LINMA1731 - Project 2019

Fish schools tracking

`https://youtu.be/dkP8NUwB2io`

A lot of fish species spend most of their life schooling, i.e., *swimming in the same direction (polarization) and with approximately the same speed as the rest of the group. This synchronized group behavior has a lot of advantages such as an improved efficiency to escape from predators and energy saving [1, 2]. Such social behavior does not require a leader or external stimuli to form a school [2]: each individual acts according to the estimated average speed and orientation of the entire group. As mentioned by Niwa [2], «* the entire group is the leader and an individual is a follower *».*

The goal of this project is to track one or several individuals of a fish school. This work will be divided into two parts. In the first part, you will derive estimators for parameters characterizing the fish trajectories. In the second part, you will implement a particle filter enabling you to track the fish individually using the parameter estimates.

Part I - Average speed estimation

For the purpose of this project, we assume that the speed of each fish in a school at time step i is a random variable V_i following a Gamma distribution, as suggested in [1]. This distribution is characterized by two parameters: a shape parameter $k > 0$ and a scale parameter $s > 0$. The parameters are the same for every fish and are time invariant. The aim of this first part is to estimate these two parameters using empirical observations [1] v_i.

Let v_i be i.i.d. realisations of a random variable following a Gamma distribution $\Gamma(k, s)$ (with $i = 1, \ldots, N$). We first assume that the shape parameter k is known.

a) Derive the maximum likelihood (ML) estimator of $\theta := s$ based on N observations.

b) Show that this estimator is:

- (asymptotically) unbiased ;
- efficient ;
- asymptotically normal ;
- consistent.

Let $V_i \sim \Gamma(k, s)$ (for $i = 1, \ldots, N$) with both k and s now unknown.

c) Explain theoretically how to jointly derive ML estimators for the parameters of the distribution, *i.e.*, how to derive ML estimator for $\theta := \begin{pmatrix} k & s \end{pmatrix}^T$ from N observations. Detail your reasoning and define all unusual functions that you may need.

d) Generate N i.i.d. samples of random variable $V_i \sim \Gamma(1, 2)$. Compute estimates $\hat{k}_N$ and $\hat{s}_N$ for different values of N (*e.g.*, from $N = 1$ to $N = 1000$) using (*i*) the method of moments and (*ii*) the ML method described in question (c). Repeat M times the experiment (*e.g.*, $M = 500$). Plot the mean curves (with standard deviation) of $\hat{k}_N$ and $\hat{s}_N$. Comment on the results. How could you improve the ML method using the method of moments?

1. In statistics, a more appropriate term for *estimation of the parameters of a model* is *identification*.

Now, we would like to check if the variance of the ML estimators described in (c) asymptotically tends to the Cramér-Rao lower bound when N increases.

e) Derive the analytical expression of the Fisher information matrix $\mathbf{I}(\theta)$ for the Gamma distribution as well as the associated theoretical Cramér-Rao lower bound.

f) Based on the experiment in (d), compute the statistical (*i.e.*, empirical) covariance matrix of the estimators, namely $\mathbf{Cov}(\hat{\theta}_N)$. Show numerically that the estimator is efficient, *i.e.*, that the ratio of the elements of matrices $\mathbf{Cov}(\hat{\theta}_N)$ and $\mathbf{I}^{-1}(\theta)$ tends to 1 when N increases. You can take for instance $N = 10$, 50, 150 and 3000 and repeat the experiment $M = 10^4$ times.

Part II - Particle filtering

We now consider a school of P fish located in an aquarium. A predator (*e.g.*, a shark) is introduced in this aquarium and attempts to attack the fish school! The objective of the second part of the project is to implement a particle filter to track the positions of each animal based on noisy observations.

State model

For the sake of simplicity, the aquarium will be represented as a rectangle in $\mathbb{R}^2$ and all the animals as points in this rectangle. Every fish p (with $p = 1, \dots, P$) moves in the aquarium according to the same discrete state model:

$$\begin{cases} \boldsymbol{x}_p(i+1) & = \boldsymbol{x}_p(i) + v_i^{(p)} \boldsymbol{o}_p(i)\, t_s, \\ \boldsymbol{o}_p(i+1) & = \begin{bmatrix} \cos\alpha_p(i+1) & -\sin\alpha_p(i+1) \\ \sin\alpha_p(i+1) & \cos\alpha_p(i+1) \end{bmatrix} \boldsymbol{o}_p(i) \end{cases}$$

where $\boldsymbol{o}_p(i)$ is the orientation of fish p (unitary norm) at time index i (with $i = 1, \dots, N$), $\boldsymbol{x}_p(i)$ is its position, $V_i^{(p)} \sim \Gamma(k, s)$ as described in Part I, and t_s is the sampling period.

Turning angle $\alpha_p(i)$ of fish p at time index i is computed as a function of the fish position relatively to the positions of the predator, the other fish, and the boundaries of the aquarium. The update rules for the turning angle are detailed in [1].

The state model of the predator is the same as for the fish, except that its turning angle is computed according to different rules and that its speed is deterministic and constant.

In the project folder on Moodle, you will find a .zip file called `LINMA1731_2019_project_template`. This file contains a function called `StateUpdate` that implements the state model here above.

