TP 5 - Compte rendu de groupe

Matthias MAZET, Garance MALNOË

2025-12-08

Contents

0.1 *Préparation des données.* 2
0.2 *Estimation et stabilité de l'estimateur.* 3
0.3 *Sélection post-inférence.* 4

```
# Packages statistiques
library(glmnet)

# Packages de style
library(ggplot2)
library(ggpubr)
library(GGally)
library(tidyverse)
library(knitr)
library(kableExtra)
```

0.1 *Préparation des données.*

```
data <- read.csv("day.csv")
```

Pour simplifier les méthodes, nous supprimons la variable “dteday” qui est au format “chr”. Nous supprimons aussi “casual” et “registered” qui, en les sommant, donne la valeur de la variable à prédire “cnt”. Enfin, nous supprimons la variable “instant” qui correspond simplement à l’index de chaque observation. Nous vérifions ensuite les valeurs manquantes, la dimension des données et les valeurs aberrantes.

```
# Nettoyage
data <- data[, -c(1, 2, 14, 15)] # Suppression de variables

# Résumés statistiques des variables
kable(
  summary(data[, 1:6], digits = 2), caption = "Résumé statistiques des 6 premières variables",
  format = "latex", booktabs = TRUE, escape = FALSE, align = "r"
) %>%
  kable_styling(
    latex_options = c("striped", "HOLD_position", "scale_down"), font_size = 9
  ) %>%
  row_spec(0, bold=TRUE)
```

Table 1: Résumé statistiques des 6 premières variables

season	yr	mnth	holiday	weekday	workingday
Min. :1.0	Min. :0.0	Min. : 1.0	Min. :0.000	Min. :0	Min. :0.00
1st Qu.:2.0	1st Qu.:0.0	1st Qu.: 4.0	1st Qu.:0.000	1st Qu.:1	1st Qu.:0.00
Median :3.0	Median :1.0	Median : 7.0	Median :0.000	Median :3	Median :1.00
Mean :2.5	Mean :0.5	Mean : 6.5	Mean :0.029	Mean :3	Mean :0.68
3rd Qu.:3.0	3rd Qu.:1.0	3rd Qu.:10.0	3rd Qu.:0.000	3rd Qu.:5	3rd Qu.:1.00
Max. :4.0	Max. :1.0	Max. :12.0	Max. :1.000	Max. :6	Max. :1.00

```
kable(
  summary(data[, 7:12], digits = 2), caption = "Résumé statistiques des 6 dernières variables",
  digit = 3, format = "latex", booktabs = TRUE, escape = FALSE, align = "r"
) %>%
  kable_styling(
    latex_options = c("striped", "HOLD_position", "scale_down"), font_size = 9
  ) %>%
  row_spec(0, bold=TRUE)
```

Table 2: Résumé statistiques des 6 dernières variables

weathersit	temp	atemp	hum	windspeed	cnt
Min. :1.0	Min. :0.059	Min. :0.079	Min. :0.00	Min. :0.022	Min. : 22
1st Qu.:1.0	1st Qu.:0.337	1st Qu.:0.338	1st Qu.:0.52	1st Qu.:0.135	1st Qu.:3152
Median :1.0	Median :0.498	Median :0.487	Median :0.63	Median :0.181	Median :4548
Mean :1.4	Mean :0.495	Mean :0.474	Mean :0.63	Mean :0.190	Mean :4504
3rd Qu.:2.0	3rd Qu.:0.655	3rd Qu.:0.609	3rd Qu.:0.73	3rd Qu.:0.233	3rd Qu.:5956
Max. :3.0	Max. :0.862	Max. :0.841	Max. :0.97	Max. :0.507	Max. :8714

