1 Filtering introduction

We consider the stochastic differential equation model:

$$\begin{aligned} dX_t &= f(X_t;\theta)dt + g(X_t;\theta)dW_t \\ Y_t &= X_t + \epsilon_t \end{aligned}$$

The first equation is a stochastic differential equation with drift function $f(X_t;\theta)$, diffusion function $g(X_t;\theta)$ and Brownian motion W_t. We consider X_t as the latent variable and Y_t as the observed variable, where there's another

observation noise ϵ_t. The observation noise is assumed to be distributed normally with mean 0, variance

σ_ϵ^2.

The aim of filtering in general is to estimate the posterior density of the state variables given the observed variables. Particle filtering tries to estimate sequentially the value of x_k given the observed values $y_0, \cdots, y_k$, $k \leq L$ (the length of the time series). The particle filter provides an approximation to the posterior density $p(x_k|y_0, \cdots, y_k)$, in contrast to MCMC which computes the full posterior $p(x_0, \cdots, x_k|y_0, \cdots, y_k)$. The nonlinear filtering equation follows a recursive updation - prediction step, $p(x_k|y_0, \cdots, y_{k-1}) \rightarrow p(x_k|y_0, \cdots, y_k) \rightarrow p(x_{k+1}|y_0, \cdots, y_k)$. The particle filter is an approximation to the full posterior computation but can be very accurate given enough data points.

Our aim is to infer the posterior distribution $p(\vec{x}, \theta, \sigma_\epsilon^2|\vec{y})$, which is proportional to the likelihood function and the prior

$$p(\vec{x}, \theta, \sigma_\epsilon^2|\vec{y}) \propto p(\vec{y}|\vec{x}, \theta, \sigma_\epsilon^2)p(\vec{x}, \theta, \sigma_\epsilon^2)$$

$\vec{x}$ is the vector of latent system variables and $\vec{y}$ is the vector of observed variables. Since y_t depends on x_t and ϵ_t, it is independent of θ and other latent or observed variables. The likelihood function can be rewritten as,

$$p(\vec{y}|\vec{x}, \theta, \sigma_\epsilon^2) = \prod_{j=0}^{L} p(y_{t_j}|x_j, \sigma_\epsilon^2) = \prod_{j=0}^{L} \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} e^{-\frac{(y_{t_j} - x_j)^2}{2\sigma_\epsilon^2}}$$

The prior on the other hand can be reduced to multiplicative factors of the priors of the parameters,

$$p(\vec{x}, \theta, \sigma_\epsilon^2) = p(\vec{x}|\theta)p(\theta)p(\sigma_\epsilon^2)$$

Till now, the first term of the prior, $p(\vec{x}|\theta)$ used to be the likelihood function we would calculate using the DTQ method, by applying quadrature. The final posterior can now be written as,

$$p(\vec{x}, \theta, \sigma_\epsilon^2|\vec{y}) \propto \left[\prod_{j=0}^{L} p(y_{t_j}|x_j, \sigma_\epsilon^2)\right] p(\vec{x}|\theta)p(\theta)p(\sigma_\epsilon^2)$$

Since we need the gradients of the posterior for optimization, it's easier to take the logarithm on both sides and take derivatives of the summation terms

$$\log p(\vec{x}, \theta, \sigma_\epsilon^2 | \vec{y}) \propto \log \left[\prod_{j=0}^{L} p(y_{t_j} | x_j, \sigma_\epsilon^2) \right] + \log p(\vec{x}|\theta) + \log p(\theta) + \log p(\sigma_\epsilon^2)$$

$$\propto \underbrace{\left[\sum_{j=0}^{L} \log p(y_{t_j} | x_j, \sigma_\epsilon^2) \right]}_{①} + \underbrace{\left[\sum_{j=0}^{L} \log p(x_{j+1} | x_j, \theta) \right]}_{②\text{: computed by DTQ}} + \underbrace{\log p(x_0)}_{③} + \underbrace{\log p(\theta)}_{④} + \underbrace{\log p(\sigma_\epsilon^2)}_{⑤}$$

The time series is of length L, where an current data point is i. So while in the process of optimizing the objective function, we go one - by - one from $0 \leq i \leq L$. The current data point is t_i with the observed data Y_{t_i}, the estimated data x_i and the estimation to be made as x_{i+1}. So we are in the interval $[t_i, t_{i+1}]$ where the parameters estimated at t_i in the previous iterations are the initial conditions for this iteration.

