Gaussian Process Demo

David Arthur, Krishnan Raman

November 2, 2020

Let $Y(t) = f(t) + \epsilon(t)$ where $\epsilon(t) \sim N(0, \sigma^2)$

We put a prior over the entire function $f(t)$. This is a Gaussian Process prior.

$$f(t) \sim \mathcal{N}(0, \tau^2 K(t, t'))$$

where $K(t, t')$ is the squared exponential covariance function $\exp\left(-\frac{|t-t'|^2}{2l^2}\right)$ Let $\boldsymbol{t}$ be the vector of time points where we have observed the process and let $\boldsymbol{t'}$ be the vector of time points where we haven't observed the process and for which we wish to make predictions.

Since $Y(\boldsymbol{t})|f(\boldsymbol{t})$, $f(\boldsymbol{t})$ is normal and $f(\boldsymbol{t'})$ is normal, it can be shown that:

$$\begin{bmatrix} Y(\boldsymbol{t}) \\ f(\boldsymbol{t}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \sigma^2 I_n + \Sigma_{\boldsymbol{t}} & \Sigma_{\boldsymbol{t}} \\ \Sigma_{\boldsymbol{t}} & \Sigma_{\boldsymbol{t}} \end{bmatrix} \right)$$

where $\Sigma_f = \tau^2 K(t, t')$ calculated at each pair of time points in $\boldsymbol{t}$. Using properties of Gaussians, one can show that the posterior of $f(\boldsymbol{t})$ is

$$f(\boldsymbol{t})|Y(\boldsymbol{t}) \sim \mathcal{N}(\Sigma_{\boldsymbol{t}}(\sigma^2 I_n + \Sigma_{\boldsymbol{t}})^{-1} Y(\boldsymbol{t}), \Sigma_{\boldsymbol{t}} - \Sigma_{\boldsymbol{t}}(\sigma^2 I_n + \Sigma_{\boldsymbol{t}})^{-1} \Sigma_{\boldsymbol{t}})$$

Similarly the posterior predictive distribution for $f(\boldsymbol{t'})$ is

$$f(\boldsymbol{t'})|Y(\boldsymbol{t}) \sim \mathcal{N}(\Sigma_{\boldsymbol{t'},\boldsymbol{t}}(\sigma^2 I_n + \Sigma_{\boldsymbol{t}})^{-1} Y(\boldsymbol{t}), \Sigma_{\boldsymbol{t}} - \Sigma_{\boldsymbol{t'},\boldsymbol{t}}(\sigma^2 I_n + \Sigma_{\boldsymbol{t}})^{-1} \Sigma_{\boldsymbol{t},\boldsymbol{t'}})$$

Using the following priors:

$\sigma^2 \sim \mathrm{IG}(a_{\sigma^2}, b_{\sigma^2})$

$\tau^2 \sim \mathrm{IG}(a_{\tau^2}, b_{\tau^2)}$

we got the following conditional distributions for σ^2 and τ^2:

$\sigma^2|\cdot \sim \mathrm{IG}(a_{\sigma^2} + n/2, b_{\sigma^2} + 0.5(Y(\boldsymbol{t}) - f(\boldsymbol{t}))'(Y(\boldsymbol{t}) - f(\boldsymbol{t}))$

$\tau^2|\cdot \sim \mathrm{IG}(a_{\tau^2} + n/2, b_{\tau^2} + 0.5(f(\boldsymbol{t}))' K_{\boldsymbol{t}}{}^{-1}(f(\boldsymbol{t}))$

With the prior for l^2 of $l^2 \sim \mathrm{G}(a_{l^2}, b_{l^2})$, then if we propose a new l^2 from a normal distribution centered at the old l^2, we accept this proposal with probability