Il n’y a aucune valeurs manquantes (`sum(is.na(data))` $= 0$), aucune variables ne semblent avoir de valeurs aberrantes (d’après les résumés statistiques) et nous sommes dans un cas où $(n = 731) > (p = 12)$, avec n le nombre d’observations et p le nombre de variables. Aussi, la variable à prédire, “cnt”, compte le nombre d’emprunts de vélos réalisés dans une journée. Nous pouvons donc bien appliquer une régression de Poisson pour modéliser une variable de comptage. Dans ce but, nous séparons les données en X les variables explicatives, et y la variable à prédire.

```r
# Format matriciel pour glmnet
X <- as.matrix(data[colnames(data) != "cnt"])
y <- as.matrix(data$cnt)
```

0.2 *Estimation et stabilité de l'estimateur.*

Nous ajustons un modèle de Poisson via `glmnet`. Pour cela, nous précisons `family = "poisson"` dans les fonctions de `glmnet`. Nous conservons arbitrairement la valeur de `lambda.1se` obtenue par validation croisée.

```r
# Seed pour la reproductibilité
set.seed(5)

# Fit du modèle
lambda_1se <- cv.glmnet(x = X, y = y, alpha = 1, family = "poisson")$lambda.1se
mod_lasso <- glmnet(x = X, y = y, alpha = 1, lambda = lambda_1se, family = "poisson")
```

Nous obtenons une valeur de lambda.1se = 99.094, et les variabbles sélectionnées avec les données de base et cette valeur sont : season, yr, holiday, weekday, weathersit, temp, atemp, windspeed.

Regardons maintenant la stabilité de sélection de variables sur 100 itérations. Pour chaque itération, nous regardons quelles variables sont sélectionnées au `lambda.1se` exhibé précédemment et nous stockons l'information dans une matrice.

```r
set.seed(5)

# Nombre d'itérations
n_it <- 100

# Matrices des variables sélectionnées à chaque itérations
mat_vars <- matrix(data = 0, ncol = ncol(X), nrow = n_it)
colnames(mat_vars) <- colnames(X)

# Itérations boostrap
for (i in 1:n_it) {
  # Individus bootstrap
  ind_al <- sample(1:nrow(data), size = nrow(data), replace = TRUE)
  X <- as.matrix(data[ind_al, colnames(data) != "cnt"])
  y <- data[ind_al, "cnt"]
  # Fit du modèle
  mod_lasso <- glmnet(x = X, y = y, alpha = 1, lambda = lambda_1se, family = "poisson")
  # Variables sélectionnées à lambda.1se
  mat_vars[i, which(coef(mod_lasso) != 0)[-1] - 1] <- 1
}
```

```r
# Stabilité des variables
### Dataset de résultat au format long
df_lasso <- t(mat_vars) %>%
  as.data.frame() %>%
  mutate(variable = rownames(.)) %>%
  pivot_longer(cols = -variable, names_to = "iteration", values_to = "selected") %>%
  mutate(iteration = as.numeric(gsub("V", "", iteration)))
### Figure ggplot
ggplot(df_lasso, aes(x = iteration, y = variable, fill = selected)) +
  geom_tile() +
  scale_fill_gradientn(colours = c("white", "black"), limits = c(0, 1), name = "Sélection") +
  labs(x = "Itération", y = "Indices des variables") +
  theme_light() +
  theme(
```

```
    axis.text = element_text(size = 8),
    axis.line = element_line(colour = "grey"),
    panel.border = element_blank(),
    panel.grid = element_blank(),
    legend.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```

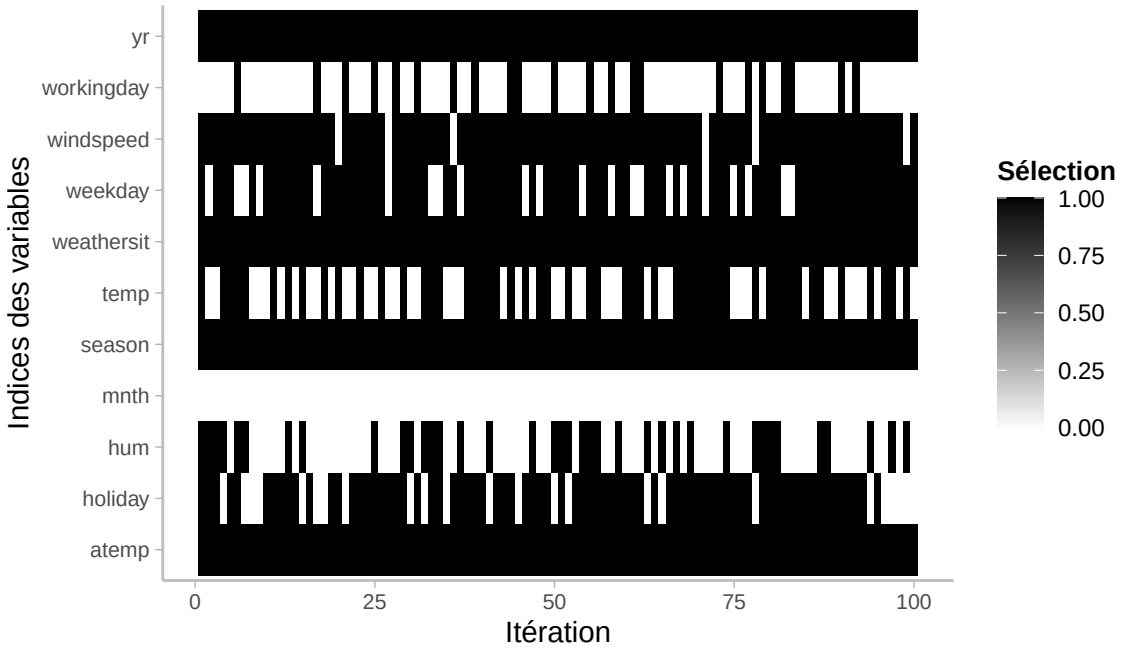

Figure 1: Stabilité à lambda.1se au fil des itérations

