AE567 Inference, Estimation, and Learning
University of Michigan March 13 2024

Project 2 Part 2: Bayesian Inference for Nonlinear Models

Due March 27, 2024 by 11:59 pm on Canvas

In this project we will consider the topic of parameter inference in dynamical systems.

The deliverable for this project is a report detailing your approach and answers to all of the questions in each section.

Please submit a single PDF file that contains your text and figures. All code should be in the *appendix*, unless you want to provide a BRIEF 5 line snippet that contains only the algorithm you are describing within the text. In the latter case, make absolutely sure that this algorithm follows the narrative of the text. For every figure that you obtain, please make sure that the axes are labeled, any relevant legend is shown, and a caption describes what is being plotted.

The preference is that a formal typed report is submitted. However, a scanned written report that is legible will be accepted. Illegible reports or reports that are primarily/entirely code with no text will not be evaluated.

The same guidelines outlined in Project 1 and the Syllabus apply here.

1 Delayed Rejection Adaptive Metropolis

In this section you will create the Delayed Rejection Adaptive Metropolis (DRAM) algorithm.

1. Implement the Adaptive Metropolis (AM) algorithm with the efficient covariance updating scheme.
2. Implement a delayed rejection algorithm with two levels. Assume that it is initialized with a high level proposal with covariance C and the second level ℓ reduces this covariance by γ. For instance if $\gamma = \frac{1}{2}$ and we have a two level scheme then the covariances of each proposal are C and $\frac{1}{2}C$.
3. Implement a DRAM approach where the top level proposal is the AM algorithm and the subsequent level has a scaled version of the AM covariance matrix.
4. Justify and check your implementation by showing its performance on the simple banana shaped distribution. In this problem the posterior is two dimensional $\pi(x_1, x_2)$. so that

$$\left(x_1, x_2 + (x_1^2 + 1)\right) \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}\right) \tag{1}$$

5. Compare the standard Metropolis Hastings, AM, DR, and DRAM, algorithms on the above example and describe which algorithm works best. Justify your answer by presenting the following results (similar to what we did in class)
   (a) Plots of 1D and 2D marginals
   (b) Autocorrelation plots
   (c) Integrated autocorrelation values
   (d) Acceptance ratios
   (e) Visual inspection of mixing.
6. **Do not forget burn-in!**

2 Apply DRAM for Bayesian Inference

In the next problem we will learn the parameters of some dynamical system. You have two options for the dynamical system which you can choose to study:

1. A satellite attitude dynamics application: this is a model of 3D rotational dynamics. The goal of learning is to learn some control parameters and some products of inertia. The dynamics are given in Section A.

2. A benchmark 3 state model for the spread of disease (SIR model). The dynamics are given in Section B.

Both models are effectively the same amount of work with respect to the learning aspect – since a majority of the work will be on inference. They are identical with regard to the inference procedures/methodologies that you are learning in class. The model for the satellite is, however, more complicated to set up. If you do both problems: we will treat the SIR model as the graded problem and then provide up to 10 points in extra credit for correct answers for the Satellite example.
For this problem you will use the DRAM algorithm to learn the parameters of your chosen dynamics model. Note that because the dynamics are deterministic, we do not need to infer the state at every time period. Use a standard Gaussian prior for each parameter $\theta_i \sim \mathcal{N}(0, 1)$. **Answer the following questions for your chosen problem. Note that each dynamical system has two learning problems, do the following for both.**

1. What is the likelihood model? Please describe how you determine this.

2. What is the form of the posterior?

3. How did you tune your proposal? Think carefully about what a good initial point and initial covariance could be?

4. Analyze your results using the same deliverables as you used in Section 1.
5. Plot the true parameters on your plots of the marginals for reference.
6. Plot the prior and posterior predictives of the dynamics (separately): for some prior/posterior samples, run the dynamics and plot them in a transparent light gray on top of your "truth" dynamics and data. These are essentially the probabilistic predictions of your model before and after you have accounted for the data. How do they look compared to the "truth." How does this plot differ between the prior and posteriors?