The terms bracketed above are as follows:

1. The log likelihood function is a gaussian with mean x_j and variance σ_ϵ^2 because of the noise relation, $Y_t = X_t + \epsilon_t$. This term is independent of previous observations, previous estimated timeseries values and the parameters of the SDE. In this computation, y_{t_j}, x_j are known and the previously estimated values of $\theta, \sigma_\epsilon^2$ are considered the initial guesses for this iteration of the optimizer.

$$①= \sum_{j=0}^{i} \log p(y_{t_j} | x_j, \sigma_\epsilon^2) = \sum_{j=0}^{i} \log \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} + \sum_{j=0}^{i} \left(- \frac{(y_{t_j} - x_j)^2}{2\sigma_\epsilon^2} \right), \forall i \mid 0 \leq i \leq L$$

$$\frac{\partial ①}{\partial x_i} = \frac{(y_{t_i} - x_i)}{\sigma_\epsilon^2}, \frac{\partial ①}{\partial \sigma_\epsilon^2} = \sum_{j=0}^{j'} \left[\frac{1}{\sigma_\epsilon^2} \log \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} - \frac{(y_{t_j} - x_j)^2}{2\sigma_\epsilon^4} \right], \forall j' \mid 0 \leq j' \leq L$$

2. The second term is the one computed by the DTQ method. This is the bottleneck computation because it can not be written down explicitly in the general case. In the DTQ method, we use the Euler - Maruyama approximation to the SDE and apply a quadrature over the interval so that it can do the computation more accurately even in the case of long non - equispaced intervals. In the DTQ method, if we have n intermediate spatial grid points in the interval $[t_{j'}, t_{j'+1}]$, the following steps are involved

   (a) 1^{st} step is exact and involves going from the delta distribution at $x_{j'}$ to the normal distribution centered around $x_{j'}$. This is represented as the column vector $\vec{g}(x_{j'})$.

   (b) Starting from the initial grid along $x_{j'}$ we need to propagate the computations along the spatial grids. There are $n-2$ repeated matrix multiplications for forward progression in time represented as A^{n-2}. The matrix A remains the same irrespective of the interval.

   (c) To avoid interpolation on the grid that we created, the last step involves multiplication by the vector which represents the grid along the final spatial point, $\vec{\Gamma}(x_{j'+1})$

For computing the likelihood $p(\vec{x} \mid \theta)$, we use the Markov property satisfied by the Euler-Maruyama approximation of the SDE x

$$
\begin{aligned}
p(x_{j'}, t_{n+1}) &= \int_y G(x_{j'}, y) p(y, t_n) dy \\
\vec{\Gamma}(x_2) &= (G(x_2, -Mk), G(x_2, (-M+1)k), \cdots, G(x_2, Mk)) \\
\frac{\partial \vec{\Gamma}}{\partial x_2} &= \left(\frac{\partial G}{\partial x_2}(x_2, -Mk), \frac{\partial G}{\partial x_2}(x_2, (-M+1)k), \cdots, \frac{\partial G}{\partial x_2}(x_2, Mk) \right)
\end{aligned}
$$

The grid is $(-Mk, (-M+1)k, \cdots, (M-1)k, Mk)$ In the above explanation, $G(a, b)$ is described as,

$$
G(a, b) = \frac{1}{\sqrt{2\pi g^2(b;\theta)h}} \exp\left(- \frac{(a - b - f(b;\theta)h)^2}{2g^2(b;\theta)h} \right)
$$

So considering one of the grid points, $ik, -M \leq i \leq M$, we get,

$$
\begin{aligned}
G(x_2, ik) &= \frac{1}{\sqrt{2\pi g^2(ik;\theta)h}} \exp\left(- \frac{(x_2 - b - f(ik;\theta)h)^2}{2g^2(ik;\theta)h} \right), \forall i \\
\frac{\partial G}{\partial x_2}(x_2, ik) &= - \frac{(x_2 - ik - f(ik;\theta)h)}{g^2(ik;\theta)h} G(x_2, ik), \forall i \\
\implies \frac{\partial (2)}{\partial x_i} &= \sum_{j=1}^{L} \left(\cdots, - \frac{(x_j - ik - f(ik;\theta)h)}{g^2(ik;\theta)h}, \cdots \right) +
\end{aligned}
$$

