SF2955 Computer Intensive Methods
Home Assignment 1

Group 27: Klara Zimmerman and Ville Wassberg

April 2024

In this lab, we look at sequential Monte Carlo-based mobility tracking in cellular networks. We begin by looking at a motion model to plot the trajectory of a moving target. We then look at the strength of received signals from base stations around the target, and try to simulate the trajectory based on this information. First we implement Sequential Importance Sampling (SIS), and then improve this model by adding resampling (SISR). Finally, we approximate the standard deviation of the noise variable in the hidden Markov model, using our SISR algorithm.

Problem 1

The mobility state model for a moving target is given by

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

We have simulated the trajectory $\{(X_n^1, X_n^2)\}_{n=0}^m$ of length $m = 500$, where the state of the target at time step n is given by $X_n = (X_n^1, \dot{X}_n^1, \dot{\dot{X}}_n^1, X_n^2, \dot{X}_n^2, \dot{\dot{X}}_n^2)^T$ which contains the positions, velocities and accelerations of the target.

The update of X_{n+1} was done using equation (1), where Z is the driving command, which can take any value in $\{(0,0)^T, (3.5,0)^T, (0,3.5)^T, (0,-3.5)^T, (-3.5,0)^T\}$, according to a transition probability matrix P, and W_n is the noise at state n, which is a bivariate mutually independent normal distributed random variable. Our trajectory is plotted in figure 1. The initial position vector X_0 has a normal distribution, and Z_0 is uniformly distributed over the set of driving commands.

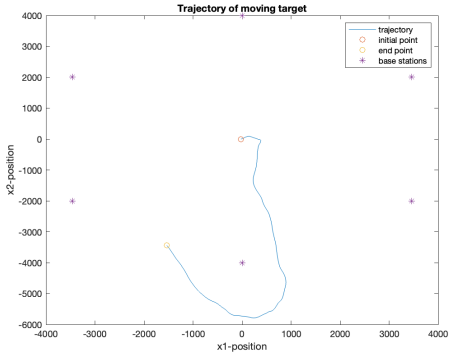

Figure 1: Example of what a trajectory can look like, with $m = 500$.

This looks like a reasonable trajectory of a moving target. In each state, the probability of the driving command continuing in the same direction as the previous command is 16 times greater than the target switching directions. This leads to a quite smooth trajectory.

A sequence is a Markov chain if the next state depends only on the current

state of the sequence, and no other states prior to it. This is called the Markov property.

In our problem, X_{n+1} depends on the current state X_n, as well as the current driving command Z_n. This means that $\{X_n\}_{n\in N}$ is not a Markov chain, as Z_n is not included in X_n. However, if we let $\tilde{X}_n = (X_n^T, Z_n^T)^T$, we see that we get a Markov chain, as $\{\tilde{X}_n\}_{n\in N}$ only depends on its own previous state, plus a noise variable.

Problem 2

We now view the observation model

$$Y_n^l = v - 10\eta log_{10}||(X_n^1, X_n^2)^T - \pi_l|| + V_n^l \tag{2}$$

where $\{V_n^l\}_{l=1}^s$ are independent Gaussian noise variables with mean $\mu_V = 0$ and standard deviation $\varsigma = 1.5$. We want to find the transition density $p(y_n|\tilde{x}_n)$ of $Y_n|\tilde{X}_n$.

Since V_n^l is Gaussian distributed, we know that the linear transformation Y_n^l will also have a Gaussian distribution. Given the position of the trajectory, $(X_n^1, X_n^2)^T$ which is contained in $\tilde{X}_n$, the mean of Y_n^l will be $v-10\eta log_{10}||(X_n^1, X_n^2)^T - \pi_l|| + \mu_V$. Thus, for each base station l with position π_l, we get,

$$Y_n^l|\tilde{X}_n \sim N(v - 10\eta log_{10}||(X_n^1, X_n^2)^T - \pi_l||, \varsigma^2).$$

Since the positions $\{\pi_l\}$ of the 6 base stations are independent, we can multiply the distributions for each l, and finally get

$$Y_n|\tilde{X}_n \sim \prod_{l=1}^{6} N(v - 10\eta log_{10}||(X_n^1, X_n^2)^T - \pi_l||, \varsigma^2).$$

Further, we see that $\{\tilde{X}_n, Y_n\}_{n\in N}$ forms a hidden Markov model, where $(\tilde{X}_n)_{n\geq 0}$ is the state process, and $(Y_n)_{n\geq 0}$ the observation process. That is, the model is only partially observed, and each observation Y_m is conditionally dependent on the corresponding hidden state $\tilde{X}_m$.

