qqq

by

Grace G. Hsu

Project Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in the

Department of Statistics and Actuarial Science

Faculty of Science

SIMON FRASER UNIVERSITY

Summer 2018

Approval

Name:	**Grace G. Hsu**
Degree:	**Master of Science (Statistics)**
Title:	**qqq**
Examining Committee:	**Chair:** qqq Professor
	Professor Senior Supervisor Professor
Date Defended:	**2018 qqq**

Table of Contents

Chapter 1

Report

1.1 Select notation and setup

1.1.1 Covariance functions

Let

- x_i be a row vector. That is, $x_i = \begin{bmatrix} x_{(i,1)} & x_{(i,2)} & \cdots & x_{(i,d)} \end{bmatrix}$
- $d = \dim(x)$
- $d_1, d_2 \in \{0, 1, 2, ..., \dim(x)\}$, where d_i indicates the dimension to which the derivative is taken for the i-th argument of $k(\cdot, \cdot)$
- $g(\cdot)$ is a correlation function.

$\boxed{d_1 = d_2 = 0} \Rightarrow$ `"matern"/"sqexp"`

$$Cov\left(Y\left(x_1\right), Y\left(x_2\right)\right) = k^{00}\left(x_1, x_2\right) \tag{1.1}$$

$$= \sigma^2 \prod_{m=1}^{d} g\left(x_{(1,m)}, x_{(2,m)}; l_m\right) \tag{1.2}$$

$\boxed{d_1 > 0, d_2 = 0 \text{ or } d_1 = 0, d_2 > 0} \Rightarrow$ `"matern1"/"sqexp1"`

$$\frac{\partial}{\partial x_{(1,i)}} Cov\left(Y\left(x_1\right), Y\left(x_2\right)\right) = k^{i0}\left(x_1, x_2\right) \tag{1.3}$$

$$= \sigma^2 \frac{\partial}{\partial x_{(1,i)}} \prod_{m=1}^{d} g\left(x_{(1,m)}, x_{(2,m)}; l_m\right) \tag{1.4}$$

$$= \sigma^2 \left[\prod_{\substack{m=1\\ m\neq i}}^{d} g\left(x_{(1,m)}, x_{(2,m)}; l_m\right)\right]\left[\frac{\partial}{\partial x_{(1,i)}} g\left(x_{(1,i)}, x_{(2,i)}; l_i\right)\right] \tag{1.5}$$

$$\frac{\partial}{\partial x_{(2,i)}} Cov\left(Y\left(x_1\right), Y\left(x_2\right)\right) = k^{0i}\left(x_1, x_2\right) \tag{1.6}$$

$$= -k^{i0}\left(x_1, x_2\right) \tag{1.7}$$

$\boxed{d_1, d_2 > 0} \Rightarrow$ `"matern2"/"sqexp2"`

$$\frac{\partial^2}{\partial x_{(2,j)} \partial x_{(1,i)}} Cov\left(Y\left(x_1\right), Y\left(x_2\right)\right) = k^{ij}\left(x_1, x_2\right) \tag{1.8}$$

$$= \begin{cases} \sigma^2 \left[\prod\limits_{\substack{m=1\\ m\neq i,j}}^{d} g\left(x_m, x'_m; l_m\right)\right]\left[\frac{\partial^2}{\partial x_{(2,i)}\partial x_{(1,i)}} g\left(x_{(1,i)}, x_{(2,i)}; l_i\right)\right], i = j \\ \sigma^2 \left[\prod\limits_{\substack{m=1\\ m\neq i,j}}^{d} g\left(x_m, x'_m; l_m\right)\right]\left[\frac{\partial}{\partial x_{(2,j)}} g\left(x_{(1,j)}, x_{(2,j)}; l_j\right)\right]\left[\frac{\partial}{\partial x_{(1,i)}} g\left(x_{(1,i)}, x_{(2,i)}; l_i\right)\right], i \neq j \end{cases} \tag{1.9}$$

1.1.2 Multiple inputs

$k^{d_1 d_2}(x_1, x_2)$ has been defined so that both arguments are row vectors. That is, $x_i = \begin{bmatrix} x_{(i,1)} & x_{(i,2)} & \cdots & x_{(i,d)} \end{bmatrix}$, and so $d = \dim(x)$.
Now I extend the definition of $k(\cdot,\cdot)$ to the function $K^{d_1 d_2}(\cdot,\cdot)$, which allows for multiple inputs. Let

$$\underbrace{X}_{n_i \times d} = \begin{bmatrix} x_1 \\ \vdots \\ x_{n_i} \end{bmatrix} = \begin{bmatrix} x_{(1,1)} & x_{(1,d)} \\ \vdots & \vdots \\ x_{(n_i,1)} & x_{(n_i,d)} \end{bmatrix} \tag{1.10}$$

Then define

$$\underbrace{K^{d_1 d_2}\left(\overbrace{X}^{n_1 \times d}, \overbrace{X'}^{n_2 \times d}\right)}_{n_1 \times n_2} = K^{d_1 d_2}\left(\begin{bmatrix} x_1 \\ \vdots \\ x_{n_1} \end{bmatrix}, \begin{bmatrix} {x_1}' \\ \vdots \\ {x_{n_2}}' \end{bmatrix}\right) = \left[k^{d_1 d_2}\left(x_i, {x_j}'\right)\right]_{ij}. \tag{1.11}$$