Let's simplify each of the terms before we take the derivatives

$$
\begin{aligned}
(2\text{a}) &= p(x_{j+1} | x_j, \theta) = \vec{\Gamma}(x_{j+1}).A^{n-2}.\vec{g}(x_j) \\
\frac{\partial (2\text{a})}{\partial x_i} &= \frac{\partial \vec{\Gamma}}{\partial x_i}(x_i).A^{n-2}.\vec{g}(x_{i-1}) \\
\frac{\partial (2)}{\partial x_i} &= \sum_{j=1}^{L} \frac{\frac{\partial \vec{\Gamma}}{\partial x_i}(x_i).A^{n-2}.\vec{g}(x_{i-1})}{\vec{\Gamma}(x_i).A^{n-2}.\vec{g}(x_{i-1})} \\
&= \sum_{j=1}^{L} \frac{1}{\vec{\Gamma}(x_i)} \frac{\partial \vec{\Gamma}}{\partial x_i}(x_i) \\
\frac{\partial (2)}{\partial \theta} &= \text{computed in DTQ code}
\end{aligned}
$$

We need to find the derivatives of the 4 circled terms above, with respect to 3 parameters, $\{x_i, \{\theta_1, \theta_2, \theta_3\}, \sigma_\epsilon^2\}$

Table 1: Derivatives

$\frac{\partial}{\partial}$	(1)	(2)	(3)	(4)	(5)
x_i	✓	✓		0	0
$\{\theta_1, \theta_2, \theta_3\}$	0	✓	0		0
σ_ϵ^2	✓	0	0	0	

The priors used for the parameters $\{x, \{\theta_1, \theta_2, \theta_3\}, \sigma_\epsilon^2\}$ are as follows:

$$\begin{aligned}\log p(x_0) &= \log \mathcal{N}(x = x_0, \mu = y_0, \sigma^2 = \sigma_\epsilon^2)\\ \log p(\theta) &= \log \mathcal{N}(x = \theta_1, \mu = 0.5, \sigma^2 = 1) + \log \mathcal{N}(x = \theta_2, \mu = 2, \sigma^2 = 10)\\ \log p(\sigma_\epsilon^2) &= \log(\text{Exp}(\lambda = 1)) = \log(\lambda) - \lambda\sigma_\epsilon^2\end{aligned}$$

For a general log Normal pdf,

$$\begin{aligned}f(x, \mu, \sigma^2) &= \log\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\right] = -\log(\sqrt{2\pi\sigma^2}) - \frac{(x-\mu)^2}{2\sigma^2}\\ \frac{\partial f(x, \mu, \sigma^2)}{\partial x} &= -\frac{(x-\mu)}{\sigma^2}\end{aligned}$$

So the derivatives of the priors with respect to the parameters are,

$$\begin{aligned}\frac{\partial \log p(x_0)}{\partial x_0} &= -\frac{(x_0 - y_0)}{2\sigma_\epsilon^2}\\ \frac{\partial \log p(\theta)}{\partial \theta_1} &= -(\theta_1 - 0.5), \frac{\partial \log p(\theta)}{\partial \theta_2} = -\frac{(\theta_2 - 2)}{100}, \frac{\partial \log p(\theta)}{\partial \theta_3} = 0\\ \frac{\partial \log p(\sigma_\epsilon^2)}{\partial \sigma_\epsilon^2} &= -\lambda\end{aligned}$$

2 Implementation

In this paper we try to solve the inference problem for a time series. The time series is generated by a stochastic differential equation with drift function $f(X;\theta)$ and diffusion function $g(X;\theta)$. The observations obtained from this time series are noisy. Our goal is to not only infer the values of θ and the distribution of noise, but we also want to find the estimated time series. We consider the parameters of the SDE (θ), the parameters of the noise (σ_ϵ^2) and the estimated time series, as the parameters of the model. We want to get a MAP estimate of the log posterior of the parameters given the observed time series, $p(\vec{X}, \theta, \sigma_\epsilon^2)$ for a given time series $\{\vec{t}, \vec{Y}\}$ where $\vec{Y}$ are the observations on the times given by $\vec{t}$.
One methodology to approach the problem (as investigated in the BigMine 16 paper) is to consider the time series as a parameter and infer the distribution of the estimated states using Metropolis algorithm. In this paper we try to find an approximate value by solving for the optimal θ and σ_ϵ^2, one data point at a time. The iterative filtering approach is an approximation of the full posterior computation but given enough data points it gets more accurate. The advantage of using this approach is that it's an online algorithm. Since the computations are done one data point at a time, it doesn't require the full timeseries and it also doesn't require any recomputation when new data is added. In the full posterior computation, each additional data point requires the whole computation to be repeated.