Problem 3

Here we implement the sequential importance sampling, SIS, without the usage of resampling, for sampling from

$$f(\tilde{x}_n|\mathbf{y}_{0:n}),\ n = 1, \ldots, m$$

for the given observation stream in the file RSSI-measurements.mat and find the estimates $\{(\tau_n^1, \tau_n^2)\}_{n=0}^m$. To do this we use the prior dynamics $q(\tilde{x}_n|\tilde{x}_{n-1}) \sim$

$\mathcal{N}(x_{n+1}; A\tilde{x}_n, \Psi_w^2)$ as proposal kernel, where $A = (\Phi, \Psi_z)$.

Notice that the expected trajectory plotted below is not very long, which is likely to be due to the weights converging towards zero fast in time; making the SIS algorithm unable to predict a trajectory.

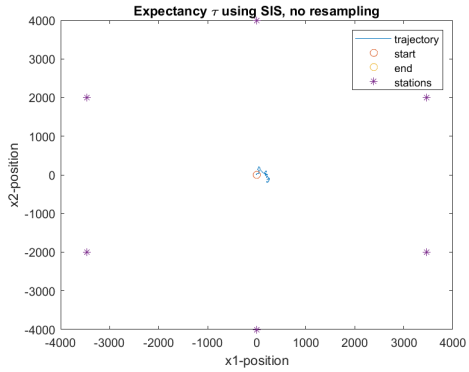

Figure 2: Expectancy of trajectory given m RSSI signals together with the basis stations.

As we can see, the histograms of the importance weights for SIS (figure 3) are skewed and the effective sample size plot (figure 4) suggests there are some particles that are much more influential than others which likely will dominate the algorithm and lead to poor approximation. Also, the weights gets less uniform over time. Due to the absence of resampling the variance among the weights increases, leading to the issue of some particles dominating others. This issue will be overcome in problem 4 below when using resampling among the weights.

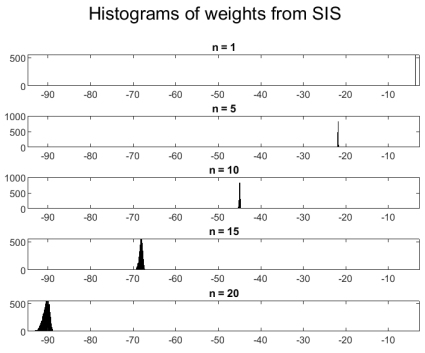

Figure 3: Skewed histograms of the importance weight for some states.

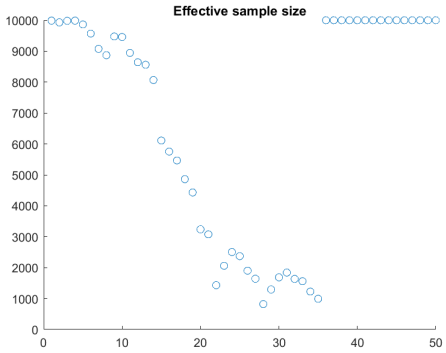

Figure 4: Effective sample size from using SIS method.

Moreover, the method was vectorised in order to avoid as much for loops as possible, and the computation time is approximately 5 seconds for a particle size of 10000.

Problem 4

Next, we implemented the sequential importance sampling with resampling (SISR) algorithm. Through resampling according to weights, particles with small weights are "killed off", and particles with large weights are duplicated. We thus avoid the problem of skewed weights, as we now get uniformly weighted samples (figure 7). Our expected trajectory using this method is plotted in figure 5, where we simulated $N = 10000$ particles at each of the $m = 500$ time steps.