Please comment on the following:

- For Satellite: What is the difference between the two parameter inference problems? How does the posterior predictive change? Are there any notable differences?
- For SIR: How does non-identifiability affect the Bayesian approach? In many models it may not be clear by inspection that certain parameters are non-identifiable. How can you use the Bayesian approach to probe whether this might be the case?

3 Extra Credit

Find COVID-19 or any other disease data, and try to fit the SIR model for some time period.

1. Document where you got the data, and how you "cleaned it". Describe the dataset.
2. After doing inference and some period of time, provide a posterior predictive showing its prediction over that period of time, and slightly in the future.
3. Hint: Many papers have done this, so perhaps try to reproduce a result that you found.

A Satellite Dynamics

The goal is to learn the parameters of a control system and the moments / products of inertia from observations of only the attitude (orientation) of the satellite. This example is motivated by e.g., the desire to reverse engineering a system's control system from purely external observations. There are two learning problems that you will use with these dynamics

1. Problem 1: Parameters are the control coefficients (k_1, k_2)
2. Problem 2: Parameters are the control coefficients *and* a product of inertia (k_1, k_2, J_{12})

A.1 Equations

This model assumes that the satellite is a rigid body. The system has seven states: four states due to the quaternian (q_1, q_2, q_3, q_4) and three states due to the angular velocity $(\omega_1, \omega_2, \omega_3)$. Here are the dynamics for the states

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (q_4\omega_1 - q_3\omega_2 + q_2\omega_3) \\ (q_3\omega_1 + q_4\omega_2 - q_1\omega_3) \\ (-q_2\omega_1 + q_1\omega_2 + q_4\omega_3) \\ (-q_1\omega_1 - q_2\omega_2 - q_3\omega_3) \end{bmatrix}$$
$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = J^{-1} \begin{bmatrix} -\left(\omega_2\left(J_{31}\omega_1 + J_{32}\omega_2 + J_{33}\omega_3\right) - \omega_3\left(J_{21}\omega_1 + J_{22}\omega_2 + J_{23}\omega_3\right)\right) + \tau_1 \\ -\left(\omega_3\left(J_{11}\omega_1 + J_{12}\omega_2 + J_{13}\omega_3\right) - \omega_1\left(J_{31}\omega_1 + J_{32}\omega_2 + J_{33}\omega_3\right)\right) + \tau_2 \\ -\left(\omega_1\left(J_{21}\omega_1 + J_{22}\omega_2 + J_{23}\omega_3\right) - \omega_2\left(J_{11}\omega_1 + J_{12}\omega_2 + J_{13}\omega_3\right)\right) + \tau_3, \end{bmatrix}$$

where

$$J = \begin{bmatrix} 20 & J_{12} & 0.9 \\ J_{12} & 17.0 & 1.4 \\ 0.9 & 1.4 & 15 \end{bmatrix} \tag{2}$$

is the inertia matrix and $\tau = (\tau_1, \tau_2, \tau_3)$ are the torques applies to the system. These torques are determined by the following control law[1]

$$\tau = -k_1 M^T \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} - k_2 \begin{bmatrix} \mathtt{sat}(\omega_1) \\ \mathtt{sat}(\omega_2) \\ \mathtt{sat}(\omega_3) \end{bmatrix}, \qquad M = \frac{1}{2}\left(q_4 I_{3\times 3} + \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \right), \tag{3}$$

where

$$\mathtt{sat}(\omega_i) = \begin{cases} \mathtt{sign}(\omega_i) & \text{if } |\omega_i| > 1 \\ \omega_i & \text{otherwise} \end{cases} \tag{4}$$

This control law is design to bring about the satellite to an orientation with $q_1 = q_2 = q_3 = 0$, and this is what you should see in your simulation.

