CIS 520, Machine Learning, Fall 2015: Assignment 3
Due: Thursday, October 1st, 11:59pm, PDF to Canvas

Instructions. Please write up your responses to the following problems clearly and concisely. We encourage you to write up your responses using LaTeX; we have provided a LaTeX template, available on Canvas, to make this easier. **Submit your answers in PDF form to Canvas. We will not accept paper copies of the homework.**

Collaboration. You are allowed and encouraged to work together. You may discuss the homework to understand the problem and reach a solution in groups up to size **two students.** However, *each student must write down the solution independently, and without referring to written notes from the joint session.* **In addition, each student must write on the problem set the names of the people with whom you collaborated.** You must understand the solution well enough in order to reconstruct it by yourself. (This is for your own benefit: you have to take the exams alone.)

1 Linear Regression and LOOCV [30 points]

In the last homework, you learned about using cross validation as a way to estimate the true error of a learning algorithm. A solution that provides an almost unbiased estimate of this true error is *Leave-One-Out Cross Validation* (LOOCV), but it can take a really long time to compute the LOOCV error. In this problem, you will derive an algorithm for efficiently computing the LOOCV error for linear regression using the *Hat Matrix*. [1]

Assume that there are n given training examples, $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$, where each input data point X_i, has m real-valued features. The goal of regression is to learn to predict Y from X. The *linear* regression model assumes that the output Y is a weighted *linear* combination of the input features with weights given by $\mathbf{w}$, plus some Gaussian noise.

We can write this in matrix form by stacking the data points as the rows of a matrix X so that x_{ij} is the j-th feature of the i-th data point. Then writing Y, $\mathbf{w}$ and ϵ as column vectors, we can express the linear regression model in matrix form as follows:

$$Y = X\mathbf{w} + \epsilon$$

where:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, X = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1m} \\ x_{21} & x_{22} & \ldots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nm} \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}, \quad \text{and } \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Assume that ϵ_i is normally distributed with variance σ^2. We saw in class that the maximum likelihood estimate of the model parameters $\mathbf{w}$ (which also happens to minimize the sum of squared prediction errors) is given by the *Normal equation*:

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T Y$$

[1] Unfortunately, such an efficient algorithm may not be easily found for other learning methods.

Define $\hat{Y}$ to be the vector of predictions using $\hat{\mathbf{w}}$ if we were to plug in the original training set X:

$$\begin{aligned}\hat{Y} &= X\hat{\mathbf{w}} \\ &= X(X^TX)^{-1}X^TY \\ &= HY\end{aligned}$$

where we define $H = X(X^TX)^{-1}X^T$ (H is often called the *Hat Matrix*).
As mentioned above, $\hat{\mathbf{w}}$, also minimizes the sum of squared errors:

$$\text{SSE} = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Now recall that the Leave-One-Out Cross Validation score is defined to be:

$$\text{LOOCV} = \sum_{i=1}^{n}(Y_i - \hat{Y}_i^{(-i)})^2$$

where $\hat{Y}^{(-i)}$ is the estimator of Y after removing the i-th observation (i.e., it minimizes $\sum_{j\neq i}(Y_j - \hat{Y}_j^{(-i)})^2$).

1. **[2 points]** To begin with, we should consider when it is possible to compute $\hat{\mathbf{w}}$ in this framework.

   (a) **[1 point]** Suppose $m > n$. Is $\hat{\mathbf{w}}$ well-defined? Why or why not?
   Hint: Recall that the rank of a matrix is equal to the number of linearly independent rows, which is also equal to the number of linearly independent columns. Use the fact that for two matrices A and B which can be multiplied to form the product AB, it must be the case that $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$. Furthermore, recall that a square matrix is invertible if and only if it is full-rank.

   (b) **[1 point]** Suppose $m \leq n$. Give a condition on X which guarantees that $\hat{\mathbf{w}}$ will **not** be well-defined and explain why not. (Don't assume X is a square matrix.)

   For the rest of question 1, assume $\hat{\mathbf{w}}$ is well-defined.

2. **[3 points]** What is the complexity of computing the LOOCV score naively? (The naive algorithm is to loop through each point, performing a regression on the $n-1$ remaining points at each iteration.)

   Hint: The complexity of matrix inversion for a $k \times k$ matrix is $O(k^3)$. (There are faster algorithms out there but for simplicity we'll assume that we are using the naive $O(k^3)$ algorithm.)

3. **[3 points]** Write $\hat{Y}_i$ in terms of the elements of H and Y. You may find it useful to use shorthand such as H_{ab} to denote the entry in row a, column b of H.

4. **[4 points]** Show that $\hat{Y}^{(-i)}$ is also the estimator which minimizes SSE for Z where

$$Z_j = \begin{cases} Y_j, & j \neq i \\ \hat{Y}_i^{(-i)}, & j = i \end{cases}$$

   Hint: Try to start by writing an expression for the SSE of Z; it should look very similar to the definition of SSE for Y that was given in the introduction section of this question. Then, manipulate terms until you can argue that substituting $\hat{Y}^{(-i)}$ for $\hat{Z}$ would minimize this expression.

