Bayesian Scientific Computing
Exercise for Day 4

Daniela Calvetti, Erkki Somersalo

Case Western Reserve University
Department of Mathematics, Applied Mathematics and Statistics

Lyngby, December 2019

Numerical Differentiation

The problem is to estimate a the derivative of a function f in the interval $[0,1]$ from noisy observations

$$b_j = f(t_j) + \varepsilon_j, \quad 1 \le j \le n-1, \quad , t = jh = \frac{j}{n}.$$

As in the lecture notes, write $g(t) = f'(t)$, and

$$b_j = \int_0^{t_j} g(\tau) d\tau + \varepsilon_j \approx \frac{1}{n} \sum_{k=1}^{j} g(t_k),$$

or, in matrix form, denoting $g(t_k) = x_k$, we have

$$b = \mathsf{A}x + \varepsilon, , \quad \mathsf{A} = \frac{1}{n} \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & & \ddots & \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

Numerical Differentiation

We look for the Bayesian solution of the problem using the Whittle-Mátern prior discussed previously. Assuming independent normally distributed noise components, the prior and likelihood are

$$\pi_{B|X} \propto \exp\left(-\frac{1}{2\sigma^2}\|b - \mathsf{A}x\|^2\right),$$

$$\pi_X(x) \propto \exp\left(-\frac{1}{2\gamma^2}\|\mathsf{M}_\lambda x\|^2\right),$$

where

$$\mathsf{M}_\lambda = \left(\mathsf{L}_1 - \frac{1}{\lambda^2}\mathsf{I}\right), \quad \lambda = \text{correlation length}.$$

Numerical Differentiation

Task 1: Generate smooth data. Assume that the true solution $g_{\rm true}$ is a Gaussian curve, given by

$$g_{\rm true}(t) = \frac{12}{\sqrt{\pi}} \exp\left(-(6t-3)^2\right),$$

and so the derivative is

$$f(t) = \int_{-\infty}^{t} g_{\rm true}(s)ds = 1 + {\rm erf}(6t-3),$$

where ${\rm erf}$ is the error function (you find it in Matlab with the name `erf`).

Numerical Differentiation

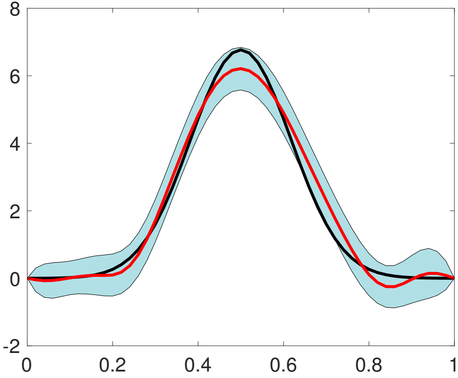
Task 2: *Using the Theorems in Lecture notes for Day 3, implement the mean and covariance of the posterior distribution. Use discretization* $n = 50$*.*

Task 3: *Plot the posterior mean solution, and the posterior belief envelopes of width 1 and 2 standard deviations. More precisely, once you have calculated the posterior covariance matrix* C*, the diagonal entries give the posterior variances of the solutions,*

$$\eta_j^2 = \textit{posterior variance of } X_j = \mathsf{C}_{jj}.$$

The posterior belief envelope of one standard deviation is the "ribbon" between curves passing through points $\overline{x}_j \pm \eta_j$, $1 \leq j \leq n-1$*. Test the results with different noise levels. You should get something like the figure on the next page.*

Numerical Differentiation



Numerical Differentiation

Solution: The prior is defined by using the WM prior,

$$\pi_X(x) \propto \exp\left(-\frac{1}{2\gamma^2}\|\mathsf{M}_\lambda x\|^2\right) = \exp\left(-\frac{1}{2}x^{\mathsf{T}}\left(\frac{1}{\gamma^2}\mathsf{M}_\lambda^{\mathsf{T}}\mathsf{M}_\lambda\right)x\right),$$

showing that the prior precision matrix is

$$\mathsf{D}^{-1} = \frac{1}{\gamma^2}\mathsf{M}_\lambda^{\mathsf{T}}\mathsf{M}_\lambda.$$

