10-701 Machine Learning - Spring 2012
Problem Set 2

Out: February 15th
In: February 29th

TA: Yang Xu (yx1@cs.cmu.edu)
School Of Computer Science, Carnegie Mellon University

Homework will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in figuring out answers and helping each other solve the problems. We will be assuming that, as participants in a graduate course, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration. You also must indicate on each homework with whom you collaborated. Homework is due at the beginning of class on the due date.

1 Logistic regression [25 points]

1.1 Logistic *vs* linear regression [7 points]

In both logistic (LR) and linear regressions (R), given input X, the goal is to predict the response Y. The difference is that LR is typically used for classification whereas R is used for regression.

1. Propose a simple modification to R that makes it amenable to classification (instead of regression) tasks. Comment on whether such a proposal is superior than LR or not and briefly explain why. [**3 points**]

2. Recall in R where $y = wx$ (w being the estimated linear coefficient for a one-dimensional variable x), a unit change in x would induce a multiplicative w change in y. In LR, y and wx are linked by the sigmoid function. Explain how you would interpret the w coefficients in logistic regression. Suppose $w = 2$, calculate the change in the odds of the classes induced by a unit change in x, assuming there are two available classes. [**4 points**]

1.2 Logistic *vs* naive Bayes [12 points]

Suppose in a binary classification problem, the input variable $X = [x^1, ..., x^M]$ is M-dimensional and the response variable Y is a class indicator (0 or 1). In this section, you will work in steps to establish a connection between logistic regression and Gaussian naive Bayes.

1. Write down expressions of the class conditional probability for each class, $P(Y = 1|X)$ and $P(Y = 0|X)$, for logistic regression. [**2 points**]
2. Using the Bayes rule, derive the posterior probabilities for each class, $P(Y = 1|X)$ and $P(Y = 0|X)$, for naive Bayes. [**2 points**]
3. Assuming a Gaussian likelihood function in each of N dimensions, write down the full likelihood $f(X|Y)$ for naive Bayes. [**2 points**]
4. Assuming a uniform prior on the two classes and using the results from part 2 and 3, derive a full expression for $P(Y = 1|X)$ for naive Bayes. [**2 points**]
5. Show that with appropriate manipulation and parameterization, $P(Y = 1|X)$ in naive Bayes from part 4 is equivalent to $P(Y = 1|X)$ for logistic regression in part 1. [**4 points**]

1.3 Loss function [6 points]

Write down the loss function, or the negative log likelihood, for logistic regression. Denote y as the class indicator, x as the predictor vector, w as the coefficient vector and N as the number of data points. Derive the derivative of the loss function with respect to w (*hint: first derive the derivative of the sigmoid function* $\sigma(u)$ *with respect to a generic input* u.).

2 Learning theory [20 points]

2.1 PAC learning [16 points]

Imagine yourself as an apprentice chef in a restaurant. Your first task is to figure out how to make a salad. The rules are supposedly simple: 1) you are free to combine any of the ingredients as they are 2) you can also slice any of the ingredients into two distinct pieces before mixing them. Since you have learnt PAC learning theory, you wonder how much effort you would need to figure out the makeup in a salad.

1. [**5 points**] Suppose that a naive chef makes salads following only rule 1. Given N available ingredients and that each salad made out of these constitutes a distinct hypothesis. How large would the hypothesis space be? Explain how you arrive at your answer.
2. [**3 points**] Suppose that a more experienced chef follows both rules when making a salad. How large is the hypothesis space now? Explain.
3. [**8 points**] An experienced chef decides to train you to discern the makeup of a salad by showing you the salad samples he has made. There are 6 available ingredients. If you would like to learn any salad at 0.01 error with probability 99%, how many sample salads would you want to see? Show your workings in clear steps.

2.2 VC dimensions [4 points]

Consider a 2D space or $x1$–$x2$ plane. What is the VC dimension of circles where points inside are labeled as 1's and those outside as 0's? Draw an example scenario with minimal number of points where these circles would fail to shatter the space.