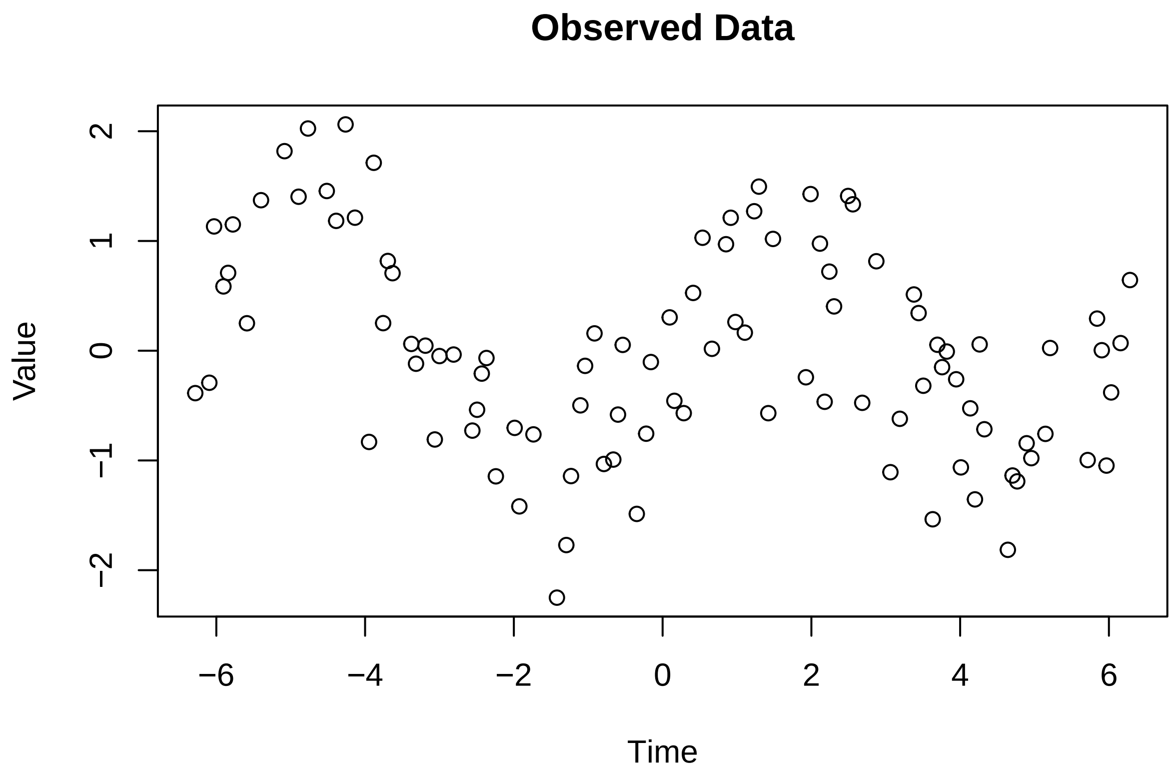
$\min\left\{1, \frac{p(f(\boldsymbol{t})|(l^2)^*)p((l^2)^*)}{p(f(\boldsymbol{t})|l^2)p(l^2)}\right\}$

