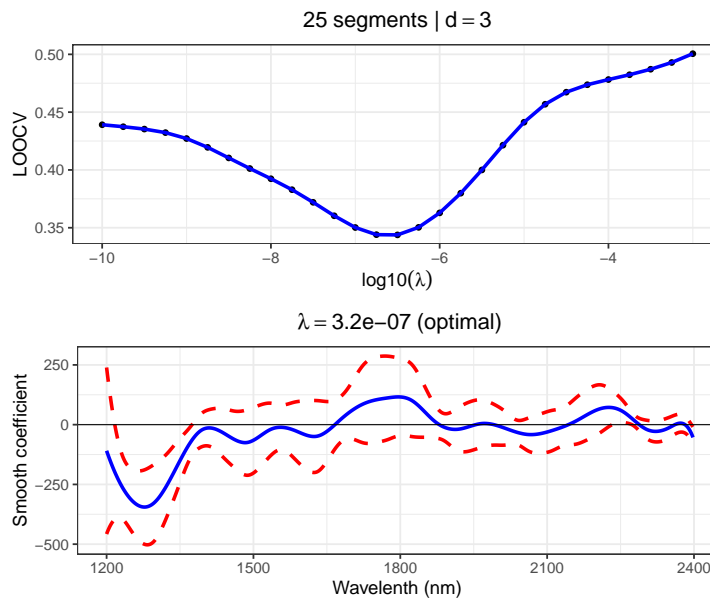


Penalized signal regression optimally tuned with CV (Biscuit data)



Tracking LOOCV as a function of $\log_{10}(\lambda)$ for the biscuit data (top), and the optimal smooth coefficient vector (with twice standard error bands) corresponding to minimum LOOCV (bottom). R code in `f-psropt.R`

```
# Penalized signal regression optimally tuned with CV (Biscuit 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)
```

```
# Get the data
library(fds)
data(nirc)
```

```
iindex = nirc$x
X = nirc$y
sel = 50:650 #1200 <= x & x<= 2400
X = X[sel, ]
iindex = iindex[sel]
dX = diff(X)
diindex = iindex[-1]
y = as.vector(labc[1, 1:40]) #fat
out = 23 # Denoted as outlier by Brown et al. (2001)
dX = t(dX[, -out])
y = y[-out]
```

```
# Design parameters
pord = 3
n = 25
```

```
llamb = seq(from = -10, to = -3, by = 0.25)
lambin = 10^llamb
r = length(llamb)
cv_ = rep(-999, r)
for (i in 1:r) {
  psrloop = psSignal(y, dX, diindex, nseg = n,
```

```

        lambda = lambin[i], pord = pord)
    cv_[i] = psrloop$cv
}

# Optimal signal fit
lamopt = lambin[which(cv_ == min(cv_))]
fit2 = psSignal(y, dX, diindex, nseg = 25, lambda = lamopt, pord = pord)
lower2=fit2$beta-2*fit2$stdev_beta
upper2=fit2$beta+2*fit2$stdev_beta

# Titles and dataframes for ggplot
titl1 = bquote(. (n) ~ "segments |" ~ d == .(pord))
titl2 = bquote(lambda == .(round(lamopt,8)) ~ "(optimal)")
F1=data.frame(x=llamb,y=cv_)
F2=data.frame(beta=fit2$beta,lower=lower2, upper=upper2)

# Make the plots
plt1 = ggplot(F1, aes(x = x, y = y)) +
  geom_point(size = 1.2) +
  geom_line(aes(x = x, y = y), data = F1, size = 1, colour = I("blue"), lty = 1) +
  labs(x=expression(paste("log10"(lambda))), y="L00CV")+
  ggtitle(titl1) +
  JOPS_theme()

plt2 = ggplot() +
  geom_line(aes(x = diindex, y = beta), data = F2, size = 1, colour = I("blue"), lty = 1) +
  geom_line(aes(x = diindex, y = upper), data = F2, size = 1, colour = I("red"), lty = 2) +
  geom_line(aes(x = diindex, y = lower), data = F2, size = 1, colour = I("red"), lty = 2) +
  geom_hline(yintercept = 0, size = 0.3) +
  xlab("Wavelength (nm) ") + ylab("Smooth coefficient") +
  ggtitle(titl2) +
  JOPS_theme()

# Save the plots
grid.arrange(plt1, plt2, ncol = 1, nrow = 2)

```
