STAT 6910-001 – Principles of ML – Homework #2

Due: 5:00 PM 9/27/19

1. **Maximum Likelihood Estimation (14 pts)**

Consider a random variable $\mathbf{X}$ (possibly a vector) whose distribution (pdf or pmf) belongs to a parametric family. The density or mass function may be written as $f(x;\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is called the parameter, and can be either a scalar or vector. For example, in the univariate Gaussian distribution, $\boldsymbol{\theta}$ can be a two dimensional vector consisting of the mean and the variance. Suppose the parametric family is known, but the value of the parameter is unknown. It is often of interest to estimate this parameter from observations of x.

Maximum likelihood estimation is one of the most important parameter estimation techniques. Let $x_1, \ldots, x_n$ be i.i.d. (independent and identically distributed) realizations drawn from $f(x;\boldsymbol{\theta})$. By independence, the joint distribution of the observations is the product

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(x_i;\boldsymbol{\theta}).$$

Viewed as a function of $\boldsymbol{\theta}$, this quantitiy is called the likelihood of $\boldsymbol{\theta}$. It is often more convenient to work with the log-likelihood,

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(x_i;\boldsymbol{\theta}).$$

A maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ is any parameter

$$\hat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log f(x_i;\boldsymbol{\theta}).$$

If the maximizer is unique, $\hat{\boldsymbol{\theta}}$ is called *the* maximum likelihood estimate of $\boldsymbol{\theta}$.

(a) Let $X_1, \ldots, X_n$ be i.i.d. sample from a Poisson distribution with parameter λ, i.e.

$$P(X = x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}.$$

i. (2 pts) Write down the likelihood function $L(\lambda)$.

ii. (2 pts) Write down the log-likelihood function $\ell(\lambda)$.

iii. (3 pts) Find the maximum likelihood estimate (MLE) of the parameter λ.

(b) (7 pts) Let $X_1, \ldots, X_n$ be an i.i.d. sample from an exponential distribution with the density function

$$p(x;\beta) = \frac{1}{\beta} e^{-\frac{x}{\beta}},\ 0 \le x < \infty.$$

Find the MLE of the parameter β. Given what you know about the role that β plays in the exponential distribution, does the MLE make sense? Why or why not?

2. **Logistic regression Hessian (20 pts).** Determine a formula for the gradient and the Hessian of the regularized logistic regression objective function. Argue that the objective function

$$J(\boldsymbol{\theta}) = -\ell(\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||^2$$

is convex when $\lambda \geq 0$, and that for $\lambda > 0$, the objective function is strictly convex.
Hints: The following conventions and properties regarding vector differentiation may be useful. The properties can be easily verified from definitions. Try to avoid long, tedious calculations.

- If $f : \mathbb{R}^n \to \mathbb{R}$, then we adopt the convention

$$\frac{\partial f(\boldsymbol{z})}{\partial \boldsymbol{z}} := \nabla f(\boldsymbol{z}).$$

- If $f : \mathbb{R}^n \to \mathbb{R}^m$, adopt the convention

$$\frac{\partial f(\boldsymbol{z})}{\partial \boldsymbol{z}^T} := \left(\frac{\partial f(\boldsymbol{z})}{\partial \boldsymbol{z}}\right)^T.$$

- Given these conventions, it follows that the Hessian H of J is

$$H = \frac{\partial}{\partial \boldsymbol{\theta}^T}\left(\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right),$$

  which is often denoted more concisely as

$$\frac{\partial^2 J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^T}.$$

- (One form of a multivariate chain rule): If $f(\boldsymbol{z}) = g(h(\boldsymbol{z}))$ where $g : \mathbb{R} \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}$, then

$$\nabla f(\boldsymbol{z}) = \nabla h(\boldsymbol{z}) \cdot g'(h(\boldsymbol{z})).$$

3. **Bayesian spam filtering (20 pts)**. In this problem, you will apply the naive Bayes classifier to the problem of spam detection using a benchmark database assembled by researchers at Hewlett-Packard. Do not use a built-in package or library that directly implements the naive Bayes classifier. You must code it up. If you have questions about a specific package or library, please ask on Piazza. Download the file `spambase.data` from Canvas in the "Files/Data" section and issue the following commands to load the data. In Matlab:

```
z = dlmread('spambase.data',',');
rng(0); % initialize the random number generator
rp = randperm(size(z,1)); % random permutation of the indices
z = z(rp,:); % shuffle the rows of the data matrix
x = z(:,1:end-1);
y = z(:,end);
```

In Python:

```
import numpy as np
z = np.genfromtxt('spambase.data',dtype=float, delimiter=',')
np.random.seed(0) #Seed the random number generator
rp = np.random.permutation(z.shape[0]) #random permutation of the indices
z = z[rp,:]  #shuffle the rows of the data matrix
x = z[:,:-1]
y=z[:,-1]
```

In R:

```
#Code created for UNIX/Mac Environment, R Version = 3.5
```

```
set.seed(2^16-1) #Set Seed

#Assuming you have downloaded the spambase data to your Downloads folder.  Change as necessary
setwd("~/Downloads")

#Read the spambase data.  The following 2 lines will create a R dataframe with 58 columns.
#The Outcome Variable is named Y
#All other feature variables are named V1 through V58
spam_base <- read.csv("spambase.data", header = F)

#Randomly Shuffle the Dataframe on Rows
spam_base <- spam_base[sample(nrow(spam_base)),]

#Prepare feature set x as a dataframe and outcome vector y separately
x = spam_base[,-58] #Excluding the last column which is y
y = spam_base[,58] #Selecting only the last column y
```

Note: If you copy and paste the above, you may need to delete and retype the single quote to avoid an error. This has to do with the optical character recognition of pdfs on some systems.

