Statistical Machine Learning: Coursework 5

General Instructions

- Work in groups of at least one and at most two.
- Include your student number and your name on your assignment.
- In your solutions, you should present your R output or snippets of R code as you deem appropriate. Make sure to present your results clearly.
- You should submit your work as a single PDF file on Blackboard by Thursday the 27^{th} March at 12 noon.

Additional Information

- There are **40 marks** available in total. Your mark out of 40 will be scaled by 2.5 to give the mark out of 100 that you see on Blackboard.

Problem 1: Theory (20 marks)

In this half of the assignment, we shall investigate some aspects of the singular value decomposition, and its relationship to effective linear dimension reduction. Throughout this section, the norm $\|\cdot\|$ will always refer to the standard Euclidean norm, unless otherwise noted.

(i) **(5 marks)** Fix dimensions $1 \leq r \ll p \in \mathbb{N}$, and let $\mathbf{A}$ be a matrix of shape $(p \times r)$ which has full column rank, but whose columns are **not necessarily orthonormal**. Define the subspace $\mathcal{V} \subseteq \mathbb{R}^p$ as the *range* of $\mathbf{A}$, i.e.

$$\mathcal{V} = \{\mathbf{A}z : z \in \mathbb{R}^r\}.$$

Recalling the definition of the projection operator $\text{proj}_{\mathcal{V}}$ as

$$\text{proj}_{\mathcal{V}}(x) := \arg\min_{v \in \mathcal{V}} \|x - v\|,$$

(α) Show that $\text{proj}_{\mathcal{V}}$ takes the form $\text{proj}_{\mathcal{V}}(x) = \mathbf{B}x$ for some matrix $\mathbf{B}$, giving an exact formula for $\mathbf{B}$ in terms of $\mathbf{A}$, with justification.

(β) Verify that the projection operator is *non-expansive*, i.e.

$$\forall x, x' \in \mathbb{R}^p, \qquad \|\text{proj}_{\mathcal{V}}(x) - \text{proj}_{\mathcal{V}}(x')\| \leq \|x - x'\|.$$

As part of your proof, you may wish to consider choosing a specific basis for $\mathcal{V}$; if you do so, then you should justify why such a basis exists. You do not need to discuss how you would find such a basis in practice.

(*ii*) **(5 marks)** Let $\mathbf{X}$ be a data matrix of shape $(n \times p)$. In the first step of the greedy approach to performing linear dimension reduction upon $\mathbf{X}$, we solve the problem

$$\max_{v} \|\mathbf{X}v\| \qquad \text{such that } v \in \mathbf{R}^p,\ \|v\| = 1$$

Consider now the following modified optimisation problem

$$\max_{u,v} u^\top \mathbf{X} v \qquad \text{such that } u \in \mathbf{R}^n,\ v \in \mathbf{R}^p,\ \|u\| = \|v\| = 1$$

Show that this modified problem has the same optimal value as our original problem. Moreover, show that if $(u_\star, v_\star)$ are an optimal solution for this modified problem, then $v_\star$ is optimal for our original problem.

(*iii*) **(5 marks)** Recall that the *operator norm* of an $(m \times n)$ matrix $\mathbf{C}$ is defined by

$$\|\mathbf{C}\|_{\text{op}} := \max\left\{\|\mathbf{C}z\| : z \in \mathbb{R}^n,\ \|z\| \leq 1\right\}.$$

Let $\mathbf{E}$ be an arbitrary $(m \times n)$ matrix of rank $r \leq \min\{m, n\}$ which has singular value decomposition

$$\mathbf{E} = \mathbf{L}\mathbf{D}\mathbf{R}^\top,$$

where $\mathbf{D}$ is an $(r \times r)$ diagonal matrix with diagonal entries $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_r > 0$, and $\mathbf{L}$, $\mathbf{R}$ are each matrices of appropriate shape with orthonormal columns. Prove carefully that $\|\mathbf{E}\|_{\text{op}} = \delta_1$.

(*iv*) **(5 marks)** As in part (ii), let $\mathbf{X}$ be a data matrix of shape $(n \times p)$, which we now assume to satisfy $n > p$ and be of full column rank. Notate the singular value decomposition of $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where as usual, $\mathbf{U}$ is of shape $(n \times p)$, $\mathbf{V}$ is of shape $(p \times p)$, and $\Sigma = \text{diag}\left(\sigma_1, \sigma_2, \cdots, \sigma_p\right)$ is a $(p \times p)$ diagonal matrix with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > 0$.

