TP 5 - Régression de Poisson - Compte rendu de groupe

Matthias MAZET, Garance MALNOË

2025-12-13

Contents

```r
# Packages statistiques
library(glmnet)

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

1 *Préparation des données.*

Nous récupérons le jeu de données *bike sharing* à partir du fichier CSV disponible sur le site UCI.

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

Nous supprimons la variable `dteday` (jour étudié) qui se trouve au format "chr" et qui ne nous semble pas pertinente. Nous supprimons également les variables `casual` et `registered` qui correspondent aux deux types de location possibles et dont la somme est égale à la variable à prédire `cnt`. Enfin, nous supprimons aussi la variable `instant` qui correspond simplement à l'index de chaque observation.

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

Nous vérifions ensuite les valeurs manquantes, la dimension des données et les valeurs aberrantes.

```r
sum(is.na(data)) # Valeurs manquante
```

```
## [1] 0
```

```r
dim(data) # Dimensions du jeu de données
```

```
## [1] 731  12
```

Le jeu de données ne compte aucune valeur manquante (`sum(is.na(data))` $= 0$). Il est composé de $n = 731$ observations de $p = 12$ variables (une variable à prédire et onze variables prédictives). Il est bon de noter que nous sommes dans un cas où $n > p$.

Nous vérifions finalement la présence de données aberrantes.

```r
# Résumés statistiques des variables
kable(
  summary(data[, 1:6], digits = 2), caption = "Résumé statistique 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 statistique 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

```r
kable(
  summary(data[, 7:12], digits = 2), caption = "Résumé statistique 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 statistique 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

D'après les résumés statistiques, aucune variable ne semble avoir de valeur aberrante qui serait à exclure **(Tab. 1, Tab. 2)**.

Dans le cadre de notre étude, nous souhaitons prédire la variable `cnt`, qui compte le nombre d'emprunts de vélos réalisés dans une journée et est donc une variable de comptage. Nous pouvons donc bien appliquer une régression de Poisson afin de la modéliser.

2 *Estimation et stabilité de l'estimateur.*

Nous souhaitons ajuster un modèle de régression de Poisson tout en appliquant une régularisation LASSO afin d'effectuer une sélection des variables prédictives. Pour cela, nous utilisons le package `glmnet` en précisant l'argument `family = "poisson"` dans l'appel des fonctions. Nous choisissons de conserver la valeur de `lambda.1se` obtenue par validation croisée comme meilleure valeur du paramètre de régularisation.

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

# Seed pour la reproductibilité
set.seed(2)

# 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")
```

Le paramètre de régularisation retenu est `lambda.1se` = 99.094, et les variables sélectionnées via ce modèle entraîné sur l'ensemble des données de bases sont : season, yr, holiday, weekday, weathersit, temp, atemp, windspeed.

Nous souhaitons maintenant analyser la stabilité de la sélection des variables. Pour cela, nous effectuons un bootstrap sur les données avec 100 itérations. Pour chaque itération, nous regardons quelles variables ont été sélectionnées lorsque l'on choisit comme paramètre de régularisation le `lambda.1se` obtenu par validation croisée sur les nouvelles données. Nous obtenons finalement une matrice nous indiquant, pour chaque itération, quelles variables ont été sélectionnées et ainsi estimer la fréquence de sélection de chaque variable lorsque l'on choisi `lambda.1se` comme paramètre de régularistation à chaque fois.

```r
set.seed(2) # Reproductibilité

# 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_new <- as.matrix(data[ind_al, colnames(data) != "cnt"])
  y_new <- data[ind_al, "cnt"]
  # Nouvelle CV et Fit du modèle
  lambda_1se_i <- cv.glmnet(x = X_new, y = y_new, alpha = 1, family = "poisson")$lambda.1se
  mod_lasso_i <- glmnet(x = X_new, y = y_new, alpha = 1, lambda = lambda_1se_i, family = "poisson")
  # Variables sélectionnées à lambda.1se
  mat_vars[i, which(coef(mod_lasso_i) != 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 = "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"
  )
```

