

# Assignment 4: CS 763, Computer Vision

Due 3rd April before 11:55 pm

**Remember the honor code while submitting this (and every other) assignment. You may discuss broad ideas with other student groups or ask me for any difficulties, but the code you implement and the answers you write must be from members of the group. We will adopt a zero-tolerance policy against any violation.**

**Submission instructions:** You should ideally type out all the answers in Word (with the equation editor) or using LaTeX. In either case, prepare a pdf file. Put the pdf file and the code for the programming parts all in one zip file. The pdf file should contain instructions for running your code. Name the zip file as follows: A4-IdNumberOfFirstStudent-IdNumberOfSecondStudent.zip. If you are doing the assignment alone, the name of the zip file should be A4-IdNumber.zip. Late assignments will be assessed a penalty of 50% per day late. Please preserve a copy of all your work until the end of the semester.

1. In this exercise, you will implement the Adaboost method for creating a strong binary classifier from a series of weaker classifiers. You will work with some synthetic datasets and also with the MNIST dataset containing images of digits.

Consider a training set consisting of  $N$  input vectors  $\{\mathbf{x}_j\}_{j=1}^N$  in a  $d$ -dimensional space, and their respective labels  $\{y_j\}_{j=1}^N$  where  $\forall j, y_j \in \{-1, +1\}$ . You will assign a scalar weight to each input vector. Before the first iteration of Adaboost, these weights will be set to be equal in value. In each round  $t$ , you will pick the best classifier from the following family of weak classifiers:  $h_t(\mathbf{x}; i, p, \theta) = \text{sign}(p(x_i - \theta))$  where  $x_i$  is the  $i^{\text{th}}$  element of  $d$ -dimensional input vector  $\mathbf{x}$ , the parameter  $p \in \{-1, +1\}$  and  $\theta$  is a real-valued threshold parameter. Basically, this classifier assigns input vector  $\mathbf{x}$  the label '+1' if either (1)  $x_i > \theta$  and  $p = +1$ , or (2)  $x_i \leq \theta$  and  $p = -1$ . Otherwise it assigns  $\mathbf{x}$  the label '-1'. The best classifier refers to the classifier producing the least weighted error on the training set, i.e. least value of  $\epsilon = \sum_{j=1}^N w_j I(h_t(\mathbf{x}_j) \neq y_j)$  where  $I(\cdot)$  is an indicator function that returns 1 if the predicate passed as a parameter is true, and returns 0 otherwise. Note that the search for the best classifiers involves picking the tuple  $(i, p, \theta)$ . After picking the best classifier, you will update the weights following the method in the standard Adaboost algorithm. This entire process is repeated for  $T$  rounds. You need to specify the value of  $T$  but  $T = 30$  to  $40$  is sufficient. The final classifier after  $T$  rounds will have the form  $H(\mathbf{x}) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}))$ .

You will work with the following datasets.

- (a) A dataset containing 2000 points in 2D drawn from a  $[0,1]$  bounded uniform random distribution. Label all the points lying on or inside a rectangle bounded by the lines  $x = 0.3$ ,  $x = 0.7$ ,  $y = 0.3$ ,  $y = 0.7$  as +1 and the rest as -1. Randomly divide this dataset into disjoint sets of 1000 training points and 1000 test points (called dataset1).
- (b) A dataset containing 2000 points in 2D drawn from a  $[0,1]$  bounded uniform random distribution. Label all the points satisfying any of the following conditions as '+1' and the rest as '-1': (1) lying on or inside a rectangle bounded by the lines  $x = 0.3$ ,  $x = 0.7$ ,  $y = 0.3$ ,  $y = 0.7$  as +1, (2) with x-coordinate between 0.15 and 0.25 or between 0.75 and 0.85, (3) with y-coordinate between 0.15 and 0.25 or between 0.75 and 0.85. Randomly divide this dataset into disjoint sets of 1000 training points and 1000 test points (called dataset2).
- (c) A dataset containing 2000 points in 2D drawn from a zero-mean Gaussian distribution of standard deviation 2. Label all the points whose distance from the origin is less than 2 as +1 and the rest as

- 1. Randomly divide this dataset into disjoint sets of 1000 training points and 1000 test points (called dataset3).
- (d) A dataset containing 2000 points in 2D drawn from a zero-mean Gaussian distribution of standard deviation 2. Label all the points whose distance from the origin is either less than 2, or between 2.5 and 3, as +1 and the rest as -1. Randomly divide this dataset into disjoint sets of 1000 training points and 1000 test points (called dataset4).
- (e) The MNIST database is a popular dataset containing images of handwritten digits from 0 to 9. It can be downloaded from <http://yann.lecun.com/exdb/mnist/>. The dataset contains four files each in 'idx' format, namely train-images-idx3-ubyte.gz which contains the training data consisting of 60000 images of size 28 by 28 each, train-labels-idx1-ubyte.gz which contains the respective labels of the training images, t10k-images-idx3-ubyte.gz which contains 10000 images of size 28 by 28 each for testing, with their respective labels (for ground truth) marked out in t10k-labels-idx1-ubyte.gz. A MATLAB script for reading these files into memory is uploaded here: [https://www.cse.iitb.ac.in/~ajitvr/CS763\\_Spring2017/HW4/readMNIST.m](https://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/HW4/readMNIST.m). You will need to gunzip each of the four files before calling this (parameter-free) function readMNIST. You should perform training on the first 5000 images from the training set.

In the first experiment on this dataset, label the images belonging to the digit '2' as '+1' and all the others as '-1'. Thus your job is to determine whether a given image contains the selected digit '2' or not.

In the second experiment, label the images belonging to the digits '2' and '3' as +1 and all others as '-1'. Thus your job is to determine whether a given image contains the selected digit '2' or '3' or not.

For each of the five datasets (totally 6 experiments), do the following after each round of Adaboost: (1) estimate and print the training error of the strong classifier created thus far, (2) estimate and print the error of the strong classifier created thus far on the test set. For the first four datasets, also plot the test points with their associated labels using the MATLAB function called 'scatter' (in each round of Adaboost). Sample scatter plots for the first four datasets can be found at [http://www.cse.iitb.ac.in/~ajitvr/CS763\\_Spring2017/HW4/](http://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/HW4/) (named as dataset1.jpg and so on). You do not need to include the scatter plot images from each iteration in your report. Finally, for all five datasets (totally 6 experiments), plot a graph of the test set error versus the number of rounds of Adaboost and include it in your report. Do you notice something peculiar with the fourth dataset? Explain what you would do to remedy that situation (there is no need to implement). [10+10+10+10+40+10+10 = 100 points]