ADVANCED SIMULATION METHODS

COURSEWORK 2, SPRING 2024-2025

This assignment will be done individually. You should prepare a **PDF** report written primarily in a in a report/paper format, to be submitted through Blackboard, Coursework 2 container (under Courseworks folder). You can also write this in IPython notebook format, if you like, however, you should export a readable PDF from this. By submitting the coursework, you will be agreeing that this assessed coursework is your own work unless otherwise acknowledged, includes no plagiarism, you have not discussed it with anyone else except when seeking clarifications with the module lecturer, and have not shared any code with anyone else prior to submission. In your answers, you can quote material from the notes and shared IPython notebooks, providing your own original explanations.

This course work accounts for 60% of the final grade. The report will be marked from 0 to 60. Page limit is 40 pages. The deadline is **1PM, 14 May 2025**. To enable anonymous marking, please put your CID and not your name into the filename of the submission or into the manuscript. Marking scheme will consider the correctness of your answers, organization and clarity of your report, as well as code quality and efficiency in your implementation.

Important Note 1: You are free to use the code I shared during the course as linked below. However, the code's correctness is your responsibility. The code I shared may have deficiencies as they are often simplified implementations. In other words, you should fully understand the code and modify it before you use in your coursework, and in case of typos and mistakes, you should correct it.

Important Note 2: There are many points in this coursework where you need to make choices of parameters and the setting. As soon as the general direction of the question is clear to you, you are free to make these choices. Each question aims to dig into some qualitative aspect of the methods we have seen, therefore, while the parameters I suggest below are a good setting, they are not an absolute requirement. However, you should justify these choices in your report and note that the parameters you choose should be reasonable, and the results you obtain should be interpretable.

Important Note 3: Please pay attention in the questions that ask you to "describe" methods. This means that the method should be introduced with reasonable (but not excessive) amount of detail. Directly providing the code of the method does not constitute a description. Unclear, short, and incomplete descriptions will not be awarded full marks.

Q1: MULTIPLE INTERACTING TARGET TRACKING: ALGORITHMS AND COMPARISONS

In this part, we will work with a modification of the target tracking problem discussed in one of the course Jupyter notebooks. We will modify this model to investigate the performance of stochastic filtering methods in higher dimensions. Consider the following "base" HMM:

$$\begin{aligned} \pi_0(x_0) &= \mathcal{N}(x_0; m_0, P_0), \\ \tau(x_t|x_{t-1}) &= \mathcal{N}(x_t; Ax_{t-1}, Q), \\ g(y_t|x_t) &= \mathcal{N}(y_t; Hx_t, R). \end{aligned}$$

where

$$A = \begin{bmatrix} I_2 & \kappa I_2 \\ \mathbf{0}_2 & 0.99 I_2 \end{bmatrix}, \quad H = I_4, \quad R = rI_4, \quad Q = \begin{bmatrix} \frac{\kappa^3}{3} I_2 & \frac{\kappa^2}{2} I_2 \\ \frac{\kappa^2}{2} I_2 & \kappa I_2 \end{bmatrix}$$

and $\kappa = 0.1, r = 0.1, m_0 = [0, 0, -20, 20], P_0 = I_4$, and $T = 400$. We have thus $y_n \in \mathbb{R}^4$ and $x_n \in \mathbb{R}^4$. Note that this is slightly different than the model in the notebook as we have data y_n has the same dimension as x_n. Assume that, as a modeller, you are interested in using this model to track multiple targets. In this case, it is standard to consider an extended state space $x_n \in \mathbb{R}^{4K}$ for K objects, where objects simultaneously evolve. A standard way to do this is to build a high-dimensional linear Gaussian model using the model above, constructing block diagonal matrices. For example, say for K objects, we build

$$\bar{A}_K = I_K \otimes A, \quad Q_K = I_K \otimes Q, \quad \bar{H}_K = I_K \otimes H, \quad \bar{R}_K = I_K \otimes R,$$

where I_K is the $K \times K$ identity matrix. Let us assume further that, to model interactions between objects, we perturb the transition noise matrix Q_K by adding a small noise matrix L:

$$\bar{Q}_K = Q_K + LL^\top,$$

where L is a zero mean Gaussian random matrix with entries having standard deviation $\sigma_L = 0.5$. With this, we have a full linear Gaussian hidden Markov model of $4K$ dimensions with parameters $\bar{A}_K, \bar{Q}_K, \bar{H}_K, \bar{R}_K$. Throughout, we will refer to this new HMM as multiple-object HMM (MOHMM).

1. Write a generic function to generate data from the model you described in the question. Test and demonstrate this function for $K = 1, 2, 5$ objects. To provide the demonstration, provide the plots of observed data and first two dimensions of true trajectories of the each object **on the same plot**.

(2 marks)

2. **Kalman filtering for MOHMM**

   (a) Describe the Kalman filtering procedure for the model you developed in Part 1 by describing the method and the pseudocode within this setting.

