Documentation for the RMHMC Source

Andrei Kramer

November 29, 2016

This text provides illuminates the mathematical considerations and definitions for the implementation of the Riemannian Manifold Hamiltonian Monte Carlo (RMHMC) algorithm in the software package.

1 Model Specifications

We consider a deterministic ODE model and a stochastic measurement model. System states are captured by the state variables $x(t) \in \mathbb{R}^n$. The model parameters $\theta \in \mathbb{R}^m$ describe interactions between the model state variables and are unknown. A second set of parameters $u \in \mathbb{R}^l$ describes the conditions of an experimental setup. These parameters are considered inputs and we assume that they can be set by the experimenter. They can be external parameters, e.g. the temperature, or describe modifications to the system, e.g. inhibitions to some of the interactions. The known time derivative of $x(t)$ defines the model:

$$\dot{x} = f(t, x; \theta, u)\,, \tag{1}$$

$$y(t_j; \theta, u) = Cx(t_j; \theta, u) + \varkappa(t_j)\,, \tag{2}$$

where $y(t; \theta, u) \in \mathbb{R}^k$ is the output of the measurement process, which is recorded at T time points. The linear output function, characterised by the real matrix $C \in \mathbb{R}^{k \times n}$, models the capabilities of the measurement setup. The elements of this matrix are typically unknown. The measurements are obscured by the noise process $\varkappa_i(t_j) \sim \mathcal{N}(0, \sigma_{ij}^2)$ $(i = 1, \ldots, n; j = 1, \ldots, T)$. In addition, the observations might be done in unknown units, such that a reference Experiment might be needed to interpret any numerical values of $y(t_j)$. A typical example of C is:

$$C = \begin{pmatrix} c_1 & 2c_1 & 0 \\ 0 & c_2 & c_2 \end{pmatrix}. \tag{3}$$

Assuming that C is always scaled equally per row (in this way), with unknown row coefficients c_i, we can eliminate these unknown (but uninteresting) elements by taking

row wise ratios between experiments:

$$\tilde{y}_i(t_j;\theta,u_b) = \frac{y_i(t_j,\theta,u_b)}{y_i(t_j;\theta,u_0)}\,, \tag{4}$$

where $u_b \in \mathbb{R}^l (b = 1,\ldots,n_\mathrm{E})$ is any particular experimental setup (n_E is the number of experiments) and u_0 characterises the *reference* experiment setup. In consequence we have to recalculate the standard deviation of $\tilde{y}_i(t_j;\theta,u_b)$ from the standard deviations[1] of $y_i(t_j,\theta,u_b)$ and $y_i(t_j;\theta,u_0)$:

$$\tilde{\sigma}_{i,j} \approx \left|\frac{\partial\tilde{y}_i(t_j;\theta,u_b)}{\partial y_i(t_j;\theta,u_b)}\right|\sigma_{ij,b} + \left|\frac{\partial\tilde{y}_i(t_j;\theta,u_b)}{\partial y_i(t_j;\theta,u_0)}\right|\sigma_{ij,0}\,, \tag{5}$$

and in this case:

$$\tilde{\sigma}_{i,j} \approx \left|\frac{\tilde{y}_i(t_j;\theta,u_b)}{y_i(t_j;\theta,u_b)}\right|\sigma_{ij,b} + \left|\frac{\tilde{y}_i(t_j;\theta,u_b)}{y_i(t_j;\theta,u_0)}\right|\sigma_{ij,0}\,. \tag{6}$$

Alternatively, a time series measurement can be normalised at one of the points:

$$\tilde{y}_i(t_j;\theta,u_b) := \frac{y_i(t_j;\theta,u_b)}{y_i(t_{k(b)};\theta,u_b)}\,, \tag{7}$$

where k is the given normalisation time index for each experiment b. In this method, there is no control u_0. An example with $k = 3$ is listed in Table 1. Similar consequences apply here with regard to measurement error. Even more generally, each y_i can be normalised by another state variable, e.g.:

$$\tilde{y}_i(t_j;\theta,u_b) := \frac{y_i(t_j;\theta,u_b)}{y_{l(i,b)}(t_{k(i,b)};\theta,u_b)}\,. \tag{8}$$

Both k and l must be specified, for each output i and each experiment b.

2 Sensitivities

Given reference experiment normalisation method, the sensitivity of $\tilde{y}(t)$ in terms of the (known) $y(t;\theta,u)$ sensitivities $S(t;\theta,u)$ is:

$$\begin{aligned}\partial_{\theta_j}\tilde{y}_i(t;\theta,u_b) &= \frac{{S_i}^j(t;\theta,u_b)y_i(t;\theta,u_0) - y_i(t;\theta,u_b){S_i}^j(t;\theta,u_0)}{y_i(t;\theta,u_0)^2} \\ &= \frac{{S_i}^j(t;\theta,u_b) - \tilde{y}_i(t;\theta,u_b){S_i}^j(t;\theta,u_0)}{y_i(t;\theta,u_0)}\,.\end{aligned} \tag{9}$$

The code for this operation is located in the function `Likelihood` and is organized such, that if the data is absolute and does not require normalization, then ${S_i}^j(t;\theta,u_0)$ is set to 0 for all i,j and the reference $y_i(t;\theta,u_0) = 1$ for all i. Similar calculations apply for the other two methods.

