TP5 noté individuel

Garance Malnoë

2025-12-05

Librairies

Préparation des données

Importation du jeu de données

```
day <- read.csv("~/Ecole/M2/S1/Stat grande dimension/Partie 2 - Modèles pénalisés/day.csv")
# A modifier avec votre path
```

Vérifiez les dimensions des données et assurez-vous qu'elles sont cohérentes avec le modèle.

```
View(day)
```

Il est nécessaire d'enlever les variables "instant","dteday" car il s'agit de l'instance et de la date. On supprime également les colonnes "casual" et "registered" car leur somme donne la variable que l'on souhaite prédire "cnt".

```
day <- day[,-c(1,2,14,15)]
```

Les modèles de régression de Poisson sont utilisés lorsque l'on souhaite modéliser des variables de comptage. La variable d'intérêt dans ce jeu de données, "cnt", compte le nombre d'emprunt de vélos réalisés dans une journée, le modèle de régression de Poisson est adapté.

Vérifions les données manquantes

```
sum(is.na(day))
```

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

Il n'y en a pas.

```
dim(day)
```

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

Le jeu de donénes est composé de 731 observations de 11 variables explicatives et de la variable "cnt" à expliquer.

Vérifions la présence de données aberrantes :

```
summary(day)
```

```
##      season            yr              mnth          holiday
##  Min.   :1.000   Min.   :0.0000   Min.   : 1.00   Min.   :0.00000
##  1st Qu.:2.000   1st Qu.:0.0000   1st Qu.: 4.00   1st Qu.:0.00000
##  Median :3.000   Median :1.0000   Median : 7.00   Median :0.00000
##  Mean   :2.497   Mean   :0.5007   Mean   : 6.52   Mean   :0.02873
##  3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:10.00   3rd Qu.:0.00000
```

```
##  Max.   :4.000   Max.   :1.0000   Max.   :12.00   Max.   :1.00000
##     weekday        workingday       weathersit         temp
##  Min.   :0.000   Min.   :0.000   Min.   :1.000   Min.   :0.05913
##  1st Qu.:1.000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.33708
##  Median :3.000   Median :1.000   Median :1.000   Median :0.49833
##  Mean   :2.997   Mean   :0.684   Mean   :1.395   Mean   :0.49538
##  3rd Qu.:5.000   3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:0.65542
##  Max.   :6.000   Max.   :1.000   Max.   :3.000   Max.   :0.86167
##      atemp              hum           windspeed            cnt
##  Min.   :0.07907   Min.   :0.0000   Min.   :0.02239   Min.   :  22
##  1st Qu.:0.33784   1st Qu.:0.5200   1st Qu.:0.13495   1st Qu.:3152
##  Median :0.48673   Median :0.6267   Median :0.18097   Median :4548
##  Mean   :0.47435   Mean   :0.6279   Mean   :0.19049   Mean   :4504
##  3rd Qu.:0.60860   3rd Qu.:0.7302   3rd Qu.:0.23321   3rd Qu.:5956
##  Max.   :0.84090   Max.   :0.9725   Max.   :0.50746   Max.   :8714
```

Il ne semble pas y en avoir, les minimums et maximum font sens.

Nous séparons le jeu de données day en : y, les données à expliquer, et X les variables explicatives.

```r
y <- day[,12]
X <- day[,-12]

X <- as.matrix(X) # Transformation en matrice nécessaire pour glmnet.
```

Estimation et stabilité de l'estimateur

Modèle de regression de Poisson avec régularisation Lasso

```r
# On fait une cross-validation pour choisir le paramètre lambda le + adapté
set.seed(1) # Reproductibilité
lasso_cv <- cv.glmnet(
  X, y,
  alpha = 1, # lasso donc alpha=1
  nfolds=10, # K = 10 folds
  family="poisson" # Pour avoir une modèle de poisson
)
lambda.min <- lasso_cv$lambda.min
lambda.1se <- lasso_cv$lambda.1se

# On récupère le modèle correspondant
lasso.min <- glmnet(X, y, alpha = 1, lambda = lambda.min, family="poisson")
lasso.1se <- glmnet(X, y, alpha = 1, lambda = lambda.1se, family="poisson")
```

```r
lambda.min
```

```
## [1] 20.3787
```

```r
lambda.1se
```

```
## [1] 119.3585
```

Le lambda.min obtenu par la cross-validation est de 20.3787 et le lambda.1se est de 119.3585.

