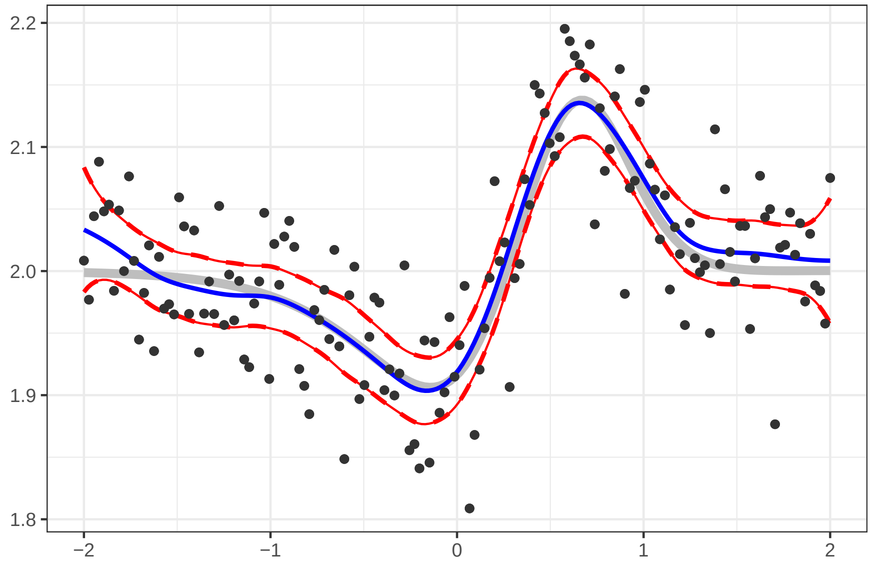
Bayesian P-splines, based on example Jullion and Lambert



Smoothing of simulated data with Bayesian P-splines (20 cubic segments, second order penalty). The true curve is represented by the thick grey line. It consists of sums and differences of logistic functions. The thinner blue line shows the fit and the red broken lines the two standard deviation lines above and below it. The number of Markov steps was 1000. `R` code in `f-jullion.R`

```
# Bayesian P-splines, based on example Jullion and Lambert
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

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

efun = function(x, a, b) 1/(1 + exp(a * (x - b)))

# Simulate data
m = 150
set.seed(23)
xlo = -2
xhi = 2
x = seq(xlo, xhi, length = m)
y0 = efun(x, -4, 0.3) + efun(x, 3, 0.2) + efun(x, -4, 0.7) + efun(x,
    5, 0.8)
y = y0 + rnorm(m) * 0.05

# Bspline parameters
nseg = 20
B = bbase(x, xlo, xhi, nseg, 3)
n = ncol(B)

# Roughness penalty
E = diag(n)
d = 2
D = diff(E, diff = d)
P = t(D) %*% D

# Initialize
ndraw = 1000
```

```
v0 = v1 = rep(0, ndraw)
sig2 = 0.1
tau2 = 1
BB = t(B) %*% B
By = t(B) %*% y
yy = t(y) %*% y
A = matrix(0, n, ndraw)
# Run Markov chain
for (it in 1:ndraw) {

    # Update coefficients
    U = BB/sig2 + P/tau2
    Ch = chol(U)
    a0 = solve(Ch, solve(t(Ch), By))/sig2
    a = solve(Ch, rnorm(n)) + a0
    A[, it] = a

    # Update and save error variance
    r2 = yy - 2 * t(a) %*% By + t(a) %*% BB %*% a
    sig2 = c(r2/rchisq(1, m))
    v0[it] = sig2

    # Update and save roughness variance
    r = D %*% a
    tau2 = c(sum(rˆ2)/rchisq(1, n - d))
    v1[it] = tau2
}

# Compute meand curve on grid
am = apply(A[, -(1:100)], 1, mean)
xg = seq(-2, 2, length = 200)
Bg = bbase(xg, xlo, xhi, nseg, 3)
mu = Bg %*% am

# Local standard deviations
Mu = Bg %*% A
s = apply(Mu, 1, sd)

# Plot data and curve
Data = data.frame(x = x, y = y, y0 = y0)
Dfit = data.frame(x = xg, mu = mu, lo = mu - 2 * s, hi = mu + 2 * s)
plt1 = ggplot(Data, aes(x = x, y = y)) +
  geom_point(aes(x = x, y = y), size = 1.5, color = grey(0.20)) +
  geom_line(data = Data, aes(x = x, y = y0), size = 2, color = ’grey’) +
  geom_line(data = Dfit, aes(x = x, y = mu), size = 1, color = ’blue’) +
  geom_line(data = Dfit, aes(x = x, y = lo), size = 1, color = ’red’,  linetype = 2) +
  geom_line(data = Dfit, aes(x = x, y = hi), size = 1, color = ’red’, linetype = 2) +
  geom_line(data = Dfit, aes(x = x, y = lo), size = 0.5, color = ’red’,  linetype = 1) +
  geom_line(data = Dfit, aes(x = x, y = hi), size = 0.5, color = ’red’, linetype = 1) +
  geom_point(aes(x = x, y = y), size = 1.5, color = grey(0.20)) +
  xlab(’’) + ylab(’’) +
  JOPS_theme()

# Save graph
plot(plt1)
```