ENSIMAG – GRENOBLE INP – UGA
Statistical Analysis and Document Mining
Pedro L. C. Rodrigues
Alexandre Wendling

YEAR 2023-2024

pedro.rodrigues@inria.fr
alexandre.wendling@univ-grenoble-alpes.fr

Final **practical** exam – Duration: 3h

This exam is composed of four parts.

The maximum final grade is 20.0 points.

Be sure to **read through all of the questions** before starting to solve the exam.

All handwritten documents are allowed, as well as the electronic documents on your computer. No calculator or mobile phone allowed.

No internet connection is available from your computer.

To do this practical exam, you have to use `RStudio`. Open each `rmd` file located in the `$HOME/exam/` directory. Answer the questions directly in each `rmd` file and ensure that you can knit it into an `html` file, and save the `rmd` file. The different `rmd` files correspond to each of the parts of the exam.

Any answer not contained in the `rmd` files will not be taken into account, unless explicitly mentioned by you or the teacher in the `rmd` file (for example you can state: please see figure `myfig.png`).

The contents of the `$HOME/exam/` directory are

- `exam.pdf` (this document)
- `Part1.rmd`, `Part2.rmd`, `Part3.rmd`, `Part4.rmd` (the files you have to complete and save)
- `/references` (reference materials that might be useful in the exam)

> ! You should regularly **save the current state** of your whole home directory using the icons on your desktop. To do that: (1) Save the files currently open in your editor (e.g. `Rstudio`) then (2) Save your current work and continue the exam. **Before leaving the room** save your current work and stop the session. Please note that the backup operation is based on `rsync`, and does not store every intermediate saved version (in this order!). If your PC crashes, your latest backup can be recovered with the help of the teaching staff.

Good luck!

▶ Part 1: Multiple choice questions (6.0 points)

For each question, write the letter in your `Part1.rmd` file corresponding to the correct answer.

Each question has **exactly one correct** answer.

– Question 1: True or false (0.5 points each)

For each item below you should write in your `Part1.rmd` file whether the sentence is True or False.

(a) Unlabeled data can be used for detecting overfitting.

(b) PCA and spectral embedding perform eigendecomposition on two different matrices. However, the dimensions of these matrices are the same.

(c) Since classification is a special case of regression, logistic regression is a special case of linear regression.

(d) The largest eigenvector of the covariance matrix is the direction of minimum variance in the data.

(e) A random forest is an ensemble learning method that attempts to lower the bias error of decision trees.

(f) As model complexity increases, bias will decrease while variance will increase.

(g) Consider a cancer diagnosis classification problem where almost all of the people being diagnosed don't have cancer. The probability of correct classification is the most important metric to optimize.

(h) The more features that we use to represent our data, the better the learning algorithm will generalize to new data points.

– Question 2: Ridge regularization (credits to EPFL CS-433) (1.0 points)

Assume we have N training samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ where each $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$.

For $\lambda \geq 0$, we consider the following loss function:

$$\mathcal{L}_\lambda(\boldsymbol{\beta}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2 + \lambda\|\boldsymbol{\beta}\|_2^2$$

and let $C_\lambda = \min_{\boldsymbol{\beta}\in\mathbb{R}^p} \mathcal{L}_\lambda(\boldsymbol{\beta})$ denote the optimal loss value. Which of the following statements is **true**?

(A) C_λ is a non-increasing function of λ.

(B) For $\lambda = 0$, the loss $\mathcal{L}_0$ is non-convex and might have several minimizers.

(C) C_λ is a non-decreasing function of λ.

(D) None of the above statements are true.

– Question 3 (credits to Berkeley CS-189) (1.0 points)

Which of the following are true for the k-nearest neighbor (k-NN) algorithm?

(A) k-NN can be used for both classification and regression

(B) The decision boundary looks smoother with smaller values of k.

(C) As k increases, the variance usually increases

(D) None of the above.