There are a total of $n_1 + n_2$ inputs, resulting in an $n_1 \times n_2$ matrix. Note that the dimension of each of the inputs must all be the same d. Also,

$$K^{d_1 d_2}\left(X, X'\right) = X^{d_1 d_1}\left(X', X\right)^T. \tag{1.12}$$

1.1.3 Matern correlation function and its derivatives

If $g(\cdot,\cdot)$ is from the Matern class of covariance functions with parameter 5/2 there is a more simple form. The subscripts for the inputs in $g(\cdot,\cdot)$ are dropped for legibility.
Let $\theta = \sqrt{5}/l$:

$$g\left(x,x'\right)=\left(1+\theta\left|x-x'\right|+\frac{1}{3}\theta^{2}\left|x-x'\right|^{2}\right)\exp\left\{-\theta\left|x-x'\right|\right\} \tag{1.13}$$

$$\frac{\partial}{\partial x}g\left(x,x'\right)=-\frac{1}{3}\theta^{2}\left(x-x'\right)\left[1+\theta\left|x-x'\right|\right]\exp\left\{-\theta\left|x-x'\right|\right\} \tag{1.14}$$

$$\frac{\partial^{2}}{\partial x'\partial x}g\left(x,x'\right)=\frac{1}{3}\theta^{2}\left[1+\theta\left|x-x'\right|-\theta^{2}\left(x-x'\right)^{2}\right]\exp\left\{-\theta\left|x-x'\right|\right\} \tag{1.15}$$

See calculations of derivatives of $g(\cdot,\cdot)$ below.

First derivative

Recall: $g\left(x,x'\right)=\left[1+\theta\left|x-x'\right|+\frac{1}{3}\theta^{2}\left|x-x'\right|^{2}\right]\exp\left\{-\theta\left|x-x'\right|\right\}$

$$\frac{\partial}{\partial x}g\left(x,x'\right)=\left[\theta sign\left(x-x'\right)+\frac{2}{3}\theta^{2}\left(x-x'\right)\right]\exp\left\{-\theta\left|x-x'\right|\right\} \tag{1.16}$$

$$+\left[1+\theta\left|x-x'\right|+\frac{1}{3}\theta^{2}\left(x-x'\right)^{2}\right]\exp\left\{-\theta\left|x-x'\right|\right\}\left(-\theta sign\left(x-x'\right)\right) \tag{1.17}$$

$$=\exp\left\{-\theta\left|x-x'\right|\right\}\left[-\frac{1}{3}\theta^{2}\left(x-x'\right)-\frac{1}{3}\theta^{3}\left(x-x'\right)^{2}sign\left(x-x'\right)\right] \tag{1.18}$$

$$=\exp\left\{-\theta\left|x-x'\right|\right\}\left(-\frac{1}{3}\theta^{2}\left(x-x'\right)\right)\left[1+\theta\left(x-x'\right)sign\left(x-x'\right)\right] \tag{1.19}$$

$$=-\frac{1}{3}\theta^{2}\left(x-x'\right)\left[1+\theta\left|x-x'\right|\right]\exp\left\{-\theta\left|x-x'\right|\right\} \tag{1.20}$$

Second derivative

$$\frac{\partial^2}{\partial x'\partial x}g\left(x,x'\right) = \frac{\partial}{\partial x'}\left[-\frac{1}{3}\theta^2\left(x-x'\right)\left[1+\theta\left|x-x'\right|\right]\exp\left\{-\theta\left|x-x'\right|\right\}\right] \tag{1.21}$$

$$= \frac{\partial}{\partial x'}\left[\left[-\frac{1}{3}\theta^2\left(x-x'\right)-\frac{1}{3}\theta^3{(x-x')}^2 sign\left(x-x'\right)\right]\exp\left\{-\theta\left|x-x'\right|\right\}\right] \tag{1.22}$$

$$= \left[\frac{1}{3}\theta^2-\frac{2}{3}\theta^3\left(x-x'\right)(-1)\,sign\left(x-x'\right)\right]\exp\left\{-\theta\left|x-x'\right|\right\} \tag{1.23}$$

$$+\left[-\frac{1}{3}\theta^2\left(x-x'\right)-\frac{1}{3}\theta^3{(x-x')}^2 sign\left(x-x'\right)\right]\exp\left\{-\theta\left|x-x'\right|\right\}\left(\theta sign\left(x-x'\right)\right) \tag{1.24}$$

$$= \exp\left\{-\theta\left|x-x'\right|\right\}\left[\frac{1}{3}\theta^2+\frac{2}{3}\theta^3\left|x-x'\right|-\frac{1}{3}\theta^3\left|x-x'\right|-\frac{1}{3}\theta^4{(x-x')}^2\right] \tag{1.25}$$

$$= \exp\left\{-\theta\left|x-x'\right|\right\}\left[\frac{1}{3}\theta^2+\frac{1}{3}\theta^3\left|x-x'\right|-\frac{1}{3}\theta^4{(x-x')}^2\right] \tag{1.26}$$

$$= \frac{1}{3}\theta^2\left[1+\theta\left|x-x'\right|-\theta^2{(x-x')}^2\right]\exp\left\{-\theta\left|x-x'\right|\right\} \tag{1.27}$$

