# CIS 520, Machine Learning, Fall 2017: Assignment 3
# Due: Sunday, 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 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:

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

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 $b \in \mathbb{R}$

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

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

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

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 $p$, $\alpha_i$, and $\beta_i$ for each $i \in \{1, \cdots, n\}$. Call these estimates $\hat{p}$, $\hat{\alpha}_i$, and $\hat{\beta}_i$, respectively.

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

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

$$h(\mathbf{x}) = \text{sign}\big(\hat{\mathbf{Pr}}(1|\mathbf{x}) - \hat{\mathbf{Pr}}(-1|\mathbf{x})\big)\,. \tag{2}$$

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 (2), and use the fact that log is an increasing function.

# 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.

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 $\mathbf{w}_j$.

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

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

# 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 $w$?

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 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 $w_i$.

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.

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 $\lambda$ 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 $\lambda$ for which the estimate $\hat{w}$ satisfies

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

# 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.

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 $\lambda$ in RIC($Err/2\sigma^2 + \lambda||w||_0$) is $\log_2 f$. In this situation, what would be a better alternative to $\lambda$?

# 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) $\text{Err}_3 =$

   (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) $\text{ERR\_bits}_3 =$

   (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) $\text{AIC\_bits}_3 =$

   (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) $\text{BIC\_bits}_3 =$

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

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.