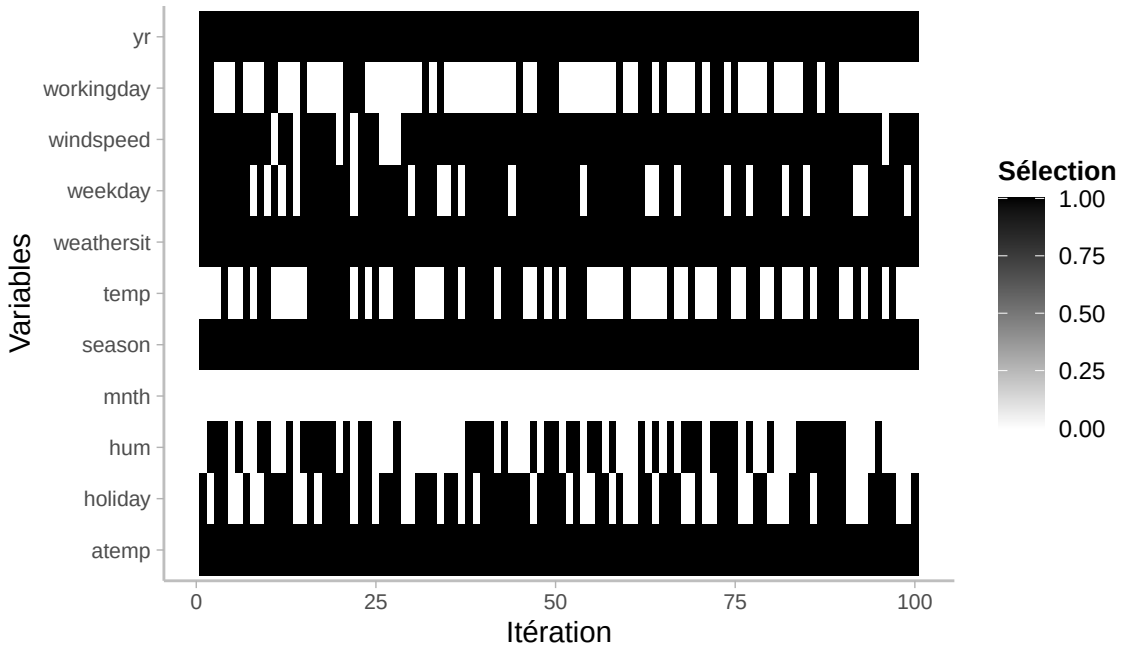

Figure 1: Stabilité à lambda.1se au fil des itérations

```
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	0.62	0.79	0.28	1	0.45	1	0.49	0.92

La figure précédente **(Fig. 1)** présente les variables sélectionnées à chaque itération, avec 1 = "la variable a été sélectionnée" et 0 sinon (la légende de couleur devrait être discrète (0 et 1) mais ggplot n'accepte pas ce format sur une heatmap). A partir de cette figure et de la table **(Tab. 3)**, nous observons différents niveaux de sélection. D'une part, 5 variables semblent stables en termes de sélection : les variables `season`, `yr`, `weathersit` et `atemp` sont systématiquement sélectionnées, et la variable `mnth` ne l'est jamais. D'autre part, les autres variables (`holiday`, `weekday`, `workingday`,`temp`, `hum` et `windspeed`) sont sélectionnées de manière instable (entre 5% et 95% du temps).

La méthode LASSO, en choisissant `lambda.1se` comme paramètre de régularisation, présente donc une instabilité sur la sélection des variables. Cette instabilité est potentiellement due à la présence de multicolinéarité entre certaines variables.

3 *Sélection post-inférence.*

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

Nous souhaitons maintenant construire un intervalle de confiance pour $\hat{\beta}$ l'estimateur du maximum de vraisemblance restreint à $\hat{S}$, l'ensemble des variables sélectionnées par `glmnet` avec pour paramètre de régularisation `lambda.1se`. Pour

cela, nous utilisons une approche naïve basée sur la normalité asymptotique de l'estimateur du maximum de vraisemblance pour obtenir un intervalle de confiance au niveau 1 - α pour chaque coefficient $\hat{\beta}_j$ de $\hat{\beta}$:

$$\hat{IC}_\alpha^j = \left[\hat{\beta}_j - \frac{q_{1-\frac{\alpha}{2}}\hat{\sigma}_j}{\sqrt{n}}; \hat{\beta}_j + \frac{q_{1-\frac{\alpha}{2}}\hat{\sigma}_j}{\sqrt{n}}\right]$$

Les coefficients non nuls sélectionnés par LASSO sur les données originales avec lambda = `lambda.1se` concernent les variables season, yr, holiday, weekday, weathersit, temp, atemp, windspeed. Nous construisons donc le modèle de régression de Poisson restreint à ces variables.