1.1.4 Setup

Assume

$$\left[y^*,y_1^\delta,y_2^\delta,...,{y_d}^\delta,y\,\middle|\,\sigma^2,l\right]\sim N\left(0,\Sigma\right) \tag{1.28}$$

$$\Sigma = \begin{bmatrix} K^{00}\left(X^*,X^*\right) & K^{01}\left(X^*,X_1^\delta\right) & K^{02}\left(X^*,X_2^\delta\right) & \dots & K^{0d}\left(X^*,X_d^\delta\right) & K^{00}\left(X^*,X\right) \\ K^{10}\left(X_1^\delta,X^*\right) & K^{11}\left(X_1^\delta,X_1^\delta\right) & K^{12}\left(X_1^\delta,X_2^\delta\right) & \dots & K^{10}\left(X_1^\delta,X_d^\delta\right) & K^{10}\left(X_1^\delta,X\right) \\ K^{20}\left(X_2^\delta,X^*\right) & K^{21}\left(X_2^\delta,X_1^\delta\right) & K^{22}\left(X_2^\delta,X_2^\delta\right) & \dots & K^{10}\left(X_2^\delta,X_d^\delta\right) & K^{20}\left(X_2^\delta,X\right) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ K^{d0}\left(X_d^\delta,X^*\right) & K^{d1}\left(X_d^\delta,X_1^\delta\right) & K^{d2}\left(X_d^\delta,X_2^\delta\right) & \dots & K^{dd}\left(X_d^\delta,X_d^\delta\right) & K^{d0}\left(X_d^\delta,X\right) \\ K^{00}\left(X,X^*\right) & K^{01}\left(X,X_1^\delta\right) & K^{02}\left(X,X_2^\delta\right) & \dots & K^{0d}\left(X,X_d^\delta\right) & K^{00}\left(X,X\right) \end{bmatrix} \tag{1.29}$$

where

- $X(n \times d), X^*(n^* \times d), X_k^\delta(n_k \times d)$: input sets
- $y = y(X)$: observed target function values (training set)
- $y^* = y(X^*)$: unobserved function values
- $y_k^\delta = \frac{\partial}{\partial x_k} y\left(X^\delta\right) = y(X_k^\delta)$: vectors of partial derivatives in the kth input dimension; $k \in \{1, 2, ..., d\}$
- σ^2: constant variance parameter
- l: length-scale parameter
- $K^{d_1 d_2}(X, X')$: covariance functions. The superscripts are $d_1, d_2 \in \{0, 1, 2, ..., d\}$, where d_i indicates the dimension to which the derivative is taken for the i-th argument of $K(\cdot, \cdot)$ (explicit formulas given earlier).

The posterior density

For 2-dimensional problems (qqq):

$$\begin{aligned}
\left[l, \sigma^2, y^*, y_1^\delta, y_2^\delta \,|y\right] &\propto \left[y, y^*, y_1^\delta, y_2^\delta \,\middle|\sigma^2, l\right] \left[\sigma^2, l\right] && (1.30)\\
&\propto \left[y^*, y_1^\delta, y_2^\delta \,\middle|y, \sigma^2, l\right] \left[\left[y \,\middle|\sigma^2, l\right]\right] \left[\sigma^2, l\right] && (1.31)
\end{aligned}$$

The goal is to evaluate densities on the RHS. Specifically,

$$\left[y^*, y_1^\delta, y_2^\delta \,\middle|y, \sigma^2, l\right] \sim N\left(m, S\right)$$

due to the properties of the multivariate normal distribution (`https://en.wikipedia.org/wiki/Multivariate_normal_distribution#Conditional_distributions`). That is, the formulas from Kriging follow immediately.

To calculate m, S, start by dividing the covariance matrix Σ into the appropriate blocks:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \text{ where} \tag{1.32}$$

$$\Sigma_{11} = \begin{bmatrix} K^{00}\left(x^*, x^*\right) & K^{10}\left(x_1^\delta, x^*\right)^T & K^{10}\left(x_2^\delta, x^*\right)^T \\ K^{10}\left(x_1^\delta, x^*\right) & K^{11}\left(x_1^\delta, x_1^\delta\right) & K^{11}\left(x_2^\delta, x_1^\delta\right)^T \\ K^{10}\left(x_2^\delta, x^*\right) & K^{11}\left(x_2^\delta, x_1^\delta\right) & K^{11}\left(x_2^\delta, x_2^\delta\right) \end{bmatrix} \tag{1.33}$$