(α) Suppose that the singular values of $\mathbf{X}$ satisfy $\sigma_k = k^{-\alpha}$ for some $\alpha > 0$. Writing $\mathbf{X}_k$ for the truncated SVD of $\mathbf{X}$ of order k, show that the relative error in the approximation $\mathbf{X} \approx \mathbf{X}_k$ can be bounded as

$$\|\mathbf{X} - \mathbf{X}_k\|_{\text{op}} \leq \rho_k \cdot \|\mathbf{X}\|_{\text{op}}$$

for some factor $\rho_k \in [0, 1]$ which you should identify explicitly.

(β) Fix some $\epsilon \in [0, 1]$, and suppose that we want to find some approximating matrix $\widehat{\mathbf{X}}_\epsilon$ satisfying

$$\left\|\mathbf{X} - \widehat{\mathbf{X}}_\epsilon\right\|_{\text{op}} \leq \epsilon \cdot \|\mathbf{X}\|_{\text{op}},$$

and such that the rank of $\widehat{\mathbf{X}}_\epsilon$ is as small as possible. Explain how to use the SVD of $\mathbf{X}$ to find the optimal $\widehat{\mathbf{X}}_\epsilon$, justifying its optimality. How does the rank of $\widehat{\mathbf{X}}_\epsilon$ behave as a function of ϵ and α?
(You may wish to refer to the lecture notes or slides on "matrix norms" to justify your claims of optimality.)

Problem 2: Practical (20 marks)

In this half of the assignment, we will explore the use of randomised dimension reduction strategies as part of a machine learning pipeline. The dataset we shall use for this task, `sonar.csv`, can be found on Blackboard, alongside this document.

The `sonar` dataset consists of the results of a number of experiments, each involving bouncing acoustic sonar signals off of either i) a metal cylinder (denoted `M` in the data) or ii) a roughly cylindrical rock (denoted `R` in the data). The rows of the data are each made up of 60 numbers in the range 0.0 to 1.0, describing the signal which was transmitted in the experiment in question; the final column records the label of the observation (i.e. whether the object was a metal cylinder or a rock). As such, this data should be thought of as defining a classification problem.

To begin with, load in the data, split it into testing data and training data, and then into features and labels. For this task, you may find it useful to refer back to the R demo from Lecture 7.

(i) **(5 marks)** Write a function `my_gauss_sketch` which receives as input a pair of natural numbers (r, p) which satisfy $1 \leq r \leq p$, and outputs a matrix $\mathbf{S}$ of shape $(r \times p)$ with iid Gaussian entries, with variance chosen so that the *expected squared norm* of each row is equal to p.

(ii) **(5 marks)** Write a function `my_sse` which receives as input the triple of natural numbers (s, r, p) which satisfy $1 \leq s \leq r \leq p$, and outputs a matrix $\mathbf{S}$ of shape $(r \times p)$, such that

1. The rows of $\mathbf{S}$ are independent,
2. Each row of $\mathbf{S}$ contains only s non-zero entries, and
3. Each of these non-zero entries is equally likely to be either of $\pm\sqrt{\frac{p}{s}}$.

(When using these random embedding matrices in practice, it is recommended to take $s = \min\{r, 8\}$).

(questions continue on the next page)

(*iii*) **(5 marks)** Using either of your methods from part (i) and part (ii), and making use of the `knn` function from the `caret` package (as in e.g. CW2),

(α) Randomly embed the data features into a subspace of dimension $r = 10$, and fit a K-Nearest Neighbours classifier which uses the embedded features $z_i = \mathbf{S}x_i$ to predict the label, performing a sweep over K to pick a good value of K.

(β) Repeating this experiment $E = 10$ times (i.e. generating independent replicates $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \cdots, \mathbf{S}^{(10)}$), comment on the variability of the accuracy of your classifiers from one embedding to the next. Does it appear that the specific matrix $\mathbf{S}$ which we generate matters very much?

(γ) For comparison, run K-Nearest Neighbours directly on the original features, again performing a sweep over K to pick a good value of K. Does this offer much of an improvement over the embedding-based classifier?

(*iv*) **(5 marks)** Since the data set at hand is not too high-dimensional, it is feasible for us to compute the singular value decomposition of the original feature matrix using `svd`.

(α) Using the `svd` function to examine the structure of our original data set, and making particular reference to the singular values (which can be obtained via e.g. `svd(features_matrix)$d`), comment on why the accuracy of our embedding-based classifiers is not necessarily surprising.