**Advanced Statistics for Machine Learning**

**Exam 2024**

**Samd Guizani**

**(A23 – SPOC)**

**Table of contents**

[I. Exercise 1 3](#_Toc180696422)

[1. Question 1 3](#_Toc180696423)

[2. Question 2 5](#_Toc180696424)

[a. Model involving all the explanatory variables 6](#_Toc180696425)

[b. Model with constraint 7](#_Toc180696426)

[c. Models’ comparison 8](#_Toc180696427)

[II. Exercise 2 10](#_Toc180696428)

[1. Question 1 11](#_Toc180696429)

[a. Dataset description and preprocessing 11](#_Toc180696430)

[b. Exploratory data analysis 11](#_Toc180696431)

[c. Candidate models development 12](#_Toc180696432)

[d. Model performance evaluation and comparison 17](#_Toc180696433)

[2. Question 2 19](#_Toc180696434)

[III. References 20](#_Toc180696435)

# Exercise 1

## Question 1

This problem is a constrained least square optimization of a multiple linear regression (MLR). So, the aim is to find a vector of coefficients such that:

*N.B.: (1/2) coefficient in J(β) is introduced to avoid carrying multiplication by 2 when later taking the derivative.*

Where *R* is a q x (p+1) matrix and *r* is a vector of size q, q being the number of constraints and also the rank of *R*. So, *R* and *r* define a set of q linear constraints on the coefficients of the vector *β* (*R* holds the weights on the constrained coefficients and *r* holds the weighted sum to reach).

First, let us check the existence and uniqueness of a solution:

* The set is a **closed set** as it is defined by equality constraints.
* ***J(β)* is continuous** and ***α*-convex** as it is a squared norm.

We conclude that there exists at least a minimum of *J* on *K*.

Moreover:

* *K* is a **convex set**, indeed considering 2 points *β1* and *β2* in *K* i.e. and considering any point , we get:

So, *βθ* belongs to *K*.

* ***J(β)* is strictly convex** (being *α*-convex)

We conclude that there exists at most one minimum of *J* on *K*.

To find the minimum, since J and the linear constraints are differentiable and the constraints are regular, we use the Lagrange multipliers to find the minimum of *J*:

Where *λ* is a vector of size q.

The derivatives of *J* and *F* are:

Then:

To minimize *J(β)*, we must find the critical point of the Lagrangian, such that:

From the 1st equation, we deduce an expression of :

being the unconstrained solution of the minimization problem.

Substituting in the 2nd equation, we deduce an expression of *λ*:

Finally, injecting the expression of *λ* in the expression of , we obtain:

## Question 2

In the Ozone dataset, 2 variables are categorical: pluie (with 2 levels “Pluie” and “Sec”) and vent (with 4 levels "Est”, “Nord”, “Ouest” and “Sud”). To facilitate calculations and comparison of standard R software lm function with our computation of multiple linear regression (constrained or not), the categorical variables have been preprocessed using a one-hot key encoding. In this encoding, each categorical variable is converted to *L-1* binary variables, where *L* corresponds to the number of levels of the original categorical variable. To illustrate the process, the 6 first observations are reported hereafter, before and after one-hot key encoding.

*Before pre-processing*:A screenshot of a computer

Description automatically generated

*After pre-processing*:A screenshot of a computer

Description automatically generated

### Model involving all the explanatory variables

Applying R software lm function, we obtain a model predicting maxO3 from all the explanatory variables. The summary is:

> LM = lm(formula = maxO3 ~ ., data = Dataset\_ozone)

> summary(LM)

Call:

lm(formula = maxO3 ~ ., data = Dataset\_ozone)

Residuals:

Min 1Q Median 3Q Max

-51.814 -8.695 -1.020 7.891 40.046

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 16.26536 15.94398 1.020 0.3102