   (b) Implement the Kalman filter for the MOHMM you developed in Part 1. Demonstrate your implementation by running the Kalman filter for $K = 1, 2, 3, \ldots, 20$ objects. To demonstrate the results, (i) provide the plot of NMSEs w.r.t. the state dimension, (ii) provide the plots the states and their estimates **on the same plot** for $K = 2$ (for a sanity check and demonstration that your filter is working). Discuss the results and the effect of the state dimension on NMSEs.

(5 marks)

3. **Bootstrap Particle Filtering for MOHMM**

   (a) Describe the bootstrap particle filter (BPF) for this model by describing the method and the pseudocode within this setting.

   (b) Implement the BPF for the MOHMM you developed in Part 1. Demonstrate your implementation by running the BPF for $K = 1, 2, 3, \ldots, 20$ objects by fixing $N = 500$. To demonstrate the results, (i) provide the plot of NMSEs w.r.t. the state dimension, (ii) provide the plots the states and their estimates **on the same plot** for $K = 2$ (for a sanity check and demonstration that your filter is working). Discuss the results and the effect of growing state dimension for fixed N.

(c) Discuss your results in comparison to Part 3(b) by plotting the NMSEs w.r.t. the state dimension for both Kalman filter and BPF. Discuss the differences in performance between the two methods and the effect of the growing state dimension on performance.

(d) Identify the smallest K where the BPF fails (typically NMSEs $0.3 - 0.5$ or above). Fix K, and now run the BPF for

$$N = 100, 500, 1000, 5000, 10000$$

particles. Note that you need to run M (sufficient) Monte Carlo simulations for each N. Provide the plot of average NMSEs w.r.t. N in this setting. What is the rate of convergence? Provide a discussion and justification.

(6 marks)

4. **Optimal Particle Filtering for MOHMM**

(a) Using Lemmas 3.1 and 3.2 from the notes, derive the optimal proposal and corresponding weight expressions for optimal particle filtering in the MOHMM[1]. Finally, provide the pseudocode of the full optimal particle filter (OptPF).

(b) Implement OptPF using the formulas you have derived in part (a) for the MOHMM. Demonstrate your implementation by running the OptPF for $K = 1, 2, 3, \ldots, 20$ objects by fixing $N = 500$ for M (sufficient) Monte Carlo simulations. To demonstrate the results, (i) provide the plot of NMSEs w.r.t. the state dimension, (ii) provide the plots the states and their estimates **on the same plot** for $K = 2$ (for a sanity check and demonstration that your filter is working). Discuss the results and the effect of growing state dimension for fixed N. Discuss the differences between results in this part and ones you have found in Part 3.

(c) For $K = 1, 2, 3, 4$, provide the plots of effective sample sizes (ESS) of OptPF compared to the ESS of the BPF for $N = 500$ as run above. Specifically, plot the ESS over time for OptPF and BPF for each K (i.e. 4 different plots).

(10 marks)

Q2: MARGINAL LIKELIHOOD COMPUTATION AND PARAMETER INFERENCE

This question aims at the demonstration and investigation of marginal likelihood computations using particle filtering approaches and their exact counterparts. For this, we consider a 1-dimensional HMM

$$\begin{aligned}
\pi_0(x_0) &= \mathcal{N}(x_0; m_0, P_0), \\
\tau(x_t|x_{t-1}) &= \mathcal{N}(x_t; \theta x_{t-1}, Q), \\
g(y_t|x_t) &= \mathcal{N}(y_t; H x_t, R).
\end{aligned}$$

where $\theta \in [-1, 1]$, $Q > 0$, $H \in \mathbb{R}$, $R > 0$, $m_0 \in \mathbb{R}$, $P_0 > 0$. For the purposes of this exercise, you will simulate your own datasets from this model.

[1] In particular, **do not use** the expression given in Example 4.1 as no marks will be allocated in this case.

1. For fixed parameters, describe the algorithm and the procedure for computing the exact marginal likelihood $Z_\star = p(y_{1:T})$.

(2 marks)

2. Fix (appropriate) parameters $\theta, Q, H, R, m_0, P_0$. Simulate a dataset of length $T = 10, 20, 50, 100, 200, 500, 1000$ from the model above. Implement the procedure you described in previous part for computing the exact marginal likelihood for each dataset. Provide the exact log-likelihoods for each dataset.

(2 marks)

3. Describe and detail the procedure to estimate the marginal likelihood using the bootstrap particle filter (BPF).

(1 marks)

4. Implement the procedure to estimate the marginal likelihoods using the BPF. To demonstrate the implementation, you should use the same values and the **generated data** in Part 2 of this question. For a complete demonstration, perform the following steps.

   (a) Fix $N = 100$. For every T, let us denote the marginal likelihood estimate for N particles at Mth simulation as $Z_T^{N,M}$. Denote the averaged estimate over Monte Carlo simulations as $Z_T^N = \frac{1}{M}\sum_{m=1}^M Z^{N,m}$. Using these estimates, demonstrate empirically that:

   $$\frac{Z_T^N}{Z_{T,\star}} \to 1, \text{ as } M \to \infty,$$

   as M grows for every T in Part 2. Provide a commentary on the relationship between M and T. How should you choose M as T grows?