[1] Zou, An-Min, Anton H.J. de Ruiter, and Krishna Dev Kumar. "Finite-Time Output Feedback Attitude Control for Rigid Spacecraft under Control Input Saturation." Journal of the Franklin Institute 353, no. 17 (November 2016): 4442–70. https://doi.org/10.1016/j.jfranklin.2016.08.013.

A.2 Generating reference data

1. The "True" parameter settings are $(k_1, k_2, J_{12}) = (5, 5, 1.2)$.

2. The initial conditions are $(q_1, q_2, q_3, q_4, \omega_1, \omega_2, \omega_3) = (-0.6, 0.4, -0.2, q_4(0), 1.2, -1.5, 0.2)$ where $q_4(0) = \sqrt{1 - q_1(0)^2 - q_2(0)^2 - q_3(0)^2}$

3. Solve this system with a built-in timestepping scheme, i.e., ODE45 in Matlab or `scipy.integrate.solve_ivp` in python using the "RK45" method.

4. Take 50 data points using linearly spaced points between $t \in [0, 60]$. Your data consists of measurements of each of the (q_1, q_2, q_3) corrupted by independent zero mean noise with standard deviation 0.05.

5. Plot the trajectories of each state on a separate plot. Label the axes.

6. Plot the data on the appropriate plots.

B SIR Model

Here we consider an SIR model, a common benchmarking problem for MCMC algorithms. This model describes the evolution of a disease in a population by evolving the (S)usceptible, (I)nfected and (R)ecovered portions of the population. In this model it is assumed that humans infect each other directly rather than through a disease vector such as a bird or mosquito. You will setup two versions of this problem. One in which all the parameters are identifiable, and the second in which they are not. The equations are virtually identical, except for how the new parameters enter.

1. Problem 1 (Identifiable): Parameters are $\theta = (\beta, r, \delta)$
2. Problem 2 (Not-identifiable): Parameters are $\theta = (\gamma, \kappa, r, \delta)$

The dynamics for each are described next.

B.1 Equations for version 1: Identifiable Version

This model has three parameters $\theta = (\beta, r, \delta)$ and the evolution is given by the following nonlinear ODE

$$\frac{dS}{dt} = \delta N - \delta S - \beta IS \tag{5}$$

$$\frac{dI}{dt} = \beta IS - (r + \delta)I \tag{6}$$

$$\frac{dR}{dt} = rI - \delta R, \tag{7}$$

where N is the total population, in our case we fill fix this to $N = 1000$. We will use the initial conditions $S(0) = 900$, $I(0) = 100$, and $R(0) = 0$.

B.2 Equations for version 2: Non-Identifiable Version

This model is identical but has four parameters $\theta = (\gamma, \kappa, r, \delta)$ and the evolution is given by the following nonlinear ODE

$$\frac{dS}{dt} = \delta N - \delta S - \gamma\kappa IS \tag{8}$$

$$\frac{dI}{dt} = \gamma\kappa IS - (r + \delta)I \tag{9}$$

$$\frac{dR}{dt} = rI - \delta R, \tag{10}$$

where N is the total population, in our case we fill fix this to $N = 1000$. We will use the initial conditions $S(0) = 900$, $I(0) = 100$, and $R(0) = 0$.

B.3 Generate a reference simulation

1. The "True" parameter settings are $\theta = (0.02, 0.6, 0.15)$, use the "identifiable" version.
2. Solve this system with a built-in timestepping scheme, i.e., ODE45 in Matlab or `scipy.integrate.solve_ivp` in python using the "RK45" method.
3. Take 61 data points using linearly spaced points between $t \in [0, 6]$. Your data consists of measurements of only I corrupted by zero mean noise with standard deviation 50.
4. Plot the trajectories of each state on a separate plot. Label the axes.
5. Plot the data on the appropriate plot.

You will use these 61 noisy measurements of I for the inference problem.