KTH ROYAL INSTITUTE OF TECHNOLOGY

SF2955

COMPUTER INTENSIVE METHODS IN MATHEMATICAL STATISTICS

Home assignment 1

David Ahnlund
000531-4778

Rame Dawli
970720-5333

May 8, 2023

Problem 1

The dynamic of the moving target in $\mathbb{R}^2$ is described by the motion model

$$X_{n+1} = \Phi X_n + \Psi_z Z_n + \Psi_w W_{n+1}, \quad n \in \mathbb{N} \tag{1}$$

where for each n, $X_n = (X_n^1, \dot{X}_n^1, \ddot{X}_n^1, X_n^2, \dot{X}_n^2, \ddot{X}_n^2)$ is a state vector containing the target's position, velocity and acceleration. $\{Z_n\}_{n\in\mathbb{N}^*}$ is the driving command modeled by a bivariate Markov chain. $\{W_n\}_{n\in\mathbb{N}^*}$ are bivariate, mutually independent and normally distributed noise variables. Moreover, for all $n \in \mathbb{N}$, $\tilde{X}_n := (X_n^T, Z_n^T)^T$.

To answer the question whether $\{X_n\}_{n\in\mathbb{N}}$ and $\{\tilde{X}_n\}_{n\in\mathbb{N}}$ are Markov chains or not, we have to define what a Markov chain is first. A stochastic process is defined as a Markov chain if the probability for each event to occur depends only on the state of the system at the previous step. This is given by the Markov property as

$$\Pr(X_{n+1} = x | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \Pr(X_{n+1} = x | X_n = x_n) \tag{2}$$

Given the motion model in **1**, we see that X_{n+1} depends both on the state of the system at the previous state X_n, and the driving command Z_n. W_{n+1} is a normal distributed noise that does not affect whether X_n is a Markov chain or not. Since X_{n+1} depends on both X_n and Z_n, we conclude that $\{X_n\}_{n\in\mathbb{N}}$ dose not follow the property of a Markov chain presented in **2** and therefore it is **not a Markov chain**.

For $\tilde{X}_n$, Z_n in now included in the matrix, which solve the problem in $\{X_n\}_{n\in\mathbb{N}}$, namely the dependence on both X_n and Z_n. Thus, the vector $\tilde{X}_{n+1} := (X_{n+1}^T, Z_{n+1}^T)^T$ depends only on its previous state X_n, Z_n and W_{n+1} which we already mentioned that it does not have an impact on the chain. Thus, we conclude the $\{\tilde{X}_n\}_{n\in\mathbb{N}}$ **is a Markov chain** based on **2**.

Figure 1 shows plotted trajectory for the moving target, which we conclude it to be a reasonable movement for a particle in $\mathbb{R}^2$.

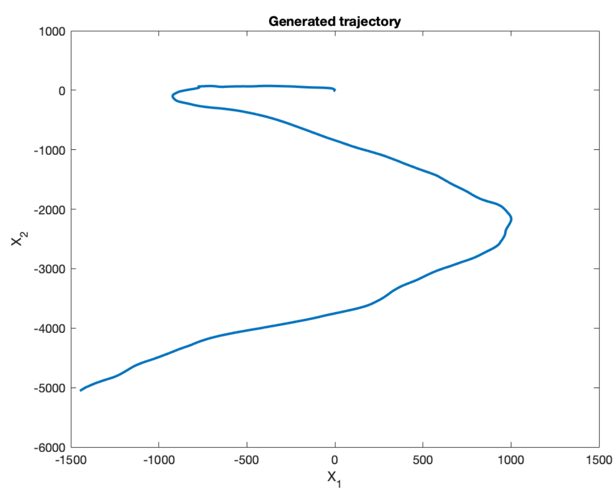

Figure 1: Generated trajectory, m = 500

Problem 2

A hidden Markov model, HMM, is defined as the canonical Markov chain $(X_n, Y_n)_{n\geq 0}$ where

1. The Markov chain $(\tilde{X}_n, Y_n)_{n\geq 0}$ is assumed to be only partially observed through the observation process $(Y_n)_{n\geq 0}$

2. The process $(\tilde{X}_n)_{n\geq 0}$ is referred to as state process

In this case, $\tilde{X}_n$ is a state process containing position, velocity, acceleration and driving command described in Problem 1. Furthermore, $\tilde{X}_n$ is only observed by the received signal strength indication, RSSI, received from 6 different base stations and given as the following:

$$\mathrm{Y}_n^l = \upsilon - 10\,\eta \log_{10} ||(X_n^1, X_n^2)^T - \vec{\pi}_l|| + V_n^l \tag{3}$$

Thus, $\{(\tilde{X}_n, Y_n)\}_{n\in\mathbb{N}}$ satisfies both requirements and can be seen as a hidden Markov model.