   (b) Fix $T = 500$ and increase N to $100, 500, 1000, 2000, 4000, 6000, 8000, 10000$. Plot the ratio $Z_T^N / Z_{T,\star}$ for each N in the same plot against increasing M. Discuss the impact of N on convergence w.r.t. M. Provide a commentary.

(6 marks)

5. Now assume we would like to conduct parameter inference for this model. Assume that R, Q, H are known and we want to infer θ. In this part, you should set some $\theta_\star$ simulate long sequences, i.e., choose a large T.

   (a) Describe and implement nested particle filter (NPF) and apply it to estimate θ for the model above, assuming a prior $\theta \sim \mathcal{U}(-1, 1)$. You should explicitly provide the NPF design parameters, e.g., the choice of jittering kernel and its parameters. Provide the estimates of θ over time and plot it against the true parameter $\theta_\star$.

   (b) Provide an alternative two-layer approach to the NPF exploiting the fact that the model is linear and Gaussian. Develop this importance sampler, describe the derivation and logic in detail, and finally provide the detailed and clear pseudocode.

   (c) Implement this importance sampler you have developed in Part 5(b) and estimate the parameter θ for the model above. Provide the estimates of θ over time and plot it against the true parameter $\theta_\star$.

(d) Discuss and compare the convergence performance of the NPF in part (a) and the importance sampler in part (c).

(14 marks)

Q3: BAYESIAN PARAMETER ESTIMATION IN STOCHASTIC VOLATILITY MODEL

Consider the Lorenz 63 model for $n \geq 1$:

$$\begin{aligned}
x_{1,n} &= x_{1,n-1} - \gamma \mathsf{s}(x_{1,n-1} - x_{2,n-1}) + \sqrt{\gamma}\varepsilon_{1,n}, \\
x_{2,n} &= x_{2,n-1} + \gamma(\mathsf{r}x_{1,n-1} - x_{2,n-1} - x_{1,n-1}x_{3,n-1}) + \sqrt{\gamma}\varepsilon_{2,n}, \\
x_{3,n} &= x_{3,n-1} + \gamma(x_{1,n-1}x_{2,n-1} - \mathsf{b}x_{3,n-1}) + \sqrt{\gamma}\varepsilon_{3,n},
\end{aligned}$$

where $\gamma > 0$ is the time step, $\mathsf{s}, \mathsf{r}, \mathsf{b}$ are the parameters of the system, and $(\varepsilon_{i,n})_{n\geq 1}$ for $i = 1, 2, 3$ are i.i.d. standard normal random variables. Using the setting linked above in the notebook (with appropriate modifications as necessary), generate a 2-dimensional dataset of observations using the parameters $\mathsf{s}_\star = 10, \mathsf{r}_\star = 28, \mathsf{b}_\star = 8/3$ with a starting point $x_0 = [-5.91652, -5.52332, 24.5723]$. To be specific, generate the observations with specified γ and σ_y (you can choose this as in the notebook or some other value but it has to be fixed and the generated states/data must make sense as in the notebook) and

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

With a prespecified T (you will also choose this), generate a dataset of observations $y_{1:T}$. In this assignment, you should design and place an appropriate prior on θ as required and motivate this prior distribution.

1. Describe and implement the particle MCMC method to infer the posterior distribution $p(\theta|y_{1:T})$. Demonstrate the method by running the particle MCMC for the dataset you have generated. To demonstrate the implementation, (i) provide the trace plots of the parameters $(\mathsf{s}, \mathsf{r}, \mathsf{b})$ against the true values, (ii) provide posterior distributions via `hist2d` plots for 2D marginal distributions on (r, s), (r, b), and (s, b), (iii) provide the autocorrelation functions for each parameter. (6 marks)

2. Describe and implement a nested particle filter (NPF). You should explicitly provide the NPF design parameters, e.g., the choice of jittering kernel and its parameters. Demonstrate the method by running the NPF for this model. You may a larger T for this part. To demonstrate the implementation, (i) provide the estimates of $(\mathsf{s}, \mathsf{r}, \mathsf{b})$ over time and plot it against the true parameter values, (ii) provide posterior distributions via `hist2d` plots for 2D marginal distributions on (r, s), (r, b), and (s, b) using N particles. (6 marks)

APPENDIX A

Let $x \in \mathbb{R}^d$ be the known (ground truth) quantity and $\bar{x} \in \mathbb{R}^d$ be its estimate. We define the NMSE for this case as:

$$\mathsf{NMSE}(x, \bar{x}) = \frac{\|x - \bar{x}\|^2}{\|x\|^2}.$$

Similarly, given a sequence of d-dimensional known (ground truth) variables $x_{0:T}$ and their estimates $\bar{x}_{0:T}$, we define the NMSE for this case as:

$$\mathsf{NMSE}(x_{0:T}, \bar{x}_{0:T}) = \frac{\sum_{n=0}^{T} \|x_n - \bar{x}_n\|^2}{\sum_{n=0}^{T} \|x_n\|^2}.$$