Avd. Matematisk statistik

KTH Matematik

HOME ASSIGNMENT 1, SF2955 COMPUTER INTENSIVE METHODS IN MATHEMATICAL STATISTICS

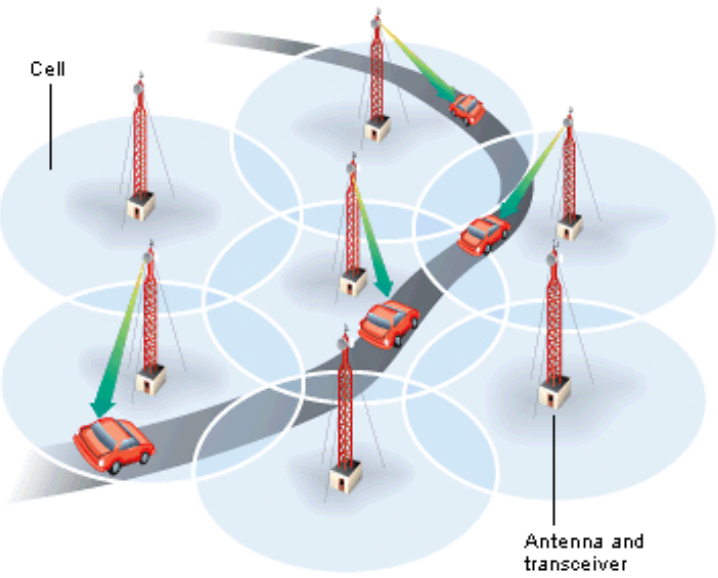
Teacher: Jimmy Olsson
All MATLAB-files needed are available through the course home page.
The following is to be submitted in Canvas by **Thursday 3 May, 12:00:00**:

- A report, named `group number-HA1-report.pdf`, of **maximum 7 pages** in `pdf` format. The report should provide detailed solutions to all problems. The presentation should be self-contained and understandable without access to the code.
- *All* your `m`-files (or similar depending on your language of choice) along with a file named `group number-HA1-matlab.m` that runs your analysis.

Discussion between groups is permitted, as long as your report reflects your own work.

Sequential Monte Carlo-based mobility tracking in cellular networks

19 april 2018



A hidden Markov model for mobility tracking

Motion model

Consider a target moving in $\mathbb{R}^2$ according to some dynamics described by the model

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

where

- for each n, $\mathbf{X}_n = (X_n^1, \dot{X}_n^1, \ddot{X}_n^1, X_n^2, \dot{X}_n^2, \ddot{X}_n^2)^\intercal$ is a state vector containing the target's positions (X_n^1, X_n^2) (in m), velocities $(\dot{X}_n^1, \dot{X}_n^2)$ (m s^{-1}) and accelerations $(\ddot{X}_n^1, \ddot{X}_n^2)$ (m s^{-2}) along the x_1 and x_2 directions, respectively.

- $\{\mathbf{Z}_n\}_{n\in\mathbb{N}^*}$ is the *driving command* modeled by a bivariate Markov chain taking on the values

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

  In the first state in the set (2), the driver does not add any velocity to the target; in the remaining states, the driver adds velocity to the target by steering in the east, north, west and south directions, respectively. The chain evolve, independently of everything else, according to the transition probability matrix

$$\mathbf{P} = \frac{1}{20}\begin{pmatrix} 16 & 1 & 1 & 1 & 1 \\ 1 & 16 & 1 & 1 & 1 \\ 1 & 1 & 16 & 1 & 1 \\ 1 & 1 & 1 & 16 & 1 \\ 1 & 1 & 1 & 1 & 16 \end{pmatrix}.$$

- $\{\mathbf{W}_n\}_{n\in\mathbb{N}^*}$ are bivariate, mutually independent normally distributed noise variables; more specifically, each $\mathbf{W}_n$ is $N(\mathbf{0}_{2\times1}, \sigma^2\mathbf{I})$-distributed with $\sigma = .5$.

- $\mathbf{\Phi}$, $\mathbf{\Psi}_z$ and $\mathbf{\Psi}_w$ are matrices given by

$$\mathbf{\Phi} = \begin{pmatrix} \tilde{\mathbf{\Phi}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \tilde{\mathbf{\Phi}} \end{pmatrix} \quad \text{and} \quad \mathbf{\Psi}_\bullet = \begin{pmatrix} \tilde{\mathbf{\Psi}}_\bullet & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \tilde{\mathbf{\Psi}}_\bullet \end{pmatrix},$$

  where

$$\tilde{\mathbf{\Phi}} = \begin{pmatrix} 1 & \Delta_t & \Delta_t^2/2 \\ 0 & 1 & \Delta_t \\ 0 & 0 & \alpha \end{pmatrix}, \quad \tilde{\mathbf{\Psi}}_z = \begin{pmatrix} \Delta_t^2/2 \\ \Delta_t \\ 0 \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{\Psi}}_w = \begin{pmatrix} \Delta_t^2/2 \\ \Delta_t \\ 1 \end{pmatrix}, \tag{3}$$

  with $\Delta_t = 0.5$ (s) being the sampling discretization period and $\alpha = .6$ the correlation between subsequent acceleration values. Consequently, the target's shift in velocity is modeled as the superposition of the driving command and a random, correlated acceleration.

