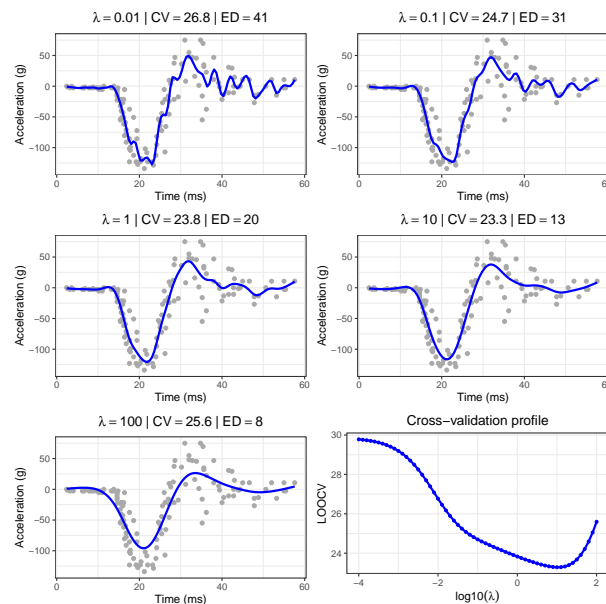


Tuning with cross-validation (Motorcycle data)



P-spline smooth fits to motorcycle acceleration data for various λ (20 segments, $d = 2$). CV is minimized for $\lambda \approx 10$ (bottom, right). R code in `f-cv-plot.R`

```
# Tuning with cross-validation (Motorcycle data)
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

library(ggplot2)
library(gridExtra)
library(JOPS)
library(MASS)

# Get the data
data(mcycle)
x = mcycle$times
y = mcycle$accel

# Basis parameters
nseg = 50
bdeg = 3
pord = 2

# Explore 5 lambdas
lambdas = 10 ^ (-2:2)
F1 = data.frame(x,y)
plts = list()
for (k in 1:length(lambdas)) {
  # Fit P-splines
  fit = psNormal(x, y, nseg = nseg, bdeg = bdeg, pord = pord, lambda = lambdas[k])
  F2 = data.frame(xg = fit$xgrid, yg = fit$ygrid)
  txt = bquote(lambda == .(lambdas[k]) ~'|' ~
    'CV'==.(round(fit$cv, 1)) ~'|' ~ 'ED' == .(round(fit$effdim, 0)))
  # Create plot
  plt = ggplot(F1, aes(x = x, y = y)) +
    geom_point(data = F1, size = 1.5, color = "darkgray") +
    geom_line(aes(x = xg, y = yg), data = F2, size = 1, colour = "blue", lty = 1) +
    xlab("Time (ms)") + ylab("Acceleration (g)") +
    ggtitle(txt) +
```

```
JOPS_theme()

# Add plot to list
plts[[k]] = plt
}

# Calculate CV on log grid for lambda
llambs = seq(from = -4, to = 2, by = .1)
lambin = 10 ^ llambs
cvs = NULL
for (lambda in lambin) {
  fit = psNormal(x,y, nseg = nseg, bdeg = bdeg, pord = pord, lambda = lambda)
  cvs = c(cvs, fit$cv)
}
F3 = data.frame(x = llambs, y = cvs)
plt6 = ggplot(data = F3, aes(x = x, y = y)) +
  geom_line( size = 1, colour = "blue", lty = 1) +
  geom_point( size = 1, colour = "blue") +
  xlab(expression(paste("log10"(lambda)))) + ylab("LOOCV") +
  ggtitle("Cross-validation profile")+
  JOPS_theme()
plts[[6]] = plt6

# Show plots and save pdf
grid.arrange(grobs = plts, ncol = 2, nrow = 3)
```
