Ensimag – Grenoble INP – UGA
Statistical Analysis and Document Mining
Pedro L. C. Rodrigues
Jacopo Iollo

Year 2022-2023

pedro.rodrigues@inria.fr
jacopo.iollo@inria.fr

Final **written** exam – Duration: 2h

This exam is composed of four parts.

All students should do Part 1.

You should then choose to answer only **two parts out of the remaining three**.

For example, you can do just Part 1, Part 2, and Part 4.

Or maybe Part 1, Part 3, and Part 4.

And so on.

The maximum final grade is 20.0 points.

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

Good luck!

▶ Part 1: Multiple choice questions (8.0 points)

For each question, cross the letter corresponding to the correct answer.

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

– Question 1: Robustness to outliers (credits to EPFL CS-433) (1.0 points)

We consider a classification problem on linearly separable data. Our dataset had an outlier – a point that is very far from the other datapoints in distance. We trained the linear discriminant analysis (LDA), logistic regression and 1-nearest-neighbour classifiers on this dataset. We tested trained models on a test set that comes from the same distribution as training set, but doesn't have any outlier points. After that we removed the outlier and retrained our models.

After retraining, which classifier will **not change** its decision boundary around the test points.

(A) Logistic regression.

(B) 1-nearest-neighbors classifier.

(C) LDA.

(D) None of them.

– Question 2: Bias-variance decomposition (credits to EPFL CS-433) (1.0 points)

Consider a regression model where data (x, y) is generated by input $x \in \mathbf{R}$ uniformly sampled between $[0, 1]$ and $y = x + \varepsilon$, where ε is random noise with mean 0 and variance 1. Two models are carried out for regression: model $\mathcal{A}$ is a trained quadratic function $g_{\mathcal{A}}(x, \boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2$ and model $\mathcal{B}$ is a constant function $g_{\mathcal{B}}(x) = \frac{1}{2}$.

Compared to model $\mathcal{B}$, model $\mathcal{A}$ has

(A) Higher bias, higher variance.

(B) Lower bias, higher variance.

(C) Higher bias, lower variance.

(D) Lower bias, lower variance.

– Question 3: Linear regression (credits to EPFL CS-433) (1.0 points)

Assume we are doing linear regression with mean-squared loss and L2-regularization on four one-dimensional data points. Our prediction model can be written as $f(x) = ax + b$ and the optimization problem can be written as

$$a^\star, b^\star = \underset{a,b}{\operatorname{argmin}} \sum_{i=1}^{4} \Big(y_i - f(x_i)\Big)^2 + \lambda a^2$$

Assume that our data points (x_i, y_i) are $\{(-2, 1), (-1, 3), (0, 2), (3, 4)\}$.

What is the optimal value for the bias, $b^\star$?

(A) Depends on the value of λ.

(B) 3

(C) 2.5

(D) None of the above answers.

– Question 4: PCA (credits to EPFL CS-433) (1.0 points)

Which of the following transformations to a data matrix $\mathbf{X}$ will affect the principal components obtained through PCA?

(A) Adding a constant value to all elements of $\mathbf{X}$.

(B) Multiplying one of the features of $\mathbf{X}$ by a constant.

(C) Adding an extra feature to $\mathbf{X}$ (i.e. an extra column) that is constant across all data points.

(D) None of the above answers.

– Question 5: 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:

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

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 6: Logistic regression (credits to EPFL CS-433) (1.0 points)

Consider the logistic regression loss $\mathcal{L} : \mathbb{R}^p \to \mathbb{R}$ for a binary classification task with data $(\mathbf{x}_i, y_i) \in \mathbb{R}^p \times \{0, 1\}$:

$$\mathcal{L}(\boldsymbol{\beta}) = \frac{1}{N} \sum_{i=1}^{N} \left(\log\left(1 + e^{\mathbf{x}_i^\top \boldsymbol{\beta}}\right) - y_i \mathbf{x}_i^\top \boldsymbol{\beta}\right)$$

Which of the following is a gradient of the loss $\mathcal{L}$?