$$\Sigma_{12} = \Sigma_{21}^T = \begin{bmatrix} K^{00}(x, x^*)^T \\ K^{01}\left(x, x_1^\delta\right)^T \\ K^{01}\left(x, x_2^\delta\right)^T \end{bmatrix} = \begin{bmatrix} K^{00}\left(x^*, x\right) \\ K^{10}\left(x_1^\delta, x\right) \\ K^{10}\left(x_2^\delta, x\right) \end{bmatrix} \tag{1.34}$$

$$\Sigma_{22} = K^{00}\left(x, x\right). \tag{1.35}$$

Then,

$$m = 0 + \Sigma_{12}\Sigma_{22}^{-1}\left(y - 0\right) = \begin{bmatrix} K^{00}\left(x^*, x\right) \\ K^{10}\left(x_1^\delta, x\right) \\ K^{10}\left(x_2^\delta, x\right) \end{bmatrix} K^{00}(x, x)^{-1} y \tag{1.36}$$

$$S = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{1.37}$$

1.2 Generating N particles

Recall that qqq

$$\pi_t\left(l, \sigma^2, y^*, y^\delta\right) \propto \left[y^*, y^\delta \middle| y, l, \sigma^2\right] \left[y \middle| l, \sigma^2\right] \left[l, \sigma^2\right] \prod_{p=1}^{N} \Phi\left(\tau_t y^{\delta(p)}\right) \tag{1.38}$$

I use Metropolis-Hastings within Gibbs to generate N particles for each parameter: $l, \sigma^2, y^*, y^\delta$.

Algorithm 1 Generating N particles

$\boxed{\mathit{input}:}$

- initial values: $l_0, \sigma_0^2, y_0^*, y_0^\delta$
- training set: y
- input sets: x, x^*, x^δ
- step size for l: $v_t, t = 1, ..., M$

$\boxed{\mathit{loop}:}$

for $i := 1, ..., N$ **do**

  $\left(l^{(i)}, \sigma^{2(i)}, y^{*(i)}, y^{\delta(i)}\right) \leftarrow \left(l^{(i-1)}, \sigma^{2(i-1)}, y^{*(i-1)}, y^{\delta(i-1)}\right)$

$\boxed{l:}$

  sample $l^{new} \sim N\left(l^{(i-1)}, v_i\right)$

  while any entry of $l^{new} < 0.05$ **do**

    sample $l^{new} \sim N\left(l^{(i-1)}, v_i\right)$

  $l^{(i)} \leftarrow l^{new}$ with probability $p = min\{1, HR\}, HR = \frac{\pi_0\left(l^{new}, \sigma^{2(i)}, y^{*(i)}, y^{\delta(i)}\right)}{\pi_0\left(l^{(i)}, \sigma^{2(i)}, y^{*(i)}, y^{\delta(i)}\right)}$ i.e.

  if $\log(HR) > \log(u), u \sim Unif(0,1)$ **then**

    $l^{(i)} \leftarrow l^{new}$

$\boxed{\sigma^2:}$

  sample $\sigma^{2,new} \sim \chi^2(\sigma^{2(i-1)})$

  while $\sigma^{2,new} \leq 1.7$ **do**

    sample $\sigma^{2,new} \sim \chi^2(\sigma^{2(i-1)})$

  $\sigma^{2(i)} \leftarrow \sigma^{2,new}$ with probability $p = min\{1, HR\}, HR = \frac{\pi_0\left(l^{(i)}, \sigma^{2,new}, y^{*(i)}, y^{\delta(i)}\right)}{\pi_0\left(l^{(i)}, \sigma^{2(i)}, y^{*(i)}, y^{\delta(i)}\right)}$

$\boxed{\left(y^*, y^\delta\right):}$

  sample $\left(y^*, y^\delta\right)^{(i)} \sim N\left(\mu_i, \tau_i^2\right)$ where μ_i and τ_i^2 are calculated using $\sigma^{2(i)}, l^{(i)}$:

$$\mu_i = \begin{bmatrix} K^{00}\left(X^*, X\right) \\ K^{\bullet 0}\left(X^\delta, X\right) \end{bmatrix} K^{00}(X, X)^{-1} y \text{ and}$$

$${\tau^2}_i = \begin{bmatrix} K^{00}\left(X^*, X^*\right) & K^{\bullet 0}\left(X^\delta, X^*\right)^T \\ K^{\bullet 0}\left(X^\delta, X^*\right) & K^{\bullet\bullet}\left(X^\delta, X^\delta\right) \end{bmatrix} \text{ (see qqq Setup)}$$

return : N particles $\left(l^{(1)}, \sigma^{2(1)}, y^{*(1)}, y^{\delta(1)}\right), ..., \left(l^{(N)}, \sigma^{2(N)}, y^{*(N)}, y^{\delta(N)}\right)$

1.3 Imposing monotonicity

Recall that qqq

$$\pi_t\left(l, \sigma^2, y^*, y^\delta\right) \propto \left[y^*, y^\delta \middle| y, l, \sigma^2\right] \left[y \middle| l, \sigma^2\right] \left[l, \sigma^2\right] \prod_{p=1}^{N} \Phi\left(\tau_t y^{\delta(p)}\right) \tag{1.39}$$