```
pourc_selection <- apply(mat_vars, FUN = mean, MARGIN = 2)

kable(
  t(pourc_selection), caption = "Fréquence de sélection de chaque variable",
  format = "latex", booktabs = TRUE, escape = FALSE, align = "r"
) %>%
  kable_styling(latex_options = c("striped", "HOLD_position", "scale_down"), font_size = 9) %>%
  row_spec(0, bold=TRUE)
```

Table 3: Fréquence de sélection de chaque variable

season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed
1	1	0	0.76	0.78	0.22	1	0.53	1	0.38	0.94

La figure précédente **(Fig. 1)** présente les variables sélectionnées à chaque itération, avec 1 = variable sélectionnée et 0 sinon (la légende de couleur devrait être discrète mais ggplot ne l'accepte pas sur une heatmap). On peut constater que certaines variables ("yr" ou "season" par exemple) sont toujours sélectionnées, tandis que d'autres le sont aléatoirement suivant l'itération ("hum" ou "temp" par exemple). La variable "mnth" n'est à l'inverse jamais sélectionnée. La méthode présente donc une légère instabilité sur la sélection de variables, ce qui peut témoigner de multicolinéarité entre certaines variables (notamment celles sélectionnées quelquefois mais pas toutes).

0.3 *Sélection post-inférence.*

1. *Construction d'un intervalle de confiance.*

```r
alpha <- .05
n <- nrow(data)
# Estimation des beta par glmnet
beta_hat <- coef(mod_lasso)[-1]
sd_beta_hat <- numeric(length(beta_hat))
for (i in 1:length(beta_hat)) {
  sd_beta_hat[i] <- sd(beta_hat[i])
}

# Intervalle de confiances
borne_inf <- beta_hat - (qnorm(1 - alpha/2)*sd_beta_hat / sqrt(n))
borne_sup <- beta_hat + (qnorm(1 - alpha/2)*sd_beta_hat / sqrt(n))
```