CIS 520, Machine Learning, Fall 2015: Assignment 3

Yu-Cheng Lin

October 1, 2015

Collaborator:

Type Collaborator Name Here

1 Linear Regression and LOOCV

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*:

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

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

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

where we define $H = X(X^T X)^{-1} X^T$ (H is often called the *Hat Matrix*).

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

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

   (a) 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.

   Ans: If $m > n$, $\hat{\mathbf{w}}$ will not be well defined. This is because that if $m > n$, $X^T X^{-1}$ will have rank n at most while it's a $m \times m$ matrix, and such matrix is not invertible because it's not full rank.

   (b) 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.)

   Ans: If k columns in X are redundant, $\hat{\mathbf{w}}$ might not be well defined because $X^T X$ will have rank $n - k$. If $n - k < m$, $X^T X^{-1}$ will not be invertible.

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

2. 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.)

   Ans: Computing LOOCV naively will require $O(n)$ complexity in the outer loop and $O(n^2 m)$ in the inner loop. Since $n > m$, the matrix multiplication in the inner loop actually dominates the inversion in terms of complexity. Overall, the naive approach should yield a computation complexity of $O(n^3 m)$

   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. 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.

   Ans:

$$\hat{Y}_i = \sum_{J=1}^{m} H_{ij} Y_j$$

4. 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.

Ans: We first write the SSE for Z

$$SSE_Z = \sum_{j=1}^{n} (Z - \hat{Y}^{(-i)})^2$$

Now substitute Z

$$SSE_Z = \begin{cases} \sum_{j=1}^{n} (Y_j - \delta_{ij}\hat{Y}_i^{(-i)})^2 & j \neq i \\ (\hat{Y}_i^{(-i)} - \hat{Y}_i^{(-i)})^2 & j = i \end{cases}$$

As we can see, when $j = i$, the SSE is zero, otherwise it takes the same form as the SSE of Y. In conclusion, $\hat{Y}^{(-i)}$ for $\hat{Z}$ would minimize Z.

5. 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.
Ans:

$$\hat{Y}_i^{(-i)} = \sum_{j=1}^{m} HijZ_j$$

6. 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.

Ans:

$$\begin{aligned} \hat{Y}_i - \hat{Y}_i^{(-i)} &= \sum_{J=1}^{m} H_{ij}Y_j - \sum_{j=1}^{m} HijZ_j \\ &= \sum_{J=1}^{m} H_{ij}Y_j - (\sum_{j=1}^{m} H_{ij}Y_j - H_{ii}Y_i + H_{ii}\hat{Y}_i^{(-i)} \\ &= H_{ii}Y_i - H_{ii}\hat{Y}_i^{(-i)} \end{aligned}$$

7. 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.

use the result from (6) and solve for $\hat{Y}_i^{(-i)}$:

$$\begin{aligned} \hat{Y}_i - \hat{Y}_i^{(-i)} &= H_{ii}Y_i - H_{ii}\hat{Y}_i^{(-i)} \\ \hat{Y}_i - HiiY_i &= (1 - H_{ii})\hat{Y}_i^{(-i)} \\ \hat{Y}_i^{(-i)} &= \frac{\hat{Y}_i - H_{ii}Y_i}{1 - H_{ii}} \end{aligned}$$

plug this $\hat{Y}_i^{(-i)}$ in the LOOCV formula and do some algebra:

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

The complexity then is just $O(n^2m)$ since we avoided the outerloop.

2 Logistic regression and Naive Bayes

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. 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) $\mathbf{Pr}(Y \mid X)$ Logistic regression optimizes this function, which is discriminative

   (d) $\mathbf{Pr}(Y)$

   (e) $\mathbf{Pr}(X)$

   (f) $\mathbf{Pr}(Y)\mathbf{Pr}(X)$

   (g) $\mathbf{Pr}(X, Y)$ Naive Bayes optimizes this function, which is generative

   (h) None of the above (provide the correct formula in this case)

2. 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.

