Explorations and Experiments on INLA with NHANES data, PartII

BIOS 7720

Randy

April 20, 2021

## Random Walk and Smoothing functions

is the samll and equal space for ;  
 derivative of is continuous;  
 is the order backward difference operator:

## Random Walk and Smoothing functions

## simulation and comparison

library(INLA)  
library(brinla)  
library(tidyverse)  
library(mgcv)  
  
set.seed(555)  
n <- 100  
x <- seq(0, 1, , n)  
f.true <- (sin(2 \* pi \* x^3))^3  
y <- f.true + rnorm(n, sd = 0.2)

## RW1 and RW2

data.inla <- data.frame(y = y, x = x)  
formula1 <- y ~ -1 + f(x, model = "rw1", constr = FALSE)  
system.time(result1 <- inla(formula1, data = data.inla))

## user system elapsed   
## 0.43 0.61 2.67

formula2 <- y ~ -1 + f(x, model = "rw2", constr = FALSE)  
system.time(result2 <- inla(formula2, data = data.inla))

## user system elapsed   
## 0.39 0.66 2.44

# names(inla.models()$latent)

## RW1 and RW2

head(result1$summary.random$x) %>% round(4)

## ID mean sd 0.025quant 0.5quant 0.975quant mode kld  
## 1 0.0000 0.0139 0.1329 -0.2471 0.0138 0.2754 0.0135 0  
## 2 0.0101 0.0605 0.1151 -0.1661 0.0605 0.2867 0.0606 0  
## 3 0.0202 0.0827 0.1109 -0.1353 0.0826 0.3007 0.0825 0  
## 4 0.0303 0.1124 0.1117 -0.1062 0.1120 0.3329 0.1112 0  
## 5 0.0404 -0.0182 0.1127 -0.2419 -0.0175 0.2016 -0.0160 0  
## 6 0.0505 0.0553 0.1090 -0.1590 0.0552 0.2697 0.0551 0

head(result2$summary.random$x) %>% round(4)

## ID mean sd 0.025quant 0.5quant 0.975quant mode kld  
## 1 0.0000 0.0504 0.1373 -0.2202 0.0506 0.3199 0.0509 0  
## 2 0.0101 0.0583 0.1073 -0.1526 0.0582 0.2693 0.0581 0  
## 3 0.0202 0.0635 0.0904 -0.1145 0.0635 0.2412 0.0635 0  
## 4 0.0303 0.0645 0.0832 -0.0991 0.0645 0.2281 0.0645 0  
## 5 0.0404 0.0600 0.0809 -0.0988 0.0599 0.2193 0.0597 0  
## 6 0.0505 0.0563 0.0805 -0.1015 0.0560 0.2152 0.0556 0

## cubic

#### smooth\_spline() ----------  
fit.ss <- smooth.spline(x, y)  
res <- (fit.ss$yin - fit.ss$y) / (1 - fit.ss$lev)  
fhat3 <- fit.ss$y   
## lower bound  
f.lb3 <- fhat3 - 2 \* sd(res) \* sqrt(fit.ss$lev)   
## upper bound  
f.ub3 <- fhat3 + 2 \* sd(res) \* sqrt(fit.ss$lev)