2.1 Statistical Model

We consider the stochastic differential equation model:

$$\begin{aligned}dX_t &= f(X_t;\theta)dt + g(X_t;\theta)dW_t\\ Y_t &= X_t + \epsilon_t\end{aligned}$$

The first equation is a stochastic differential equation with drift function $f(X_t;\theta)$, diffusion function $g(X_t;\theta)$ and Brownian motion W_t. We consider X_t as the latent variable and Y_t as the observed variable, where there's another observation noise ϵ_t. The observation noise is assumed to be distributed normally with mean 0, variance σ_ϵ^2.

Considering the length of the timeseries as L, the data is given by $\{y_0, y_1, \cdots, y_L\}$ where $y_j = Y_{t_j}$ is the observation on time t_j.

3 Gradient Calculation for the Filtering Model

Since we need the gradients of the posterior for optimization, it's easier to take the logarithm on both sides and take derivatives of the summation terms

The time series is of length M, where an current data point is m. So while in the process of optimizing the objective function, we go one - by - one from $0 \leq m \leq M$. The current data point is t_m with the observed data y_{t_m}, the estimated data x_m and the estimation to be made as x_{m+1}. So we are in the interval $[t_m, t_{m+1}]$ where the parameters estimated at t_m in the previous iterations are the initial conditions for this iteration.

$$
\begin{aligned}
\mathbf{B1} &= p(x_{j+1}|x_j,\theta) = \vec{\Gamma}(x_{j+1}).A^{n-2}.\vec{g}(x_j) \\
\frac{\partial \mathbf{B1}}{\partial x_i} &= \frac{\partial \vec{\Gamma}}{\partial x_i}(x_i).A^{n-2}.\vec{g}(x_{i-1}) \\
\frac{\partial \mathbf{B}}{\partial x_i} &= \sum_{j=1}^{L} \frac{\frac{\partial \vec{\Gamma}}{\partial x_i}(x_i).A^{n-2}.\vec{g}(x_{i-1})}{\vec{\Gamma}(x_i).A^{n-2}.\vec{g}(x_{i-1})} \\
&= \sum_{j=1}^{L} \frac{1}{\vec{\Gamma}(x_i)} \frac{\partial \vec{\Gamma}}{\partial x_i}(x_i) \\
\frac{\partial \mathbf{B}}{\partial \theta} &= \text{computed in DTQ code}
\end{aligned}
$$

We need to find the derivatives of the 4 circled terms above, with respect to 3 parameters, $\{x_i, \{\theta_1, \theta_2, \theta_3\}, \sigma_\epsilon^2\}$

Table 2: Derivatives

$\frac{\partial}{\partial}$	**A**	**B**	**C**	**D**	**E**
x_i	✓	✓		0	0
$\{\theta_1, \theta_2, \theta_3\}$	0	✓	0		0
σ_ϵ^2	✓	0	0	0	

The priors used for the parameters $\{x, \{\theta_1, \theta_2, \theta_3\}, \sigma_\epsilon^2\}$ are as follows:

$$
\begin{aligned}
\log p(x_0) &= \log \mathcal{N}(x = x_0, \mu = y_0, \sigma^2 = \sigma_\epsilon^2) \\
\log p(\theta) &= \log \mathcal{N}(x = \theta_1, \mu = 0.5, \sigma^2 = 1) + \log \mathcal{N}(x = \theta_2, \mu = 2, \sigma^2 = 10) \\
\log p(\sigma_\epsilon^2) &= \log(\text{Exp}(\lambda = 1)) = \log(\lambda) - \lambda\sigma_\epsilon^2
\end{aligned}
$$

For a general log Normal pdf,

$$f(x,\mu,\sigma^2) = \log\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\right] = -\log(\sqrt{2\pi\sigma^2}) - \frac{(x-\mu)^2}{2\sigma^2}$$

$$\frac{\partial f(x,\mu,\sigma^2)}{\partial x} = -\frac{(x-\mu)}{\sigma^2}$$

So the derivatives of the priors with respect to the parameters are,

$$\frac{\partial \log p(x_0)}{\partial x_0} = -\frac{(x_0 - y_0)}{2\sigma_\epsilon^2}$$

$$\frac{\partial \log p(\theta)}{\partial \theta_1} = -(\theta_1 - 0.5), \frac{\partial \log p(\theta)}{\partial \theta_2} = -\frac{(\theta_2 - 2)}{100}, \frac{\partial \log p(\theta)}{\partial \theta_3} = 0$$

$$\frac{\partial \log p(\sigma_\epsilon^2)}{\partial \sigma_\epsilon^2} = -\lambda$$