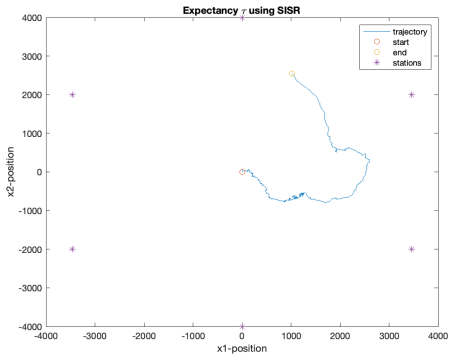

Figure 5: The expected trajectory, $\{(\tau_n^1, \tau_n^2)\}$, using SISR algorithm

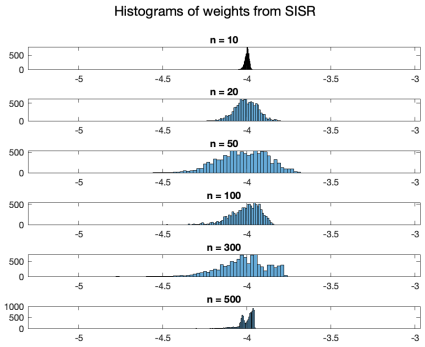

Figure 6: The weights of the particles when using resampling, at some time instances $n = 10, 20, 50, 300, 500$.

We want to investigate the driver's actions, by looking at what the most probable driving command is at each state n. Since $\tilde{X}_n$ is a Markov chain including Z_n, we know that the driving command at any given state only depends on the previous state X_{n-1}. When running our SISR algorithm, at each time n, we summed the weights of each particle with the same driving command. The driving command with the larges accumulated weight can thus be considered to be the most probable driving command at each state. These values are presented in figure 7.

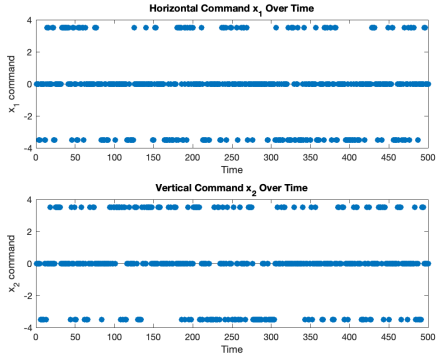

Figure 7: The most likely driving commands at each time instance m, in x_1- and x_2 directions respectively

As we can see, by resampling the particles at each time step, we successfully avoided the problem of skewed weights, and could simulate the trajectory from start to finish point. The conclusion from looking at the most probable driving commands at each time instance is that they seem to be spread quite uniformly. As expected, each command is likely to be followed by the same command.

Problem 5

To achieve a more realistic model we calibrate the sequential Monte Carlo algorithm to find the most suitable parameters, in this case the standard deviation of the observed noise. The standard deviation ς is sampled from a grid between zero and three, with increment 0.01. For very small standard deviations, $\varsigma < 0.3$, the weights were zero, which meant the algorithm could not run. Therefore we only looked at $0.3 \leq \varsigma < 3$.

The SISR algorithm is used to estimate the normalised log-likelihood function $\varsigma \mapsto m^{-1} \ln f_\varsigma(\mathbf{y}_{0:m})$. Then we choose the ς that maximises this log-likelihood. The approximate maximum likelihood estimate, $\hat{\varsigma}_m$, obtained was

$$\hat{\varsigma}_m = 2.15.$$

Finally, below is the plot of the expected trajectory produced by this estimated std, with signals from the file RSSI-measurements-unknown-sigma.mat.

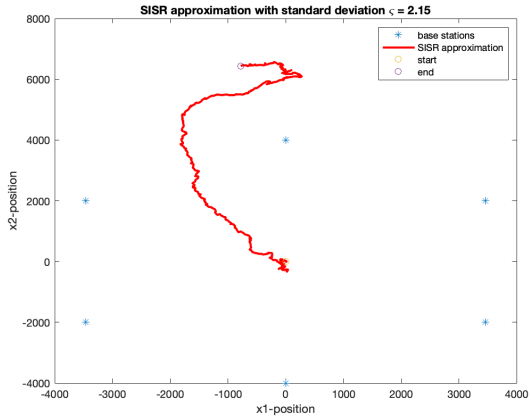

Figure 8: The simulated trajectory with approximated $\varsigma = 2.15$.