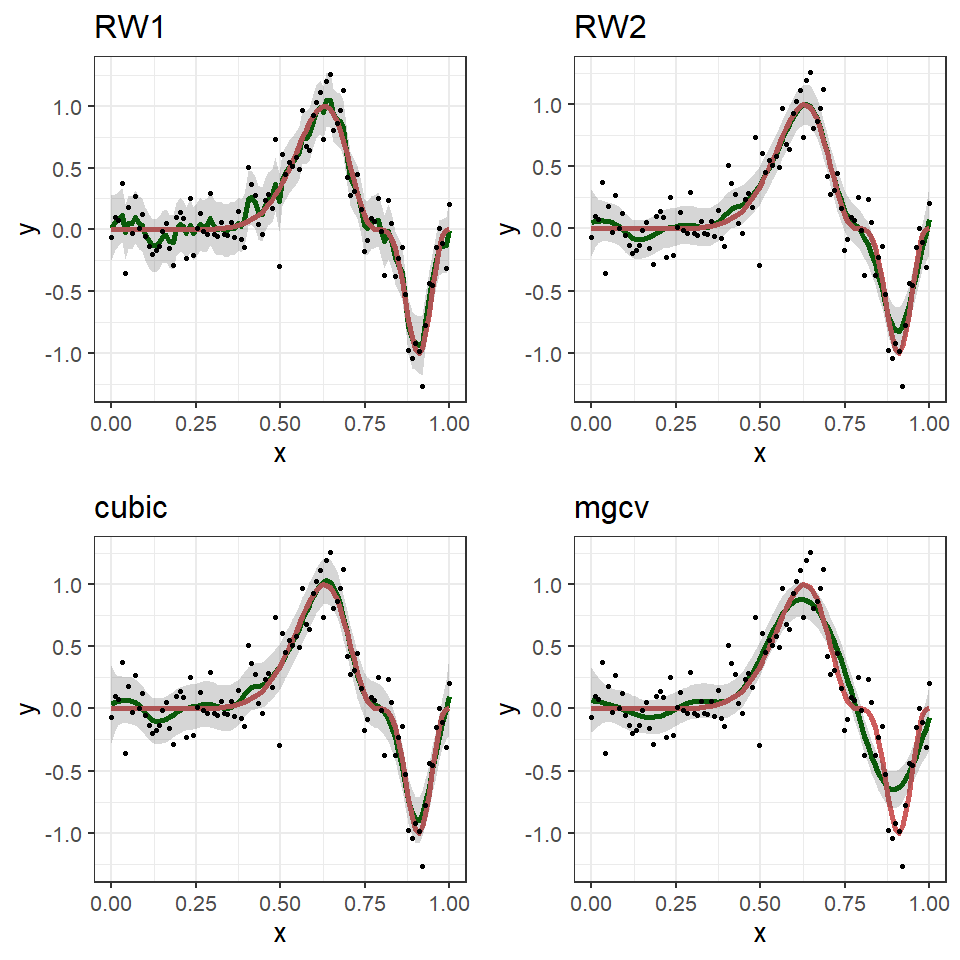
## mgcv gam

#### gam() Sun Apr 04 09:19:51 2021 ---------  
system.time(fit.gam <- gam(y ~ s(x)))

## user system elapsed   
## 0.09 0.00 0.13

res.gam <- predict(fit.gam, se.fit = TRUE)  
## fitted curve  
fhat4 <- res.gam$fit   
## lower bound  
f.lb4 <- res.gam$fit - 2 \* res.gam$se.fit   
## upper bound  
f.ub4 <- res.gam$fit + 2 \* res.gam$se.fit

gridExtra::grid.arrange(plot1, plot2, plot3, plot4, nrow = 2)



## the cherry on the top - prior

* To allow the data to speak for themselves,  
  weakly informative or non-informative priors  
  are often used in this situation.

a1 <- 3  
b1 <- 5e-5  
a2 <- -0.5  
b2 <- 5e-5  
  
lgprior1 <- list(prec = list(param = c(a1, b1)))  
lgprior2 <- list(prec = list(param = c(a2, b2)))  
  
formula5 <- y ~ -1 + f(x, model = "rw2",   
 constr = FALSE,   
 hyper = lgprior2)  
  
system.time(  
result5 <- inla(formula5,   
 data = data.inla,   
 ## used to specify the prior on delta  
 ## and hyper in f() in f() on tao  
 control.family = list(hyper = lgprior1))  
)

## user system elapsed   
## 0.33 1.00 3.16

## the cherry on the top - prior

* the resulting posterior distributions do not differ much  
  from those given by the default priors

## posterior mean  
fhat <- result5$summary.random$x$mean   
## 2.5% percentile  
f.lb <- result5$summary.random$x$"0.025quant"  
## 97.5% percentile  
f.ub <- result5$summary.random$x$"0.975quant"   
  
  
data.plot5 <- data.frame(y = y, x = x,   
 f.true = f.true,   
 fhat = fhat,   
 f.lb = f.lb,   
 f.ub = f.ub)  
  
plot5 <- data.plot5 %>%  
 ggplot(aes(x = x, y = y)) +  
 geom\_line(aes(y = fhat), size = 2, color = "darkgreen") +  
 geom\_line(aes(y = f.true), size = 2, color = "indianred") +  
 geom\_ribbon(aes(ymin = f.lb, ymax = f.ub), alpha = 0.2) +  
 geom\_point(aes(y = y)) +  
 theme\_bw(base\_size = 20) +  
 ggtitle("prior")

gridExtra::grid.arrange(plot2, plot4, plot5, nrow = 1)