(A) $\nabla \mathcal{L}(\boldsymbol{\beta}) = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_i \frac{e^{\mathbf{x}_i^\top \boldsymbol{\beta}}}{1 + e^{\mathbf{x}_i^\top \boldsymbol{\beta}}} - y_i \mathbf{x}_i^\top \boldsymbol{\beta}\right)$

(B) $\nabla \mathcal{L}(\boldsymbol{\beta}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \left(y_i - \frac{e^{\mathbf{x}_i^\top \boldsymbol{\beta}}}{1 + e^{\mathbf{x}_i^\top \boldsymbol{\beta}}}\right)$

(C) $\nabla \mathcal{L}(\boldsymbol{\beta}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \left(\frac{1}{1 + e^{-\mathbf{x}_i^\top \boldsymbol{\beta}}} - y_i\right)$

(D) None of the above.

– Question 7: Linear regression (credits to Berkeley CS-189) (1.0 points)

In linear regression, we model $p(y \mid \mathbf{x}) \sim \mathcal{N}(\beta^\top \mathbf{x} + \beta_0, \sigma^2)$. The irreducible error in this model is

(A) σ^2

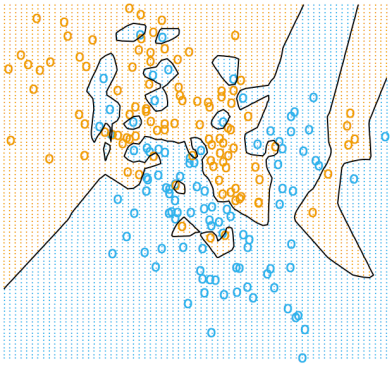
(B) $\mathbb{E}[y \mid \mathbf{x}]$

(C) $\mathbb{E}[(y - \mathbb{E}[y \mid \mathbf{x}])^2 \mid \mathbf{x}]$

(D) None of the above.

– **Question 8: Classifier boundary (credits to EPFL CS-189) (1.0 points)**

Which of these classifiers could have generated the decision boundary here below



(A) Logistic regression

(B) 1-NN

(C) Quadratic discriminant analysis

(D) None of the above.

▶ Part 2: Weighted linear regression (6.0 points)

Suppose we have a regression dataset with N pairs $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. We wish to fit a linear model $f(\mathbf{x}_i) = \boldsymbol{\beta}^\top \tilde{\mathbf{x}}_i$ where $\boldsymbol{\beta}$ is a vector with entries $\beta_0, \beta_1, \ldots, \beta_p$ and $\tilde{\mathbf{x}}_i^\top = [1\ \mathbf{x}_i^\top]$.

Suppose we minimize the following cost function:

$$\mathcal{L}(\boldsymbol{\beta}) = \frac{1}{2} \sum_{i=1}^{N} w_i \Big(y_i - \boldsymbol{\beta}^\top \tilde{\mathbf{x}}_i \Big)^2$$

where $w_i > 0$ are known real-valued scalars.

(a) Calculate the gradient of the loss function and make it equal to zero. You should write an expression in matrix-vector form similar to the expressions for least-squares given in the lectures. **(2.5 points)**

(b) Discuss the conditions under which the solution $\beta^\star$ is unique. **(1.5 points)**

(c) Assuming that these conditions hold, write down the expression for the unique solution. **(1.0 points)**

(d) Interpret the role of the weights w_i in the results of the regression. In other words, explain how adding them to the usual linear regression model can be useful for certain situations. **(1.0 points)**

▶ Part 3: Principal component analysis (6.0 points)

We consider the dataset `cars04`, which describes several properties of different car models in the market in 2004. Each observation (i.e. car) is described by 11 features (i.e. properties) listed in Table 1.

