TP Régression logistique bayésienne

1 Mini rappel : régression logistique bayésienne

On observe $(x_i, y_i)_{i=1}^n$ avec $x_i \in \mathbb{R}^p$ et $y_i \in \{0, 1\}$. La régression logistique s'écrit :

$$\mathbb{P}(y_i = 1 \mid x_i, \beta) = p_i, \qquad \text{logit}(p_i) = \log \frac{p_i}{1 - p_i} = x_i^\top \beta,$$

soit

$$p_i = g(x_i^\top \beta), \qquad g(t) = \frac{1}{1 + e^{-t}}.$$

Vraisemblance

Conditionnellement à β, les y_i sont indépendants et

$$p(y \mid X, \beta) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1 - y_i}.$$

Bayésien : prior et posterior

On place un prior sur les coefficients, gaussien (= régularisation de type ridge) :

$$\beta \sim \mathcal{N}(0, \sigma^2 I_p).$$

La loi a posteriori est

$$p(\beta \mid y, X) \propto p(y \mid X, \beta)\, p(\beta),$$

qui n'est pas conjugée en logistique : on l'approxime par MCMC

Remarque On peut considérer σ comme un hyperparamètre et lui attrbuer une loi (khi-deux, inverse gamma).

Prédiction bayésienne

Pour un nouveau point x^*, on s'intéresse à

$$\mathbb{P}(y^* = 1 \mid x^*, y, X) = \int g(x^{*\top} \beta)\, p(\beta \mid y, X)\, d\beta.$$

Avec des tirages MCMC $\beta^{(s)} \sim p(\beta \mid y, X)$, on approxime

$$\widehat{p^*} \approx \frac{1}{S} \sum_{s=1}^{S} g(x^{*\top} \beta^{(s)}).$$

2 Métriques probabilistes et calibration

On note p_i la probabilité prédite pour l'observation i (sur un jeu de test), et $y_i \in \{0, 1\}$ la vraie valeur.

2.1 NLL (Negative Log-Likelihood) / log-loss

La log-loss (moyenne) est :

$$\text{NLL} = -\frac{1}{n}\sum_{i=1}^{n}\Big(y_i \log \hat{p}_i + (1-y_i)\log(1-\hat{p}_i)\Big).$$

Plus petit = meilleur. Elle pénalise fortement les probabilités "extrêmes" erronées.

2.2 Score de Brier

Le Brier score est une erreur quadratique sur les probabilités :

$$\text{Brier} = \frac{1}{n}\sum_{i=1}^{n}(\hat{p}_i - y_i)^2.$$

Plus petit = meilleur. C'est sensible à la calibration.

2.3 ECE (Expected Calibration Error)

On partitionne les prédictions en B *bins* (= classes) $\{\mathcal{B}_b\}_{b=1}^{B}$. Pour chaque bin b :

$$\text{conf}(b) = \frac{1}{|\mathcal{B}_b|}\sum_{i\in\mathcal{B}_b} p_i, \qquad \text{acc}(b) = \frac{1}{|\mathcal{B}_b|}\sum_{i\in\mathcal{B}_b} y_i.$$

L'ECE est alors :

$$\text{ECE} = \sum_{b=1}^{B}\frac{|\mathcal{B}_b|}{n}\,|\text{acc}(b) - \text{conf}(b)|.$$

Plus petit = mieux. L'ECE résume l'écart de calibration moyen pondéré par la taille des bins.

3 Code R (sans la partie bins / ECE)

ce que fait le code ci-dessous :

- charge les données (`PimaIndiansDiabetes2`),
- fait un split train/test,
- ajuste un modèle fréquentiste (`glm`),
- ajuste un modèle bayésien (`stan_glm`),
- calcule les probabilités prédites (moyenne a posteriori),
- calcule NLL, Brier et AUC (ROC).

