CIS 520, Machine Learning, Fall 2017: Assignment 3
Solutions

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 Naïve Bayes as a Linear Classifier [25 points]

In this question we will consider the problem of binary classification, where we call one class positive and the other negative (for example spam vs. non-spam), i.e. each label $y \in \{\pm 1\}$. We will also assume that each instance $\mathbf{x} = (x_1, \cdots, x_n)$ has binary attribute/feature values, i.e. each attribute/feature $x_i \in \{0, 1\}$.

Let $p = \mathbf{Pr}(y = 1)$, $\alpha_i = \mathbf{Pr}(x_i = 1|y = 1)$, and $\beta_i = \mathbf{Pr}(x_i = 1|y = -1)$. We will assume that all the attributes of each instance $\mathbf{x}$ are conditionally independent given y. Formally,

$$\mathbf{Pr}(\mathbf{x}|y) = \Pi_{i=1}^n \mathbf{Pr}(x_i|y)\,.$$

Recall that a Naïve Bayes classifier h [1] can be written as:

$$h(\mathbf{x}) = \operatorname*{argmax}_{y \in \{\pm 1\}} \hat{\mathbf{Pr}}(y|\mathbf{x})\,, \tag{1}$$

where the probability $\hat{\mathbf{Pr}}(y|\mathbf{x})$ is estimated from data. For the above problem the Naïve Bayes classifier can be written in the form of a linear classifier, i.e. for some $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)\,,$$

where the sign function returns $+1$ when $\mathbf{w}^\top \mathbf{x} + b$ is positive, and -1 otherwise. In the problem you will be asked to find such a $\mathbf{w}$ and b.

1. **[2 points]** Show that the conditional probability of $\mathbf{x}$ given y can be written as:

$$\mathbf{Pr}(\mathbf{x}|y = 1) = \Pi_{i=1}^n\ \alpha_i^{x_i} \cdot (1 - \alpha_i)^{(1-x_i)}\,,$$

and

$$\mathbf{Pr}(\mathbf{x}|y = -1) = \Pi_{i=1}^n\ \beta_i^{x_i} \cdot (1 - \beta_i)^{(1-x_i)}\,.$$

[1] Assume $h(\mathbf{x}) = -1$ in case there is a tie.

★ **SOLUTION:** Since all the features of $\mathbf{x}$ are conditionally independent given y we have that

$$\mathbf{Pr}(\mathbf{x}|y=1) = \Pi_{i=1}^{n}\ \mathbf{Pr}(x_i|y=1)\,, \tag{2}$$

and now

$$\mathbf{Pr}(x_i|y=1) = \begin{cases} \alpha_i & \text{if } x_i = 1 \\ (1-\alpha_i) & \text{if } x_i = 0\,. \end{cases} \tag{3}$$

One can verify that $\alpha_i^{x_i} \cdot (1-\alpha_i)^{(1-x_i)}$ is a compact way of writing the above expression. The same holds for the case $y = -1$.

2. [**8 points**] Given data $D = \{(\mathbf{x}_1, y_1), \cdots (\mathbf{x}_m, y_m)\}$ find the maximum likelihood estimates (MLE) of the parameters p, α_i, and β_i for each $i \in \{1, \cdots, n\}$. Call these estimates $\hat{p}$, $\hat{\alpha}_i$, and $\hat{\beta}_i$, respectively.

★ **SOLUTION:** First we will find the MLE estimate of p. The log-likelihood function L for p, given the data D, can be written as

$$L(D;p) = C_1 + \sum_{i=1}^{m} \frac{(1+y_i)}{2} \cdot \log(p) + \frac{(1-y_i)}{2} \cdot \log(1-p)\,,$$

where the term C_1 only depends on α's and β's. Then the MLE estimate $\hat{p}$ is

$$\hat{p} = \operatorname*{argmax}_p L(D;p)\,.$$

Taking the derivative of $L(D;p)$ with respect to p, we get

$$\frac{dL(D;p)}{dp} = \sum_{i=1}^{m} \frac{(1+y_i)}{2p} - \frac{(1-y_i)}{2(1-p)}\,,$$

and setting it to 0 we get

$$\hat{p} = \frac{1}{m}\sum_{i=1}^{m} \frac{(1+y_i)}{2} = \frac{\#\text{ positive data points in } D}{m}\,.$$