Here x is $n \times d$, where $n = 4601$ and $d = 57$. The different features correspond to different properties of an email, such as the frequency with which certain characters appear. y is a vector of labels indicating spam or not spam. For a detailed description of the dataset, visit the UCI Machine Learning Repository, or Google 'spambase'.

To evaluate the method, treat the first 2000 examples as training data, and the rest as test data. Fit the nave Bayes model using the training data (i.e., estimate the class-conditional marginals), and compute the misclassification rate (i.e., the test error) on the test data. The code above randomly permutes the data, so that the proportion of each class is about the same in both training and test data.

Note: On the spam detection problem, please note that you will get a different test error depending on how you quantize values that are equal to the median. It makes a difference whether you quantize values equal to the median to 1 or 2. You should quantize all medians the same way; I'm not suggesting that you try all 2^d combinations. So just make sure you try both options, and report the one that works better.

(a) (15 pts) Quantize each variable to one of two values, say 1 and 2, so that values below the median map to 1 and those above map to 2.

Construct the naive Bayes classifier using the training data and apply the trained classifier to the test data. Report the test error. As a sanity check, what would be the test error if you have always predicted the same class, namely, the majority class from the training data?

(b) (5 pts) Submit your concise, well-organized, and clearly commented code.

4. **The Bayes Classifier (46 pts).** Let X be a random variable representing a 1-dimensional feature space and let Y be a discrete random variable taking values in $\{0, 1\}$ (i.e., Y is the corresponding class label). If $Y = 0$, then the posterior distribution of X for class 0 is Gaussian with mean μ_0 and variance σ_0^2. If $Y = 1$, then the posterior distribution of X for class 1 is Gaussian with mean μ_1 and variance σ_1^2. Let $\pi_0 = \Pr(Y = 0)$ and $\pi_1 = \Pr(Y = 1) = 1 - \pi_0$. Assume that $\mu_0 < \mu_1$ and σ_0 and σ_1 are such that the weighted pdfs $\pi_0 p_0(x)$ and $\pi_1 p_1(x)$ intersect at only one point.

(a) (10 pts) Derive the Bayes classifier for this problem as a function of π_i, μ_i, and σ_i where $i \in \{0, 1\}$. **Hint**: In other words, you need to determine decision regions for each class such that the corresponding classifier is the Bayes classifier. Since the posterior distributions are 1-dimensional and

intersect at only one point, this means that the classifier can be reduced to finding a threshold for an observation x such that if x is greater than the threshold, it is assigned to one class and if it is less than the threshold, it is assigned to the other class. Find that threshold. At some point in your derivation, you will need to consider separately the cases where $\sigma_0 = \sigma_1$ and $\sigma_0 \neq \sigma_1$.

(b) (10 pts) Derive the Bayes error rate for this classification problem as a function of π_i, μ_i, and σ_i where $i \in \{0, 1\}$. You may write your solution in terms of the Q function where if Z is a standard normal random variable, then $Q(z) = \Pr(Z > z)$.

(c) (4 pts) Describe how to perform cross-validation for a classification problem.

(d) (3 pts) Set $\mu_0 = 0$, $\mu_1 = 1.5$, $\sigma_0 = \sigma_1 = \sigma = 1$, $\pi_0 = 0.3$ and $\pi_1 = 0.7$. What is the Bayes error rate for this problem?

(e) For the same parameters as in part (d) and for the sample sizes $N \in \{100, 200, 500, 1000\}$, simulate the above classification problem. If you're not sure how to simulate this data, see the files on Canvas under Homework/Starter_Code for an example in R and Python. Apply the Bayes classifier, logistic regression, and the k-nearest neighbor (nn) classifier to the simulated data. Run this simulation for 100 trials and report the following for each sample size:

i. (3 pts) The average value of k as selected using cross-validation.

ii. The classification error of each classifier (the Bayes classifier, k-nn classifier, and logistic regression). Calculate the error of the logistic regression and k-nn classifiers for each trial using cross-validation and describe how you performed cross-validation (e.g. 5-fold, 10-fold cross validation, etc.; 1 pt). You may use built-in functions for logistic regression, k-nn classifiers, and cross-validation (i.e. you do not need to code these up from scratch). Report the mean and standard deviation of the error in

   A. (4 pts) Table form and

   B. (4 pts) Graphical form. Make a plot with sample size on the x-axis and the error on the y-axis. Plot the mean and standard deviation using error bars. Plot the results for all 3 classifiers on the same plot. You may need to use log scales for better visualization. If you're not sure how to create a plot with error bars, see the files on Canvas under Homework/Starter_Code.

iii. (6 pts) Comment on your results. Do any of the errors on the classifiers match the Bayes error rate? If there are any discrepancies, explain why.
You do not need to turn in your code.