Estimation of k and s from noisy measurements

Download the file `noisy_observations.mat` available in the project folder (.zip file on Moodle). This file contains N noisy measurements of the trajectory of one fish (2D coordinates). We assume an additive Gaussian noise with i.i.d. realisations. This trajectory has been recorded with a sampling period $t_s = 0.1$ s.

a) Explain how you proceed to estimate the speed v_i of the fish at time index i (with $i = 1, \dots, N-1$) from the noisy measurements. Briefly discuss the influence of the noise on the resulting estimate. Using the speed estimations and the method developed in the first part of the project, estimate parameters k and s of the Gamma distribution by defining function `EstimateGamma` in the project template. Give their numerical values. In the rest of this project, we assume that the speed of every fish is Gamma distributed with parameters $\hat{k}_N$ and $\hat{s}_N$.

Particle filtering

To track animals using a particle filter, observations of their trajectories must be available. In the .zip file on Moodle, you will find a function called `GenerateObservations`. This function iteratively calls `StateUpdate` to generate trajectories of P fishes and of a predator. Additive Gaussian noise is then added to every coordinate to simulate the noisy measurements (because in practice, the true and exact trajectories are unknown). The observation model is hence given by

$$\boldsymbol{y}_p(i) = \boldsymbol{x}_p(i) + \boldsymbol{n}_p.$$

where $\boldsymbol{n}_i$ is the realisation of an additive Gaussian noise with zero-mean and variance equal to σ_{obs}^2 and $\boldsymbol{y}_p(i)$ is a vector containing the noisy position of fish p at time i. It is assumed that the realisations of the noise are independent from each other. The same observation model is used for the predator.

b) Write a function called `ParticleFilter`. The function tracks the animals based on the noisy measurements produced by `GenerateObservations`. You will here implement a sequential Monte Carlo filter. This type of particle filter is described in [3, §10.1 to §10.4]. A small Matlab example is available on Moodle to help you. To build the function, you can also adapt the code contained in `StateUpdate`.

   Input arguments:

   - `param`, a structure containing simulation parameters. Its fields are described in detail in the project template.

   Output arguments:

   - `x_est`, $P \times 2 \times N$ table containing the filtered (*i.e.*, estimated) positions $\boldsymbol{x}_{\text{est}}$ of the fish at every time snapshot ;
   - `xe_est`, $1 \times 2 \times N$ table containing the filtered positions of the predator at every time snapshot.

   Describe in details your particle filter in the written report. This description must contain the different steps of the Monte Carlo filter. In addition, the implemented function must be well commented and submitted on Moodle with the report.

The last three questions are dedicated to the performance evaluation of your filter. The mean square error (MSE) of the particle filter applied on P fish is defined as

$$E_{\text{MSE}} = \frac{1}{P}\frac{1}{N}\sum_{p=1}^{P}\sum_{i=1}^{N} \|\boldsymbol{x}_p(i) - \boldsymbol{x}_{\text{est},p}(i)\|_2^2.$$

For the experiments, use $P = 3$ and $N = 100$. The number of particles in your filter is denoted N_p.

c) Run `ParticleFilter` for several (and well chosen) values of N_p. Repeat the experiment $M = 100$ times for each value of N_p.

d) Run `ParticleFilter` for several (and well chosen) values of the sampling period t_s. Repeat the experiment $M = 100$ times for each value of t_s.

e) Run `ParticleFilter` for several (and well chosen) values of the standard deviation σ_{obs}. Repeat the experiment $M = 100$ times for each value of σ_{obs}.

For each experiment, (i) plot the average MSE of the particle filter as function of the parameter and (ii) comment on its influence on the MSE.

Practical details

Teaching assistants	Charles Wiame and Stéphanie Guérit
Modality	The project is carried out by groups of two students. If you need to work alone or do not know anybody to work with, please contact us to find an arrangement. Each group must register on Moodle by Friday 10 March 2019, 11.59 pm.
Supervised sessions	Week 9 (usual schedule for exercises sessions) and Week 12 (Tuesday and Thursday, both days from 1.00 pm to 2.00 pm in BARB 92 room).
Report	French or English, 10 pages maximum. Please *do not answer separately to each question*, but write an integrated, self-contained report.
Deadline	Part I : Wednesday 24 April 2019 at 11.59 pm on Moodle plateform. For the mid-term deadline, you just need to submit the report of the first part and not the code (in pdf, with the filename `LINMA1731_2019_Project_Part1_NAMEstudent1_NAMEstudent2.pdf`). Part II : Friday 17 May 2019 at 11.59 pm on Moodle plateform. The report (in pdf format) and the code (Matlab, Julia or Python) will be submitted together in a zip file named `LINMA1731_2019_Project_Part2_NAMEstudent1_NAMEstudent2.zip`. The code consists of your two functions `EstimateGamma` and `ParticleFilter`. These functions will be tested using the provided main script: pay attention to the inputs and the outputs!
Evaluation	Evaluation criteria are available on Moodle website in the submission modules.

References

[1] Huth, A., and Wissel, C. The Simulation of the Movement of Fish Schools. *Journal of Theoretical Biology 156*, 3 (1992), 365–385.

[2] Niwa, H.-S. Self-organizing Dynamic Model of Fish Schooling. *Journal of Theoretical Biology 171* (1994), 123–136.

[3] Srivastava, A. *Computational methods in statistics.* 2009.