[1] we append the input indeces $b, 0$ to the standard deviation symbol for distinction

k	t/h	y	$\tilde{y}$
0	0	20(4)	0.5(2)
1	4	30(5)	0.8(3)
2	9	35(6)	0.9(3)
3	12	37(6)	1.00

Table 1: Example data table for time-point normalisation, where the last measurement was used to define 100%. Here, y has never been fully quantified, but the ratios $\tilde{y}$ between the measurements $j = 0, \ldots, 2$ and $k = 3$ are reproducible by a replicated experiment.

3 Sensitivity Gradient

We take the derivative of (9) for any $u_b \in \{u_1, \ldots, u_{n_E}\}$:

$$\begin{aligned}\partial_{\theta_k} \tilde{S}_i{}^j(t;\theta,u_b) = \Bigg(& \frac{\partial S_i{}^j(t;\theta,u_b)}{\partial \theta_k} - \tilde{S}_i{}^k(t;\theta,u_b) S_i{}^j(t;\theta,u_0) - \\ & \tilde{y}_i(t;\theta,u_b) \frac{\partial S_i{}^j(t;\theta,u_0)}{\partial \theta_k} \Bigg) \frac{1}{y_i(t;\theta,u_0)} \\ & + \left(\tilde{y}_i(t;\theta,u_b) S_i{}^j(t;\theta,u_0) - S_i{}^j(t;\theta,u_b) \right) \frac{S_i{}^k(t;\theta,u_0)}{(y_i(t;\theta,u_0))^2} \,. \end{aligned} \tag{10}$$

4 Sampling in logarithmic Space

Ode models typically have a stable sign structure and so parameters usually must retain a fixed sign. In some cases the model becomes unstable after certain sign flips, which is definitely the case for biological models. But even if not some models will lose their validity when the sign structure changes. For this reason we require the sign structure to be given explicitely in the model definition and pass only positive values as parameters to the model.

Sampling in logarithmic space θ and passing $\rho = \exp(\theta)$ to the ode system has the additional benefit of covering several orders of magnitude more efficiently. Unfortunately, this choice implies that we have to modify the expressions for the model sensitivities. VFGEN and CVODES provide sensitivity analysis with respect to the nominal

model parameters ρ:

$$\begin{aligned}
\dot{x} &= f(t,x;\rho,w)\,, & \rho_j &= \exp(\theta_j)\,,\\
y(t;\rho,w) &= Cx(t;\rho,w)\,,\\
\tilde{y}_i(t;\rho,w) &= \frac{y_i(t;\rho,w)}{y_i(t,\rho,u_0)}\,,\\
\tilde{S}_i{}^j(t;\rho,w) &= \partial_{\rho_j}\tilde{y}_i(t;\rho,w)\,,\\
\partial_{\theta_j}\tilde{y}(t;\rho,w) &= \frac{\partial\tilde{y}_i(t;\rho,w)}{\partial\rho_j}\frac{\partial\rho_j}{\partial\theta_j}\\
&= \tilde{S}_i{}^j(t;\rho,w)\rho_j\,.
\end{aligned} \tag{11}$$

for any input w and consequently:

$$\begin{aligned}
\partial_{\theta_j}\tilde{y}_i(t;\rho,w) &= \tilde{S}_i{}^j(t;\rho,w)\rho_j\,,\\
\partial_{\theta_k}\tilde{S}_i{}^j(t;\rho,w)\rho_j &= \frac{\partial\tilde{S}_i{}^j(t;\rho,w)}{\partial\rho_l}\overbrace{\frac{\partial\rho_l}{\partial\theta_k}}^{\rho_l\delta_{lk}}\rho_j + \tilde{S}_i{}^j(t;\rho,w)\frac{\partial\rho_j}{\partial\theta_k}\\
&= \frac{\partial\tilde{S}_i{}^j(t;\rho,w)}{\partial\rho_k}\rho_k\rho_j + \tilde{S}_i{}^j(t;\rho,w)\rho_k\delta_{jk}\,.
\end{aligned} \tag{12}$$

Note that it is possible to use `Expressions` such as this in VFGEN models:

```
<Parameter Name="\theta_1" DefaultValue="0">
<Expression Name="rho_1" Formula="exp(theta_1)"/>
```

They can be used to convert the parameters from logspace, then use these expressions to define fluxes. But, since `Expressions` need to be recalculated at every step this is very wasteful. On the other hand, VFGEN will then compute correct sensitivities ($dx_i/d\theta_j$). To save calls to the exp function, the software does this conversion before calling the solver and transforms the sensitivities as described here.