We will now find the MLE estimate of α_i. The log-likelihood function L for α_i, given the data D, can be written as

$$L(D;\alpha_i) = C_2 + \sum_{j:y_j=1} x_{ji} \cdot \log(\alpha_i) + (1-x_{ji}) \cdot \log(1-\alpha_i)\,.$$

where the term C_2 only depends on p, β's and $\alpha_{i'}$ for $i' \neq i$. Taking the derivative of $L(D;\alpha_i)$ with respect to α_i, we get

$$\frac{dL(D;\alpha_i)}{d\alpha_i} = \sum_{j:y_j=1} \frac{x_{ji}}{\alpha_i} - \frac{(1-x_{ji})}{(1-\alpha_i)}\,,$$

and setting it to 0 we get

$$\hat{\alpha_i} = \frac{\sum_{j:y_j=1} x_{ji}}{\sum_{j:y_j=1} 1} = \frac{\#\text{ positive data points in } D \text{ which have } i\text{-th component } 1}{\#\text{ positive data points in } D}\,.$$

Similarly, the MLE estimate of β_i is

$$\hat{\alpha_i} = \frac{\sum_{j:y_j=-1} x_{ji}}{\sum_{j:y_j=-1} 1} = \frac{\#\text{ negative data points in } D \text{ which have } i\text{-th component } 1}{\#\text{ negative data points in } D}\,.$$

Common Mistake 1: One mistake that is common is to write

$$L(D;\alpha_i) = C_2 + \sum_j x_{ji} \cdot \log(\alpha_i) + (1 - x_{ji}) \cdot \log(1-\alpha_i)\,.$$

and to miss the fact that the summation above should only be over $\{j : y_j = 1\}$

Common Mistake 2: Some of you have gotten the expression for MLE of α_i which depends on all components from 1 through n of the data points. Note that α_i should only depend on the i-th component of the data points.

3. [**3 points**] Let $\hat{\mathbf{Pr}}(y|\mathbf{x})$ be the probability distribution of y given $\mathbf{x}$ corresponding to the MLE estimates $\hat{p}$, $\hat{\alpha}_i$'s, and $\hat{\beta}_i$'s. Using Equation (1) show that $h(\mathbf{x})$ can be written as

$$h(\mathbf{x}) = \text{sign}\big(\hat{\mathbf{Pr}}(1|\mathbf{x}) - \hat{\mathbf{Pr}}(-1|\mathbf{x})\big)\,. \tag{4}$$

★ **SOLUTION:** It is easy to verify that the argmax function takes values $+1$ when $\hat{\mathbf{Pr}}(1|\mathbf{x}) > \hat{\mathbf{Pr}}(-1|\mathbf{x})$, and -1 otherwise.

4. [**12 points**] Using Bayes rule and the form of $\hat{\mathbf{Pr}}(y|\mathbf{x})$ show that

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)\,,$$

and find the value of $\mathbf{w}$ and b.

Hint: You need to take the log of both $\hat{\mathbf{Pr}}(1|\mathbf{x})$ and $\hat{\mathbf{Pr}}(-1|\mathbf{x})$ in Equation (4), and use the fact that log is an increasing function.

★ **SOLUTION:** First we will use Bayes rule to see that

$$h(\mathbf{x}) = \mathsf{sign}\Big(\frac{\hat{\mathbf{Pr}}(1)\cdot\hat{\mathbf{Pr}}(\mathbf{x}|1)}{\hat{\mathbf{Pr}}(\mathbf{x})} - \frac{\hat{\mathbf{Pr}}(-1)\cdot\hat{\mathbf{Pr}}(\mathbf{x}|-1)}{\hat{\mathbf{Pr}}(\mathbf{x})}\Big) \tag{5}$$

$$= \mathsf{sign}\big(\hat{\mathbf{Pr}}(1)\cdot\hat{\mathbf{Pr}}(\mathbf{x}|1) - \hat{\mathbf{Pr}}(-1)\cdot\hat{\mathbf{Pr}}(\mathbf{x}|-1)\big)\,. \tag{6}$$