▶ Part 2: Multiple linear regression (5.0 points)

(a) Set the seed of the random generator to 0 (`set.seed(0)`). Simulate $6000 \times 201 = 1206000$ independent random variables with the standard normal distribution. Store them into a matrix, then into a data frame with 6000 lines and 201 columns. Each of these columns is referred to as a “variable”. Useful commands: `rnorm, matrix, data.frame`. **(0.5 points)**

(b) Define a Gaussian multiple linear regression model using the last 200 variables to predict the first one and write a mathematical equation (do not write `R` code!) to define this regression model. Write a second mathematical equation defining the true generative model associated with the data. Compare both models and discuss. **(1.0 points)**

(c) Estimate the parameters of the linear model using the last 200 variables to predict the first one. Compute the number of coefficients assessed as significantly non-zero at level 5%. Comment the result. Useful commands: `summary(reg)$coefficients`. **(0.5 points)**

(d) Simulate a dataset of size $N = 1000$ of the following generating model:

$$\begin{aligned} X_{1,i} &= \varepsilon_{1,i} \\ X_{2,i} &= 3X_{1,i} + \varepsilon_{2,i} \\ Y_i &= X_{2,i} + X_{1,i} + 2 + \varepsilon_{3,i} \end{aligned}$$

where $i \in \{1, \ldots, N\}$ and the ε_{ij} are independent $\mathcal{N}(0,1)$ random variables. For a given i, what is the distribution of $(X_{1,i}, X_{2,i})$? Plot the clouds of points of the simulated values of $(X_{1,i}, X_{2,i})_{i=1,\ldots,n}$. What is its shape? Can you write an analytical formula for it? **(1.0 points)**

(e) Let us consider the following two regression models:

$$\begin{aligned} \text{Model A:} \quad Y_i &= \alpha_1 X_{1,i} + \alpha_0 + \tilde{\varepsilon}_{A,i} \\ \text{Model B:} \quad Y_i &= \beta_2 X_{2,i} + \beta_0 + \tilde{\varepsilon}_{B,i} \end{aligned}$$

where $\tilde{\varepsilon}_{A,i} \sim \mathcal{N}(0, \sigma_A^2)$ and $\tilde{\varepsilon}_{B,i} \sim \mathcal{N}(0, \sigma_B^2)$. What should be the values of $\hat{\alpha}_0, \hat{\alpha}_1, \hat{\sigma}_A^2, \hat{\beta}_0, \hat{\beta}_2, \hat{\sigma}_B^2$ when $N \to \infty$? Consider $N = 1000$ and check whether the estimates of the parameters are close to the true values that you've calculated. Now do `set.seed(3)` and simulate again a dataset $X_{1,i}, X_{2,i}, Y_i$ for $n = 10$. Estimate the parameters. What happens? **(1.0 points)**

(f) Let us now consider the full model

$$Y_i = \gamma_2 X_{2,i} + \gamma_1 X_{1,i} + \gamma_0 + \varepsilon_i$$

where $i \in \{1, \ldots, n\}$ and the ε_i are independent $\mathcal{N}(0, \sigma^2)$ random variables. For the previously simulated data with $n = 10$, estimate $\hat{\gamma}_0, \hat{\gamma}_1, \hat{\gamma}_2, \hat{\sigma}^2$ and compare them with the parameters obtained in item (b). What can you say about the effects of X_1 and X_2 on Y? And about their correlation? **(1.0 points)**

▶ Part 3: Classification (5.0 points)

Consider a simulated dataset which you will generate as follows:

(1) Set the seed of your `R` script with `set.seed(42)`.

(2) For each data point i, sample its label from a Bernoulli distribution $y_i \sim \mathcal{B}(p)$, i.e. $y_i = 1$ with probability p and $y_i = 0$ with probability $1 - p$.

> 💡 To sample a random variable B from $\mathcal{B}(p)$ you can first sample U from an uniform distribution with function `runif` from the `stats` package and then $B = \mathbf{1}(U < p)$ where $\mathbf{1}(\cdot)$ is an indicator function.

(3) Then, depending on the label $y_i \in \{0, 1\}$ the associated data point $\mathbf{x}_i \in \mathbb{R}^2$ is sampled as follows:

$$y_i = 0 \quad \Rightarrow \quad \mathbf{x}_i \sim 0.5\, \mathcal{N}(\boldsymbol{\mu}_0^{(a)}, \boldsymbol{\Sigma}_0) + 0.5\, \mathcal{N}(\boldsymbol{\mu}_0^{(b)}, \boldsymbol{\Sigma}_0)$$

$$y_i = 1 \quad \Rightarrow \quad \mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ with pdf

$$p_{\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}(x) = \frac{1}{2\pi\sqrt{\det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right)$$

and

$$\boldsymbol{\mu}_0^{(a)} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \boldsymbol{\mu}_0^{(b)} = \begin{bmatrix} 0 \\ +1 \end{bmatrix} \quad \boldsymbol{\mu}_1 = \begin{bmatrix} \varepsilon \\ 0 \end{bmatrix} \quad \boldsymbol{\Sigma}_0 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}$$

> 💡 To sample a p-dimensional vector $\mathbf{x}$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, you can use the function `mvrnorm` from the `MASS` package. Note that to generate samples from a Gaussian mixture model such as the one for $y_i = 0$ you need to first sample from a Bernoulli $\mathcal{B}(0.5)$ and then, depending on its value, sample from the first or second term of the mixture.