T9 0.03917 1.16496 0.034 0.9732

T12 1.97257 1.47570 1.337 0.1844

T15 0.45031 1.18707 0.379 0.7053

Ne9 -2.10975 0.95985 -2.198 0.0303 \*

Ne12 -0.60559 1.42634 -0.425 0.6721

Ne15 -0.01718 1.03589 -0.017 0.9868

Vx9 0.48261 0.98762 0.489 0.6262

Vx12 0.51379 1.24717 0.412 0.6813

Vx15 0.72662 0.95198 0.763 0.4471

maxO3v 0.34438 0.06699 5.141 1.42e-06 \*\*\*

ventNord 0.53956 6.69459 0.081 0.9359

ventOuest 5.53632 8.24792 0.671 0.5037

ventSud 5.42028 7.16180 0.757 0.4510

pluieSec 3.24713 3.48251 0.932 0.3534

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.51 on 97 degrees of freedom

Multiple R-squared: 0.7686, Adjusted R-squared: 0.7352

F-statistic: 23.01 on 14 and 97 DF, p-value: < 2.2e-16

We can calculate (with their 95% confidence intervals), the unconstrained least square solution, as following:

> hbeta = solve(t(X) %\*% X) %\*% t(X) %\*% Y

> print(hbeta\_ci)

Est. Coef. Std. Error Lower Bound Upper Bound

(Intercept) 16.26535597 15.94398012 -15.3790310 47.9097430

T9 0.03916979 1.16495679 -2.2729470 2.3512865

T12 1.97257424 1.47570493 -0.9562915 4.9014400

T15 0.45030800 1.18707252 -1.9057023 2.8063183

Ne9 -2.10975486 0.95985471 -4.0148008 -0.2047090

Ne12 -0.60559218 1.42633808 -3.4364784 2.2252941

Ne15 -0.01717804 1.03589370 -2.0731403 2.0387842

Vx9 0.48260889 0.98762405 -1.4775515 2.4427692

Vx12 0.51379495 1.24716744 -1.9614872 2.9890771

Vx15 0.72662334 0.95197628 -1.1627861 2.6160327

maxO3v 0.34437835 0.06699148 0.2114188 0.4773379

ventNord 0.53956395 6.69459345 -12.7473509 13.8264788

ventOuest 5.53631722 8.24792304 -10.8335269 21.9061613

ventSud 5.42028442 7.16180048 -8.7939071 19.6344759

pluieSec 3.24713025 3.48251475 -3.6646975 10.1589580

We can confirm that the calculated coefficients are identical to lm() function results.

### Model with constraint

If we apply a constraint on the variables T9, T12 and T15 such that , we obtain the following solution:

> hbeta\_c =

+ hbeta +

+ solve(t(X) %\*% X) %\*%

+ t(R) %\*%

+ solve(R %\*% solve(t(X) %\*% X) %\*% t(R)) %\*%

+ (r - R %\*% hbeta)

> print(hbeta\_c\_ci)

Est. Coef. Std. Error Lower Bound Upper Bound

(Intercept) 59.42206714 17.016295 25.6494312 93.1947031

T9 -1.87215990 1.243306 -4.3397785 0.5954587

T12 1.17078960 1.574954 -1.9550576 4.2966368

T15 0.70137031 1.266909 -1.8130939 3.2158346

Ne9 -2.97625939 1.024410 -5.0094295 -0.9430893

Ne12 -1.37664117 1.522267 -4.3979192 1.6446369

Ne15 0.08932548 1.105563 -2.1049109 2.2835618

Vx9 -0.16837184 1.054047 -2.2603631 1.9236195

Vx12 0.53454939 1.331046 -2.1072083 3.1763071

Vx15 0.77110593 1.016002 -1.2453760 2.7875879

maxO3v 0.46895526 0.071497 0.3270535 0.6108570

ventNord -5.14140434 7.144840 -19.3219330 9.0391244

ventOuest 3.94210314 8.802639 -13.5286978 21.4129040

ventSud 6.50590711 7.643469 -8.6642624 21.6760766

pluieSec 5.28638433 3.716732 -2.0902997 12.6630683

### Models’ comparison

The 2 models can be compared using statistical metrics as presented in the following table:

|  |  |  |
| --- | --- | --- |
|  | Unconstrained MLR | Constrained MLR |
| R² | 0.7686 | 0.7364 |
| Adjusted R² | 0.7352 | 0.6983 |
| Residuals Standard Error | 14.51 | 15.48 |