The initial state vector $\mathbf{X}_0$ is assumed to be $N(\mathbf{0}_{6\times1}, \mathrm{diag}(500, 5, 5, 200, 5, 5))$-distributed and the initial driving command $\mathbf{Z}_0$ is supposed to be uniformly distributed over the set (2).

Problem 1

Aquaint yourself with the mobility state model and make sure that you understand the rationale behind the equation (1). Is $\{\mathbf{X}_n\}_{n\in\mathbb{N}}$ a Markov chain? For all $n \in \mathbb{N}$, let $\tilde{\mathbf{X}}_n = (\mathbf{X}_n^\intercal, \mathbf{Z}_n^\intercal)^\intercal$; is $\{\tilde{\mathbf{X}}_n\}_{n\in\mathbb{N}}$ a Markov chain? Implement a MATLAB code simulating a trajectory $\{(X_n^1, X_n^2)\}_{n=0}^m$ of some arbitrary length m and plot the same. Does it look like a reasonable trajectory of a moving target?

Observation model

As the target is moving, it measures online the pilot signal strenghts (i.e., the *received signal strength indication*, RSSI) from the basis stations of a cellular network. The network comprises $s = 6$ basis stations (BS), whose positions $\{\boldsymbol{\pi}_\ell\}_{\ell=1}^s$ in the plane are known and found in the file `stations.mat`. The RSSI (measured in dB) that the mobile unit receives from the ℓth BS at time $n \in \mathbb{N}$ can be modeled as

$$Y_n^\ell = \upsilon - 10\eta \log_{10} \|(X_n^1, X_n^2)^\intercal - \boldsymbol{\pi}_\ell\| + V_n^\ell, \tag{4}$$

where $\| \bullet \|$ denotes the Euclidean distance, $\upsilon = 90$ (dB) is the *base station transmission power*, $\eta = 3$ is the so-called *slope index* and $\{V_n^\ell\}_{\ell=1}^s$ are independent Gaussian noise variables with mean zero and standard deviation $\varsigma = 1.5$ (dB). We denote by $\mathbf{Y}_n = (Y_n^1, \ldots, Y_n^s)^\intercal$ the RSSIs received at time n from all the BSs in the network.

Problem 2

Convince yourself that $\{(\tilde{\mathbf{X}}_n, \mathbf{Y}_n)\}_{n\in\mathbb{N}}$ forms a *hidden Markov model* (see Lecture 5) and find the transition density $p(\mathbf{y}_n \mid \tilde{\mathbf{x}}_n)$ of $\mathbf{Y}_n \mid \tilde{\mathbf{X}}_n$.

Mobility tracking using SMC methods

The file `RSSI-measurements.mat` contains a stream $\mathbf{y}_{0:m} = (\mathbf{y}_0, \ldots, \mathbf{y}_m)$, $m = 500$, of RSSI measurements on a moving target with an unknown trajectory. Our aim is to estimate, by processing the measurements in succession, the positions of the target by means of *optimal filtering*. This means that we wish to estimate sequentially the expectated positions

$$\tau_n^1 = \mathbb{E}\left[X_n^1 \mid \mathbf{Y}_{0:n} = \mathbf{y}_{0:n}\right] \quad \text{and} \quad \tau_n^2 = \mathbb{E}\left[X_n^2 \mid \mathbf{Y}_{0:n} = \mathbf{y}_{0:n}\right]$$

for $n = 0, 1, 2, \ldots$ Since the model is nonlinear, we will in this project apply sequential Monte Carlo (SMC) methods for this purpose. More specifically, we will evolve a particle sample $\{(\tilde{\mathbf{X}}_{0:n}^i, \omega_n^i)\}_{i=1}^N$ targeting sequentially, as new measurements appear for $n = 0, 1, 2, \ldots$, the densities

$$f(\tilde{\mathbf{x}}_{0:n} \mid \mathbf{y}_{0:n}) = \frac{f(\tilde{\mathbf{x}}_{0:n}, \mathbf{y}_{0:n})}{f(\mathbf{y}_{0:n})} = \frac{q(\tilde{\mathbf{x}}_0) p(\mathbf{y}_0 \mid \tilde{\mathbf{x}}_0) \prod_{k=1}^n p(\mathbf{y}_k \mid \tilde{\mathbf{x}}_k) q(\tilde{\mathbf{x}}_k \mid \tilde{\mathbf{x}}_{k-1})}{f(\mathbf{y}_{0:n})} \tag{5}$$