Now it is easy to verify that

$$h(\mathbf{x}) = \mathsf{sign}\Big(\log\big(\hat{\mathbf{Pr}}(1)\cdot\hat{\mathbf{Pr}}(\mathbf{x}|1)\big) - \log\big(\hat{\mathbf{Pr}}(-1)\cdot\hat{\mathbf{Pr}}(\mathbf{x}|-1)\big)\Big)\,, \tag{7}$$

since log is an increasing function. Now writing the probabilities explicitly we get

$$h(\mathbf{x}) = \mathsf{sign}\Big(\log(\hat{p}) + \sum_{i=1}^{n}(x_i\cdot\log(\hat{\alpha}_i) + (1-x_i)\cdot\log(1-\hat{\alpha}_i)) - \log(1-\hat{p}) - \sum_{i=1}^{n}(x_i\cdot\log(\hat{\beta}_i) + (1-x_i)\cdot\log(1-\hat{\beta}_i))\Big) \tag{8}$$

$$= \mathsf{sign}\Big(\sum_{i=1}^{n} x_i \log\big(\frac{\hat{\alpha}_i\cdot(1-\hat{\beta}_i)}{\hat{\beta}_i\cdot(1-\hat{\alpha}_i)}\big) + \sum_{i=1}^{n}\log(\frac{1-\hat{\alpha}_i}{1-\hat{\beta}_i}) + \log(\frac{\hat{p}}{1-\hat{p}})\Big) \tag{9}$$

$$= \mathsf{sign}(\mathbf{w}^\top\mathbf{x} + b)\,, \tag{10}$$

here

$$w_i = \log\big(\frac{\hat{\alpha}_i\cdot(1-\hat{\beta}_i)}{\hat{\beta}_i\cdot(1-\hat{\alpha}_i)}\big)\,,$$

and

$$b = \log(\frac{\hat{p}}{1-\hat{p}}) + \sum_{i=1}^{n}\log(\frac{1-\hat{\alpha}_i}{1-\hat{\beta}_i})\,.$$

2 Multiclass Logistic Regression [25 points]

In this question, we will see how we can extend the logistic regression model from HW2 (which was used for binary classifiction) to multi-class classification. Let's say we have C different classes, and for a class j we have :

$$\mathbf{P}(Y = j \mid X = \mathbf{x}) = \frac{\exp\{\mathbf{w}_j^T x\}}{\sum_{k=1}^{C} \exp\{\mathbf{w}_k^T x\}} \quad \forall j \in \{1, 2, .., C\}$$

where as usual $\mathbf{x}$ is a vector of features, and $\mathbf{w}_j$ is the weight vector assigned to class j. Our objective is to estimate the weights using gradient ascent (just like we did last week), but this time there will not be any coding involved. We will also add a regularization term to the loss function to avoid overfitting.

1. [**15points**] Suppose that the training matrix is of dimensions $M \times N$, which is to say that you have M data points and each data point has N features. Write down the log likelihood, $L(\mathbf{w}_1, ..., \mathbf{w}_C)$. Now add a $L2$ regularization term. Please show all your steps and write a justification for each step.

   ★ **SOLUTION:** Let $\mathbb{I}_{mj}$ be an indicator which is 1 if the m^{th} datapoint belonged to class j, 0 otherwise. Let Y_m be the random variable denoting the label of the m^{th} datapoint, and y_m be the label of the m^{th} datapoint. We can thus write the likelihood function as:

$$\begin{aligned} l(\mathbf{w}_1, ..., \mathbf{w}_C) &= \prod_{m=1}^{M} \mathbf{P}(Y_m = y_m \mid \mathbf{X}, \mathbf{w}) \\ &= \prod_{m=1}^{M} \prod_{j=1}^{C} \mathbf{P}(Y_m = j | \mathbf{X}, \mathbf{w})^{\mathbb{I}_{mj}} \\ &= \prod_{m=1}^{M} \prod_{j=1}^{C} \left(\frac{\exp\{\mathbf{w}_j^T \mathbf{x}_m\}}{\sum_{k=1}^{C} \exp\{\mathbf{w}_k^T \mathbf{x}_m\}} \right)^{\mathbb{I}_{mj}} \end{aligned}$$

   where the first equality follows from the independence of data and the second equality follows from the application of the indicator.

   Taking log and adding the L2 regularization term:

$$L(\mathbf{w}_1, ..., \mathbf{w}_C) = \sum_{m=1}^{M} \sum_{j=1}^{C} \mathbb{I}_{mj} [\mathbf{w}_j^T \mathbf{x}_m - \ln \sum_{k=1}^{C} \exp\{\mathbf{w}_k^T \mathbf{x}_m\}] - \frac{\lambda}{2} ||\mathbf{w}_j||^2$$

   Note that the regularization term must be negative as we want the function to be concave.

2. [**5points**] Next, derive the expression for the j^{th} index in the vector gradient (i.e. partial derivative) $L(\mathbf{w}_1, ..., \mathbf{w}_C)$, with respect to $\mathbf{w}_j$.

