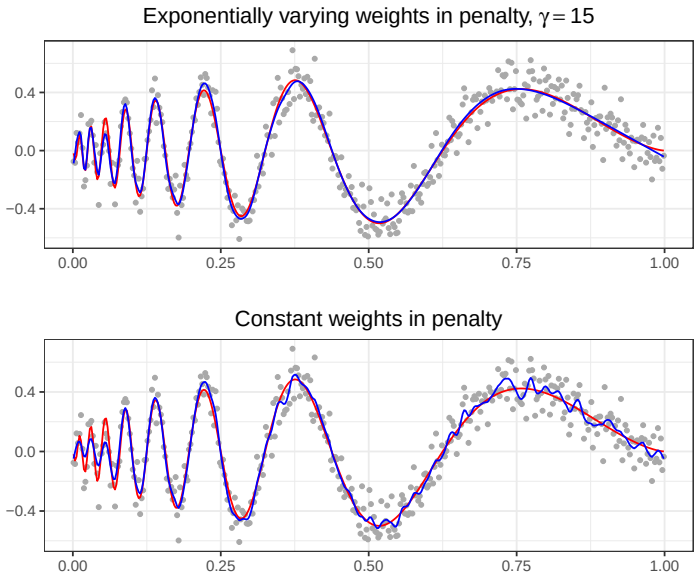
Smoothing of a swept sine, with a variable penalty (Simulated data)



Smoothing with a variable penalty. The data were simulated by adding noise with a normal distribution to a "swept sine" with decreasing frequency and variable amplitude, depicted by the red line in both panels. The identity matrix was used as the basis. The blue line in the lower panel shows the output of standard P-splines. In the upper panel, variable weights were used according to $v_i = \lambda \exp(\gamma i)$. Both γ and the smoothing parameter λ were selected from a grid by cross-validation. R code in `f-swept-sine.R`

```
# Smoothing of a swept sine, with a variable penalty (simulated data)
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

library(JOPS)
library(spam)
library(ggplot2)
library(gridExtra)

# Simulate data
np = 6;
m = 400;
x = ((1:m) - 0.5)/ m;
q = 2 ^ ((9 - 4 * np) / 5);
y0 = sqrt(x * (1 - x)) * sin(2*pi * (1 + q) / (x + q));
set.seed(123)
sigma = 0.1;
y = y0 + rnorm(m) * sigma;

# Prepare penalty stuff
E = diag.spam(m)
D = diff(E, diff = 2)
u = seq(0, 1, length = m - 2)

# Smoothing with varying smoothness
las = seq(-2,2 , by = 0.1)
Z = NULL
cvs2 = NULL
gammas = 0:20
for (gamma in gammas) {
  cvs = NULL
  v = exp(gamma * u);
```

```
  V2 = diag.spam(v);
  P2 = t(D) %*% V2 %*% D;
  # Vary lambda
  for (lambda in 10 ^ las) {
    H = solve(E + lambda * P2);
    z = H %*% y
    r = (y - z) / (1 - diag(H))
    cv = sqrt(mean(r ^ 2))
    cvs = c(cvs, cv)
    # cat(log10(lambda), cv, '\n')
  }

  # Pick minimum CV for current gamma
  k = which.min(cvs)
  lambda = 10 ^ las[k]
  z = solve(E + lambda * P2, y);
  Z = cbind(Z, z)
  cat(gamma, cvs[k], las[k], '\n')
  cvs2 = c(cvs2, cvs[k])
}

# Pick overal minmum of CV
k2 = which.min(cvs2)
gamma = gammas[k2]
z = Z[, k2]

# First column of Z contains fit with gamma = 0
z0 = Z[, 1]

DF = data.frame(x = x, y = y, y0 = y0, z = z, z0 = z0)

plt1 = ggplot(data = DF) +
        geom_point(aes(x = x, y = y), size = 1, color = 'darkgrey') +
        geom_line(aes(x = x, y = y0), color = 'red') +
        geom_line(aes(x = x, y = z), color = 'blue') +
        xlab('') + ylab('') +
        ggtitle(bquote("Exponentially varying weights in penalty," ~ gamma == .(gamma))) +
        JOPS_theme()

plt2 = ggplot(data = DF) +
        geom_point(aes(x = x, y = y), size = 1, color = 'darkgrey') +
        geom_line(aes(x = x, y = y0), color = 'red') +
        geom_line(aes(x = x, y = z0), color = 'blue') +
        xlab('') + ylab('') +
        ggtitle(paste('Constant weights in penalty')) +
        JOPS_theme()

plot(plt1)

plots = grid.arrange(plt1, plt2, nrow = 2, ncol = 1)
plot(plots)
```