   ANS: Let's start with the probabilities

$$\begin{aligned}
P(X_i, Y = j) &= \theta_{ij} \\
P(Y = 1) &= \pi \\
P(Y = 0) &= 1 - \pi
\end{aligned}$$

The goal of logistic regression is to maximize the following target function

$$P(Y=1|X_{1,2,3\ldots,n}) = \frac{P(X_{1,2,3\ldots,n}|Y=1)P(Y=1)}{P(X_{1,2,3\ldots,n})}$$

marginalize to obtain

$$= \frac{P(X_{1,2,3\ldots,n}|Y=1)P(Y=1)}{P(X_{1,2,3\ldots,n}|Y=1)(Y=1)+P(X_{1,2,3\ldots,n}|Y=0)(Y=0)}$$

$$= \frac{1}{1+\frac{P(X_{1,2,3\ldots,n}|Y=0)P(Y=0)}{P(X_{1,2,3\ldots,n}|Y=1)P(Y=1)}}$$

$$= \frac{1}{1+\exp(\ln(\frac{P(Y=0)}{P(Y=1)})+\ln(\frac{P(X_{1,2,3\ldots,n}|Y=0)}{P(X_{1,2,3\ldots,n}|Y=1}))}$$

$$= \frac{1}{1+\exp(\ln(\frac{1-\pi}{\pi})+\ln(\frac{\prod_{i=1}^{n}\theta_{i0}}{\prod_{i=1}^{n}\theta_{i1}}))}$$

$$= \frac{1}{1+\exp(\ln(\frac{1-\pi}{\pi})+\sum_{i=1}^{n}\ln(\frac{\theta_{i0}}{\theta_{i1}}))}$$

from here, we can plug in the definition of Bernoulli distribution for θ_{ij}

$$\theta_{ij} = \theta_{ij}^{X_i}(1-\theta_{ij})^{1-X_i}$$

We obtain

$$\frac{1}{1+\exp(\ln(\frac{1-\pi}{\pi})+\sum_{i=1}^{n}\ln(\frac{\theta_{i0}}{\theta_{i1}}))} = \frac{1}{1+\exp(\ln(\frac{1-\pi}{\pi})+\sum_{i=1}^{n}\ln(\frac{\theta_{i0}^{X_i}(1-\theta_{i0})^{1-X_i}}{\theta_{i1}^{X_i}(1-\theta_{i1})^{1-X_i}}))}$$

the terms inside the natural log can be simplified as

$$\ln(\frac{\theta_{i0}^{X_i}(1-\theta_{i0})^{1-X_i}}{\theta_{i1}^{X_i}(1-\theta_{i1})^{1-X_i}}) = \ln((\frac{\theta_{i0}}{\theta_{i1}})^{X_i}(\frac{1-\theta_{i0}}{1-\theta_{i1}})^{1-Xi})$$

$$= \frac{\frac{\theta_{i0}}{\theta_{i1}}}{\ln(\frac{\theta_{i0}}{\theta_{i1}})}X_i + \frac{\frac{1-\theta_{i0}}{1-\theta_{i1}}}{\ln(\frac{1-\theta_{i0}}{1-\theta_{i1}})}(1-X_i)$$

Which is in form

$$w_iX_i + K_i$$

Ultimately, plugging this term back in the original equation, we will get the objective function of logistic regression

$$\frac{1}{1+\exp(w_0+\sum_{i=1}^{n}w_iX_i)}$$

3 Double-counting the evidence

1. 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.)

ANS: We need five parameters to use this Naive Bayes classifier.

$$\begin{aligned} P(X_1 = T|Y = T) \\ P(X_2 = T|Y = T) \\ P(X_1 = T|Y = F) \\ P(X_2 = T|Y = F) \\ P(Y = T) \end{aligned}$$

2. 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.