On pourrait ajouter une variance à σ^2 et le modèle serait complètement bayésien.

```
############################################################
# glm vs stan_glm (sans bins / ECE code)
############################################################

options(repos = c(CRAN = "https://cloud.r-project.org"))
options(mc.cores = parallel::detectCores())

suppressPackageStartupMessages({
  library(mlbench)
  library(dplyr)
  library(ggplot2)
  library(pROC)
  library(rstanarm)
})
```

```r
set.seed(1)

# ------------------------------------------------------------
# 1) Donnees + split train/test
# ------------------------------------------------------------
data(PimaIndiansDiabetes2)
df <- na.omit(PimaIndiansDiabetes2) %>%
  mutate(y = ifelse(diabetes == "pos", 1, 0))

n <- nrow(df)
idx <- sample.int(n, size = floor(0.7 * n))
train <- df[idx, ]
test <- df[-idx, ]
y_test <- test$y

form <- y ~ glucose + mass + age

# ------------------------------------------------------------
# 2) Frequentialiste : glm
# ------------------------------------------------------------
m_glm <- glm(form, data = train, family = binomial(link = "logit"))
p_glm <- predict(m_glm, newdata = test, type = "response")

# ------------------------------------------------------------
# 3) Bayesien : stan_glm (rstanarm)
# Prior ~ Normal(0, 2.5) sur les pentes ; intercept Normal(0,5)
# ------------------------------------------------------------
m_bayes <- stan_glm(
  form,
  data = train,
  family = binomial(link = "logit"),
  prior = normal(0, 2.5),
  prior_intercept = normal(0, 5),
  chains = 2,
  iter = 1200,
  refresh = 0
)

# Probabilites bayesiennes : moyenne a posteriori E[p_i | data]
# posterior_epred renvoie directement des proba (S x n_test)
p_draws <- posterior_epred(m_bayes, newdata = test)
p_bayes <- colMeans(p_draws)

# ------------------------------------------------------------
# 4) Metriques : NLL, Brier, AUC
# ------------------------------------------------------------
eps <- 1e-15

nll <- function(p, y){
  p <- pmin(pmax(p, eps), 1 - eps)
  -mean(y * log(p) + (1 - y) * log(1 - p))
}

brier <- function(p, y) mean((p - y)^2)

NLL_glm <- nll(p_glm, y_test)
NLL_bayes <- nll(p_bayes, y_test)
```

```r
Brier_glm <- brier(p_glm, y_test)
Brier_bayes <- brier(p_bayes, y_test)

roc_glm <- pROC::roc(y_test, p_glm, quiet = TRUE)
roc_bayes <- pROC::roc(y_test, p_bayes, quiet = TRUE)

AUC_glm <- as.numeric(pROC::auc(roc_glm))
AUC_bayes <- as.numeric(pROC::auc(roc_bayes))

cat("\n--- Resultats (test) ---\n")
cat(sprintf("GLM : NLL=%.3f | Brier=%.3f | AUC=%.3f\n", NLL_glm, Brier_glm, AUC_glm))
cat(sprintf("Bayes : NLL=%.3f | Brier=%.3f | AUC=%.3f\n", NLL_bayes, Brier_bayes,
    AUC_bayes))

# ------------------------------------------------------------
# 5) histogramme des proba
# ------------------------------------------------------------
pred_df <- data.frame(
  p = c(p_bayes, p_glm),
  model = rep(c("Bayes (stan_glm)", "Frequentiste (glm)"), each = length(p_glm))
)

fig_hist <- ggplot(pred_df, aes(x = p)) +
  geom_histogram(bins = 25) +
  facet_wrap(~model, ncol = 1) +
  scale_x_continuous(limits = c(0, 1)) +
  labs(
    title = "Distribution des probabilites predites (test)",
    x = "Probabilite predite P(y=1|x)",
    y = "Effectif"
  ) +
  theme_minimal(base_size = 12)

print(fig_hist)
```