COMS 4771 HW1

Due: Tue Feb 13, 2018 at 11:59pm

You are allowed to work in groups of (at max) three students. Only one submission per group is required by the due date. Name and UNI of all group members must be clearly specified on the homework. You must cite all the references you used to do this homework. You must show your work to receive full credit.

1 **[Maximum Likelihood Estimation]** Here we shall examine some properties of Maximum Likelihood Estimation (MLE).

(i) Consider the density $p(x|\theta) \propto \begin{cases} x^2 e^{-x/\theta} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$, for some $\theta > 0$. Suppose that n samples $x_1, \ldots, x_n$ are drawn i.i.d. from $p(x|\theta)$. What is the MLE of θ given the samples?

(ii) Consider the density $p(x|\theta) \propto \begin{cases} 1 & \text{if } -\theta \leq x \leq \theta \\ 0 & \text{otherwise} \end{cases}$, for some $\theta > 0$. Suppose that n samples $x_1, \ldots, x_n$ are drawn i.i.d. from $p(x|\theta)$. What is the MLE of θ given the samples?

(iii) Recall the Gaussian density: $p(x|\mu, \sigma^2) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$, for some mean parameter $\mu \in \mathbb{R}$ and variance parameter $\sigma^2 > 0$. Suppose that n samples $x_1, \ldots, x_n$ are drawn i.i.d. from $p(x|\mu, \sigma^2)$. Show that if μ is unknown, then the MLE σ^2_{ML} is **not** an unbiased estimator of the variance σ^2 for all sample sizes n. What simple modification can we make to the estimate to make it unbiased?

(iv) Show that for the MLE θ_{ML} of a parameter $\theta \in \mathbb{R}^d$ and any known differentiable function $g : \mathbb{R}^d \to \mathbb{R}^k$, the MLE of $g(\theta)$ is $g(\theta_{\text{ML}})$. From this result infer the MLE for the standard deviation (σ) in the same setting as in Part (iii).

2 **[Universality of Decision Trees]**

(i) Show that any binary classifier $g : \{0,1\}^D \to \{0,1\}$ can be implemented as a decision tree classifier. That is, for any classifier g there exists a decision tree classifier T with k nodes $n_1, \ldots, n_k$ (each n_i with a corresponding threshhold t_i), such that $g(x) = T(x)$ for all $x \in \{0,1\}^D$.

(ii) What is the best possible bound one can give on the maximum height of such a decision tree T (from part (i))? For what function g is the bound tight?

3 **[Designing the optimal predictor for continuous output spaces]** We studied in class that the "Bayes Classifier" $\left(f := \arg\max_y P[Y|X]\right)$ is optimal in the sense that it minimizes generalization error over the underlying distribution, that is, it maximizes $\mathbb{E}_{x,y}[\mathbf{1}[g(x) = y]]$. But what can we say when the output space $\mathcal{Y}$ is continuous?

Consider predictors of the kind $g : \mathcal{X} \to \mathbb{R}$ that predict a real-valued output for a given input $x \in \mathcal{X}$. One intuitive way to define the quality of of such a predictor g is as

$$Q(g) := \mathbb{E}_{x,y}[(g(x) - y)^2].$$

Observe that one would want a predictor g with the lowest $Q(g)$.

Show that if one defines the predictor as $f(x) := \mathbb{E}[Y|X = x]$, then $Q(f) \leq Q(g)$ for any g, thereby showing that f is the optimal predictor with respect to Q for continuous output spaces.

4 **[Analyzing iterative optimization]** In this problem, we will analyze the Richardson iteration (from HW0) for finding $\beta \in \mathbb{R}^d$ that (approximately) minimizes $\|A\beta - b\|_2^2$ for a given matrix $A \in \mathbb{R}^{n \times d}$ and vector $b \in \mathbb{R}^n$.

Recall the Richardson iteration, given as follows:

- Initially, $\beta^{(0)} = (0, \ldots, 0) \in \mathbb{R}^d$ is the zero vector in $\mathbb{R}^d$.
- For $k = 1, 2, \ldots, N$:
  - Compute $\beta^{(k)} := \beta^{(k-1)} + \eta A^\mathsf{T}(b - A\beta^{(k-1)})$.

Above, $\eta > 0$ is a fixed positive number called the step size, and N is the total number of iterations. Define $M := A^\mathsf{T}A$ and $v := A^\mathsf{T}b$.

(i) Show that the matrix M is symmetric positive semi-definite.

