Homework #2

ECE 461/661: Introduction to Machine Learning for Engineers
Prof. Carlee Joe-Wong and Prof. Virginia Smith
Due: Tuesday, September 25, 2018 at 8:30am PT/11:30am ET

Please remember to show your work for all problems and to write down the names of any students that you collaborate with. The full collaboration and grading policies are available on the course website: https://18661.github.io/.

Your solutions should be uploaded to Gradescope (https://www.gradescope.com/) in PDF format by the deadline. We will not accept hardcopies. If you choose to hand-write your solutions, please make sure the uploaded copies are legible. Gradescope will ask you to identify which page(s) contain your solutions to which problems, so make sure you leave enough time to finish this before the deadline. We will give you a 30-minute grace period to upload your solutions in case of technical problems.

1 Linear Algebra Warm Up *[10 points]*

Suppose A, B, and X are $n \times n$ matrices with A, X, and $A - AX$ invertible, and suppose

$$(A - AX)^{-1} = X^{-1}B \tag{1}$$

(a) Is B invertible? Explain why.

(b) Solve for X. If you need to invert a matrix, explain why that matrix is invertible.

2 Linear Regression with Heterogeneous Noise *[20 points]*

In the standard linear regression model, we consider the model where the observed response variable y is the prediction perturbed by noise, namely

$$y = \mathbf{x}^\top \boldsymbol{\beta} + \varepsilon$$

where ε is a Gaussian random variable with mean 0 and variance σ^2. Notably, we are assuming that for all observations in the training data, the corresponding noises are identically and independently distributed. In other words, for the n-th observation $\mathbf{x}_n$, the observed response is

$$y_n = \mathbf{x}_n^\top \boldsymbol{\beta} + \varepsilon_n$$

where $\varepsilon_n \sim \mathcal{N}(0, \sigma^2)$.

This assumption is not always applicable. For example, consider the problem of predicting the sales price of a house based on features about that house. It could be that $\mathbf{x}_n$ is a feature that represents the *neighborhood* that a house resides in, and that some neighborhoods have houses with very different sales prices, compared to other neighborhoods where the sales prices are more uniform.

In this case, we can model the data in the following way:

$$y_n = \mathbf{x}_n^\top \boldsymbol{\beta} + \varepsilon_n$$

where ε_n are independently distributed but **do not have to be identically distributed**. In particular, each one could have a different variance, namely, $\varepsilon_n \sim \mathcal{N}(0, \sigma_n^2)$.

(a) Suppose our training dataset contains $\{(\mathbf{x}_n, y_n), n = 1, 2, \ldots, N\}$ such observations. Write down the log-likelihood function of the data. This function should be a function of the data as well as $\boldsymbol{\beta}$ and all σ_n.

(b) Derive the maximum likelihood estimate of $\boldsymbol{\beta}$, and express it in terms of the data as well as all the σ_n. You should assume each σ_n is known to you — you do not need to estimate it from the data.

3 Linear Regression with Smooth Coefficients *[20 points]*

Consider a dataset with n data points $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbb{R}^{p\times 1}$, drawn from the following linear model:

$$y = \mathbf{x}^\top \boldsymbol{\beta} + \varepsilon,$$

where ε is a Gaussian noise. Suppose that we would like the elements of $\boldsymbol{\beta}$ to have a natural ordering, such that the difference $(\beta_i - \beta_{i+1})^2$ cannot be large, for $i = 1, \ldots, p-1$.

(a) State this condition on $\boldsymbol{\beta}$ as a regularizer. Write the new optimization problem for finding $\boldsymbol{\beta}$ by combining both this regularization and L_2 regularization.

(b) Find the optimal $\boldsymbol{\beta}$ by solving the problem in part (a).

4 High Dimensional Linear Regression - Ridge Regression *[30 points]*

4.1 Derivation of the Ridge Regression estimator

A variation of the Least Squares estimation problem considers the following optimization problem:

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y}_n - \mathbf{X}_n\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \tag{2}$$

where $\lambda > 0$ is a regularization parameter. The regularizing term penalizes large components in $\boldsymbol{\beta}$ which leads to shrinking $\boldsymbol{\beta}$ (i.e., $\boldsymbol{\beta}$ has a smaller norm).