The components of observations vector $(Y_0, ..., Y_n)$ are conditionally independent given the corresponding states $(X_0, ..., X_n)$ such that the conditional distribution of each (Y_M) $0 \leq m \leq n$, depends only on X_M. [Theorem Lecture 5]

$$\Pr(Y_n|\tilde{X_n}) = \{\text{Ind}\} = \prod_{i=1}^{n} \Pr(Y_i|\tilde{X_i}) \tag{4}$$

Given 3, since V_n^l is Gaussian random vector with mean zero and standard deviation ζ:

$$\mathbb{E}(Y_n^l|\tilde{X}_n) = \mathbb{E}(\upsilon - 10\,\eta \log_{10} ||(X_n^1, X_n^2)^T - \vec{\pi}_l|| + V_n^l) = \upsilon - 10\,\eta \log_{10} ||(X_n^1, X_n^2)^T - \vec{\pi}_l|| \tag{5}$$

$$\mathbb{V}(Y_n|\tilde{X}_n) = \mathbb{V}(V_n^l) = \zeta^2 \tag{6}$$

thus,

$$(Y_n^l|\tilde{X}_n) \sim \mathcal{N}(\upsilon - 10\,\eta \log_{10} ||(X_n^1, X_n^2)^T - \vec{\pi}_l||, \zeta^2) \tag{7}$$

Mobility tracking (Problem 3 and 4)

SIS implementation

Due to the properties of the filter density of *Hidden Markov Models*, the weight updating in sequential importance sampling simplifies significantly.

We have the unnormalized density

$$z(\tilde{x}_{0:n}|y_{o:n}) = q(\tilde{x}_0)p(y_0|\tilde{x}_0)\prod_{k=1}^{n} p(y_k|\tilde{x}_k)q(\tilde{x}_k|\tilde{x}_{k-1})$$

The weight updating in SIS goes as follows:

$$\omega_{n+1}^i = \frac{z_{n+1}(X_{0:n+1}^i)}{z_n(X_{0:n}^i)g_n(X_{n+1}^i|X_{0:n}^i)}\omega_n^i$$

With $q(x_{n+1}|x_n)$, in SIS, being set to $g_n(x_{n+1}|x_{0:n})$ to weight update gets simplified to

$$\omega_{n+1}^i = p(y_{n+1}|X_{n+1}^i)\omega_n^i = \mathcal{N}(y_{n+1}; \left[\upsilon - 10\,\eta \log_{10} ||(X_{n+1}^1, X_{n+1}^2)^T - \vec{\pi}||\right]_{l=1}^{6}, \zeta^2 \mathrm{I}_{6\times 6}) \cdot \omega_n^i \tag{8}$$

Before the estimation of the trajectory of the actual Y-values, the SIS-implementation was first tested on a generated trajectory from the dynamic model, using the corresponding Y-values from eq. 3. This yields the results in Figure 2.

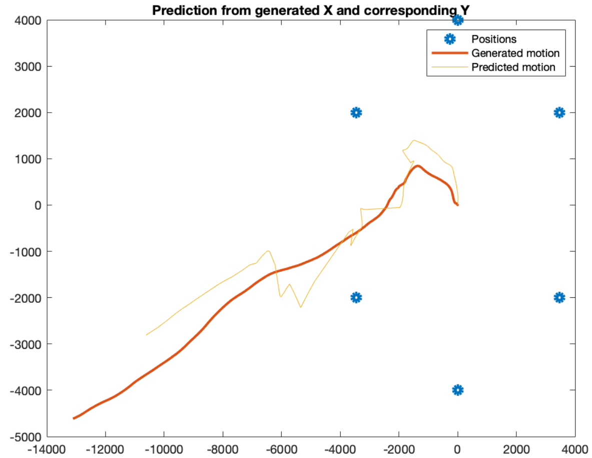

Figure 2: Estimated path of pre-known X-values (N = 10 000), without resampling

As seen in Figure 2, the predicted trajectory relatively quickly diverged from the actual X-values. This problem is well known and is due to the fact that no resampling was carried out, which leads to numerical stability issues.

Using this implementation for the actual Y-values, we can clearly see in Figure 3 how the weights quickly shifts towards only containing zeros, even though a logarithmic updated was carried out together with dividing the weights by the maximum element in the weight list. In Figure 4, the estimated trajectory is shown for $N = 10^6$.

