 by 20 (gray images). You may want to extract a much large number of positive and negative images.



  - Feature vector and weak classifiers: Use Harr feature. Compute the value of each Harr feature for each sample. *Each feature corresponds to a weak learner*. Determine the threshold between face samples and non-face samples for all the weak learners. Calculate the classification error for each weak learner and draw the best ten features (before boosting) as Figure 3 in [1].
    - You can use any publicly available code for extracting the Harr features. Show what the actual Harr features are in the code you use.
  - Algorithm: Implementing the Adaboost Algorithm (also see Table 1 in [1]).

[1] P.Viola, M.Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *CVPR* 2001.

Adaptive Boosting (AdaBoost) for Face Detection

- Project 3:
  - Evaluation: in the test dataset you created. Plot the ROC based on thresholding $F(x)$.

- Hint:
  - *Start small.* Write a program using only a small number of weak classifiers – like the two illustrated above. Train and test the program on a small set of face and non-face images. At every stage, check to make sure that your results make intuitive sense. For example, see if your program selects the same first two features that Viola found (see above).
  - Be very careful about how your program scales with the number of weak classifiers and the number of face/non-face images. (This will depend on the details of how you write the code and the type of computer you are using). If your program scales badly, then only use a limited number of weak classifiers and a small number of faces/non-face images.
  - The classic mistake for this type of problem is to write a large amount of code and then try testing it with 50,000 weak classifiers on all the face and non-face images. Check the code first with only a few classifiers – so that you have a good guess of what classifier should be selected. Then try scaling up – and stop if the program scales badly or ceases to give intuitive results.