Throughout, assume that the eigenvalues of M, denoted by $\lambda_1, \ldots, \lambda_d$, satisfy $\lambda_i < 1/\eta$ for all $i = 1, \ldots, d$.

(ii) Prove (e.g., using mathematical induction) that, for any positive integer N,

$$\beta^{(N)} = \eta \sum_{k=0}^{N-1} (I - \eta M)^k v.$$

(Here, for a square matrix B, we have $B^0 = I$, $B^1 = B$, $B^2 = BB$, $B^3 = BBB$, and so on.)

(iii) What are the eigenvalues of $\eta \sum_{k=0}^{N-1} (I - \eta M)^k$? Give your answer in terms of $\lambda_1, \ldots, \lambda_d$, η, and N.

(iv) Let $\hat{\beta}$ be any vector in the range of M satisfying $M\hat{\beta} = v$. Prove that

$$\|\beta^{(N)} - \hat{\beta}\|_2^2 \leq e^{-2\eta\lambda_{\min}N}\|\hat{\beta}\|_2^2,$$

where $\lambda_{\min}$ is the smallest non-zero eigenvalue of M.
Hint: You may use the fact that $1 + x \leq e^x$ for any $x \in \mathbb{R}$.

This implies that as the number of iterations N increases, the difference between our estimate $\beta^{(N)}$ and $\hat{\beta}$ decreases exponentially!

5 **[A comparative study of classification performance of handwritten digits]** Download the datafile hw1data.mat from the class website. This datafile contains 10,000 images (each of size 28x28 pixels = 784 dimensions) of handwritten digits along with the associated labels. Each handwritten digit belongs to one of the 10 possible categories $\{0, 1, \ldots, 9\}$. There are two variables in this datafile: (i) Variable X is a 10,000x784 data matrix, where each row is a sample image of a handwritten digit. (ii) Variable Y is the 10,000x1 label vector where the i^{th} entry indicates the label of the i^{th} sample image in X.

Special note for those who are not using Matlab: Python users can use `scipy` to read in the mat file, R users can use `R.matlab` package to read in the mat file, Julia users can use `MAT.jl` package.

To visualize this data (in Matlab): say you want to see the actual handwritten character image of the 77th datasample. You may run the following code (after the data has been loaded):

```
figure;
imagesc(1-reshape(X(77,:),[28 28])');
colormap gray;
```

To see the associated label value:

```
Y(77)
```

(i) Create a probabilistic classifier (as discussed in class) to solve the handwritten digit classification problem. The class conditional densities of your probabilistic classifier should be modeled by a Multivariate Gaussian distribution. It may help to recall that the MLE for the parameters of a Multivariate Gaussian are:

$$\vec{\mu}_{\text{ML}} := \frac{1}{n}\sum_{i=1}^{n}\vec{x}_i$$

$$\Sigma_{\text{ML}} := \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i - \vec{\mu}_{\text{ML}})(\vec{x}_i - \vec{\mu}_{\text{ML}})^{\mathsf{T}}$$

You must submit your code via Courseworks to receive full credit.

(ii) Create a k-Nearest Neighbor classifier (with Euclidean distance as the metric) to solve the handwritten digit classification problem.

You must submit your code via Courseworks to receive full credit.

(iii) Which classifier (the one developed in Part (i) or the one developed in Part (ii)) is better? You must justify your answer with appropriate performance graphs demonstrating the superiority of one classifier over the other. Example things to consider: you should evaluate how the classifier behaves on a holdout 'test' sample for various splits of the data; how does the training sample size affects the classification performance.

(iv) As discussed in class, there are several metrics one can use in a Nearest Neighbor classification. Do a similar analysis to justify which of the three metrics: L_1, L_2 or L_∞ is better for handwritten digit classification problem.

6 **[Understanding model complexity and overfitting]** Here we will empirically study the tradeoff between model complexity and generalizability.

(i) Build a decision tree classifier for the digit dataset that we already introduced in Question 5. In building your decision tree, you may use any reasonable uncertainty measure to determine the feature and threshold to split at in each cell. Make sure the depth of the tree is adjustable with hyperparameter K.

You must submit your code via Courseworks to receive full credit.

(ii) Ensure that there is a random split between training and test data. Plot the training error and test error as a function of K.

(iii) Do the trends change for different random splits of training and test data?

(iv) How do you explain the difference in the behavior of training and testing error as a function of K?

(v) Based on your analysis, what is a good setting of K if you were deploy your decision tree classifier to classify handwritten digits?