Stabilité de la sélection des variables

```r
set.seed(1) # Reproductibilité
```

```r
# Noms des variables explicatives, pour les vecteurs de résultat
vars <- colnames(day)[colnames(day) != "cnt"]

# Nombre de répétition pour le bootstrap, à prendre suffisamment grand.
n_rep <- 1000

p <- length(vars)
n <- nrow(day)

# Fonction pour récupérer les variables sélectionnées
get_support <- function(model) {
  as.matrix(coef(model))[-1, , drop = FALSE] != 0  # enlève intercept
}

# Initialisation des vecteurs / matrices de résultats
path_lasso   <- array(0, dim = c(p, n_rep), dimnames = list(vars, NULL))

for (r in 1:n_rep) {
  # Ré-échantillonage
  set.seed(100 + r)
  idx <- sample(1:n,size=n,replace=TRUE)
  new_day <- day[idx, ]
  new_y <- new_day[,12]
  new_X <- new_day[,-12]
  new_X <- as.matrix(new_X) # Transformation en matrice nécessaire pour glmnet.

  # Lasso
  ## fit pour chaque lambda
  lasso_fit <- glmnet(new_X, new_y, alpha = 1, family="poisson", lambda = lambda.1se)
  ## variables sélectionnées pour chaque fit
  path_lasso[, r] <- get_support(lasso_fit)
}

# On compte pour chaque variable pour chaque
#le nombre moyen de fois où elle est incluse
stab_lasso <- apply(path_lasso, c(1), mean)
```

```r
stab_lasso
```

```
##     season         yr       mnth    holiday    weekday workingday weathersit
##      1.000      1.000      0.005      0.476      0.679      0.159      1.000
##       temp      atemp        hum  windspeed
##      0.504      1.000      0.296      0.914
```

