Assignment 4: CS 763, Computer Vision

Due 29th March 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^{th} element of d-dimensional input vector $\mathbf{x}$, the parameter $p \in \{-1,+1\}$ and θ 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 $I(.)$ 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, θ). 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_Spring2016/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. Label the images belonging to the digit '2' as '+1' and all the others as '-1'. Thus for this database, your job is to determine whether a given image contains the selected digit '2' or not.

For each of the five datasets, 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_Spring2016/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, 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 you do to remedy that situation (there is no need to implement). [4+4+4+4+10+4 = 30 points]

Answer: See sample code in the files adaboost.m and adaboost_digits2.m in the homework folder.

The fourth dataset is peculiar because even after several rounds (more than 800), the error does not reduce below 17%. The axis-parallel classifiers (in some books, these are called 'decision stumps') may be too weak to separate between the two classes - as one of the classes consists of points inside a circle and an outer ring. The situation may be remedied by using a slightly 'stronger' weak classifier - for example, you could use classifiers whose boundaries make various angles with the X and Y axes as opposed to only axis-parallel classifiers, or you could use classifiers of the following form: $h_t(x;\theta) = 1$ if $\|x\|^2 \leq \theta, h_t(x;\theta) = -1$ if $\|x\|^2 > \theta$ (i.e. you put a threshold on the squared distance from the origin).

2. In this exercise, we will see a beautiful application: recovering the structure of a wavy water surface under the influence of wind (under certain assumptions), and also reconstructing a distortion-free image of any objects under the water surface. To this end, consider a water-tank with a still, perfectly horizontal water surface of height h_0 above the tank-floor (= XY plane). A ray of light $\mathbf{s} = (0,0,-1)$ hits the water surface at point P and goes through without refraction striking the tank-floor at point Q. However, usually, the water surface is wavy (say, due to wind) and will not have a constant height. We will denote the height and surface normal at any surface point as $h(x,y)$ and $\mathbf{n}(x,y) = (n_x(x,y), n_y(x,y), n_z(x,y))$ respectively. The ray $\mathbf{s}$ will now get refracted at P and strike the tank-floor at point $Q' \neq Q$. Let $\mathbf{r}$ be the refracted ray, α be the angle of incidence, i.e. angle between $\mathbf{n}$ and $\mathbf{s}$, and β be the angle of refraction, i.e. angle between $\mathbf{n}$ and $\mathbf{r}$. We know that (a) $\mathbf{r}$ lies in the plane spanned by $\mathbf{s}$ and $\mathbf{n}$, and that (b) $\frac{\sin\alpha}{\sin\beta}$ = refractive index of water = κ. We will assume that the water surface acts as a purely transparent surface and that there are no reflections or specularities (this is a simplifying assumption that is not valid in many real-life scenarios). Draw a simple ray diagram and answer the following:

(a) Write the mathematical relation between $\mathbf{n}(x,y)$ and $h(x,y)$.
Ans: $\mathbf{n}(x,y) = \mathbf{t}_x(x,y) \times \mathbf{t}_y(x,y)$ where $\mathbf{t}_x(x,y) = (1,0,h_x(x,y))$ and $\mathbf{t}_y(x,y) = (0,1,h_y(x,y))$ are

tangent vectors in the X and Y directions. Thus, $\mathbf{n}(x,y) = (-h_x(x,y), -h_y(x,y), 1)$.

(b) Let $\mathbf{r}_x$ and $\mathbf{r}_y$ be the x and y components of the vector $\mathbf{r}$. Prove that $(\mathbf{r}_x, \mathbf{r}_y)$ is parallel to $(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y})$.
Ans: As $\mathbf{r}, \mathbf{s}, \mathbf{n}$ are coplanar, we have $\mathbf{r} = a\mathbf{s} + b\mathbf{n}$ where a and b are scalars. This gives us $\mathbf{r}_x = b\mathbf{n}_x = -bh_x, \mathbf{r}_y = b\mathbf{n}_y = -bh_y$ since $\mathbf{s} = (0,0,01)$. Thus we can conclude that:

$$(\mathbf{r}_x, \mathbf{r}_y) = -b(h_x, h_y) = -b\nabla h. \tag{1}$$

(c) Express the magnitude of vector QQ' in terms of α, β and h_0. What is the relation between QQ' and $\mathbf{r}$?
Ans: QQ' is same as the vector $(\mathbf{r}_x, \mathbf{r}_y)$, the projection of the refracted ray on the XY plane (i.e. the tank-floor). By drawing a ray diagram one can see that $\|QQ'\| = h(x,y)\tan(\alpha-\beta) \approx h_0 \tan(\alpha - \beta)$. The last approximate equality follows if we assume that change in height due to fluctuations is small at any point.

