BUS197287: Advanced Machine Learning Summer 2021 Imperial College Business School
Martin Haugh

MOCK Final Exam

Exam Duration: 2 Hours
Total Marks Available: 95

Instructions

1. This is a closed-book exam.
2. All answers and explanations must be provided in the answer book. This is also true of Question 2 which requires you to fit LDA and QDA models using the computer.
3. Keeps your answers succinct and to the point. Long rambling answers with irrelevant details will work against you.

Advice: Read the entire exam carefully before starting on your answers.

Question 1. (10 marks)

Suppose we want to solve for the optimal λ in the LASSO regression

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 - \lambda \|\mathbf{x}\|_1$$

via cross-validation when we have a total of pn IID training points. The typical way to do this as follows:

- Choose a set $\Lambda = \{\lambda_1, \ldots, \lambda_m\}$ of possible choices for λ.
- Randomly split the data into p folds: $D_1, \ldots, D_p$ where each fold contains n datapoints.
- For each $\lambda_i \in \Lambda$, and $k = 1, \ldots, p$, train the Lasso classifier on $\{D_1, \ldots, D_{k-1}, D_{k+1}, \ldots, D_p\}$ and test on D_k to obtain the k^{th} sample error, $\mathcal{E}_k(\lambda_i)$
- Use the p values to obtain the mean error $\mathcal{E}(\lambda_i) := \frac{1}{p}\sum_{k=1}^{p} \mathcal{E}_k(\lambda_i)$ for $i = 1, \ldots, m$.
- Let i^* be the value of i that corresponds to the minimum mean error, i.e. $i^* = \operatorname{argmin}_{i \in \{1,\ldots,m\}} \mathcal{E}(\lambda_i)$

Given this procedure answer the following questions:

(a) Assuming the data $D = \cup D_k$ was IID, is $\mathcal{E}(\lambda_i)$ an unbiased estimate of the true error for λ_i when $(p-1)n$ IID training points are used to learn the regression function? Justify your answer. (5 marks)

(b) Is $\mathcal{E}(\lambda_{i^*})$ an unbiased estimate of the true error of λ_{i^*}? Justify your answer. (5 marks)

Question 2. (15 marks)

Open the `R` Notebook *LDA_v_QDA_Mock_Exam.rmd*. The first code chunk reads in two separate data-sets. The first set is the training data-set and consists of 1,000 observations of $(y, \mathbf{x})$ where $y \in \{1, 2\}$ is the target / class and $\mathbf{x} \in \mathbb{R}^8$ is a feature vector. The target y is in the first column and is labelled "class" in the training set data-frame. The test-set contains 9,000 observations of the same random vector $(y, \mathbf{x})$. (The second code chunk provides a summary of the training data while the third chunks prints out the first few rows of the training data.)

(a) Fit an LDA model to the training data. Report the confusion matrix of your fitted model on the test set. (5 marks)

(b) Fit a QDA model to the training data. Report the confusion matrix of your fitted model on the test set. (5 marks)

Question 3. (5 marks)

Indicate which of (a) through (d) is correct and be sure to justify your answer. (Just one or two lines is sufficient.) The lasso, relative to ordinary least squares, is:

(a) More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

(b) More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

(c) Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

(d) Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Question 4. (20 marks)

Consider the situation where we have data from two classes – class 0 and class 1. We make the following mixture model for data from class 0 which places a normal density on each data-point:

$$\begin{aligned} p(\mathbf{x} \mid c=0) &= \frac{1}{N_0} \sum_{n \in \text{class } 0} \mathrm{N}\left(\mathbf{x} \mid \mathbf{x}_n, \sigma^2 \mathbf{I}\right) \\ &= \frac{1}{N_0} \frac{1}{\left(2\pi\sigma^2\right)^{d/2}} \sum_{n \in \text{class } 0} e^{-(\mathbf{x}-\mathbf{x}_n)^\top (\mathbf{x}-\mathbf{x}_n)/(2\sigma^2)} \end{aligned}$$

where d is the dimension of a data-point $\mathbf{x}$ and N_0 is the number of training points in class 0. This is a so-called *Parzen estimator* and it models the data as a uniform weighted sum of normal distributions centred on the training points.

Similarly, for data from class 1 we assume

$$p(\mathbf{x} \mid c=1) = \frac{1}{N_1} \frac{1}{\left(2\pi\sigma^2\right)^{d/2}} \sum_{n \in \text{class } 1} e^{-(\mathbf{x}-\mathbf{x}_n)^\top (\mathbf{x}-\mathbf{x}_n)/(2\sigma^2)}.$$

(a) Explain how you would classify a new point $\mathbf{x}^*$? (10 marks)

(b) Explain clearly how would you choose an appropriate value of σ^2? (5 marks)

(c) What sort of classifier would you obtain in the limit $\sigma^2 \rightarrow 0$? (It's ok to guess!) (5 marks)

Question 5. (10 marks)

Consider the following process for training a binary classifier $(\mathbf{x}, k)$ where $\mathbf{x} \in \mathbb{R}^2$ and $k \in \{1, 2\}$ is the class. We first plot the data in $\mathbb{R}^2$ and note that it is not even approximately linearly separable. We then perform a simple transformation $\mathbf{x} \rightarrow \mathbf{z} \in \mathbb{R}^2$ and note that the transformed data is now close to being linearly separable. We therefore decide to use the transformed data to construct our classifier. Towards this end we split the data, $\mathbf{z}_1, \ldots, \mathbf{z}_n$, into a training set and a test set. We train a linear classifier on the training set and then use the test set to estimate its generalization error.