Algorithm 2 SMC for monotone emulation

$\boxed{\textit{input}:}$

- a sequence of constraint parameters: $\tau_1, ..., \tau_M$ where $\tau_1 = 0$
- step size for l: $v_t, t = 1, ..., M$ qqq
- step size for $\left(y^*, y^\delta\right)$: $q_t, t = 1, ..., M$ qqq

Generate N particles for each parameter via Algorithm 1: $\left(l, \sigma^2, y^{*(1)}, y^\delta\right)^{1:N} = \left(l^{(1)}, \sigma^{2(1)}, y^{*(1)}, y^{\delta(1)}\right), ..., \left(l^{(N)}, \sigma^{2(N)}, y^{*(N)}, y^{\delta(N)}\right)$

$\left(l, \sigma^2, y^*, y^\delta\right)_1^{1:N} \leftarrow \left(l, \sigma^2, y^*, y^\delta\right)^{1:N}$

Give all particles equal weight: $W_1^{1:N} \leftarrow 1/N$

$\boxed{\textit{loop}:}$

for $t := 2, ..., M$ **do**

$\quad\left(l, \sigma^2, y^*, y^\delta\right)_t^{1:N} \leftarrow \left(l, \sigma^2, y^*, y^\delta\right)_{t-1}^{1:N}$

$\boxed{\textit{weights}:}$

$\quad$ Weight calculation: $W_t^i \leftarrow W_{t-1}^i \tilde{w}_t^i$,

$\quad$ where $\tilde{w}_t^i = \dfrac{\prod_{p=1}^N \Phi\left(\tau_t y^{\delta(p)}\right)}{\prod_{p=1}^N \Phi\left(\tau_{t-1} y^{\delta(p)}\right)}$ for all $i = 1, ..., N$ particles

$\quad$ Normalise W_t^i: $W_t^i \leftarrow \frac{W_t^i}{\sum_{i=1}^N W_t^i}$

$\quad$ Use one of the two following re-sampling options:

$\quad\boxed{\textit{1. re-sampling based on ESS}:}$

$\quad$ **if** $ESS_t \leq N/2$, where $ESS_t = 1/\sum_{i=1}^N W_t^i$ **then**

$\quad\quad$ re-sample particles $\left(l, \sigma^2, y^*, y^\delta\right)_t^{1:N}$ with weights $W_t^{1:N}$

$\quad\quad$ $W_t^{1:N} \leftarrow 1/N$

$\quad\boxed{\textit{2. re-sampling always}:}$

$\quad\quad$ re-sample particles $\left(l, \sigma^2, y^*, y^\delta\right)_t^{1:N}$ with weights $W_t^{1:N}$

$\quad$ **if** the first re-sampling option is used **then**

$\quad\quad$ Proceed to algorithm 3 for sampling until finished $t = M$

return : N particles $\left(l, \sigma^2, y^*, y^\delta\right)_M^{1:N}$

Algorithm 3 SMC sampling

continuously adapt step sizes :

$a_{l,t} \leftarrow$ the acceptance rate of the l particles at time t

$a_{\left(y^*,y^\delta\right),t} \leftarrow$ the acceptance rate of the $\left(y^*, y^\delta\right)$ particles at time t

if $t = 2$ **then**

  $q_t \leftarrow \begin{bmatrix} \hat{v}ar\left(\left([l]_1\right)_1^{1:N}\right) & 0 \\ 0 & \hat{v}ar\left(\left([l]_2\right)_1^{1:N}\right) \end{bmatrix}$

  $v_t = 0.1$

if $t > 2$ and $t \leq \left\lceil \frac{M}{3} \right\rceil$ **then**

  if $a_{l,t-1} < 0.25$ or $a_{l,t-1} > 0.4$ **then**

    $v_t \leftarrow v_{t-1} \frac{a_{l,t-1}}{0.25}$

  if $a_{\left(y^*,y^\delta\right),t-1} < 0.25$ or $a_{l,t-1} > 0.4$ **then**

    $q_t \leftarrow q_{t-1} \frac{a_{\left(y^*,y^\delta\right),t-1}}{0.25}$

sampling :

Sample $\left(l, \sigma^2, y^*, y^\delta\right)_t^{1:N}$

for $i := 1, ..., N$ **do**

l :

  sample $l^{new} \sim N\left(l_{t-1}^{(i)}, v_t\right)$

  while any entry of $l^{new} < 0.05$ **do**

    sample $l^{new} \sim N\left(l_{t-1}^{(i)}, v_t\right)$

  $l_t^{(i)} \leftarrow l^{new}$ with probability $p = min\{1, HR\}, HR = \frac{\pi_t\left(l^{new}, \sigma^{2(i)}{}_t, y^{*(i)}{}_t, y^{\delta(i)}{}_t\right)}{\pi_t\left(l^{(i)}, \sigma^{2(i)}{}_t, y^{*(i)}{}_t, y^{\delta(i)}{}_t\right)}$ i.e.