(d) Now, let us assume that the water surface fluctuations are very small. Hence α, β and change in height are very small. Use this to prove that $(\mathbf{r}_x, \mathbf{r}_y) = h_0(1 - 1/\kappa)(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y})$.
Ans: $\|QQ'\| = h_0 \tan(\alpha - \beta) = h_0 \frac{\tan\alpha - \tan\beta}{1 + \tan\alpha\tan\beta}$. As α and β are small, $\tan\alpha \approx \sin\alpha$ and same for β. Hence $\|QQ'\| = h_0 \frac{\sin\alpha - \sin\beta}{1 + \sin\alpha\sin\beta} = h_0 \frac{\sin\alpha - \sin\alpha/\kappa}{1 + \sin^2\alpha/\kappa} = h_0 \sin\alpha(1 - \frac{1}{\kappa})$. The last equality follows as $\sin^2\alpha \approx 0$. But $\sin\alpha = \sqrt{1 - \cos^2\alpha}$ and $\cos\alpha = (0,0,1) \cdot \frac{(n_x, n_y, n_z)}{\sqrt{n^x + n^2 y + n_z^2}} = \frac{n_z}{\sqrt{n_x^2 + n_y^2 + n_z^2}}$. Hence we have

$$\|QQ'\| = h_0(1 - \frac{1}{\kappa})\frac{\sqrt{n_x^2 + n_y^2}}{\sqrt{n_x^2 + n_y^2 + n_z^2}} = h_0(1 - \frac{1}{\kappa})\frac{\sqrt{h_x^2 + h_y^2}}{\sqrt{h_x^2 + h_y^2 + 1}} \approx h_0(1 - \frac{1}{\kappa})\sqrt{h_x^2 + h_y^2}. \tag{2}$$

The last approximate equality is because $h_x \approx h_y \approx 0$ as the fluctuations are small, and hence the denominator is very negligibly greater than 1. From equations 1 and 2, we can see that $b = -h_0(1 - \frac{1}{\kappa})$. This gives us:

$$(\mathbf{r}_x, \mathbf{r}_y) = h_0(1 - \frac{1}{\kappa})(h_x, h_y). \tag{3}$$

The beauty of this is that it tells us that the image distortion QQ' is parallel to $(\mathbf{r}_x, \mathbf{r}_y)$ and that in turn is directly proportional to ∇h.

(e) Now, given the video of a static scene on the tank floor beneath a wavy water surface undergoing small fluctuations, suggest a method to recover $h(x,y)$ at any time time instant. You may assume that the height at any point averaged over time is h_0. [3 + 3 + 4 + 4 + 6 = 20 points]
Ans: Let the superscript t denote the t^{th} frame, $1 \leq t \leq T$. Let us suppose we track a given image point L across all T frames of the video. Let the coordinates of the corresponding points in the different frames be $L_1 = (x_1, y_1), L_2 = (x_2, y_2), ..., L_T = (x_T, y_T)$. Let point $\bar{L} = (\bar{x}, \bar{y})$ where $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t, \bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$. The height of the point on the water surface directly above L averaged over time is h_0, i.e. the height gradient averaged over time is 0. Therefore, we can conclude that the average apparent distortion of the physical entity at L is zero. This is because we have seen that the 2D distortion vector at any time t is parallel to the gradient of the height function, i,.e. parallel to $(h_{t,x}, h_{t,y})$. Now, from Equation 3, we have $L_t\bar{L} = h_0(1 - \frac{1}{\kappa})(h_{t,x}, h_{t,y})$. We see here that $\bar{L}$ plays a similar role as Q. We know $L_t\bar{L}, h_0, \kappa$ and hence can find the surface normals at any time instant. The surface normals can then be integrated to obtain the height h_t at time frame t.

Useful formulae: $\sin(A+B) = \sin A\cos B + \cos A \sin B$; $\cos(A+B) = \cos A\cos B - \sin A \sin B$; $\tan(A+B) = \frac{\tan A + \tan B}{1 - \tan A \tan B}$; $\tan(A-B) = \frac{\tan A - \tan B}{1 + \tan A \tan B}$. For small A, $\sin A \approx \tan A$.