Table – Comparison of unconstrained vs. constrained MLR performance

As expected, the constrained model shows lower R² values and higher Residuals Standard Error. Indeed, the unconstrained least square minimization yields a set of coefficients , that allows the predictions to be as close as possible to the observed datapoints. Hence, the Residuals Standard Error is the lowest and the R²/Adjusted R² are the highest. On the other hand, constraining the coefficients of the variables T9, T12 and T15, yields a set of coefficients for which the predictions are more distant from the observed dataset, compared to the unconstrained solution. Consequently, Residuals Standard Error grows and R²/Adjusted R² are lower.

On the following figure, the 2 models can be graphically compared by plotting the predicted response of each model against the actual response.



Figure – Comparative plot of constrained vs. unconstrained MLR models

It is to be observed that the constrained and unconstrained models are very close, despite the restrictions applied to the coefficients of the variables T9, T12 and T15. This may be explained by the correlations that exist among the explanatory variables which can be evaluated through the correlation matrix of the explanatory variables:

> # Correlation matrix of explanatory variables

> print(round(cor(X[,(1:p)+1]), 2))

T9 T12 T15 Ne9 Ne12 Ne15 Vx9 Vx12 Vx15 maxO3v ventNord ventOuest ventSud pluieSec

T9 1.00 0.88 0.85 -0.48 -0.47 -0.33 0.25 0.22 0.17 0.58 -0.19 -0.09 0.26 0.38

T12 0.88 1.00 0.95 -0.58 -0.66 -0.46 0.43 0.31 0.27 0.56 -0.22 -0.10 0.26 0.44

T15 0.85 0.95 1.00 -0.59 -0.65 -0.57 0.45 0.34 0.29 0.57 -0.20 -0.09 0.23 0.42

Ne9 -0.48 -0.58 -0.59 1.00 0.79 0.55 -0.50 -0.53 -0.49 -0.28 -0.11 0.32 -0.08 -0.39

Ne12 -0.47 -0.66 -0.65 0.79 1.00 0.71 -0.49 -0.51 -0.43 -0.36 -0.16 0.37 -0.11 -0.42

Ne15 -0.33 -0.46 -0.57 0.55 0.71 1.00 -0.40 -0.43 -0.38 -0.31 -0.18 0.29 0.02 -0.29

Vx9 0.25 0.43 0.45 -0.50 -0.49 -0.40 1.00 0.75 0.68 0.34 -0.02 -0.39 0.20 0.42

Vx12 0.22 0.31 0.34 -0.53 -0.51 -0.43 0.75 1.00 0.84 0.22 0.22 -0.64 0.12 0.30

Vx15 0.17 0.27 0.29 -0.49 -0.43 -0.38 0.68 0.84 1.00 0.19 0.20 -0.53 0.04 0.21

maxO3v 0.58 0.56 0.57 -0.28 -0.36 -0.31 0.34 0.22 0.19 1.00 -0.01 -0.06 0.12 0.38

ventNord -0.19 -0.22 -0.20 -0.11 -0.16 -0.18 -0.02 0.22 0.20 -0.01 1.00 -0.56 -0.30 0.08

ventOuest -0.09 -0.10 -0.09 0.32 0.37 0.29 -0.39 -0.64 -0.53 -0.06 -0.56 1.00 -0.43 -0.25

ventSud 0.26 0.26 0.23 -0.08 -0.11 0.02 0.20 0.12 0.04 0.12 -0.30 -0.43 1.00 0.14

pluieSec 0.38 0.44 0.42 -0.39 -0.42 -0.29 0.42 0.30 0.21 0.38 0.08 -0.25 0.14 1.00