```
# Données avec uniquement les variables sélectionnées
y <- data$cnt
X_bis <- data[, -c(3,6,10,12)]

# Régression de poisson correspondante
model_poiss <- glm(y ~ . , data = X_bis, family = poisson)

# Esimateur de beta
kable(
  t(model_poiss$coefficients),
  caption = "Coefficients de l'estimateur du maximum de vraisemblance $\\hat\\beta$",
  digits = 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 4: Coefficients de l'estimateur du maximum de vraisemblance $\hat{\beta}$

(Intercept)	season	yr	holiday	weekday	weathersit	temp	atemp	windspeed
7.486	0.109	0.473	-0.168	0.015	-0.182	0.156	1.179	-0.438

Nous pouvons alors construire les intervalles de confiance pour tous les $\hat{\beta}_j$ au niveau $\alpha = 0.05$:

```
# Seuil alpha
alpha <- .05
# Coefficients beta et mise en forme
beta_true <- numeric(ncol(X_bis) + 1)
names(beta_true) <- c("(Intercept)", colnames(X_bis))
beta_true[names(coef(model_poiss))] <- coef(model_poiss)

# Écarts-types
se <- summary(model_poiss)$coefficients[, 2]

# Calcul des bornes supérieures et inférieures
n <- dim(data)[1]
borne_inf <- beta_true - qnorm(1 - alpha/2)*se/sqrt(n)
borne_sup <- beta_true + qnorm(1 - alpha/2)*se/sqrt(n)

# Résultat
kable(
  t(data.frame("borne inférieure" = borne_inf, "borne supérieure" = borne_sup)),
  caption = "Intervalles de confiance des variables sélectionnées",
  digits = 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 5: Intervalles de confiance des variables sélectionnées

	(Intercept)	season	yr	holiday	weekday	weathersit	temp	atemp	windspeed
borne.inférieure	7.485	0.109	0.473	-0.168	0.015	-0.182	0.154	1.177	-0.438
borne.supérieure	7.486	0.109	0.473	-0.167	0.015	-0.182	0.158	1.181	-0.437

Nous constatons que les intervalles de confiances sont tous très resserrés autour de la valeur du coefficient $\hat{\beta}_j$ **(Tab. 4, Tab. 5)**.

2. *Évaluation du niveau empirique.*

Nous souhaitons maintenant évaluer le niveau empirique de ces intervalles de confiance, sachant que le niveau de confiance attendu serait de 95%. Pour cela, nous itérons 100 fois le procédé suivant :

- Pour chaque individu i, nous simulons une variable $y_i \sim \mathcal{P}(\lambda_i)$, avec $\lambda_i = \mathbb{E}[Y_i|X_i]$ l'intensité de l'individu i estimé par le premier modèle construit.
- Nous ajustons ensuite un modèle régularisé en utilisant les données simulées, nous récupérons le $\hat{S}$ correspondant aux variables sélectionnées pour ces données simulées et recalculons l'estimateur de maimum de vraisemblance restraint à $\hat{S}$ pour les données simulées.
- Nous construisons de nouveaux intervalles de confiance pour les variables sélectionnées.
- Nous vérifions si la véritable valeur de $\hat{\beta}_j$ (qui a servi à la simulation) se trouve dans ce nouvel intervalle de confiance.

À la fin des 100 itérations, nous obtiendrons donc une approximation du niveau empirique de cette approche.