Variable	Meaning
`Retail`	Builder recommended price(US$)
`Dealer`	Seller price (US$)
`Engine`	Motor capacity (liters)
`Cylinders`	Number of cylinders in the motor
`Horsepower`	Engine power
`CityMPG`	Consumption in city (Miles or gallon; proportional to km/liter)
`HighwayMPG`	Consumption on roadway (Miles or gallon)
`Weight`	Weight (pounds)
`Wheelbase`	Distance between front and rear wheels (inches)
`Length`	Length (inches)
`Width`	Width (inches)

Table 1: Variable list for `cars04`

The aim of this exercise is to summarize and to interpret the data `cars04` using PCA. Using `R` we run the following instruction:

```
cars04.pca <- prcomp(cars04, scale=TRUE)
summary(cars04.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.6655 1.3726 0.92181 0.59751 0.52482 0.44491 0.37486
## Proportion of Variance 0.6459 0.1713 0.07725 0.03246 0.02504 0.01799 0.01277
## Cumulative Proportion  0.6459 0.8171 0.89439 0.92685 0.95189 0.96988 0.98266
##                           PC8     PC9    PC10    PC11
## Standard deviation     0.29434 0.25766 0.19229 0.02811
## Proportion of Variance 0.00788 0.00604 0.00336 0.00007
## Cumulative Proportion  0.99053 0.99657 0.99993 1.00000
```

(a) What is the effect of the argument `scale=TRUE` in the result of the PCA? **(1.5 points)**

The command `scale=TRUE` forces the variance of each column of the data matrix to be equal to one. This ensures that all predictors are comparable and avoids distortions in one direction.

(b) Are the first two principal components enough to summarize most of the information (i.e. variance) of the dataset? Justify in terms of the proportion of the total variance that they represent. **(1.5 points)**

The first two principal components sum 80% of the total variance of the dataset, which is quite high. Furthermore, we note that adding a 3rd component would only explain 7% more of the variance, which we could argue as being too small to justify adding it.

Principal components are linear combinations of the 11 variables. The coefficients of the first 2 principal components on the 11 feature are

```
cars04.pca$rotation[,1:2]
```

```
##                    PC1          PC2
## Retail     -0.2637504 -0.468508698
## Dealer     -0.2623186 -0.470146585
## Engine     -0.3470805  0.015347186
## Cylinders  -0.3341888 -0.078032011
## Horsepower -0.3186023 -0.292213476
```

```
## CityMPG     0.3104817  0.003365936
## HighwayMPG  0.3065886  0.010964460
## Weight     -0.3363294  0.167463572
## Wheelbase  -0.2662100  0.418177107
## Length     -0.2567902  0.408411381
## Width      -0.2960546  0.312891350
```

(c) What would be a good interpretation for these new variables in terms of the initial features of the dataset? **(1.0 points)**

PC1 represents features related to the motor of the cars. Positive values of PC1 describe cars with larger and more powerful engines (large coefficients for ***`Engine`*** *and* ***`Cylinders`****), which tend to consume more energy, and negative values of PC1 relate to cars that are more energy efficient (large coefficients of* ***`CityMPG`*** *and* ***`HighwayMPG`****).*

In PC2, large positive values relate to cars with big dimensions (***`Wheelbase`*** *and* ***`Length`****) and negative values represent expensive cars (* ***`Retail`*** *and* ***`Dealer`****).*

Figure 1 shows the projection of the dataset on its first two principal components.

(d) Interpret each quadrant of the figure. **(1.0 points)**

Wenote from the plot in Figure 1A that the efficiency of a car is in completely opposition to the size of its motor. Also, more expensive cars tend to be more compact, lighter, and with more horsepower.