ANS: Let's first write out the decision rule of Naive Bayes

$$\frac{P(Y = T)\prod_{i=1}^{n} P(X_i|Y = T)}{P(Y = F)\prod_{i=1}^{n} P(X_i|Y = F)} \geq 1$$

If the inequality above holds true, Naive Bayes will return Y = T
Following is a table of the 4 possible scenarios:

X_2/X_1	$X_1 = T$	$X_1 = F$
$X_2 = T$	$\frac{(0.5)(0.8)(0.5)}{(0.5)(0.3)(0.1)} = 13.3 > 1$	$\frac{(0.5)(0.8)(0.5)}{(0.5)(0.3)(0.9)} = 1.48 > 1$
$X_2 = F$	$\frac{(0.5)(0.2)(0.5)}{(0.5)(0.7)(0.1)} = 1.42 > 1$	$\frac{(0.5)(0.2)(0.5)}{(0.5)(0.7)(0.9)} = 0.158 < 1$

such results in the following truth table:

X_2/X_1	$X_1 = T$	$X_1 = F$
$X_2 = T$	T	T
$X_2 = F$	T	F

3. 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) Show that if Naive Bayes uses both attributes, X_1 and X_2, the error rate is 0.235.
ANS: There are four scenarios which gives me an error:

$$\begin{aligned}
P(Y = F, X_1 = T, X_2 = T) = P(Y = F)P(X_1 = T \mid Y = F)P(X_2 = T \mid Y = F) = (0.3)(0.1)(0.5) = 0.015 \\
P(Y = F, X_1 = F, X_2 = T) = P(Y = F)P(X_1 = F \mid Y = F)P(X_2 = T \mid Y = F) = (0.7)(0.1)(0.5) = 0.035 \\
P(Y = F, X_1 = T, X_2 = F) = P(Y = F)P(X_1 = T \mid Y = F)P(X_2 = F \mid Y = F) = (0.3)(0.9)(0.5) = 0.135 \\
P(Y = T, X_1 = F, X_2 = F) = P(Y = T)P(X_1 = F \mid Y = T)P(X_2 = F \mid Y = T) = (0.7)(0.5)(0.5) = 0.05
\end{aligned}$$

Sum all the probability up, we obtain:

$$0.015 + 0.035 + 0.135 + 0.05 = 0.235$$

(b) What is the error rate using only X_1?

ANS: Since we can just read the decision rule off of one variable, I will not compute the decision rule for part b and c.
There are two scenarios which gives me an error:

$$P(Y=F, X_1=T) =;\ (0.3)(0.5) = 0.15$$
$$P(Y=T, X_1=F) =;\ (0.2)(0.5) = 0.1$$

The error rate sums up to 0.25.

(c) What is the error rate using only X_2?

ANS:

$$P(Y=F, X_2=T) =;\ (0.1)(0.5) = 0.05$$
$$P(Y=T, X_2=F) =;\ (0.5)(0.5) = 0.25$$

The error rate sums up to 0.3.

(d) 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.

ANS: In this case, the error rate is lower when both X_1 and X_2 are used. Given that my assumptions of Naive Bayes are correct, obtaining each feature gives some additional information gain to help me predict the outcome better.

4. 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) Are X_2 and X_3 conditionally independent given Y?

ANS: They are not conditionally independent given Y because

$$P(X_3 = T|Y, X_2 = T) = 1$$

If they are truely conditionally independent,

$$P(X_3 = T|Y, X_2 = T) = 0.3 = P(X_3 = T|Y)$$

which is untrue

(b) 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.
Here I will save work and space by not considering $X_2 \neq X_3$ since such scenario is impossible

X_2/X_1	$X_1 = T$	$X_1 = F$
$X_2 = X_3 = T$	$\frac{(0.5)(0.8)(0.5)(0.5)}{(0.5)(0.3)(0.1)(0.1)} = 66 > 1$	$\frac{(0.5)(0.8)(0.5)(0.5)}{(0.5)(0.3)(0.9)(0.9)} = 0.822 < 1$
$X_2 = X_3 = F$	$\frac{(0.5)(0.2)(0.5)(0.5)}{(0.5)(0.7)(0.1)(0.1)} = 7.1 > 1$	$\frac{(0.5)(0.2)(0.5)(0.5)}{(0.5)(0.7)(0.9)(0.9)} = 0.087 < 1$

such results in the following truth table:

X_2/X_1	$X_1 = T$	$X_1 = F$
$X_2 = X_3 = T$	T	F
$X_2 = X_3 = F$	T	F

Now, let's calculate the error rate, note that since $X_3 = X_2$

$$P(X_3|Y, X_2) = 1$$

$$P(X_3, X_2|Y) = P(X_2|Y)$$

Again, four scenarios will give me error

$$\begin{aligned}
P(Y = F, X_1 = T, X_2 = T, X_3 = T) &= P(Y = F)P(X_1 = T \mid Y = F)P(X_2 = T \mid Y = F) \\
&= (0.3)(0.1)(0.5)(1) = 0.015 \\
P(Y = F, X_1 = F, X_2 = T, X_3 = T) &= P(Y = F)P(X_1 = F \mid Y = F)P(X_2 = T \mid Y = F) \\
&= (0.7)(0.1)(0.5)(1) = 0.035 \\
P(Y = T, X_1 = T, X_2 = F, X_3 = F) &= P(Y = T)P(X_1 = T \mid Y = T)P(X_2 = F \mid Y = T) \\
&= (0.8)(0.5)(0.5)(1) = 0.2 \\
P(Y = T, X_1 = F, X_2 = F, X_3 = F) &= P(Y = T)P(X_1 = F \mid Y = T)P(X_2 = F \mid Y = T) \\
&= (0.7)(0.5)(0.5) = 0.05
\end{aligned}$$

The error rate sums up to be

$$0.015 + 0.2 + 0.035 + 0.05 = 0.3$$

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

   ANS: When we added X_3, the problem does not satisfy the assumption of Naive Bayes anymore. We can see that the decision rule essentially throws out X_1 and becomes the same as when we only considered X_2. The error rate moves towards the error rate of only X_2 as the result.

6. Does logistic regression suffer from the same problem? Explain why or why not.

   ANS: Logistic regression does not suffer from this problem because logistic regression makes no assumption about X.

7. : 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.

   ANS: Let's apply Baye's rule to first

$$\begin{aligned}
P(X_1|Y = T) &= \frac{pP(X_1)}{0.5} \\
P(X_1|Y = F) &= \frac{(1-p)P(X_1)}{0.5} \\
P(X_2|Y = T) &= \frac{qP(X_2)}{0.5} \\
P(X_2|Y = f) &= \frac{(1-q)P(X_2)}{0.5}
\end{aligned}$$

   As we can see, when we take the fraction of one feature given $Y = T$ and $Y = F$, the $P(X)$ and $P(Y)$ cancels out. I can then write the decision rule as follows

$$\frac{P(Y = T)\prod_i^n P(Y = T|X_i = T)}{P(Y = F)\prod_i^n P(Y = F|X_i = F)} \geq 1$$