The code for this Gibbs-Sampling scheme with the Metropolis step for l^2 is below.

```
library(mvtnorm)
library(GauPro)
```

```
## Warning: package 'GauPro' was built under R version 3.5.3
```

```r
# Create time series vector of length 200

n <- 200
tt <- seq(-2*pi, 2*pi, length = n)

# Identity matrix for variance of errors

In <- diag(1, n)

# Identity matrix used for numerical stability when inverting large matrix

In_pert <- diag(0.0001, n)

# True variance and range parameters

sig2 <- 0.4
tau2 <- 0.5
l2 <- 3

# Our observed data will be a subset of 100 observations from the true process

n_sub <- 100
In_sub <- diag(1, n_sub)

ind <- sort(sample(1:n, n_sub))
tt_sub <- tt[ind]
tt_pred <- tt[-ind]

# Observed data is a sine wave function plus normal noise

Y <- sin(tt_sub) + rnorm(n_sub, 0, sqrt(sig2))

plot(tt_sub, Y, main = "Observed Data", xlab = "Time", ylab = "Value")
```




```r
# Number of iterations for MCMC sampling

niter <- 1000

# Save draws from posterior here

# f_save is posterior of function at points where we have observed the function
# fpred_save is predictive posterior for points of function where we haven't observed observations

f_save <- matrix(0, niter, n_sub)
fpred_save <- matrix(0, niter, n-n_sub)
sig2_save <- rep(0, niter)
tau2_save <- rep(0, niter)
l2_save <- rep(0, niter)

# Initial values

sig2_save[1] <- sig2
tau2_save[1] <- tau2
l2_save[1] <- l2

# Values for hyperparameters

sig2_a <- 2
sig2_b <- 2
tau2_a <- 2
```

```r
tau2_b <- 2
l2_a <- 2
l2_b <- 2
l2_sd <- 0.5

# Function to compute the squared exponential covariance function for given time points

kern <- function(tt, l2){
  D <- as.matrix(dist(tt, diag = TRUE, upper = TRUE))^2
  exp(-1/(2*l2)*D)
}

# MCMC Sampler

for (i in 2:niter){
  K <- tau2_save[i-1]*kern(tt, l2) + diag(0.0001, n)
  Kf <- K[ind, ind]
  Kfpred <- K[-ind, ind]
  Kpred <- K[-ind, -ind]

  YSig_inv <- solve(sig2_save[i-1]*In_sub + Kf, In_sub)

  Kf_Sigma <- Kf - Kf%*%YSig_inv%*%t(Kf)
  Kpred_Sigma <- Kpred - Kfpred%*%YSig_inv%*%t(Kfpred)

  Kf_Mu <- Kf%*%YSig_inv%*%Y
  Kpred_Mu <- Kfpred%*%YSig_inv%*%Y

  f_save[i,] <- rmvnorm(1, Kf_Mu, Kf_Sigma)
  fpred_save[i,] <- rmvnorm(1, Kpred_Mu, Kpred_Sigma)

  sig2_save[i] <- 1/rgamma(1, n_sub/2 + sig2_a, sig2_b + 0.5*t(Y-f_save[i,])%*%(Y-f_save[i,]))
  tau2_save[i] <- 1/rgamma(1, n_sub/2 + tau2_a, tau2_b + 0.5*t(f_save[i,])%*%solve(K[ind,ind], f_save[i

  l2_prop <- rnorm(1, l2_save[i-1], l2_sd)
  KSigma_prop <- tau2_save[i]*kern(tt_sub, l2_prop) + diag(0.0001, n_sub)
  KSigma_cur <- tau2_save[i]*kern(tt_sub, l2_save[i-1]) + diag(0.0001, n_sub)

  l2_acc_prob <- dmvnorm(f_save[i,], rep(0, n_sub), KSigma_prop, log = TRUE) +
    dgamma(l2_prop, l2_a, l2_b, log = TRUE) -
    dmvnorm(f_save[i,], rep(0, n_sub), KSigma_cur, log = TRUE) -
    dgamma(l2_save[i-1], l2_a, l2_b, log = TRUE)

  if (log(runif(1)) < l2_acc_prob){
    l2_save[i] <- l2_prop
  } else {
    l2_save[i] <- l2_save[i-1]
  }
}

burn <- 100

f_pred <- colMeans(f_save[burn:niter,])
```

```
fstar_pred <- colMeans(fpred_save[burn:niter,])

fcomb_pred <- rep(0, n)
fcomb_pred[ind] <- f_pred
fcomb_pred[-ind] <- fstar_pred

# Get fit from an R package just to compare our fit

gaupro_fit <- GauPro(tt_sub, Y)
gaupro_preds <- gaupro_fit$predict(tt)

plot(tt_sub, Y, main = "Fitted Function from Gaussian Process",
     xlab = "Time", ylab = "Value")
lines(tt, fcomb_pred, col = 'red')
lines(tt, sin(tt), col = 'blue')
lines(tt, gaupro_preds, col = "green")

legend("topright", c("True", "Our Fit", "GauPro Fit"),
       col = c("blue", "red", "green"), lwd = c(2,2,2))
```