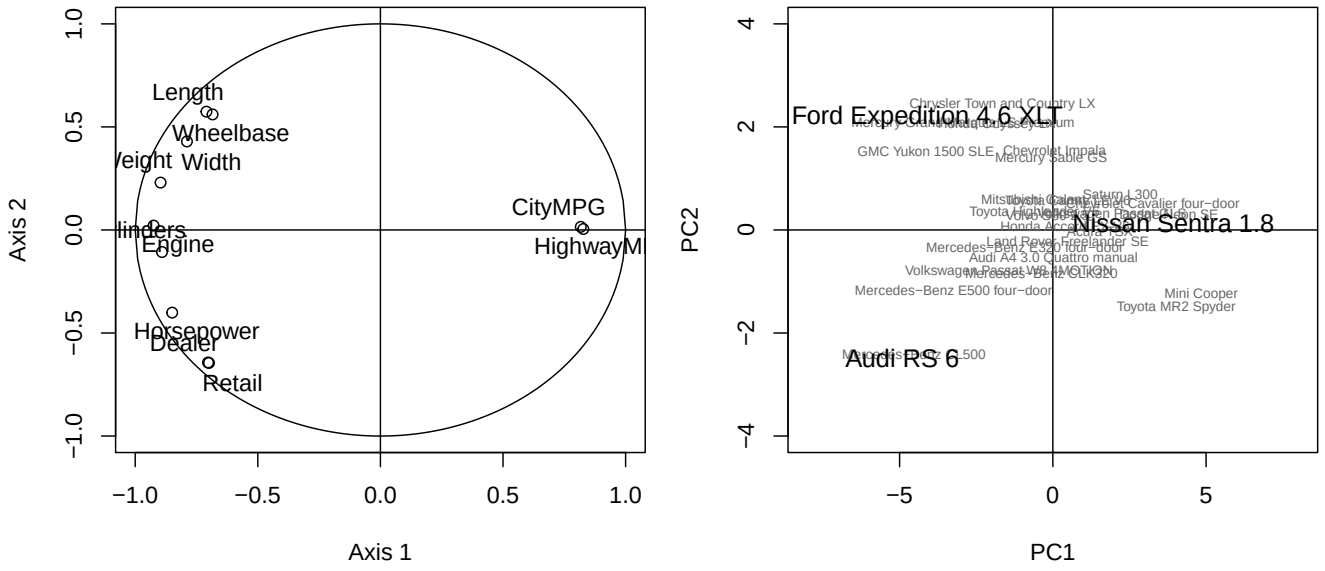
The quadrants of Figure 1B are directly related to the explanations given in item (C).

(e) Can you describe which kind of car Audi RS 6, Ford Expedition 4.6 XLT and Nissan Sentra 1.8 are? **(1.0 points)**

– Audi RS 6: It is an expensive car which is not very energy efficient. It is also not very heavy and rather compact.

– Ford Expedition 4.6 XLT: It is a big and heavy car which is not very energy efficient.

– Nissan Sentra 1.8: It is a small, compact and energy efficient car. It is more expensive than the Ford Expedition but cheaper than Audi RS 6.



(a) Variable space (b) Individual space

Figure 1: Principal component representation in the first plane of the variable and of the sample spaces.

▶ Part 4: Spectral community detection (6.0 points)

In this part, you will be asked a few questions related to the content of the article that I've asked you to read before the final exam: "*Modularity and community structure in networks*" by Mark Newman.

(a) Give two examples of applications of community detection in networks in real life. **(1.0 points)**

– Community detection can be used to detect groups within the worldwide web that might correspond to sets of web pages on related topics.

– Community detection can be used to detect groups within social networks that might correspond to social units or communities.

(b) Give the definition of modularity and explain why it is a good target quantity when investigating community structures in networks. **(1.5 points)**

The modularity is, up to a multiplicative constant, the number of edges falling within groups minus the expected number in an equivalent network with edges placed at random.

Modularity can be either positive or negative, with positive values indicating the possible presence of community structure. Thus, one can search for community structure pre- cisely by looking for the divisions of a network that have positive, and preferably large, values of the modularity.

(c) What is the interpretation in terms of community detection when the modularity matrix of a network has no positive eigenvalues? **(1.0 points)**

When the modularity matrix has no positive eigenvalues, it indicates that the graph can't be satisfactorily split into two communities and, therefore, should be left as it is.

(d) What information does the magnitude of each coordinate of the leading eigenvector of the modularity matrix convey regarding the split of a network into two communities? **(1.5 points)**

The magnitudes of the leading eigenvector indicate which vertices corresponding make the largest contributions to the modularity. These values indicate how confident/certain we can be of the the group to which the vertex has been assigned to.

(e) What is the procedure proposed by the author of the paper for splitting a network into more than two communities using his spectral algorithm? Is it capable of proposing a split into three communities? Why? **(1.0 points)**

The paper proposed a bissection procedure, where each group of the graph is further divided into new groups. Thanks to its natural stopping criterium (i.e. no positive eigenvalues when the graph should no longer be split) the algorithm can indeed detect three communities: it would first detect two, then one of the sub-groups would be further split into two whereas the other one would not.