Homework 2

Linear Regression, Perceptron, Decision Trees and Neural Networks

CMU 10-601: Machine Learning (Fall 2015)
http://www.cs.cmu.edu/~10601b/
OUT: Sep 23, 2015
DUE: Oct 1, 2015, 10:20 AM

START HERE: Instructions

- The homework is due at 10:20 am on Thursday October 1, 2015. Each student will given two late days that can be spent on any homeworks but not on projects. Once you have used up your late days for the term, late homework submissions will receive 50% of the grade if they are one day late, and 0% if they are late by more than one day.
- ALL answers will be submitted electronically through the submission website: https://autolab.cs.cmu.edu/10601-f15. You can sign in using your Andrew credentials. You should make sure to edit your account information and choose a nickname/handle. This handle will be used to display your results for any competition questions (such as the class project) on the class leaderboard.
- There are no autograded questions on this homework.
- Collaboration on solving the homework is allowed (after you have thought about the problems on your own). When you do collaborate, you should list your collaborators! You might also have gotten some inspiration from resources (books or online etc...). This might be OK only after you have tried to solve the problem, and couldn't. In such a case, you should cite your resources.
- If you do collaborate with someone or use a book or website, you are expected to write up your solution independently. That is, close the book and all of your notes before starting to write up your solution. You should also state your collaborations in your short-answer writeup. Specifically, please write down the following:

  1. Did you receive any help whatsoever from anyone in solving this assignment? Yes / No. If you answered yes, give full details: (e.g., "Jane explained to me what is asked in Question 3.4").
  2. Did you give any help whatsoever to anyone in solving this assignment? Yes / No. If you answered yes, give full details: (e.g., "I pointed Joe to section 2.3 to help him with Question 2").

  Collaboration without full disclosure will be handled severely, in compliance with CMU's Policy on Cheating and Plagiarism.

1 Linear Regression [Calvin; 40 points]

1.1 Mathematical Analysis of Linear Regression [8pts]

Suppose data are generated from the following model:

$$y = \mathbf{X}\beta + \epsilon, \qquad \text{with} \;\; \epsilon \sim \mathcal{N}(0, \sigma^2 I_n) \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n \times p}$, $\beta \in \mathbb{R}^p$, and $\sigma > 0$ are fixed constants and $y, \epsilon \in \mathbb{R}^n$ are random vectors. In this question we will analytically study the distribution of the least squares estimator

$$w = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y \tag{2}$$

when data are generated by the model given above. Given the model, the expectation and variance-covariance matrix of w, the MLE estimate, are given as:

$$\begin{aligned}\mathbf{E}[w] =& \beta \\ \mathbf{Cov}[w] =& \sigma^2 \left[(\mathbf{X}^T\mathbf{X})^{-1} \right].\end{aligned}$$

(Intuitively, this means that because w is a function of random y, it too is a random vector. The equation $\mathbf{E}[w] = \beta$ indicates that the distribution of w is centered around the true weight vector β. Meanwhile, the equation $\mathbf{Cov}[w] = \sigma^2 \left[(\mathbf{X}^T\mathbf{X})^{-1}\right]$ indicates that how w varies around β will depend on σ and $\mathbf{X}$.)

1. Show that $\mathbf{E}[w] = \beta$. (Hint 1: Remember that the only random variables in the model are ϵ, y, and w. Hint 2: You'll need to use the linearity of expectation, which was discussed in the Probability Review of Recitation 1.)

2. Suppose two columns (ie features) in $\mathbf{X}$ are identical. Would $\mathbf{Cov}[w]$ be well-defined? What about if two columns were opposites (i.e. $\mathbf{X}_i = -\mathbf{X}_j$ for $i \neq j$)? Answer each with yes/no and explain in one sentence maximum.

1.2 Empirical Investigation of Linear Regression [32pts]

In this section, we empirically investigate how the least squares estimator is affected by collinearity. We will see that with collinearity, where input features are strongly correlated (or negatively correlated), it is harder to estimate the true weight vector and to generalize to new data.