3 Mistake bounds [15 points]

Suppose you have a team of N robot agents and you wish to train them to help you make predictions in life. As a simple start, the prediction will be based on majority votes from these agents. To assure that they won't fail you in crucial tasks, you went on to analyze their prediction mistakes. Soon you find that their mistakes are curiously related to that of the best agent ...

Your strategy of training these agents on a binary classification problem is as follows:

1. Initialize all robots with equal weight $w_i = 1$ for $i = 1...N$.

2. Since each robot makes a prediction of either class ($y_i = 0$ or 1), the ensemble prediction follows the weighted majority and predicts 1 if

$$\sum_{i=1}^{N} w_i I(y_i = 1) \geq \sum_{i=1}^{N} w_i I(y_i = 0) \tag{1}$$

   and otherwise 0, where $I(\cdot)$ is the indicator function and equals 1 if its argument is true.

3. If any robot makes a mistake, you penalize them by reducing their weights by a half.

4. Go to step 2.

You discover that your best agent makes M_b prediction mistakes while the ensemble agent makes M_e mistakes. You are now going to figure out how these numbers are related. In other words, you are going to find an upper bound for M_e in terms of M_b.

1. **[3 points]** What would the weight of the best agent be after making M_b prediction errors and why? Let's denote it as w_b.

2. **[4 points]** What is the maximal ensemble weight ($\sum w_i$) after making M_e errors? Let's denote it as W_{max}.

3. [**2 points**] Write a simple inequality that relates part 1 (w_b) and 2 (W_{max}).
4. [**6 points**] Using the equality in part 3 and your solutions to part 1 and 2, derive an upper bound for the ensemble mistake M_e in terms of the mistake from the best agent M_b.

4 Guess the lean animal [40 points]

In the animal kingdom, there are lean candidates such as the monkey and chubby ones such as the giant panda. In this section, you will develop a classifier that predicts whether an animal is lean or not given some of its properties.

1. Load "LeanAnimals.mat" in MATLAB. You should see the following: "names" for animals, "properties" for properties of animals, "labs" for indicators of leanness and "D" for relational matrix (animals *vs* properties). Sort "labs" into groups of 0s and 1s and sort the rows in D accordingly. How many animals are lean? Plot the sorted matrix D in black and white using *imagesc* command. [**2 points**] Formulate the problem using logistic regression given that the goal is to predict whether an animal is lean or not. Specify clearly your inputs and outputs, and write down the expressions for the class conditional probabilities. [**4 points**]
2. Write a generic logistic regression classifier. Attach your MATLAB codes for the LR classifier ONLY in compact format in the writeup. [**15 points**]
3. Apply your LR classifier to the data and perform leave-one-out cross-validated (LOOCV) predictions on the animals. In other words, at each round you would first train the classifier on 49 animals, and predict whether the held-out animal is lean or not. Report your classification accuracies for the lean and non-lean classes in percentage separately. [**4 points**]
4. Now, instead of LOOCV, fit your classifier on the entire dataset "D". This should return a single set of weights. List them and interpret them for properties 2 to 6–do they make sense, why or why not? [**5 points**] The annotation for property 1 is missing–can you guess what property it might be given your estimated weight? [Note: no credits would be deducted or granted here so don't agonize if you are stuck.]
5. Using the "corr" function in MATLAB, compute the correlation coefficients between each of the properties and "labs"–tabulate these. Do these match well with your estimated weights in part 6–explain briefly why or why not. [**4 points**]
6. From your outputs in part 4, you should be able to compute $p(lean|animal)$ for each animal. Rank the animals by sorting their class conditionals in descending orders. You should produce a table that consist of two columns: Animal Name (sorted), Conditional Probability (p(lean—animal)). Is this how you would sort these animals? [**6 points**]