What, if anything, is "wrong" with this learning process? If you think nothing is "wrong" explain why you think the learning process is sound.

Question 6. (15 marks)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ denote (centered) sample observations of $\mathbf{x} \in \mathbb{R}^d$ and suppose we wish to solve the following problem:

$$\begin{aligned} & \min_{\mathbf{B},\mathbf{Z}} \sum_{i=1}^{n} \sum_{j=1}^{d} \left[x_{j,i} - \sum_{k=1}^{K} b_{j,k} z_{k,i} \right]^2 \\ \equiv \; & \min_{\mathbf{B},\mathbf{Z}} \|\mathbf{X} - \mathbf{B}\mathbf{Z}\|_F^2 \end{aligned} \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1 \; \cdots \; \mathbf{x}_n]$ is a $(d \times n)$ matrix whose columns are the n sample observations.

Fact: It is well known that an optimal solution to (1) is

$$\begin{aligned} \mathbf{B}^* &= \mathbf{\Gamma}_{1:K} \\ \mathbf{Z}^* &= \mathbf{P}_{1:K} = [\mathbf{p}_{1:K}^1 \; \cdots \; \mathbf{p}_{1:K}^n] \end{aligned}$$

where:

- $\mathbf{\Gamma}_{1:K}$ contains the K eigen vectors of $\mathbf{\Sigma}$, the sample variance-covariance matrix of the $\mathbf{x}_i$'s, corresponding to the K largest eigen values
- $\mathbf{p}_{1:K}^i$ are the first K principal components for the i^{th} data-point $\mathbf{x}_i$.

This is simply stating that the familiar PCA approximation $\mathbf{X} \approx \mathbf{\Gamma}_{1:K} \, \mathbf{P}_{1:K}$ is the optimal solution to (1).

Consider now the matrix completion problem where we only have partial information on the entries of the $d \times n$ matrix $\mathbf{X}$. Specifically, let $\Omega \subseteq \{1, \ldots, d\} \times \{1, \ldots, n\}$ denote the observed entries of $\mathbf{X}$. We want to find matrices $\mathbf{B}, \mathbf{Z}$ to solve

$$\equiv \min_{\mathbf{B},\mathbf{Z}} \sum_{(u,i)\in\Omega} \left(x_{ui} - \sum_{k=1}^{K} b_{uk} z_{ki} \right)^2 . \tag{2}$$

This is a non-convex optimisation problem so we can only hope in general to find good local optimal solutions to it.

(a) Explain how you could use the above fact to design an iterative algorithm to find a solution to (2) and therefore impute / estimate the missing values in $\mathbf{X}$. (To be clear, no equations are required here or in part (b) below and just a few lines will suffice.) (8 marks).

(b) Explain how you would use a least-squares algorithm to find a solution to (2) and therefore impute / estimate the missing values in $\mathbf{X}$. (7 marks).

Question 7. (20 marks)

Show that the K-means clustering algorithm can be kernelized, i.e. is amenable to the kernel trick. To get you started, let $\phi(\mathbf{x}) \in \mathbb{R}^M$ be the feature vector where $\mathbf{x} \in \mathbb{R}^m$. (Typically $M >> m$ with possible $M = \infty$.) Recall that the k^{th} cluster center at the end of iteration t satisfies

$$\mathbf{m}_k^{(t)} := \underset{\mathbf{m}\in\mathbb{R}^M}{\operatorname{argmin}} \sum_{j\in\mathcal{C}_k^{(t)}} \|\phi(\mathbf{x}_j) - \mathbf{m}\|_2^2 = \frac{1}{|\mathcal{C}_k^{(t)}|} \sum_{j\in\mathcal{C}_k^{(t)}} \phi(\mathbf{x}_j)$$

for $k = 1, \ldots, K$ where K is the number of clusters and $|\mathcal{C}_k^{(t)}|$ is the number of points (in iteration t) assigned to cluster k. The cluster assignment of data-point i in iteration $t+1$ of the algorithm is

$$\mathcal{C}^{(t+1)}(i) = \underset{k}{\operatorname{argmin}} \left\| \phi(\mathbf{x}_i) - \mathbf{m}_k^{(t)} \right\|_2^2 .$$