1. [10pts] Suppose we have fixed $\beta = [0 \quad 1]^T, \sigma = 1$, and

$$\mathbf{X} = \begin{bmatrix} -1 & -1.5 \\ 0 & 0.5 \\ 1 & 0.5 \\ 2 & 1.5 \end{bmatrix}$$

   - What is the condition number of $\mathbf{X}^T\mathbf{X}$? (A large condition number means the matrix is almost noninvertible. You can call cond() in Matlab.)
   - What is $\mathbf{E}[w]$? What is $\mathbf{Cov}[w]$? (See Question 1.1.)

     Now, simulate 50 independent y datasets using the model in Equation (1). Note that each dataset will consist of 4 y_i samples, since $\mathbf{X}$ consists of 4 samples. For each dataset, compute the least-squares estimate w using Equation (2), and the error $\|w - b\|_2$. Since we have 50 datasets, you should now have 50 (scalar) error measurements, and 50 2-dimensional w's.
   - Using the collected data, compute the empirical mean of w and the empirical covariance of w.
   - Plot a histogram of the 50 errors (eg. using hist() in Matlab). Because w is 2-dimension, you can create a scatter plot of the 50 w's, where the x-axis contains w_1 for each w and the y-axis contains w_2. Plot it (eg. using scatter() in Matlab).
   - Create two QQ-plots for w_1 and w_2 (eg. using qqplot() in Matlab).

2. [10pts] Repeat the previous experiment for

$$\tilde{\mathbf{X}} = \begin{bmatrix} -1 & -1 \\ 0 & 0 \\ 1 & 1 \\ 2 & 2.1 \end{bmatrix}$$

3. [4pts] Suppose we have test sample $\mathbf{x}_{test} = [2 \quad 0]$. Note that $\mathbf{E}[y_{test}] = 0$. Let's compare this to how our w estimates predict y_{test}.

   - Plot a histogram of the 50 predictions for y_{test} using the 50 w estimates from Part 1.
   - Repeat using the 50 w estimates from Part 2.

4. [8pts] Answer each of the following short answer questions:

   - Were $\mathbf{E}[w]$ and $\mathbf{Cov}[w]$ approximately equal to their empirical estimates? Answer yes/no and explain in one sentence or less.

- What do the scatterplots and QQ-plots tell us about the distribution of w? (What is its shape and where is it centered?) Answer in one sentence maximum.
- Which experiment produced better behavior for w? Answer "Experiment 1" or "Experiment 2", and explain in one or two sentences.
- In both experiments we have w_1 and w_2 negatively correlated with each other. If we replace $\mathbf{X}_2$ with $-\mathbf{X}_2$ in the $\mathbf{X}$ matrix, would w_1 and w_2 still be negatively correlated? (Hint: Are $\mathbf{X}_1$ and $\mathbf{X}_2$ positively or negatively correlated with each other?) Answer yes/no and explain in one or two sentences.

Along with the outputs and plots described above, you must include in the PDF all your code to receive full credit.

2 Decision Trees [Aman; 35 points]

1. Information Gain (10 pts)

   Consider the following set of training examples:

Table 1: Training examples

	Classification	X_1	X_2
1	+	T	T
2	+	T	T
3	-	T	F
4	-	F	F
5	-	F	T
6	-	F	F

   (a) What is the entropy of this collection of training examples with respect to the column classification? Briefly write the steps of your calculation.

   (b) What is the information gain of X_1 relative to these training examples? Briefly write the steps of your calculation.

2. Drawing decision trees [10 pts]

   Consider the dataset in table 2 for classifying whether a type of food is appealing or not

Table 2: Food dataset

Appealing	Temperature	Taste	Size
No	Hot	Salty	Small
No	Cold	Sweet	Large
No	Cold	Sweet	Large
Yes	Cold	Sour	Small
Yes	Hot	Sour	Small
No	Hot	Salty	Large
Yes	Hot	Sour	Large
Yes	Cold	Sweet	Small
Yes	Cold	Sweet	Small
No	Hot	Salty	Large

   (a) What is the information gain associated with choosing *Temperature* as root? What about *Taste* and *Size*? Briefly write the steps of your calculations.