```r
# Seuil
alpha <- 0.05

# Nombre de répétitions
nrep <- 100

# Compteurs pour la couverture et le nombre d'intervalles
compteur_couverture <- setNames(numeric(length(beta_true)), names(beta_true))
nb_intervalles <- setNames(numeric(length(beta_true)), names(beta_true))

# Intensité estimés pour les données originales
lambda_hat <- predict(model_poiss, type = "response")

for(i in 1:nrep){
  # Nouveau y (nouveau "cnt") simulé à partir de beta
  y_sim <- rpois(n, lambda = lambda_hat)
  # Ajustement d'un LASSO avec lambda = lambda.1se sur l'ensemble des variables prédictives
  cv_i <- cv.glmnet(as.matrix(X), y_sim, family = "poisson")
  lambda_i <- cv_i$lambda.1se
  # Récupération du support correspondant
  coef_i <- coef(cv_i, s = lambda_i)
  index_col_select <- which(as.vector(coef_i[-1, 1]) != 0)
  # Nouveau modèle de Poisson
  X_act_i <- X[, index_col_select, drop = FALSE]
  model_poiss_i <- glm(y_sim ~ . , data = as.data.frame(X_act_i), family = "poisson")
  # Nouveaux intervalles de confiances
  beta_i <- numeric(ncol(X_act_i) + 1)
  names(beta_i) <- c("(Intercept)", colnames(X_act_i))
  beta_i[names(coef(model_poiss_i))] <- coef(model_poiss_i)
  se <- summary(model_poiss_i)$coefficients[, 2]
  borne_inf_i <- beta_i - qnorm(1-alpha/2)*se/sqrt(n)
  borne_sup_i <- beta_i + qnorm(1-alpha/2)*se/sqrt(n)
  # Variables originales sélectionnées ?
  for(name in names(borne_inf_i)){
```

```
    # Oui : on augmente le compteur d'intervalles
    if(name %in% names(beta_true)){
      beta_true_j <- beta_true[name]
      inf <- borne_inf_i[name]
      sup <- borne_sup_i[name]
      nb_intervalles[name] <- nb_intervalles[name] + 1
      # Vrai valeur dans le nouvel intervalle ?
      if (beta_true_j >= inf && beta_true_j <= sup) {
        # Oui : on augmente le compteur de couverture.
        compteur_couverture[name] <- compteur_couverture[name] + 1
      }
    }
  }
}

niveau_empirique <- compteur_couverture / nb_intervalles
```

```
kable(
  t(compteur_couverture),
  caption = "Nombre de fois où le vrai $\\hat\\beta_j$ était dans le nouvel intervalle de confiance",
  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 6: Nombre de fois où le vrai $\hat{\beta}_j$ était dans le nouvel intervalle de confiance

(Intercept)	season	yr	holiday	weekday	weathersit	temp	atemp	windspeed
7	9	6	3	8	6	8	6	5

```
kable(
  t(nb_intervalles), caption = "Nombre de sélections des 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 7: Nombre de sélections des variables

(Intercept)	season	yr	holiday	weekday	weathersit	temp	atemp	windspeed
100	100	100	100	100	100	100	100	100

```
kable(
  t(niveau_empirique), caption = "Niveau empirique moyen 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 8: Niveau empirique moyen de chaque variable

(Intercept)	season	yr	holiday	weekday	weathersit	temp	atemp	windspeed
0.07	0.09	0.06	0.03	0.08	0.06	0.08	0.06	0.05

Nous obtenons un niveau de confiance moyen d'environ 6.44%, contre $100 \cdot (1 - \alpha) = 95\%$ attendu **(Tab. 8)**. Cependant, cette différence peut s'expliquer : l'hypothèse d'un modèle choisi sans voir les données est violée dans ce contexte. En effet, nous effectuons une inférence naïve sur des données déjà utilisées afin d'effectuer la sélection des variables avec le LASSO. Cela entraîne des intervalles de confiance trop optimistes (trop resserrés, comme nous l'avions remarqué précédemment) dans lesquels les "vraies" coefficients ont peu de chance de se trouver.

Ce résultat est un bon exemple des défis spécifiques à l'inférence post-sélection de variables : puisque les données ont déjà été utilisées pour la sélection de variables, il n'est plus possible des les utiliser naïvement pour l'inférence sur les coefficients. Il est donc nécessaire d'opter pour une autre approche qui prend en compte cette problématique, comme par exemple le data-splitting (séparer le jeu de données en 2, avec une partie pour la sélection de variables et l'autre pour l'inférence) ou les méthodes du package `selectiveInference` vu précédemment.