Noticeably, variables T9, T12 and T15 are correlated negatively with Ne9, Ne12 and Ne15 and positively with maxO3v. Hence, the constraint on the coefficients of T9, T12 and T15 get “distributed” on other variables. This finally leads to a different model (refer to section 2a and 2b where the coefficients of the 2 models are reported) which overall is closely comparable to the unconstrained solution.

# Exercise 2

2 datasets are provided, each of them has 1 response variable and multiple explanatory variables. The objective is to build different models for each dataset to predict the response variable, compare their performance and select the best one.

The general approach for both questions is the following:

* Step 1: Each dataset is split in a training set and a test set.
* Step 2: Exploratory data analysis is performed on the training dataset to better understand the data.
* Step 3: Several modelling options are considered and fine-tuned. At this step, only the training set is used. The fine-tuned models form a candidate list to be evaluated.
* Step 4: Thanks to statistical metrics, the performance of the candidate models is evaluated on the test set to define which model is the best.

## Question 1

### Dataset description and preprocessing

The dataset has 77 observations, 200 numeric explanatory variables and a binary categorical response variable (coded -1/1). Hence the goal is to develop a classification model.

No preprocessing is applied to the data. However, the observations have been randomly split between a training set (70% of the observations) and a test set (30% of the observations).

### Exploratory data analysis

Linear correlation coefficients of each explanatory variable against the response variable (transformed to a numeric value -1 or +1) have been computed and plotted on a bar chart (Figure 2). We notice that the 6 first variables are the most correlated to the response variable. Hence, the developed models will integrate variable selection strategies to obtain sparse, easy to explain and robust-to-noise models.



Figure – Correlation coefficients of explanatory variables vs. response variable

### Candidate models development

Several models’ options have been considered:

* Generalized Linear Model, associated with penalty approach to perform variable selection
* CART decision trees, associated with pruning to achieve variable selection
* Random Forest
* Random Forest for variable selection, followed by CART tree fitting on the selected variables

#### Generalized Linear Regression (using R package “glmnet”)

The response is a binary categorical variable. Therefore, using Multiple Linear Regression, which considers a numeric response, is not appropriate. A more suited approach is to use Generalized Linear Models and specifically a binary Logistic Regression [1] that models the probability of an observation being in class (+1) as:

An alternative writing is called “log-odds”:

The objective is to find set of coefficients that minimizes the classification error of training set observations.

When the number of variables exceeds the number of observations, fitting a reliable model can be very difficult and unreliable. Therefore, a regularized approach, using LASSO, has been chosen to develop a sparse model. The tuning of the hyperparameter has been done through cross-validation. The value that minimizes cross-validation error is 0.09965 and yielded the selection of 5 explanatory variables (Figure 3), which are V1, V2, V3, V5 and V6. These variables are among the most correlated with the response variable.



Figure – Logistic Regression misclassification error as function of hyperparameter, using a LASSO regularization

#### CART decision tree (using R package “rpart”)

The principle is to build a maximal decision tree, for which the leaves will be pure and then prune it thanks to tuning the hyperparameter cp. The tuning is performed by cross-validation (the tuned value is 0.0769).

CART allows also to provide a ranking of the explanatory variables’ importance which yielded respectively the top 4 variables V2, V3, V1 and V6. The decision tree can also be plotted to explain how new observations would be classified (Figure 4).



Figure – CART tree for classification

We can note that V1 and V3 are not actually represented on the tree. The reason is that these variables are used in the surrogate splits, which makes it possible to predict the class of an observation for which V2 or V6 values would be missing.