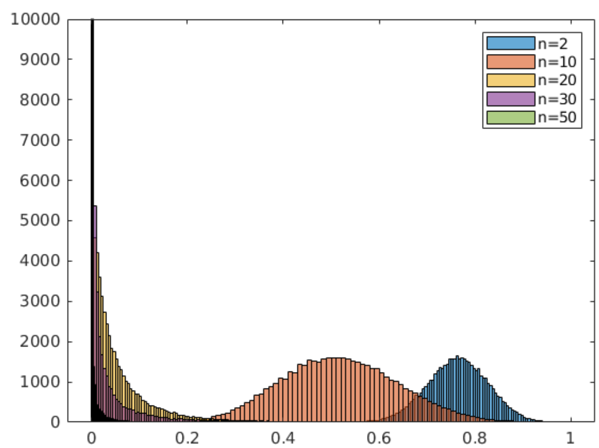

Figure 3: Weight histogram, no resampling

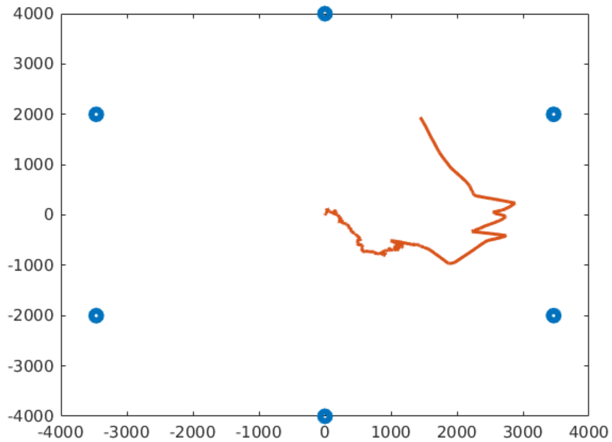

Figure 4: Estimated path (N = 10^6), no resampling

SISR implementation

When using resampling in the weight updating, the problems above resolved, as expected. In Figure 5, a generated trajectory, and its estimation using Y-values is shown. Clearly this algorithm works better. In Figure 6 this becomes clear also when observing the new weights.

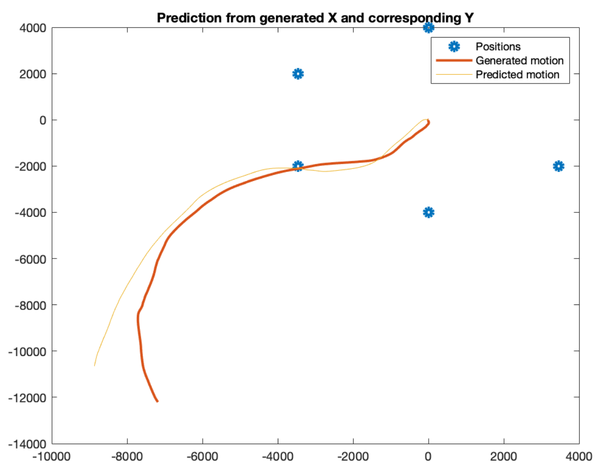

Figure 5: Estimated path of pre-known X-values, with resampling, N = 10 000

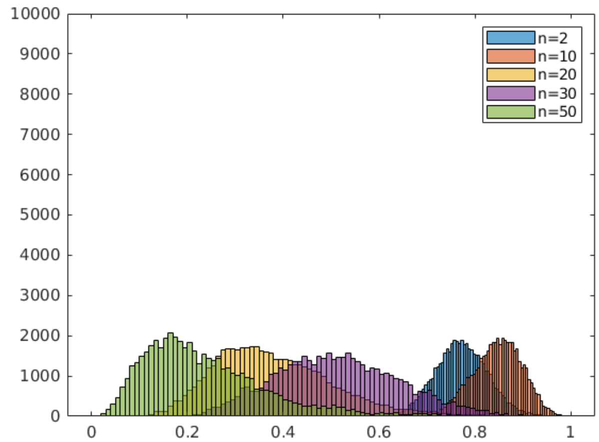

Figure 6: Weight histogram, with resampling

Using the SIS implementation with resampling, the final estimation of the mobility using the given Y-values from cellular networks is shown in Figure 7. Based on the estimated trajectory, the most likely driving command was observed to be $[0; 3.5]$.