  if $\log(HR) > \log(u), u \sim Unif(0,1)$ **then**

    $l_t^{(i)} \leftarrow l^{new}$

σ^2 :

  sample $\sigma^{2,new} \sim \chi^2(\sigma_{t-1}^{2(i)})$

  while $\sigma^{2,new} \leq 1.7$ **do**

    sample $\sigma^{2,new} \sim \chi^2\left(\sigma_{t-1}^{2(i)}\right)$

  $\sigma_t^{2(i)} \leftarrow \sigma^{2,new}$ with probability $p = min\{1, HR\}, HR = \frac{\pi_t\left(l^{(i)}{}_t, \sigma^{2,new}, y^{*(i)}{}_t, y^{\delta(i)}{}_t\right)}{\pi_t\left(l^{(i)}{}_t, \sigma^{2(i)}, y^{*(i)}{}_t, y^{\delta(i)}{}_t\right)}$

$\left(y^*, y^\delta\right)$:

  sample $\left(y^*, y^\delta\right)^{new} \sim N\left(\left(y^*, y^\delta\right)_{t-1}^{(i)}, q_t C_{t-1}^{(i)}\right)$ where

  $C_{t-1}^{(i)} = K^{0\bullet}(X^*, X^\delta; l_t^i, \sigma^2 = 1)$ is a correlation matrix with length scale l_t^i and q_t is the constant variance and step size

  $\left(y^*, y^\delta\right)^{(i)} \leftarrow \left(y^*, y^\delta\right)^{new}$ with probability

  $p = min\{1, HR\}, HR = \frac{\pi_t\left(l^{(i)}{}_t, \sigma^{2(i)}, y^{*new}, y^{\delta new}\right)}{\pi_t\left(l^{(i)}{}_t, \sigma^{2(i)}, y^{*(i)}{}_t, y^{\delta(i)}{}_t\right)}$

  return : N particles $\left(l, \sigma^2, y^*, y^\delta\right)_t^{1:N}$ and return to algorithm 2 for next iteration of t

1.4 1-dimensional inputs

(EX1) I was able to replicate Shirin's results by using: **Algorithm 2 following "always re-sampling" (lines 16-17) without any sampling (algorithm 3 skipped).**

- Algorithm 2 and always re-sampling (lines 16-17) with sampling (algorithm 3 included) may also be used. However, including the sampling does not seem to change the posterior distributions of y^* much and so this variant of the algorithm is not included.

- Similar performance can be achieved by re-sampling based on ESS with sampling if I increase the number of time steps (up to 40-60), which takes more time.

- There seems to be no clear benefit to re-sampling based on ESS and sampling when the increase in computational time does not seem to improve the performance for this example.

- Testing other functions with steeper curves and those that oscillate more (while remaining monotonic), it seems that this is where re-sampling based on ESS and sampling are needed.

  - Always resampling and skipping algorithm 3 causes all the weights to become focussed on a single particle very quickly no matter how slowly τ increases. That is, I end up with a single particle and the algorithm essentially terminates early.
  - In contrast if I re-sample based on ESS and use algorithm 3, I need to increase the number of time steps, but in return I no longer end up with a single particle.

- Since the inputs are only 1-dimensional, 100 particles with 100 burn-in seems to work fine.

- In summary, if a problem has 1-dimensional inputs

  1. Generate $N = 100$ (+ 100 burn-in) particles using algorithm 1.

2. Try algorithm 2 with $M = 20$ time steps and always re-sample.
3. If all the weight ends up on a single particle or the algorithm terminates early for some other reason, then try combinations of the following options:
   - run algorithm 2 using re-sampling based on ESS and sample with algorithm 3.
   - increase the number of time steps M.
   - increase the number of particles N.

1.5 2-dimensions

(EX2) I was able to obtain posterior distributions similar to Shirin's results by using: **Algorithm 2 following "sampling based on ESS" (lines 12-15) along with sampling (algorithm 3).**

- Differences in results include:
  - The posterior density curves for y^* (or indeed any of the parameters) do not "move" as smoothly as those of Shirin as M increases. Mine sometimes jump a lot from one time step to the next whereas hers move a similar amount each time. Hers are also all very clearly unimodal whereas my density shapes can look somewhat deformed at times.
  - We end up with different posterior means for the length-scale parameters. However, the difference is small enough that I think we can let that pass.
  - The posterior mean for the constant variance parameter σ^2 is very different. It seems like mine is approximately the square of Shirin's.

