SF2955 Computer Intensive Methods
HA1 - Group 22

Valeri Ivinskiy

April 28, 2020

Problem 1

Consider a moving target in $\mathbb{R}^2$ according to the dynamics described below:

$$\boldsymbol{X}_{n+1} = \boldsymbol{\Phi}\boldsymbol{X}_n + \boldsymbol{\Psi}_z\boldsymbol{Z}_n + \boldsymbol{\Psi}_w\boldsymbol{W}_{n+1}, n \in \mathbb{N} \tag{1}$$

where $\{Z_n\}_{n\in\mathbb{N}}$ is the driving command modeled by a bivariate Markov chain taking on values

$$\{(0,0)^T, (3.5,0)^T, (0,3.5)^T, (0,-3.5)^T, (-3.5,0)^T\} \tag{2}$$

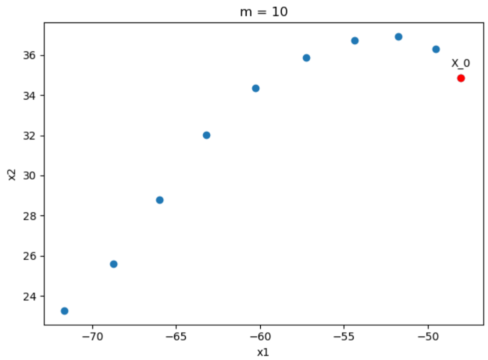
The chain evolves independently according to a transition matrix $\boldsymbol{P}$ which is defined in the project description. $\{W_n\}_{n\in\mathbb{N}}$ are bivariate mutually independent Gaussian noise, distributed as $N(0_{2x1}, \sigma^2\boldsymbol{I})$ with $\sigma = 0.5$. $\boldsymbol{\Phi}, \boldsymbol{\Psi}_z, \boldsymbol{\Psi}_w$ are also defined in the project description. The initial state vector $\boldsymbol{X_0}$ is assumed to be $N(0_{6x1}, diag(500, 5, 5, 200, 5, 5))$ distributed and the initial driving command $\boldsymbol{Z_0}$ is assumed to be uniformly distributed over the set (2).

To prove whether $\{\boldsymbol{X}_n\}_{n\in\mathbb{N}}$ is a Markov chain the Markov properties $\mathbb{P}(\boldsymbol{X}_0 \in A) = \chi(A)$ (where χ is the initial distribution) and $\mathbb{P}(\boldsymbol{X}_{n+1} \in B|\boldsymbol{X}_0, \boldsymbol{X}_1, ..., \boldsymbol{X}_n) = \mathbb{P}(\boldsymbol{X}_{n+1} \in B|\boldsymbol{X}_n)$ needs to hold. The first condition is satisfied by the definition of the problem, and thus we check the second condition:

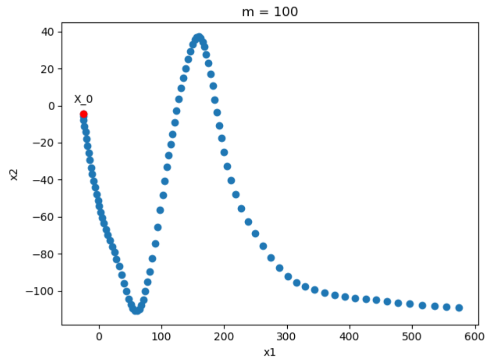
$$\begin{aligned}\mathbb{P}(\boldsymbol{X}_{n+1} \in B|\boldsymbol{X}_0, \boldsymbol{X}_1, ..., \boldsymbol{X}_n) &= \{\text{We assume that } \boldsymbol{X}_n \text{ is a Markov chain}\} = \\ &\mathbb{P}(\boldsymbol{X}_{n+1} \in B|\boldsymbol{X}_n) = \mathbb{P}(\boldsymbol{X}_{n+1}|\boldsymbol{\Phi}\boldsymbol{X}_{n-1} + \boldsymbol{\Psi}_z\boldsymbol{Z}_{n-1} + \boldsymbol{\Psi}_w\boldsymbol{W}_n)\end{aligned}$$

Here we see that since Z_n is the only Markovian factor in equation (1), and X_{n+1} depends on Z_{n-1} which leads us to the conclusion that X_n is not in fact a Markov chain. On the other hand, $\tilde{\boldsymbol{X}}_n = (\boldsymbol{X}_n^T \boldsymbol{Z}_n^T)$ is a Markov chain since now all components of $\tilde{\boldsymbol{X}}$ depend on the previous step. The transition density is given by $\tilde{\boldsymbol{X}}_{n+1}|\tilde{\boldsymbol{X}}_n \sim N(\boldsymbol{\Phi}\tilde{x}_n + \boldsymbol{\Psi}_z\boldsymbol{Z}_n, \boldsymbol{\Psi}_w\sigma^2\boldsymbol{I}\boldsymbol{\Psi}_w^T)$.