   ★ **SOLUTION:**

$$\begin{aligned} \frac{\partial L(\mathbf{w}_1, ..., \mathbf{w}_C)}{\partial \mathbf{w}_j} &= \sum_{m=1}^{M} \left[\mathbb{I}_{mj} \mathbf{x}_m - \frac{\mathbf{x}_m \exp\{\mathbf{w}_j^T \mathbf{x}_m\}}{\sum_{k=1}^{C} \exp\{\mathbf{w}_k^T \mathbf{x}_m\}} \right] - \lambda \mathbf{w}_j \\ &= \sum_{m=1}^{M} \left[\mathbb{I}_{mj} - \mathbf{P}(Y_m = j \mid \mathbf{X}, \mathbf{w}) \right] \mathbf{x}_m - \lambda \mathbf{w}_j \end{aligned}$$

3. [**2points**] Now, write down the update equation for weight vector $\mathbf{w}_j$, with η as the step size.

★ **SOLUTION:**

$$\mathbf{w}_j \leftarrow \mathbf{w}_j + \eta \sum_{m=1}^{M} [\mathbb{I}_{mj} - \mathbf{P}(Y_m = j \mid \mathbf{X}, \mathbf{w})]\mathbf{x}_m - \eta\lambda\mathbf{w}_j$$

4. [**3points**] Will the sequence of consecutive weight vectors converge? If yes, to what? Why?

★ **SOLUTION:** The consecutive weight vectors will converge as the loss function reaches a global maximum. This will happen since the loss function is concave.

3 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. (Where, to be precise the exponent p isn't there for $p = 0$.) 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 to do the main calculations. The best way to search over parameter spaces is using the Matlab function $fminsearch$.(*Note:* If you are not familiar with this function, please 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?

★ **SOLUTION:** Let $r = (Y - Xw)$.

$$\begin{aligned} \frac{\partial r^T r}{\partial w} &= -X^T(Y - Xw) = 0 \\ w &= (X^T X)^{-1} X^T Y \\ w &= [0.8891, -0.8260, 4.1902] \end{aligned}$$

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

★ **SOLUTION:** The closed form solution is $w = (X^T X + \lambda I)^{-1} X^T Y$
We use *fminsearch* in MATLAB to solve for the solution.
$w = [0.8646, -0.8210, 4.1219]$

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

★ **SOLUTION:** We can use either *lasso* or *fminsearch* in MATLAB.
Note that *lasso* in Matlab doesn't minimize the same objective function.

	OBJ1 (fminsearch)	OBJ1 (lasso)
dataset	$w = [0.8749, -0.8182, 4.1829]$	$w = [0, -0.2755, 4.1902]$

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 the L_0 norm, you will have to solve the (combinatorially many) cases where different components of w are set to zero, then add the L_0 penalty to each based on the number of features. There are 8 cases for 3 unknown w_i.

★ **SOLUTION:** For the L_0 norm, we 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.

$$w = [0.8891, -0.8260, 4.1902]$$

5. **[4 points]** Write a paragraph describing the relation between the estimates of w in the four cases (i.e. the four estimates of w from the first four parts of this question), explaining why that makes sense given the different penalties.

★ **SOLUTION:** The MLE estimate simply tries to minimize the residual error without enforcing any prior beliefs about w. Regularizing w under different norms corresponds to different prior beliefs on how we expect the w to be. The $L2$ norm corresponds to the belief that parameters w follow a Gaussian distribution with mean 0; this will tend to shrink all the weights by a constant factor. None will be zeroed out. Penalizing the $L1$ norm decreases all the parameters w_1, w_2, w_3 toward zero by a constant additive amount. If the parameters would be pushed beyond zero,they are zeroed out. In this problem, two were set to zero. Finally, the $L0$ norm encourages a sparse solution, but does not shrink those parameters that are not zeroed out. Thus the nonzero coefficients are larger than under $L1$ or $L2$.

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

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

   ★ **SOLUTION:** $||\hat{w}_{MLE}||_2^2 \;/\; ||Y - X\hat{w}_{MLE}||_2^2 \approx 19.03/3104.8 \approx 0.0061.$

   (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?

★ **SOLUTION:** The SSE will approximately double when N is doubled. Since the assumptions of linear regression are satisfied, doubling the amount of training data will not dramatically change the model when $N >> P$, so we expect approximately twice the SSE with twice the number of summands. Yes, SSE depends directly on N since SSE is a sum over N squared error terms (one for each training sample).

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?

★ **SOLUTION:** In the large N limit, we expect $||\hat{w}_{MLE}||_2^2$ to change barely at all when N is doubled. No, $||\hat{w}_{MLE}||_2^2$ does not depend directly on N since it is a sum over the P elements of the vector $\hat{w}_{MLE}$.