(a) Find the solution of the ridge regression problem.

(b) Explain why the ridge regression estimator is more robust to overfitting than the least squares estimator.

(c) How does the value of λ affect the bias and the variance of the estimator?

4.2 Implementation of the Ridge Regression Estimator

In order to demonstrate the ridge regression estimation, we use a data example prepared by Liebmann et al. (2009) (https://www.ncbi.nlm.nih.gov/pubmed/19427473). The matrix Y contains the concentration of glucose and ethanol (in g/L) for $n = 166$ alcoholic fermentation mashes of different feedstock (rye, wheat and corn). These are the two dependent variables. There are 235 covariates in X, which contain the first derivatives of near infrared spectroscopy (NIR) absorbency values at $1115 - 2285$ nm. In this problem, we will predict the glucose concentration for the given covariates. The training dataset (files: Ytraining.csv, Xtraining.csv) consists of 126 observations. The dataset is further divided into a validation set (files: Yvalidation.csv, Xvalidation.csv) and a testing set (files: Ytesting.csv, Xtesting.csv); each contains 20 observations.

In this subquestion, you will implement the Ridge Regression estimator. Your code must have the following methods:

- **fit**$(\mathbf{X}_n, \mathbf{y}_n, \lambda)$: which fits the model given a value for the regularization parameter.

- **predict(X, $\boldsymbol{\beta}$)**: which predicts the values of the dependent variable for a new set of covariates using the learned model.

The file 'ridge_base.py' contains a template with these methods that you can work from. You do not have to use the template, but you may find it helpful.

4.3 Evaluation

For the given dataset:

- Plot in the same figure, the learned coefficients $\boldsymbol{\beta}$ with respect to the regularization parameter λ, when λ ranges from 0.001 to 2 with a step of 0.002. What do you observe?
- Plot on the y-axis the RMSE (Root Mean Squared Error) of the learned model on the validation set with respect to the regularization parameter. Find the regularization parameter λ^* that achieves the minimum RMSE.
- For the λ^* found above, plot the predicted versus the real value of the glucose concentration, when the model is evaluated on the testing dataset. That is, for the 20 testing points plot the true values of glucose concentration (on x-axis) vs. predicted values of glucose concentration (on y-axis).

5 Naive Bayes *[20 points]*

The binary Naive Bayes classifier has interesting connections to the logistic regression classifier. You will show that, under certain assumptions, the Naive Bayes likelihood function is identical in form to the likelihood function for logistic regression. You will then derive the MLE parameter estimates under these assumptions.

(a) Suppose $X = \{X_1, \ldots, X_D\}$ is a continuous random vector in $\mathbb{R}^D$ representing the features and Y is a binary random variable with values in $\{0, 1\}$ representing the class labels. Assume the following:

- The label variable Y follows a Bernoulli distribution, with parameter $\pi = P(Y = 1)$.
- For each feature X_j, we have $P(X_j|Y = y_k)$ follows a Gaussian distribution of the form $\mathcal{N}(\mu_{jk}, \sigma_j)$.

Using the Naive Bayes assumption that states "*for all $j' \neq j$, X_j and $X_{j'}$ are conditionally independent given Y*", compute $P(Y = 1|X)$ and show that it can be written in the following form:

$$P(Y = 1|X) = \frac{1}{1 + \exp(-w_0 + \mathbf{w}^\top \mathbf{X})}.$$

Specifically, you need to find the explicit form of w_0 and $\mathbf{w}$ in terms of π, μ_{jk}, and σ_j, for $j = 1, \ldots, D$ and $k \in \{0, 1\}$.

(b) Suppose a training set with N examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$ is given, where $\mathbf{x}_i = (x_{i1}, \cdots, x_{iD})^\top$ is a D-dimensional feature vector, and $y_i \in \{0, 1\}$ is its corresponding label. Using the assumptions in 1.a (not the result), provide the maximum likelihood estimation for the parameters of the Naive Bayes model with Gaussian assumption. In other words, you need to provide the estimates for π, μ_{jk}, and σ_j, for $j = 1, \ldots, D$ and $k \in \{0, 1\}$.