The noise covariance matrix is

$$\Sigma = \sigma^2\mathsf{I},$$

and therefore

$$\Sigma^{-1} = \frac{1}{\sigma^2}\mathsf{I}.$$

Numerical Differentiation

Formula for posterior covariance: Use the second version,

$$\mathsf{C} = (\mathsf{A}^\mathsf{T}\Sigma^{-1}\mathsf{A} + \mathsf{D}^{-1})^{-1} = \left(\frac{1}{\sigma^2}\mathsf{A}^\mathsf{T}\mathsf{A} + \frac{1}{\gamma^2}\mathsf{M}_\lambda^\mathsf{T}\mathsf{M}_\lambda\right)^{-1},$$

and for the mean,

$$\overline{x} = (\mathsf{A}^\mathsf{T}\Sigma^{-1}\mathsf{A} + \mathsf{D}^{-1})^{-1}\mathsf{A}^\mathsf{T}\Sigma^{-1}b = \frac{1}{\sigma^2}\mathsf{C}\mathsf{A}^\mathsf{T}b.$$

Numerical Differentiation

In the following, we use the parameter values

$$\lambda = \text{correlation length} = 0.2,$$

$$\gamma = \text{prior scaling} = 500.$$

The prior scaling is lumped together with the WM matrix

```
WM = L1 - 1/lambda^2*speye(n-1);
WM = (1/gamma)*WM;
```

Numerical Differentiation

Generating the noisy data.

```
reln = 2; % relative noise level in pct
t = linspace(0,1,n+1)';
b0    = 1 + erf(-3 + 6*t);
nlev  = reln/100*max(b0);
b     = b0(2:n) + nlev*randn(1,n-1); % Keep the interior points
% For comparison, the true solution
xtrue = 6*2/sqrt(pi)*exp(-(-3 + 6*t).^2);
```

Numerical Differentiation

```
% Building the matrix for the forward problem
A = 1/n*toeplitz(ones(n-1,1),[1,zeros(1,n-2)]);
% Computing the posterior covariance ..
Gamma = inv((1/nlev^2)*A'*A + WM'*WM);
% and the posterior mean ...
xmean = (1/nlev^2)*Gamma*A'*b;
% and padding it with zeros.
xmean = [0;xmean;0];
% Marginal standard deviations of the pointwise values,
% padded with zeros,
d = [0;sqrt(diag(Gamma));0];
```

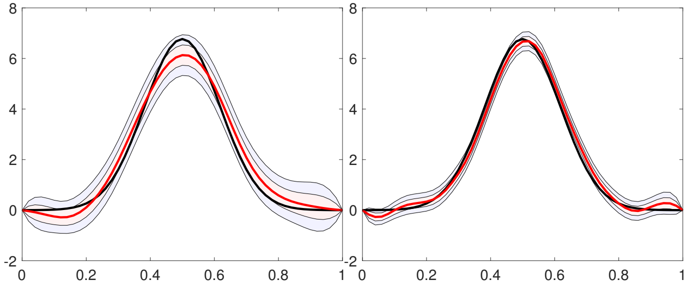
Numerical Differentiation

Plotting the posterior mean and the belief envelopes ($1\times$ STD and $2\times$STD)

```
xlow2 = xmean -2*d;
xhigh2 = xmean+2*d;
xlow1 = xmean - d;
xhigh1 = xmean+ d;

figure(5)
fill([t';t(n+1:-1:1)'],[xlow2;xhigh2(n+1:-1:1)],[0.95,0.95,1])
hold on
fill([t';t(n+1:-1:1)'],[xlow1;xhigh1(n+1:-1:1)],[1,0.95,0.95])
plot(t,xtrue,'k-','LineWidth',3)
plot(t,xmean,'r-','LineWidth',3)
hold off
set(gca,'FontSize',20)
```

Numerical Differentiation



The $1\times$ STD envelope is the red one, and the $2\times$ STD envelope is the blue one. The posterior mean is in red, while the generative true solution is the black one. The noise level was 5% on the left, and 1% on the right..