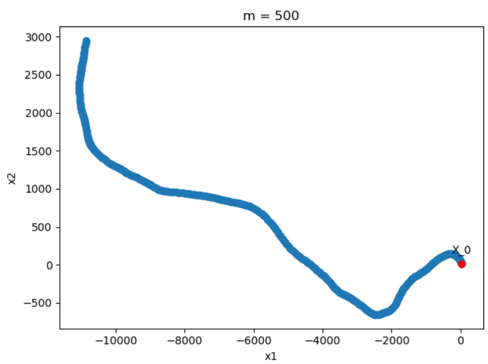
The trajectory given by this model was implemented in Python and yielded the following plots for 4 different m's:



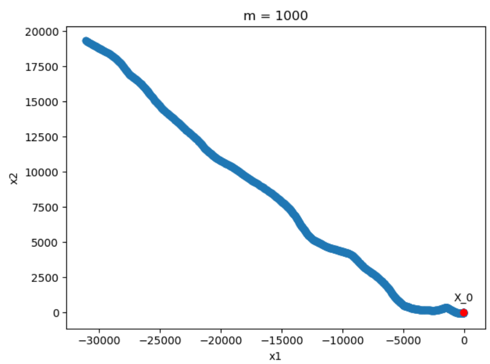
(a) m = 10



(b) m = 100



(c) m = 500



(d) m = 1000

Where we see that the implementation was successful since the trajectories for different m's indeed look like a reasonable trajectory of a moving object, with no large discrepancies anywhere.

Problem 2

The RSSI which each station ℓ receives from the moving target is given by the following model:

$$Y_n^\ell = v - 10\eta \log_{10} ||(X_n^1 X_n^2)^T - \pi_\ell|| + V_n^\ell \tag{3}$$

where $v = 90, \eta = 3$ and V_n^ℓ are independent Gaussian noise variables for each station with mean zero and standard deviation $\epsilon = 1.5$.
We want to convince ourselves that this together with $\tilde{\boldsymbol{X}}_n$, i.e. $(\tilde{\boldsymbol{X}}_n, \boldsymbol{Y}_n)_{nin\mathbb{N}}$ forms a hidden Markov model and then find the transition density $p(y_n|\tilde{x}_n)$ of $\boldsymbol{Y}_n|\tilde{\boldsymbol{X}}_n$.
To see whether this is indeed a HMM we begin by plotting a trajectory for an arbitrary time length, in this case $m = 1000$, and then also the RSSI for each stations to see whether any form of correlations are visible.

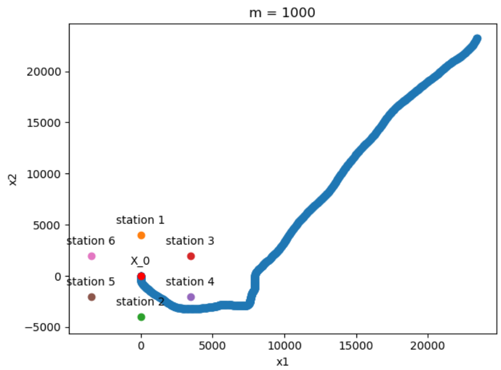
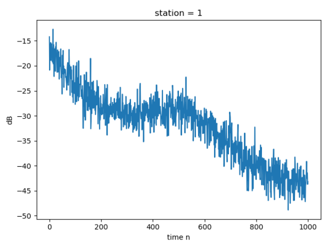
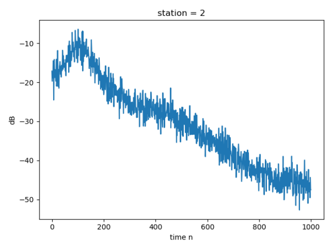


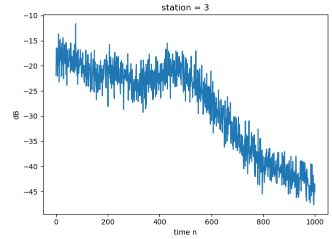
Figure 2: Trajectory with stations marked