of the *smoothing distributions* $\tilde{\mathbf{X}}_{0:n} \mid \mathbf{Y}_{0:n}$, where $q(\tilde{\mathbf{x}}_k \mid \tilde{\mathbf{x}}_{k-1})$ and $q(\tilde{\mathbf{x}}_0)$ denote the transition density and initial distribution of $\{\tilde{\mathbf{X}}_n\}_{n\in\mathbb{N}}$, respectively. The denominator $f(\mathbf{y}_{0:n})$ in (5) is intractable.[1] In other words, we will solve a larger problem, as the positions (X_n^1, X_n^2) of interest consitute only a part of the vector $\tilde{\mathbf{X}}_n$. Then, since

$$\tau_n^1 = \int x_n^1 f(x_n^1 \mid \mathbf{y}_{0:n})\, dx_n^1,$$

where the *filter density* $f(x_n^1 \mid \mathbf{y}_{0:n})$ is the marginal of (5) w.r.t. the x_n^1 component, we may use the components $\{X_n^{1,i}\}_{i=1}^N$ of the last particle generation $\{\tilde{\mathbf{X}}_n^i\}_{i=1}^N$ for approximating τ_n^1 (and similarly for τ_n^2).

[1] Note that $f(\mathbf{y}_{0:n})$ is given by the integral (w.r.t. $\tilde{\mathbf{x}}_{0:n}$) of the numerator of (5).

Problem 3

Implement the *sequential importance sampling* (SIS) *algorithm* for sampling from $f(\tilde{\mathbf{x}}_n \mid \mathbf{y}_{0:n})$, $n = 0, 1, \ldots, m$, for the observation stream in `RSSI-measurements.mat` and provide estimates of $\{(\tau_n^1, \tau_n^2)\}_{n=0}^m$.[2] Use the prior dynamics $q(\tilde{\mathbf{x}}_n \mid \tilde{\mathbf{x}}_{n-1})$ as proposal kernel. Plot the estimates in the plane together with the locations of the basis stations. In order to keep the computational time at a minimum, you should vectorize the algorithm as far as possible. In particular, since you should be able to run the algorithm for a large particle sample size, say, $N = 10{,}000$, you should avoid any `for`-loop on the particle level. Plot histograms of the importance weights and compute the *efficient sample sizes* at some selected time points. Conclusion?

Problem 4

Implement the *sequential importance sampling with resampling* (SISR) *algorithm* for sampling from the same flow of densities by adding a selection step to the algorithm designed in Problem 3. Provide again, using at least $N = 10{,}000$ particles, a plot of the estimated expected positions $\{(\tau_n^1, \tau_n^2)\}_{n=0}^m$ for the given data stream. Also this implementation should be vectorized as far as possible. Conclusion?

SMC-based model calibration

In a more realistic scenario, the parameters of the model are unknown and needs to be estimated (the standard deviation in the observation noise). In this problem we assume that all paramters except ς has been previously calibrated. The file `RSSI-measurements-unknown-sigma.mat` contains another RSSI data stream $\mathbf{y}_{0:m}$, with again $m = 500$, measured on a target with unknown $\varsigma \in (0, 3)$. In that case, one way of calibrating the ς is to maximize the normalized *log-likelihood function* $\varsigma \mapsto \ell_m(\varsigma, \mathbf{y}_{0:m}) = m^{-1} \ln L_m(\varsigma, \mathbf{y}_{0:m})$, where the likelihood $L_m(\varsigma, \mathbf{y}_{0:m}) = f_\varsigma(\mathbf{y}_{0:m})$ is the normalizing constant of the smoothing distribution (5). However, for the complicated model under consideration, $L_m(\varsigma, \mathbf{y}_{0:m})$, and thus $\ell_m(\varsigma, \mathbf{y}_{0:m})$, is intractable.

Problem 5

Perform approximative maximum likelihood estimation of ς on the bases of the RSSI data record $\mathbf{y}_{0:m}$ given in `RSSI-measurements-unknown-sigma.mat` by, first, computing using your SISR algorithm in Problem 4, pointwise Monte Carlo estimates $\ell_m^N(\varsigma_j)$ of the log-likelihood over a well-designed grid $\{\varsigma_j\}$ in the parameter space $(0, 3)$ and, second, maximizing the approximate log-likelihood by picking the ς_j corresponding to the largest $\ell_m^N(\varsigma_j)$. Note that the pointwise estimates are obtained by running, for the given data input $\mathbf{y}_{0:m}$, the SISR algorithm for each of the different ς values of the grid and estimating the log-likelihood for each run. Denote by $\widehat{\varsigma}_m$ the approximate maximum likelihood estimate obtained in this way. Finally, report $\widehat{\varsigma}_m$ as well as the estimated expected positions $\{(\tau_n^1, \tau_n^2)\}_{n=0}^m$ produced under $\widehat{\varsigma}_m$.

Good luck!

[2] When solving this problem, you are highly recommended to work initially on an artificial data record $\mathbf{Y}_{0:m} = \mathbf{y}_{0:m}$ generated by yourself and for which you know the corresponding true states $\tilde{\mathbf{X}}_{0:n}$ exactly; in this way you are able to check the correctness of your algorithm.