5. **[6 points]** Write $\hat{Y}_i^{(-i)}$ in terms of H and Z. By definition, $\hat{Y}_i^{(-i)} = Z_i$, but give an answer that includes both H and Z.

6. **[6 points]** Show that $\hat{Y}_i - \hat{Y}_i^{(-i)} = H_{ii}Y_i - H_{ii}\hat{Y}_i^{(-i)}$, where H_{ii} denotes the i-th element along the diagonal of H.
   Hint: Use the results from part 2 and 4. Substitute Z_i with Y_i and $\hat{Y}_i^{-i}$ by using its definition in part 3.

7. **[6 points]** Show that

$$LOOCV = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{Y}_i}{1 - H_{ii}} \right)^2$$

What is the algorithmic complexity of computing the LOOCV score using this formula? Note: We see from this formula that the diagonal elements of H somehow indicate the impact that each particular observation has on the result of the regression.

2 Logistic regression and Naive Bayes [20 points]

A common debate in machine learning has been over generative versus discriminative models for classification. In this question we will explore this issue by considering Naive Bayes and logistic regression.

1. **[4 points]** For input X and output Y, which of the following is the **objective function** optimized by (i) Naive Bayes, and (ii) logistic regression?

   (a) $\mathbf{Pr}(Y)/\mathbf{Pr}(X)$
   (b) $\mathbf{Pr}(X)/\mathbf{Pr}(Y)$
   (c) $\mathbf{Pr}(Y \mid X)$
   (d) $\mathbf{Pr}(Y)$
   (e) $\mathbf{Pr}(X)$
   (f) $\mathbf{Pr}(Y)\mathbf{Pr}(X)$
   (g) $\mathbf{Pr}(X, Y)$
   (h) None of the above (provide the correct formula in this case)

2. **[16 points]** Recall from the suggested reading that "the discriminative analog of Naive Bayes is logistic regression." This means that the parametric form of $P(Y \mid X)$ used by logistic regression is implied by the assumptions of a Naive Bayes classifier, for some specific class-conditional densities. In class you will see how to prove this for a Gaussian Naive Bayes classifier for continuous input values. Can you prove the same for binary inputs? Assume X_i and Y are both binary. Assume that $X_i \mid Y = j$ is $\mathsf{Bernoulli}(\theta_{ij})$, where $j \in \{0, 1\}$, and Y is $\mathsf{Bernoulli}(\pi)$.
   Hint: Start by using Bayes Rule and the assumptions of Naive Bayes to express the objective function for logistic regression in terms of the given quantities θ_{ij} and π.

3 Double-counting the evidence [30 points]

1. **[2 points]** Consider the two class problem where class label $y \in \{T, F\}$ and each training example X has 2 binary attributes $X_1, X_2 \in \{T, F\}$. How many parameters will you *need* to know/evaluate if you are to classify an example using the Naive Bayes classifier? Keep in mind that since the probability of all possible events has to sum to 1, knowing the probabilities of all except one event implies knowledge of the final event's probability already. (Don't include such final events in your count.)

2. **[2 points]** Let the class prior be $\mathbf{Pr}(Y = T) = 0.5$ and also let $\mathbf{Pr}(X_1 = T \mid Y = T) = 0.8$, $\mathbf{Pr}(X_1 = F \mid Y = F) = 0.7$, $\mathbf{Pr}(X_2 = T \mid Y = T) = 0.5$, and $\mathbf{Pr}(X_2 = F \mid Y = F) = 0.9$. (*Note:* Questions 3.2 - 3.4 all use these probabilities.) So, attribute X_1 provides slightly stronger evidence about the class label than X_2. Assume X_1 and X_2 are truly independent given Y. Write down the Naive Bayes **decision rule** given $X_1 = x_1$ and $X_2 = x_2$. Write your answer as a table listing the value of the decision, call it $f(X_1, X_2)$, for each of the 4 settings for X_1, X_2.

3. **[8 points]** For the Naive Bayes decision function $f(X_1, X_2)$, the error rate is:

$$\sum_{X_1, X_2, Y} \mathbf{1}\left(Y \neq f(X_1, X_2)\right) P(X_1, X_2, Y).$$

For this question, we will assume that the true data distribution is exactly the same as the Naive Bayes distribution, so we can write $P(X_1, X_2, Y)$ as $P(Y)P(X_1 \mid Y)P(X_2 \mid Y)$.

(a) **[2 points]** Show that if Naive Bayes uses both attributes, X_1 and X_2, the error rate is 0.235.

(b) **[2 points]** What is the error rate using only X_1?

(c) **[2 points]** What is the error rate using only X_2?

(d) **[2 points]** Give a conceptual explanation for why the error rate is (choose one) lower/higher using X_1 and X_2 together as opposed to using only a single attribute.

4. **[5 points]** Now, suppose that we create a new attribute X_3, which is an exact copy of X_2. So, for every training example, attributes X_2 and X_3 have the same value, $X_2 = X_3$.