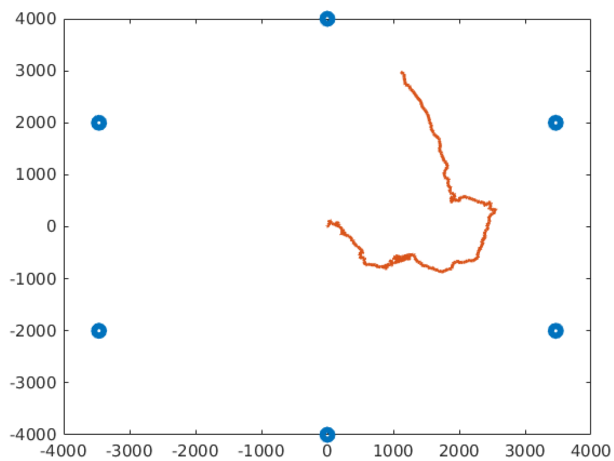

Figure 7: Estimated path (N = 10^6), with resampling

Comparison of efficient sample sizes

Another way of visualizing the bad impact on the estimation from the weight decay, is to observe the *efficient sample sizes* from the different weight. The efficient sample sizes are computed by the following formula

$$\mathrm{ESS}((\omega_n^i)_{i=1}^N) = \left(\sum_{i=1}^N \left(\frac{\omega_n^i}{\Omega_n} \right)^2 \right)^{-1} \tag{9}$$

and shows how many of the samples contribute to the estimation in an efficient manner. In Figure 8 the ESS for every weight (time steps) are shown, comparing the estimation without and with resampling, where $N = 10^6$. Clearly, not using resampling quickly makes the sample size inefficient. This due to the rapidly decaying weights.

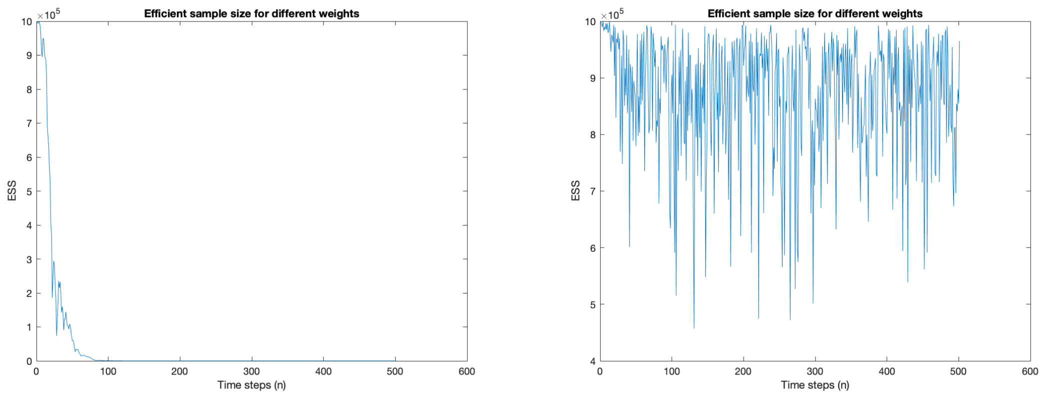

Figure 8: ESS without resampling (left figure) and with resampling (right figure)

The driver's actions were investigated by computing the driving command that was the most probable to happen at all time steps. This was done by tracking the evolution of the Markov chain through every step which is showed in Figure 9.

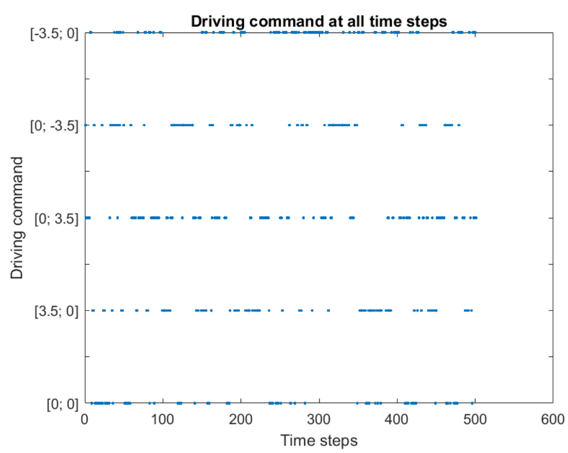

Figure 9: Driving commands among all time steps

Just by observing Figure 9, it is difficult to estimate which driving command was the most probable to occur. A more clear picture can be obtained by plotting a histogram of the driving command at all time steps. Figure 10 shows that all the five commands were chosen with almost the same probability with a small advantage for the driving command to the north as the most probable command.