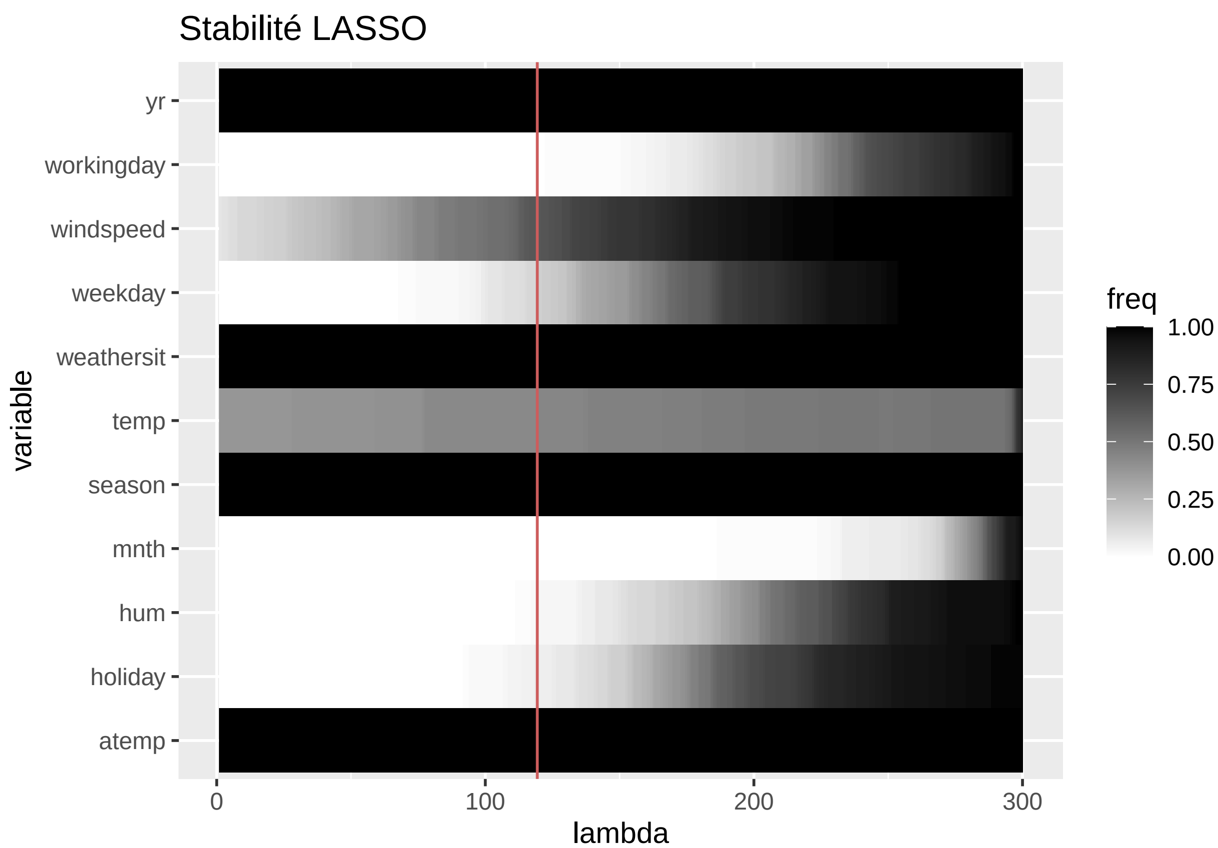
Pour ce lambda = lambda.1se, - certaines variables sont instables (en se fixant un seuil à 5% d'écart à 0 ou 1) : holiday, weekday, workingday, temp, hum et windspeed. Ces variables sont parfois sélectionnées mais pas toujours. - certaines variables sont stables : season, yr, weathersit et atemp. Soit elles sont toujours sélectionnées (valeur proche de 1), soit elles ne sont jamais sélectionnées (valeur proche de 0).

Nous pouvons également regarder ce qu'il se passe pour l'ensemble des lambda :

```r
set.seed(1) # Reproductibilité

vars <- colnames(day)[colnames(day) != "cnt"] # Noms des variables explicatives.
```

```r
n_rep <- 100 # Nombre de répétition pour le bootstrap,
#à prendre suffisamment grand.

# On définit une grille de lambda commune à toutes les répétitions,
# à prendre suffisament fine.
lambda_grid <- seq(1, 300, by=0.25)

p <- length(vars)
L <- length(lambda_grid)
n <- nrow(day)

# Fonction pour récupérer les variables sélectionnées
get_support <- function(model) {
  as.matrix(coef(model))[-1, , drop = FALSE] != 0  # enlève intercept
}

# Initialisation des vecteurs / matrices de résultats
path_lasso   <- array(0, dim = c(p, L, n_rep), dimnames = list(vars, lambda_grid, NULL))
lam_1se_lasso <- matrix(0, p, n_rep, dimnames = list(vars, NULL))

for (r in 1:n_rep) {
  # Ré-échantillonage
  set.seed(100 + r)
  idx <- sample(1:n,size=n,replace=TRUE)
  new_day <- day[idx, ]
  new_y <- new_day[,12]
  new_X <- new_day[,-12]
  new_X <- as.matrix(new_X) # Transformation en matrice nécessaire pour glmnet.

  # Lasso
  ## fit pour chaque lambda
  lasso_fit <- glmnet(new_X, new_y, alpha = 1, family="poisson", lambda = lambda_grid)
  ## variables sélectionnées pour chaque fit
  path_lasso[, , r] <- get_support(lasso_fit)
}

# On compte pour chaque variable pour chaque le nombre moyen de fois où elle est incluse
stab_lasso <- apply(path_lasso, c(1,2), mean)
```

```r
# Transformations pour les visualisation
df_lasso <- as.data.frame(stab_lasso) %>%
  mutate(variable = rownames(.)) %>%
  pivot_longer(-variable, names_to = "lambda", values_to = "freq") %>%
  mutate(lambda = as.numeric(lambda))

# Visualisaiton avec ggplot
ggplot(df_lasso, aes(x = lambda, y = variable, fill = freq)) +
  geom_tile() +
  geom_vline(xintercept = lambda.1se, color="indianred") +
  scale_fill_gradient(low = "white", high = "black") +
  labs(title = "Stabilité LASSO", x = "lambda", y = "variable")
```