(b) Draw a decision tree for the dataset by choosing appropriate features for splitting. Justify each split.

3. Decision tree theory [15 pts]

   (a) Consider an arbitrary dataset D, with the label space containing m unique labels. A decision tree is created on the dataset. What is the maximum training error you could get on such a dataset? Explain your answer briefly in one or two sentences.

   (b) Consider two decision trees that always yield the same class label, given the same test sample. Do both the trees need to have the same number of nodes? Justify your answer with an example in one or two sentences.

   (c) Can a decision tree perfectly classify any linearly separable dataset? Justify your answer briefly in one or two sentences.

3 Perceptron and Neural Nets [25 points]

3.1 Perceptron [Aman; 15 points]

1. Bounds (10 pts)

   Suppose D $= (x_1, y_1), ..., (x_T, y_T)$ is a dataset where

   - Each x_i is a sparse vector $(b_1, ..., b_{100,000})$ representing a document d_i, where $b_w = 1$ if word w is in the document d_i , and $b_w = 0$ otherwise; Each d_i has no more than 100 English words (and for the purpose of this question there are no more than 100,000 English words)
   - y_i is a label (+1 or 1) indicating the class of the document d_i;
   - there exists some vector u such that $\forall i, y_i \boldsymbol{u}.x_i > 1$

   Write down an upper bound on R, the maximum distance of an example x_i from the origin, and m, the number of mistakes made by a perceptron before converging to a correct hypothesis v_k, when trained on-line using data from D

2. Can a single perceptron compute the XOR function? Answer yes/no and explain in one or two sentences. (5 pts)

3.2 Neural Networks [Calvin; 10 points]

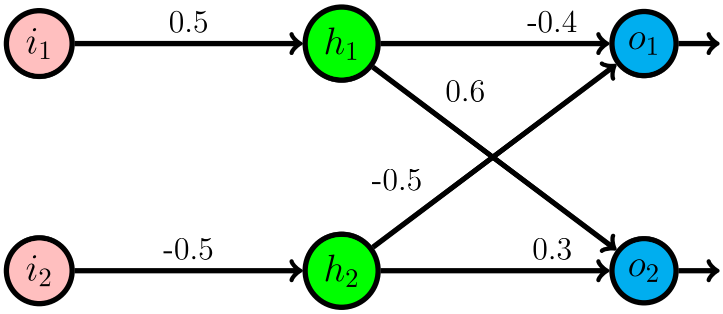

Figure 1: Neural network with one hidden layer.

In this question, we simulate backpropagation on a (very simple) feedforward neural network. The questions below refer to the neural net in Figure 1. All units have sigmoid activation functions with no bias term, so if unit z gets inputs $x_1, \ldots, x_n$, its output is

$$\sigma(z) = \frac{1}{1 + \exp(-\sum_{i=1}^{n} w_{x_i,z} x_i)}.$$

For example, the output of h_1 is $\frac{1}{1+\exp(-w_{i_1,h_1} i_1)}$. Note that the input layer nodes simply output the inputs to the network. The edge labels denote the initial network weights (eg. $w_{h_1,o_1} = -0.4$).

Consider the case where the network is given an input of $i_1 = 1.0$ and $i_2 = 0.0$.

1. What value is output by node h_1?
2. What value is output by node h_2?
3. What value is output by node o_1?
4. What value is output by node o_2?

   Now assume the correct outputs are $t_1 = 1.0$ and $t_2 = 1.0$. Using the definitions for δ_i given in lecture:

5. What is δ_{o_1}?
6. What is δ_{o_2}?
7. What is δ_{h_1}?

   Now, using these values, and a learning rate of $\eta = 0.1$:

8. What is the new value of w_{h_1,o_1}?
9. What is the new value of w_{h_1,o_2}?
10. What is the new value of w_{i_1,h_1}?