gives me $Y = T//$ So the product ends up being

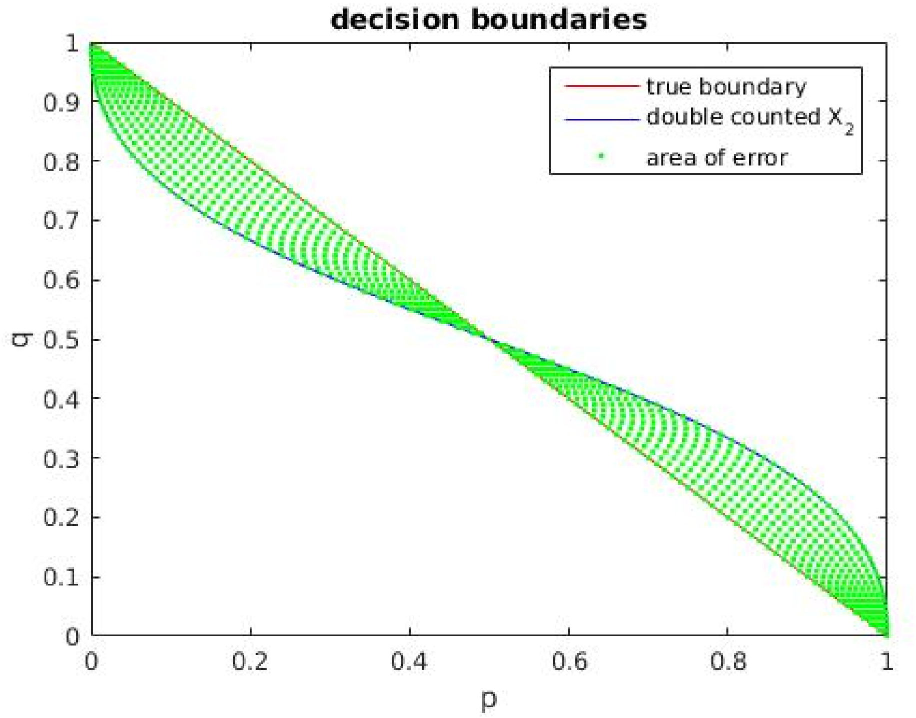
$$\begin{aligned}
\frac{pq^2}{(1-p)(1-q)^2} &\geq 1 \\
pq^2 &\geq (1-p)(1-q)^2 \\
pq^2 &\geq (1-q)^2 - p(1-q)^2 \\
pq^2 + p(1-q)^2 &\geq (1-q)^2 \\
p &\geq \frac{(1-q)^2}{q^2 + (1-q)^2}
\end{aligned}$$

(b) What is the true decision rule?

ANS: The true decision rule can be obtained by simply taking out double counting

$$p \geq \frac{(1-q)}{q + (1-q)} = 1 - q$$

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



4 Feature Selection

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 following data (also provided in matlab format). 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 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. 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?
   ANS:
   $$\hat{w}_{MLE} = w$$
   Since MLE minimizes bias, the estimated $\hat{w}$ will fall at w exactly.

2. Given $\lambda = 1$, what is $\hat{w}$ for $p = 2$?
   ANS:
   $$\hat{w} = \begin{Bmatrix} 0.8645 \\ -0.8210 \\ 4.1219 \end{Bmatrix}$$

3. Given $\lambda = 1$, what is $\hat{w}$ for $p = 1$?
   ANS:
   $$\hat{w} = \begin{Bmatrix} 0.8749 \\ -0.8182 \\ 4.1829 \end{Bmatrix}$$

4. 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.

   ANS: The case that gives me the least error is when I consider all three features. In this case, my estimated $\hat{w}$ is exactly the same as the MLE estimate, which is
   $$\hat{w} = \begin{Bmatrix} 0.8891 \\ -0.8260 \\ 4.1902 \end{Bmatrix}$$

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

   ANS: Taking the 2 norm and 1 norm as penalty both shrinks the weights towards zero. In this particular case, the two norm shrinks the weights more than the 1 norm because the 2 norm of the $\hat{w}$ vector is greater than 1. The zeroth norm did not shrink the weights at all because the 0 norm penalty, compared to the added error in the prediction when we throw away features is too small. Therefore, the objective function with 0 norm penalty did not affect the weights at all. The estimated weights can be drastically different if we increase λ to some large number, though.

6. 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) For the MLE estimate of w (as in 4.1), write down the value of the ratio

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

ANS: 0.0061

(b) i. 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 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?

ANS: Yes, as my samples increase, the vector $||Y - X\hat{w}_{MLE}||_2^2$ grows in dimension. For each dimension, I can assume that my error should be distributed the same way across the entire vector. Doubling the dimension of the vector should double my squared 2-norm of the vector as well.

ii. 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?

ANS: It should not change. Given that my first MLE estimate is very close to the true weight, the change in my estimated weight by doubling the number of samples will be marginal. The two norm of the weight should not change.

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

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

ANS:

$$\lambda = 4.63$$

$$\hat{w} = \begin{Bmatrix} 0.7865 \\ -0.8011 \\ 3.8913 \end{Bmatrix}$$

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

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

ANS:

$$\lambda = 31.83$$

$$\hat{w} = \begin{Bmatrix} 0.4794 \\ -0.6432 \\ 2.7466 \end{Bmatrix}$$