La barre rouge correspond au lambda.1se obtenu précédement sur nos données originales (lambda.1se = 119.3585).

Sélection post-inférence

Construction d'un intervalle de confiance

```
coefs.1se <- get_support(lasso.1se)
coefs.1se
```

```
##                s0
## season       TRUE
## yr           TRUE
## mnth        FALSE
## holiday      TRUE
## weekday      TRUE
## workingday FALSE
## weathersit  TRUE
## temp         TRUE
## atemp        TRUE
## hum         FALSE
## windspeed    TRUE
```

Les coefficients sélectionnés par LASSO sur les données originales avec lambda = lambda.1se sont : season, yr, holiday, weekday, weathsit, temp, atemp et windspeed.

Construisons le modèle de régression de poisson avec ces variables là uniquement :

```r
# Jeu de données avec variables sélectionnées
y <- day[,12]
X_bis <- day[,-c(3,6,10,12)] # on enlève mnth, workingday, hum qui n'ont pas été sélectionnées et cnt q

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

```
##
## Call:
## glm(formula = y ~ ., family = poisson, data = X_bis)
##
## Coefficients:
##               Estimate Std. Error  z value Pr(>|z|)
## (Intercept)  7.4857324  0.0037466 1998.008  < 2e-16 ***
## season       0.1087586  0.0005674  191.680  < 2e-16 ***
## yr           0.4732415  0.0011384  415.697  < 2e-16 ***
## holiday     -0.1677349  0.0036357  -46.136  < 2e-16 ***
## weekday      0.0147011  0.0002781   52.868  < 2e-16 ***
## weathersit  -0.1818121  0.0011061 -164.375  < 2e-16 ***
## temp         0.1561047  0.0233818    6.676 2.45e-11 ***
## atemp        1.1786281  0.0266120   44.289  < 2e-16 ***
## windspeed   -0.4378111  0.0078058  -56.088  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 668801  on 730  degrees of freedom
## Residual deviance: 167077  on 722  degrees of freedom
## AIC: 174494
##
## Number of Fisher Scoring iterations: 4
```

```r
# Esimateur de beta
model_poiss$coefficients
```

```
## (Intercept)      season          yr     holiday     weekday  weathersit
##  7.48573239  0.10875864  0.47324148 -0.16773493  0.01470114 -0.18181207
##        temp       atemp   windspeed
##  0.15610472  1.17862813 -0.43781113
```

On peut alors construire des intervalles de confiance pour tous les beta :

```r
confint(model_poiss)
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %      97.5 %
## (Intercept)  7.47838794  7.49307434
## season       0.10764652  0.10987067
## yr           0.47101040  0.47547296
## holiday     -0.17486864 -0.16061698
## weekday      0.01415613  0.01524616
## weathersit  -0.18398037 -0.17964460
## temp         0.11017152  0.20182659
```

```
## atemp        1.12659077  1.23090781
## windspeed   -0.45311214 -0.42251388
```

A verifier si cela correspond bien à la formule demandée (il me semble que oui mais je n'ai plus le temps).