(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

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

★ **SOLUTION:** Any λ between 3.25 and 7.10.

(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

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

★ **SOLUTION:** Any λ between 25.1 and 35.3.

4 Entropy and Minimum Description Length [10 points]

1. **[5 points]** You will need to transmit a sequence of n binary observations (e.g. y values), which will be "1" with probability $p_1 = 3/16$ and "0" with probability $p_0 = 13/16$. What is the minimum number of bits to code the sequence (for large n)? Please do calculate the number instead of only providing the equation.

★ **SOLUTION:** Number of bits = Entropy $= n(-p_0 \log_2 p_0 - p_1 \log_2 p_1) \approx 0.69n$

2. **[5 points]** You are doing feature selection where there are far more possible features than observations. Assume there are total of f features and roughly 3/16 of the features will be selected. The original penalty parameter λ in RIC($Err/2\sigma^2 + \lambda||w||_0$) is $\log_2 f$. In this situation, what would be a better alternative to λ?

★ **SOLUTION:** Total number of bits for transmitting features = Total entropy $\approx 0.69f$. So it requires roughly 0.69 bits per feature, or $\lambda = 0.69$.

Common Mistakes

(a) using $-\log(p)$ in this case is incorrect.

Another solution which is correct is based on piazza post @351, p(-prob log(prob) - (1-prob) log(1-prob)) $= \lambda|w|_0 = \lambda \times \frac{3}{16}p$, which gives $\lambda = 3.712$

5 MDL on a toy dataset [20 points]

We provide a data set (train_data.mat, train_y.mat, test_data.mat, test_y.mat) generated from a particular model with $N = 64$. We want to estimate

$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3$$

We want to use MDL to find the 'optimal' L_0-penalized model.

1. **[10 points]** Estimate the three linear regressions

   (We could actually try all possible subsets here, but instead we'll just try three.)

$$y_1 = w_1 x_1$$
$$y_2 = w_1 x_1 + w_2 x_2$$
$$y_3 = w_1 x_1 + w_2 x_2 + w_3 x_3$$

   For each of the three cases, what is

   (a) the sum of square error
   i) $\text{Err}_1 =$
   ii) $\text{Err}_2 =$
   iii) $\text{Err}_3 =$

   ★ **SOLUTION:** a) i. 1.2779e+03 ii. 835.0558 iii. 834.7418

   (b) 2 times the estimated bits to code the residual ($n \log \frac{Error}{n}$)
   i) $\text{ERR\_bits}_1 =$
   ii) $\text{ERR\_bits}_2 =$
   iii) $\text{ERR\_bits}_3 =$

   ★ **SOLUTION:** b) i. 276.4546 ii. 237.1666 iii. 237.1319

   (c) 2 times the estimated bits to code each residual plus model under AIC ($2 * 1$ bit to code each feature)
   i) $\text{AIC\_bits}_1 =$
   ii) $\text{AIC\_bits}_2 =$
   iii) $\text{AIC\_bits}_3 =$

   ★ **SOLUTION:** c) i. 278.4546 ii. 241.1666 iii. 243.1319

   (d) 2 times the estimated bits to code each residual plus model under BIC ($2 * (1/2) log(n)$ bits to code each feature)
   i) $\text{BIC\_bits}_1 =$
   ii) $\text{BIC\_bits}_2 =$
   iii) $\text{BIC\_bits}_3 =$

★ **SOLUTION:** d) i. 282.4546 ii. 249.1666 iii. 255.1319

2. **[5 points]** Which model has the smallest minimum description length?
   a) for AIC
   b) for BIC

   ★ **SOLUTION:** Model 2 for both (2 features)

3. **[5 points]** Included in the kit is a test data set; does the error on the test set for the three models correspond to what is expected from MDLs? Please compute the test errors and briefly explain it in one sentence.

   ★ **SOLUTION:** Yes:
   Model 1 test error: 1,778
   Model 2 test error: 1,167
   Model 3 test error: 1,173

   Common Mistakes:

   (a) $n \log \frac{\text{Error}}{n}$ is 2 times the estimated. Some students have taken 2 times that, i.e. $2 * n \log \frac{\text{Error}}{n}$ and have lost points for that.

   (b) Some students did not mention the values in part (3) and have lost some points due to that.

   (c) The values were checked against values obtained using $fminsearch$, so solutions with different values have been penalized. If you used a different function, please make a regrade request.