#### Random forest (using R package “randomForest”)

Random Forrest builds a large number of decision trees. Each tree is trained on a randomly selected set of observations, taken with replacement from the training set. In addition, only a randomly selected subset of the variables contributes to developing each tree. This approach increases the variety of the trees which improves the reliability and robustness of the predicted class.

In this case, 500 trees have been developed, each of them using 14 variables. The out-of-bag error (i.e. comparing the actual class and the predicted class of observations not used to train a given tree) is 7.41%. A confusion matrix can also be calculated, showing the split of misclassification in term of “False Negative” and “False Positive”:

Confusion matrix:

-1 1 class.error

-1 28 0 0.0000000

1 4 22 0.1538462

Random Forest also provides a ranking of the variable importance which can be plotted as follow (Figure 5). Here again, we observe that the top 4 variables V3, V1, V2 and V6 are among the most correlated to the response variable.



Figure – Random Forest variable importance plot

#### Variable selection by Random Forest and CART modelling on selected variables (using R package “VSURF”)

One drawback of Random Forest is their lack of explainability. Hence, an alternative approach is to use Random Forest for variable selection and then build a CART model using only the selected variables. This variable selection strategy through Random Forest is implemented in “VSURF” R package and relies on 3 steps:

* Elimination: aiming to reduce the number of variable thanks to a threshold established based on the mean and standard deviation of variable importance scores.
* Interpretation: aiming to further reduce the number of variables to facilitate interpretation of the link between explanatory and response variables.
* Prediction: aiming to reduce the number of variables to the minimum required to achieve a reliable prediction.

In this case (Figure 6), 20 variables have been selected after Elimination step, which have been reduced to 5 at the Interpretation step (V3, V2, V1, V6, V5) and to 4 after the Prediction step (V3, V2, V1, V5).



Figure – Output of VSURF variable selection (5 variables kept after Interpretation step, 4 variables kept after Prediction step)

Using the Interpretation or the Prediction steps’ selected variables, 2 CART trees can be developed (Figure 7):

 

Figure – CART trees based on Random Forest variable selection   
(left: post-Interpretation step, right: port-prediction step)

It can be noted that the tree developed based on the variables selected at Interpretation step is essentially the same as the one developed through CART (after applying cross-validation pruning) while the treed based on Prediction step variables uses the surrogate split variables. Therefore, comparable performance of these trees is expected.

### Model performance evaluation and comparison

The performance of the candidate models is assessed on the test set observations, using 2 metrics:

* The test set **classification error** = fraction of misclassified observations in the test set.
* The test set **confusion matrix** [2], which shows, for each class, the count of correctly and wrongly classified observations.

The following table summarizes the performance measured for the candidate models.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Tuned hyperparameters | Test set classification error (%) | Test set confusion matrix |
| Logistic regression | = 0.09965 | 0 |  |
| CART (with pruning) | cp = 0.07692 | 8.7 |  |
| Random Forest | n trees = 500  n variables/tree = 14 | 8.7 |  |
| VSURF + CART (Interpretation step) | --- | 8.7 |  |
| VSURF + CART (Prediction step) | --- | 4.3 |  |

Table – Performance metrics of classification models on test set

As a conclusion, all the tested classification algorithms managed to select a sparse subset of explanatory variables required for predicting the class of an observation. However, they differ in terms of performance. The best model in this case is the Logistic Regression which has 0 prediction error on the test set. The second best is the model combining variable selection through Random Forest (VSURF prediction variables) followed by developing a CART decision tree. It has a 4.3% error rate on the test set, with only 1 misclassified observation. The rest of the models showed an error rate of 8.7% with slight variations in the misclassified observations.

## Question 2

### Dataset description and preprocessing

### Exploratory data analysis

### Candidate models development

### Model performance evaluation and comparison

# References

1. <https://glmnet.stanford.edu/articles/glmnet.html#quick-start>
2. <https://developer.ibm.com/tutorials/awb-confusion-matrix-r/>