   (a) **[2 points]** Are X_2 and X_3 conditionally independent given Y?

   (b) **[3 points]** What is the error rate of Naive Bayes now, using X_1, X_2, and X_3? The predicted Y should be computed using the (possibly incorrect) assumption of conditional independence, and the error rate should be computed using the true probabilities.

5. **[2 points]** Why does Naive Bayes perform worse with the addition of X_3? (*Hint*: What assumption does Naive Bayes make about the inputs?)

6. **[3 points]** Does logistic regression suffer from the same problem? Explain why or why not.

7. **[8 points]** : In spite of the above fact we will see that in some examples Naive Bayes doesn't do too badly. Consider the above example i.e. your features are X_1, X_2 which are truely independent given Y and a third feature $X_3 = X_2$. Suppose you are now given an example with $X_1 = T$ and $X_2 = F$. You are also given the probabilities $\mathbf{Pr}(Y = T \mid X_1 = T) = p$ and $Pr(Y = T \mid X_2 = F) = q$, and $P(Y = T) = 0.5$. (*Note:* You should **not** use the probabilities from 3.2-3.4 in your solutions to the following.)

   (a) Prove that the decision rule is $p \geq \frac{(1-q)^2}{q^2+(1-q)^2}$ by applying Bayes rule again.

   (b) What is the true decision rule?

   (c) Plot the two decision boundaries (vary q between 0 and 1) and highlight the region where Naive Bayes makes mistakes.

4 Feature Selection [20 points]

We saw in class that one can use a variety of regularization penalties in linear regression.

$$\hat{w} = \arg\min_{w} \quad \|Y - Xw\|_2^2 + \lambda \|w\|_p^p$$

Consider the three cases, $p = 0$, 1, and 2. We want to know what effect these different penalties have on estimates of w.

Let's see this using a simple problem. Use the provided data (data.mat). Assume the constant term in the regression is zero, and assume $\lambda = 1$, except, of course, for question (1). You don't need to write code that solves these problems in their full generality; instead, feel free to use matlab functions to do the main calculations, and then just do a primitive search over parameter space by plugging in a few different values. Matlab function fminsearch will be helpful. (*Note:* If you are not familiar with function handles, please review the code from HW 2 or see Matlab documentation.)

1. **[3 points]** If we assume that the response variable y is distributed according to $y \sim N(w \cdot x, \sigma^2)$, then what is the MLE estimate $\hat{w}_{MLE}$ of w?

2. **[2 points]** Given $\lambda = 1$, what is $\hat{w}$ for $p = 2$?

3. **[2 points]** Given $\lambda = 1$, what is $\hat{w}$ for $p = 1$?

4. **[4 points]** Given $\lambda = 1$, what is $\hat{w}$ for $p = 0$? Note that since L0 norm is not a "real" norm, the penalty expression is a little different:

$$\hat{w} = \arg\min_{w} \quad \|Y - Xw\|_2^2 + \lambda \|w\|_0$$

Also for L0 norm, you have to solve all combinatorial cases separately where some certain components of w are set to zero, then add L0 accordingly. There are 8 cases for 3 unknown w_i.

5. **[4 points]** Write a paragraph describing the relation between the estimates of w in the four cases, explaining why that makes sense given the different penalties.

6. **[5 points]** When $\lambda > 0$, we make a trade-off between minimizing the sum of squared errors and the magnitude of $\hat{w}$. In the following questions, we will explore this trade-off further. For the following, use the same data from data.mat.

   (a) **[1 point]** For the MLE estimate of w (as in 4.1), write down the value of the ratio

$$||\hat{w}_{MLE}||_2^2 \; / \; ||Y - X\hat{w}_{MLE}||_2^2.$$

   (b) i. **[1 point]** Suppose the assumptions of linear regression are satisfied. Let's say that with N training samples (assume $N >> P$, where P is the number of features), you compute $\hat{w}_{MLE}$. Then let's say you do the same, this time with $2N$ training samples. How do you expect $||Y - X\hat{w}_{MLE}||_2^2$ to change when going from N to $2N$ samples? When $N >> P$, does this sum of squared errors for linear regression directly depend on the number of training samples?

   ii. **[1 point]** Likewise, if you double the number of training samples, how do you expect $||\hat{w}_{MLE}||_2^2$ to change? Does $||\hat{w}_{MLE}||_2^2$ for linear regression directly depend on the number of training samples in the large-N limit?

   (c) **[1 point]** Using any method (e.g. trial and error, random search, etc.), find a value of λ for which the estimate $\hat{w}$ satisfies

$$0.8 < ||\hat{w}||_2^2 \; / \; ||\hat{w}_{MLE}||_2^2 < 0.9.$$

   (d) **[1 point]** Using any method (e.g. trial and error, random search, etc.), find a value of λ for which the estimate $\hat{w}$ satisfies

$$0.4 < ||\hat{w}||_2^2 \; / \; ||\hat{w}_{MLE}||_2^2 < 0.5.$$