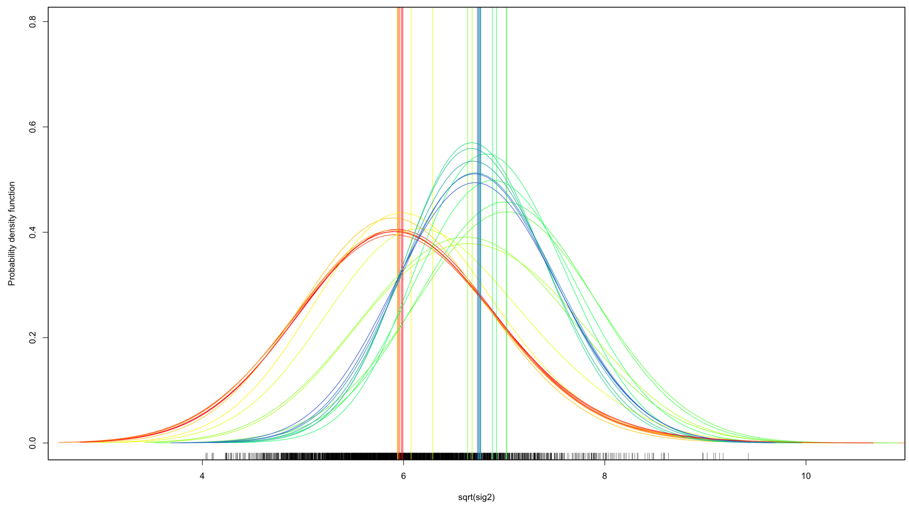

Figure 1.1: Posterior means for σ from time 2 (red) to 20 (blue); $M = 20$, $N = 3000$ (+ 1000 burn-in) qqq sizing

- The posterior means for y^* are rather lacklustre with $M = 20$ and $N = 1000$ (+ 1000 burn-in). I need to either raise N to 3000 or raise M above 40 to achieve better posterior means.
- For some reason, $N = 4000$ and $M = 20$ performs worse in terms of posterior means of y^* than $N = 3000$ and $M = 20$.
- When sampling (algorithm 3) is always included, it does not seem to matter how re-sampling is done. That is, the posterior distributions of y^* do not change much.
- If sampling is not used, then even for large M and N the algorithm terminates early due to all the weights being concentrated on a single particle. I have tested up to $M = 60$ with $N = 5000$ with similar outcomes. Looking at the posterior means of y^* a few time steps before the algorithm terminates, the results are poor, which suggests that sampling must be included for all except 1-dimensional problems.

(EX2) figures for a run with $N = 3000$ (+ 1000 burn-in), $M = 20$:

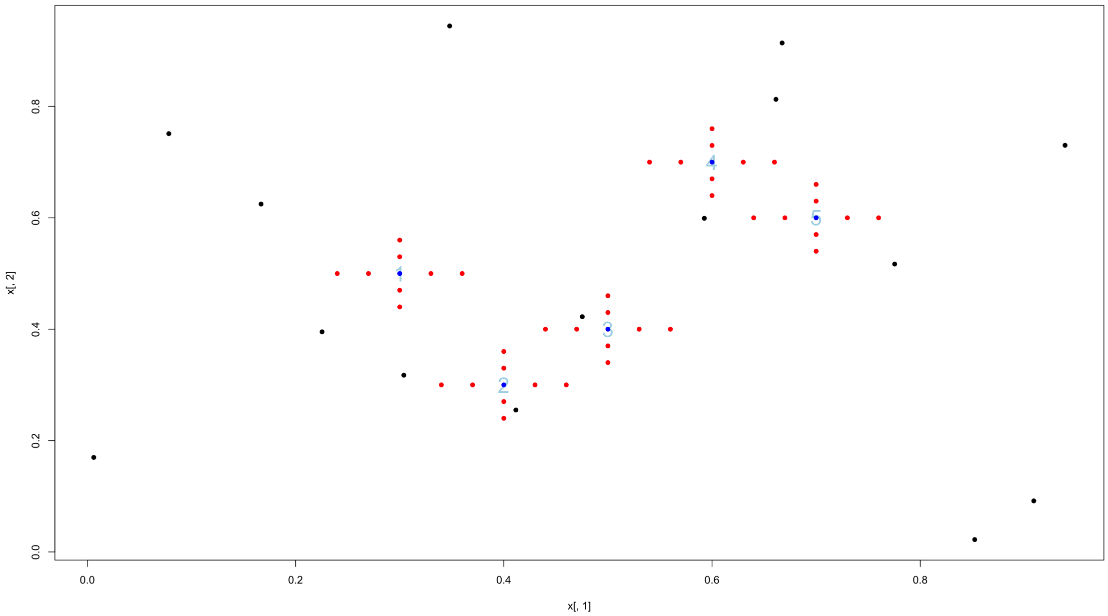

Figure 1.2: Problem setup; black is X, blue is X^*, red is X^δ.

