CS534 — Homework Assignment 1 — Due Mon April 11, 2011

Written assignment

1. Let X and Y be two independent random variables, show that:
   1. For any constants a and b, $E(aX + bY) = aE(X) + bE(Y)$
   2. For any constants a and b, $Var(aX + bY) = a^2Var(X) + b^2Var(Y)$
   3. $Cov(X, Y) = 0$

2. Let X and Y be two independent uniformly distributed random variables. $X, Y \sim U(0, 1)$. Let $Z = max(X, Y)$. What is the PDF function of Z?

3. We have three boxes colored Red, Blue and Green respectively. There are 3 apples and 6 oranges in the red box, 3 apples and 3 oranges in the green box, and 5 apples and 3 oranges in the blue box. Now we randomly select one of the boxes (with equal probability) and grab a fruit from the box. What is the probability that it is an apple? If the fruit that we got is an orange, what is the probability that we selected the green box?

4. (Probability Decision Boundary). Consider a case where we have learned a conditional probability distribution $P(y|\mathbf{x})$. Suppose there are only two classes, and let $p_0 = P(y = 0|\mathbf{x})$ and $p_1 = P(y = 1|\mathbf{x})$. Consider the following loss matrix:

predicted label $\hat{y}$	true label y	
	0	1
0	0	10
1	5	0

   It can be shown that the decision $\hat{y}$ that minimizes the expected loss is equivalent to setting a specific probability threshold θ and predicting $\hat{y} = 0$ if $p_1 < \theta$ and $\hat{y} = 1$ if $p_1 \geq \theta$. Please compute the θ for the above given loss matrix. Show a loss matrix where the threshold is 0.1.

5. (Reject Option). In many applications, the classifier is allowed to "reject" a test example rather than classifying it into one of the classes. Consider, for example, a case in which the cost of a misclassification is \$10 but the cost of having a human manually make the decision is only \$3. We can formulate this as the following loss matrix:

decision	true label y	
	0	1
predict 0	0	10
predict 1	10	0
reject	3	3

   Suppose $P(y = 1|\mathbf{x})$ is predicted to be 0.2. Which decision minimizes the expected loss? Show that in cases such as this there will be two specific thresholds θ_0 and θ_1 such that the optimal decision is to predict 0 if $p_1 < \theta_0$, reject if $\theta_0 \leq p_1 \leq \theta_1$, and predict 1 if $p_1 > \theta_1$.

decision	true label y	
	0	1
predict 0	0	10
predict 1	5	0
reject	3	3

6. (Weighted hinge loss). In our derivation of the Perceptron algorithm, we used the hinge loss to approximate the 0/1 loss. Suppose that we have a general loss matrix with the cost of a false positive being $L(1,-1) = c_0$ and the cost of a false negative $L(-1,1) = c_1$. Suppose we used

$$\tilde{J}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} z_i \max(0, -y_i \mathbf{w} \cdot \mathbf{x}_i)$$

for our approximate objective function, where $z_i = c_0$ if $y = -1$ and $z_i = c_1$ if $y = 1$. Compute the gradient using this approximation, and show how the batch Perceptron algorithm is modified to incorporate this change.

7. In class when discussing linear regression, we assume that the Gaussian noise is independently identically distributed. Now we assume the noises $\epsilon_1, \epsilon_2, \cdots \epsilon_N$ are independent but each $\epsilon_i \sim N(0, \sigma_i^2)$. Please 1) write down the log likelihood function of $\mathbf{w}$; 2) show that minimizing the log likelihood is equivalent to minimizing a weighted least square loss function $J(\mathbf{W}) = \sum_{\mathbf{i}} \mathbf{a_i}(\mathbf{y_i} - \mathbf{x_i^T w})$, and express each a_i in terms of ϵ_i; and 3) solve for $\mathbf{w_{MLE}}$.

Implementation assignment

In this assignment, you will code the *batch perceptron* and *voted perceptron* algorithms. Matlab is preferred but your implementation can be in any language of your choice. You need to submit your source code electronically together with a write-up including the contents specified below. Your implementation must contain:

1. a p_train function, which will take the training data as input, and output a weight vector for the final decision boundary.
2. a p_classify function, which will take an example x and weight vector w, and predict y.
3. a vp_train function, which will take as inputs the training data and the total number of epoches, and output a collection of weight vectors and their corresponding c's.
4. a vp_classify function{}, which will take an example x and a collection of vectors with their weights, and predict y.

Note that the training function should perform a random shuffle of the input training examples in each training epoch, such that the learning is not stuck with a particular ordering of the training examples. You will be provided with two data sets (will be posted soon). You will test your batch perceptron algorithm on the first data set (twogaussian), which contains two linearly separable gaussian classes. Your write up needs to include:

1. A plot of the classification error on the training set as a function of the number of training epoches. Note that one epoch of training goes through the full training set exactly once.
2. A scatter plot of the training data using different colors for different classes. On this scatter plot, please plot the final linear decision boundary learned by your perceptron algorithm (please also provide the weights output by your perceptron algorithm).

You will test your voted perceptron implementation on the second data set iris-twoclass. This is based on a commonly used bench-mark data set iris. In particular, we extracted two classes, and two input features from the original problem. You need to include the following in your write-up:

1. A plot of the classification error on the training set as a function of the number of training epoches (up to 100 epoches). Note that one epoch of training goes through the full training set exactly once.
2. Please visualize the final decision boundary produced by your voted perceptron algorithm (trained for 100 epoches). Note that to produce this decision boundary, one strategy is to sample a large number of $\mathbf{x}$ points on a fine grid in the input space, and compute their predicted output. You can then visualize the decision boundary by plotting these points using different colors based on their predicted classes.
3. Note that for voted perceptron, there is a simple modification people often use in order to avoid storing all intermediate weight vectors. That is to compute

$$\mathbf{w}_{avg} = \sum_{n=0}^{n=N} c_n \mathbf{w_n}$$

and use the following decision rule:

$$h(\mathbf{x}) = sgn\{\mathbf{w}_{avg} \cdot \mathbf{x}\}$$

Please compute $\mathbf{w}_{avg}$ based on the weights that are learned by your voted perceptron algorithm (100 epoches), and plot the linear decision boundary it produces on a scatter plot of the training data. Are these two decision boundaries equivalent? Why?