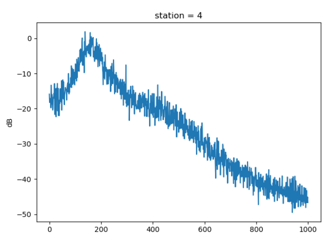
(a) Station 1



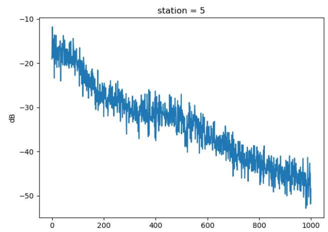
(b) Station 2



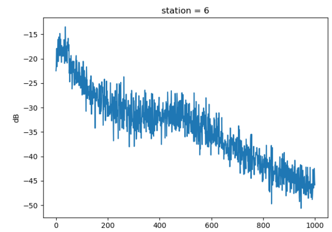
(c) Station 3



(d) Station 4



(e) Station 5



(f) Station 6

In all cases, we see that the RSSI gets weaker as the moving target gets further away from the stations. In particular, we see spikes in station 2 and station 4 as the target gets closer to them, and as the target leaves the RSSI decreases as well. There is a clear correlation between the RSSI which each station receives and the trajectory of the moving target, which leads to the conclusion that $(\tilde{\boldsymbol{X}}_n, \boldsymbol{Y}_n)_{nin\mathbb{N}}$ indeed forms a hidden Markov model.

To find the transition density, we will first look at the CDF:

$$\begin{aligned}
\mathbb{P}(\boldsymbol{Y}_n^\ell \leq y_n^\ell | \tilde{\boldsymbol{X}}_n = \tilde{x}_n) &= \mathbb{P}(v - 10\eta \log_{10} ||(X_n^1 X_n^2)^T - \pi_\ell|| + V_n^\ell \leq y_n^\ell | \tilde{\boldsymbol{X}}_n = \tilde{x}_n) \\
&= \mathbb{P}(V_n^\ell \leq y_n^\ell - v + 10\eta \log_{10} ||(X_n^1 X_n^2)^T - \pi_\ell|| | \tilde{\boldsymbol{X}}_n = \tilde{x}_n) \\
&= /\text{Since } (X_n^1 X_n^2)^T \text{ is in } \tilde{\boldsymbol{X}}_n \text{ we can remove the conditioning}/ \\
&= \mathbb{P}(V_n^\ell \leq y_n^\ell - v + 10\eta \log_{10} ||(X_n^1 X_n^2)^T - \pi_\ell||)
\end{aligned}$$

which in turn yields us the following transition density for one component:

$$\begin{aligned}
p(y_n^\ell | \tilde{x}_n) &= \frac{\partial}{\partial y_n^\ell} \mathbb{P}(V_n^\ell \leq y_n^\ell - v + 10\eta \log_{10} ||(X_n^1 X_n^2)^T - \pi_\ell||) \\
&= f_{V_n^\ell}(y_n^\ell - v + 10\eta \log_{10} ||(x_n^1 x_n^2)^T - \pi_\ell||) \qquad (4)
\end{aligned}$$

where $f_{V_n^\ell}$ is the Gaussian probability density function with mean zero and standard deviation $\epsilon = 1.5$. This means that the whole transition density is given by the following multivariate Gaussian distribution:

$$\boldsymbol{Y}_n|\tilde{\boldsymbol{X}}_n \sim N(\boldsymbol{A}_n, \epsilon^2 \boldsymbol{I}_{6x6}) \tag{5}$$

where $\boldsymbol{A}_n$ is a 6x1 vector with components $v - 10\eta log_{10}||(x_n^1 x_n^2)^T - \pi_\ell||$ for $\ell = 1, ..., 6$

Problem 3

Start by defining the transition density:

$$q(\tilde{x}_{n+1}|\tilde{x}_n) = N(\tilde{x}_{n+1}; \boldsymbol{\Phi}\tilde{x}_n + \boldsymbol{\Psi}_z \boldsymbol{Z}_n, \boldsymbol{\Psi}_w \sigma^2 \boldsymbol{I} \boldsymbol{\Psi}_w^T)$$
$$\text{since } \tilde{\boldsymbol{X}}_{n+1}|\tilde{\boldsymbol{X}}_n \sim N(\boldsymbol{\Phi}\tilde{x}_n + \boldsymbol{\Psi}_z \boldsymbol{Z}_n, \boldsymbol{\Psi}_w \sigma^2 \boldsymbol{I} \boldsymbol{\Psi}_w^T)$$