We will denote a set of N data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ simulated with ε and p as $\mathcal{D}(N \mid \varepsilon, p)$.

$$\mathcal{D}_{\text{train}} = \mathcal{D}(50 \mid 1, 0.2) \quad \text{and} \quad \mathcal{D}_{\text{test}} = \mathcal{D}(1000 \mid 1, 0.2)$$

(a) Plot the data points in $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$ using different colors to indicate the classes of each data point and different pointing symbols to indicate whether a point is from the train or test set. **(2.0 points)**

> 💡 The keyword `pch` from the `plot` function allows you to change the symbol of the scatter plot. For instance, using `pch=1` will give circles and `pch=6` are triangles pointing down.

(b) What is the mathematical expression for the optimal Bayes classifier in this setting? And for its boundary region? Do you expect it to be linear? **(1.0 points)**

> 💡 Remember that the Bayes classifier can be written in terms of the ratio of $\text{Prob}(Y = 1 \mid \mathbf{x})$ over $\text{Prob}(Y = 0 \mid \mathbf{x})$ and that the values of $\mathbf{x} \in \mathbb{R}^2$ for which this ratio is 1 are those defining its boundary.

(c) Estimate the error of the Bayes classifier on the samples from $\mathcal{D}_{\text{test}}$. How you would expect it to change in terms of ε? (**1.0 points**)

(d) Given the structure of the model generating the datasets, which classifier presented in our lectures you would expect to be the most adequate? **(0.5 points)**

(e) Train a LDA, a QDA, and a Logistic Regression classifier on $\mathcal{D}_{\text{train}}$ and estimate their errors on the samples from $\mathcal{D}_{\text{test}}$. How do their errors compare to the value obtained in (b)? **(0.5 points)**

▶ Part 4: Community detection (4.0 points)

In this part you will be using the `igraph` package from TP4 which is already installed in your PC.

Remember to load it with `library(igraph)`.

The documentation for `igraph` is available at `/references/R/igraph.pdf`

Note that some questions can be easily answered if you've read the paper "Modularity and community structure in networks" (available to you at `/references/articles/article-newman.pdf`)

We will consider a famous example of a social network from the network science literature called Wayne Zachary's "karate club" network. This network represents the pattern of friendships between members of a karate club at a US university, as determined by direct observation of the club's members over an extended period. This network is interesting because during the period of observation a dispute arose among the members of the club over whether to raise the club's fees and as a result the club eventually split into two parts, of 18 and 16 members respectively, the latter departing to form their own club. It is the two factions in this split, as reported by Zachary, that will form the ground truth for us when applying different algorithms for community detection.

Start by loading the dataset with `load('./karate.rda')` so the graph of interest will be stored at the `karate` variable of the workspace.

(a) Define in your own words the notion of modularity of a network and how it can be used to split a network into communities. Your description should include whether the modularity depends solely on the structure of the network or not. **(0.5 points)**

(b) The true factions to which each member of the karate club belongs to can be obtained via `V(karate)$Faction`. Use this information to calculate the modularity of the graph with for this ground truth setup. **(1.0 points)**

(c) Calculate the modularity matrix of the graph using the function `modularity_matrix` and obtain its eigenvectors. Interpret the magnitude of the coordinates of the leading eigenvector (i.e. the one related to the largest eigenvalue) and explain how it can be related to importance of each vertex in the graph. How can we use this eigenvector to split the graph into two communities? **(1.0 points)**

(d) Are there any nodes for which the split in (b) looks more ambiguous than others? Which aspect of the eigenvectors of the modularity matrix could be useful to check this information? **(0.75 points)**

(e) What would happen if all the eigenvalues of the modularity matrix were smaller than zero? What would this indicate in terms of the structure of the network? **(0.75 points)**