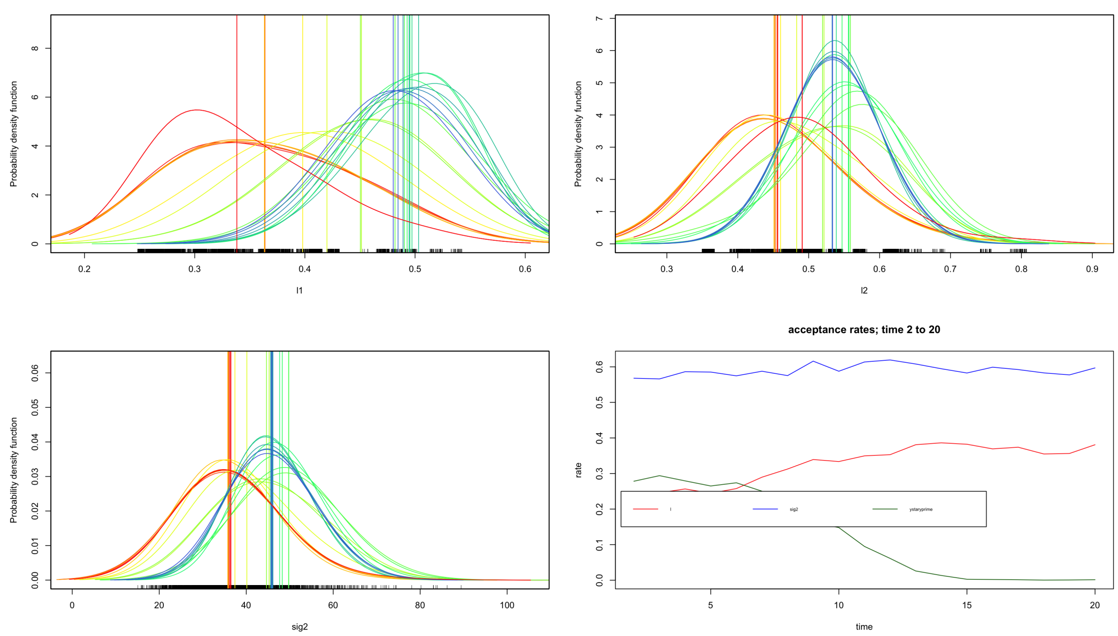

Figure 1.3: Posterior densities and means for l, σ^2 (red to blue) and acceptance rates.

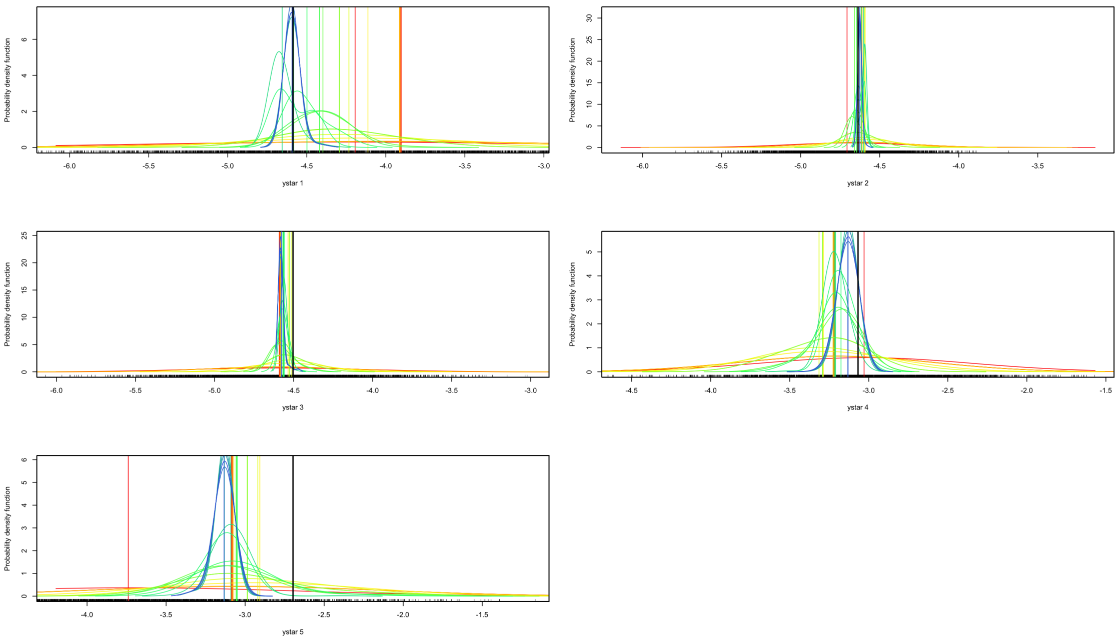

Figure 1.4: Posterior densities and means for y^* (red to blue).

1.5.1 Comparing the covariance functions

Matern

- No nugget required for any examples tried so far (EX1, EX2).

Squared exponential

- A nugget is required for EX2 (I used 10e-6) everywhere a covariance matrix needs to be calculated.

- Seems to require many more time steps to reach a good estimate (posterior mean) for y^*. That is, under the same settings as when I use the Matern, the posterior means for y^* appear much worse.

- I did not test whether a nugget is required for 1-dimensional problems

1.6 Questions

- Why does the second derivative of the Matern function have the same formula regardless of whether the derivative is in the first or second argument?
- In EX2. are the proposal distributions for l, σ^2 still (truncated) normal and chi-squared respectively as in 1-dimension?
- What's the step size for l in higher dimensions? (I'm using a diagonal matrix multiplied by a constant.)
- Is the step size for l being continuously adapted? (I stop 2/3 of the way down the time steps.)
- (EX2) Why are my posterior means for $y*$ moving in such an ugly manner compared to the paper's EX2 results? (Though the final posterior means generally look reasonable.)
- (EX2) Why is my final posterior mean for σ^2 so large (looks squared) compared to that in the paper?
- (EX2) Is EX2 following the paper's algorithm or is it only relying on re-weighting (no sampling)?

Bibliography

Appendix A

Code