Setting the smoothing density $f(\tilde{x}_{0:n}|y_{0:n}$ as z_n and the transition density $q(\tilde{x}_{n+1}|\tilde{x}_n)$ as g_n yields that the weights are defined as:

$$\omega_{n+1}^i = N(\boldsymbol{y}_{n+1}; \boldsymbol{A}_n, \epsilon^2 \boldsymbol{I}_{6x6})\omega_n^i$$
$$\text{since } \boldsymbol{Y}_n|\tilde{\boldsymbol{X}}_n \sim N(\boldsymbol{A}_n, \epsilon^2 \boldsymbol{I}_{6x6})$$

where each $\boldsymbol{y}_n$ is the observation vector of length s = 6 (number of stations).
To confirm whether the algorithm is correct, an artificial sample was created from problem 1 and 2 and the SIS was tested on it. The testing trajectory is seen below:

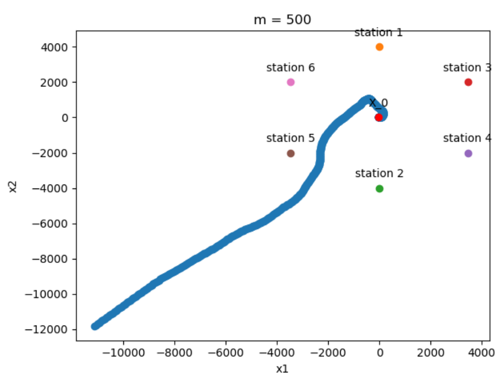
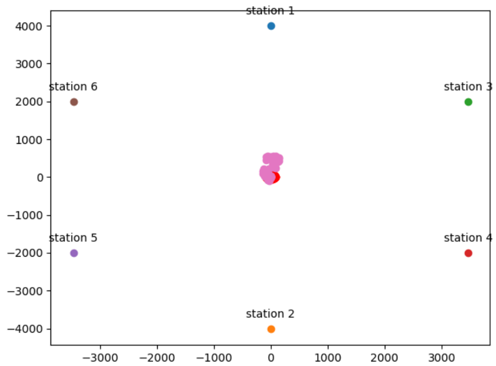


Figure 4: Testing trajectory

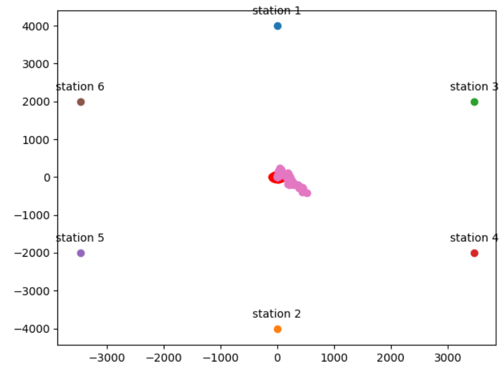
We now initialize the SIS algorithm:

$$X_0^i \sim N(0, diag(500, 5, 5, 200, 5))$$
$$\omega_0^i = N(y_0; \boldsymbol{A}_0, \epsilon^2 \boldsymbol{I}_{6x6})$$

And proceeding with the SIS algorithm with $N = 10000$ yields the following trajectories:

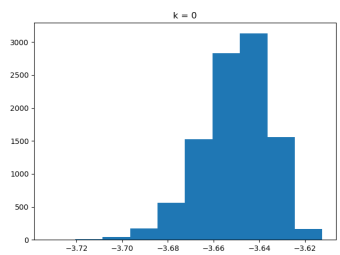


(a) Testing trajectory with SIS

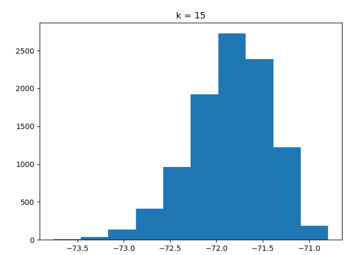


(b) Homework trajectory with SIS

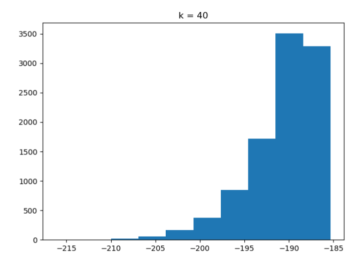
We see that the trajectory quickly degenerates and the weights become so infinitely small so they kill the trajectory. There is no trace of a real trajectory. Below are the histograms of the importance weights: It



(a) Importance weights for k=0, logarithm base 10



(b) Importance weights for k=15, logarithm base 10



(c) Importance weights for k=40, logarithm base 10

is clear that at time step 0, the weights are quite close to 0, but quickly degenerate down and by the 40th time step they are are already in the area around 10^{-190}, becoming infinitely small. Computing the efficient sample sizes for these time steps by Kish's Effective Sample Size defined as following:

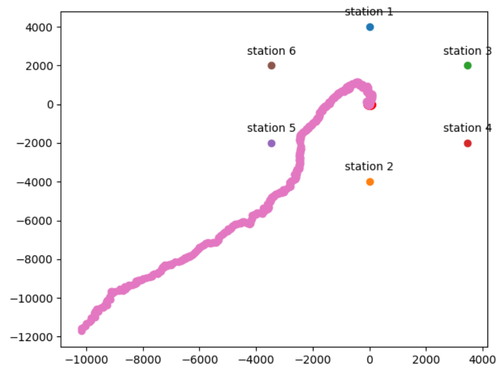
$$n_{eff} = \frac{(\sum_{i=1}^{N} \omega_i)^2}{\sum_{i=1}^{N} \omega_i^2}$$

yields $n_{eff}(0) = 9989, n_{eff}(15) = 5479, n_{eff}(40) = NaN$, by the 40th time step the denominator has become so infinitely small the efficient sample size does not exist. The closest which is computable is $n_{eff}(34) = 1160$.

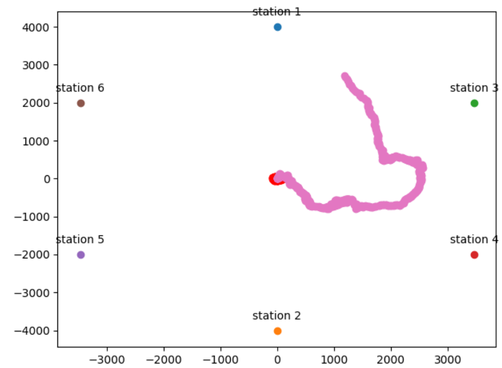
Problem 4

We now apply the resampling step in the algorithm, where we draw with replacement $(\tilde{x}'^{i}_{k})_{i=1}^{N}$ from $(\tilde{x}^{i}_{k})_{i=1}^{N}$ with probabilities $(\frac{\omega^i_k}{\Omega_k})_{i=1}^{N}$.

This yields the following trajectories:



(a) Testing trajectory with SISR



(b) Homework trajectory with SISR

We see that the testing trajectory generated by the SISR is very similar to the true trajectory, leading to the conclusion that the trajectory generated by the given station observations should be similar to the true trajectory as well. By looking at the most probable driving command at all time points (defined as the most frequent driving command at each observation) we see that the driving command tends to stay the same for multiple observations in a row, which is in line with the proposed kernel's transition matrix for Z_n.

Problem 5

The standard deviation in observation noise will be estimated by the use of maximum likelihood, as is recommended by the project description. As the likelihood function, the estimate of the normalizing constant of the smoothing distribution will be used. The estimator for the normalizing constant in SISR is given by:

$$\ell_m^N(\epsilon_j) = c_{N,n}^{SISR} = \frac{1}{N^{n+1}} \prod_{k=0}^{n} \Omega_k \tag{6}$$

and the corresponding log-likelihood function is:

$$\ln \ell_m^N(\epsilon_j) = \ln 1 - (n+1)\ln N + \sum_{k=0}^{n} \ln \Omega_k \tag{7}$$

This function is then evaluated for different $\epsilon_j \in (0,3)$ by running the SISR algorithm from Problem 4 for each of the different ϵ_j. Since the interval is open, we chose to run the algorithm on all ϵ_j from 0.5 to 2.9, with steps 0.1, producing 24 different evaluations of the log-likelihood function. Due to the nature of this task, N was chosen to be 1000 to reduce execution time.

Performing these steps produced the following MLE estimator:

$$\hat{\epsilon}_j = 2.9$$

and the estimated trajectory with this standard deviation is seen below:

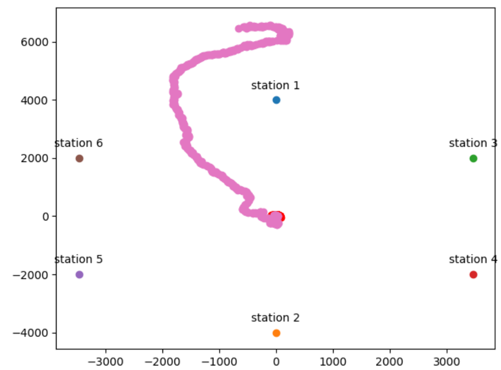


Figure 8: Estimated trajectory with $\hat{\epsilon}_j = 2.9$

(note that this trajectory was produced with the code from Problem 4, with N = 10000 particles).