Assignment 6

Swarm Intelligence

Deadline March 17, 2024

Handout for the *Natural Computing* lecture, March 5th, 2024

Johannes Textor, Inge Wortel, Koert Schreurs

In the lecture on swarm intelligence, you saw the Boid simulation model for swarming behavior [1]. And in the second lecture on evolutionary computation (Evolutionary Strategies in practice), you were introduced to the (sequential) ABC algorithm [2], which can be used to fit parameters of a simulator model to a target behaviour. You may have noticed that the Boid simulation is such a simulation model. So can we use ABC to find Boid parameters that give us some target behaviour? Let's see!

Objectives of This Exercise

1. Implement the boid simulation model of crowd motion
2. Implement the (sequential) ABC evolutionary algorithm
3. Apply the ABC algorithm to find boid parameters that yield desired behaviour
4. Analyze posterior parameter distributions and overall swarming behaviour of a fitted boids model.

Exercises

Exercise 6.1 Implement the Boids model

1. Implement your own version of Craig Reynold's Boid Algorithm for swarm motion [1], or use the example JavaScript implementation provided in the lecture. There are multiple ways you can implement Boids, but please make sure that your implementation:
   - ... is able to generate text output (for example, positions of Boids over time) for use later. If you use the JavaScript implementation, you can for instance achieve that using node.js or deno, or you can log output in an HTML textbox or similar to copy-paste into a text file.
   - ... keeps Boid speeds fixed; agents should update their direction but move at a constant speed v which is a parameter of your model.
   - ... explicitly considers what should happen at the boundary of your simulation field.
2. Analyze your implementation visually (for example by creating an animation showing Boid positions and directions over time) and quantitatively. For quantitative analysis, we suggest you look at:
   - the *order parameter* over time. The order parameter is the average normalized velocity of the Boids:

     $$O = \frac{1}{N_B} \left\| \sum_{i=1}^{N_B} \frac{\vec{v}_i}{||\vec{v}_i||} \right\| \tag{1}$$

     where the outer norm ensures that we get one non-negative value and the inner normalization ensures it lies between 0 and 1.
   - the distribution of *nearest-neighbor distances* over time, which should tell you when individuals get separated from the swarm and/or when individuals get too close to each other.
3. Using the analyses above, play with the Boid parameters (cohesion, separation, and alignment strength; exclusion and interaction radius) to verify that it is possible to reach high alignment in a run of about 300 steps (Note: make sure that the movement speed, overall Boid density, and interaction radii are in proportion to each other).
4. In such a simulation, what happens to the nearest-neighbor distance and alignment over time? Explain your observations (and don't forget to report which parameters you chose for this result).

Exercise 6.2 Evolve Boid parameters using sequential ABC

1. Using the abovementioned order parameter as a measure of Boid "alignment", implement the sequential ABC algorithm to fit the cohesion, alignment, and separation strength parameters such that, over time, boids achieve maximum alignment (i.e. an order parameter close to 1). Notes:
   - each "individual" in your ABC "population" will now consist of one Boid simulation which may contain multiple Boids.
   - Use an ABC population of $N = 20$ (accepted) Boid simulation parameters. Run simulations with each $N_B = 15$ Boids per simulation, for 300 steps.
   - For the fixed speed v, interaction radius and exclusion radius, please choose fixed values that make sense to you (they should be proportionate with respect to your field size and each other).
   - Make sure to save the accepted parameters after each iteration of ABC, so you can analyze them later.
   - You will have to decide for yourself how to set the prior, thresholds ϵ_τ, field size, boundary conditions, and initial conditions. There are multiple possible choices here, but note that if your ABC takes very long and/or does not yield a clear output distribution, then something may have gone wrong...
2. At several "generations" of your ABC, show both the (marginal) distribution of accepted parameters and the distribution of the alignment (fitness) values.
3. Choose 3 parameter sets from your ABC fitness landscape and perform some additional Boid simulation runs and analyze what happens over time (for example by assessing them visually and checking alignment and nearest-neighbor distance over time). Do the simulations robustly reach the target alignment throughout multiple runs? How different are these dynamics between the three selected parameter sets?
4. Repeat the above, but now let your Boids evolve a target alignment of 0.6 instead of 1. Is it easier or harder to achieve an alignment of 0.6 compared to an alignment of 1?
5. Analyze your final posterior distributions in more depth. In what range of the parameter landscape do you see alignment, and how is this different for a target alignment of 1 versus 0.6? Are there correlations between the parameters, and does this make sense? Can you distinguish different "strategies" in the posterior distribution? Do you see other differences? Explain your observations.
6. (Optional): so far, you have fitted simulations on alignment only. Can you design a fitness function that more strongly affects the cohesion and separation of your Boids? Try ABC again to see how this changes your posterior parameter range.

Product

Write a small report of 4-5 pages (in pdf) about this assignment. As always, this should contain:

- A (brief) introduction with a problem statement;
- A Methods section where you verbally describe what you did in sufficient detail that others can reproduce your work. Note that your specific conclusions may well depend on implementation details (such as how you chose your priors, your initial conditions, analysis choices, etc etc), so make sure to describe those implementation details clearly. Also include a link to your implementation code on Github or Gitlab, but it should be possible to understand what you did from reading the Methods alone.
- A Results section including figures, screenshots, and a written description of your observations. Here, you should first demonstrate that your implementation is valid (first exercise) before continuing to fit the parameters using ABC.
- Be sure to come to a final conclusion and discuss some potential limitations as well.

Feel free to double-check yourself using the Peer feedback rubric from assignment 2: the specific choices will be different for this assignment, but many of the principles remain the same.
Hand in your report in pdf format on Brightspace; deadline: March 17th, 2024.

References

[1] C. W. Reynolds (1987) *Flocks, herds and schools: A distributed behavioral model*, In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87, page 25–34, New York, NY, USA, 1987. Association for Computing Machinery.

[2] Sisson, Fan, and Tanaka (2007) *Sequential Monte Carlo without likelihoods*. PNAS, 104 (6) 1760-1765. doi: `https://doi.org/10.1073/pnas.0607208104`.