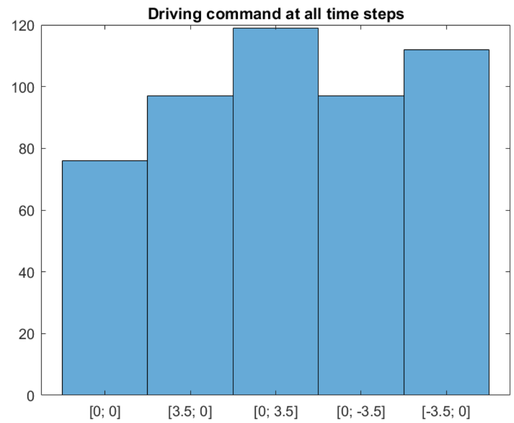

Figure 10: Histogram of driving commands among all time steps

Problem 5

The log-likelihood function is given as:

$$l_m(\zeta, Y_{0:m}) = m^{-1} \ln L_m(\zeta, Y_{0:m}) \tag{10}$$

where the likelihood $L_m(\zeta, Y_{0:m}) = f_\zeta(Y_{0:m})$ is defined as the normalizing constant of the smoothing distribution function $f(\tilde{X}_{0:n}|y_{0:n})$. The Monte-Carlo estimator of the normalizing constant c_n, with resampled particles is given in Lecture 6 as the following:

$$c_{N,n}^{SSIR} = \frac{1}{N^{n+1}} \prod_{k=0}^{n} \Omega_k \tag{11}$$

where

$$\Omega_k = \sum_{i=1}^{N} \omega_k^i \tag{12}$$

Thus, the log-likelihood function for ζ_j can be calculated by inserting 11 and 12 in 10 as:

$$l_m^N(\zeta_j) = m^{-1} \ln c_{N,n}^{SSIR} = \frac{1}{m} \ln \frac{1}{N^{m+1}} \prod_{k=0}^{n} \Omega_k = \frac{1}{m} (\sum_{k=0}^{m} \ln \sum_{i=1}^{N} \omega_k^i - (m+1) \ln(N)) \tag{13}$$

where we get ω_k^i from the SISR method implemented in problem 4 while using the new observations vector RSSI-measurements-unknown-sigma.mat. In order to get more accurate result for the estimated $\hat{\zeta}_m$, the grid was defined in a smaller interval $(2, 2.6)$ with an increment of 0.05 and a sample size $N = 20000$. We calculated then eq.13 for each value of ζ_j in the given interval and plotted $l_m^N(\zeta_j)$ to obtain $\hat{\zeta}_m$ that maximize $l_m^N(\zeta_j)$. As shown in Figure 11, $\hat{\zeta}_m$ that maximize the log-likelihood function given in eq. 13 was estimated to $\hat{\zeta}_m = 2.26$ and based on this value, the estimation for $\{\tau_n^1, \tau_n^2\}_{n=0}^{500}$ is given in Figure 12

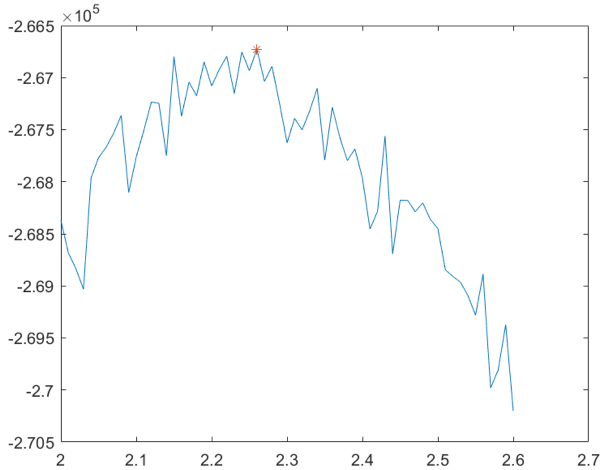

Figure 11: Log-likelihood as a function of ζ_j

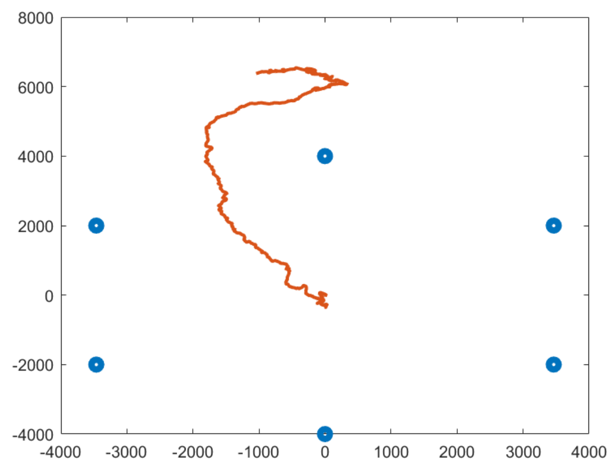

Figure 12: Esimated $\{\tau_n^1, \tau_n^2\}_{n=0}^{500}$ for $\zeta_j = 2.26$