CAAM/STAT 31310, Autumn 2024, U Chicago

CAAM 31310: Homework 5

Due Dec 7th, '24 (11:59 pm ET) on Gradescope

Cite any sources and collaborators; do not copy. See syllabus for policy.

In this last homework, we will work on formulating and solving computational problems using a variation of the two-player election model from Borgers et al 2024. Let the position (a scalar quantity) of the voting public to be distributed according to a Gaussian mixture,

$$\rho(x) = \frac{1}{2}\frac{1}{s_1\sqrt{2\pi}}e^{-\frac{(x-m_1)^2}{s_1^2}} + \frac{1}{2}\frac{1}{s_2\sqrt{2\pi}}e^{-\frac{(x-m_2)^2}{s_2^2}}.$$

The two candidates (players) contesting the election dynamically update their positions in order to win. A player wins if at time 1 if they have a vote share strictly greater than 50%. The vote share of the left player is given by the CDF of the distribution of the positions of the voting public at the average position of the players. That is, let the two players be L and R. Then, vote share of L is given by

$$S_L(L,R) = \int_{-\infty}^{(L+R)/2} \rho(x)\, dx.$$

By construction, the vote share of R is $1 - S_L$. Now consider the following dynamics of the positions of the players (the public positions do not change).

$$\frac{d}{dt}\begin{bmatrix} L \\ R \\ v_L \\ v_R \end{bmatrix} = \begin{bmatrix} v_L + (\partial S_L/\partial L) \\ v_R + (\partial S_R/\partial R) \\ F_\sigma(|L-R|) \\ -F_\sigma(|L-R|) \end{bmatrix}. \tag{1}$$

The *force* $F_\sigma(L,R)$ pulls the players' positions together if they are at a moderate distance from each other, and apart if they are too close ($< 2^{(1/6)}\sigma$). Here we use

$$F_\sigma(x) = -4\frac{d}{dx}\left(\frac{\sigma^{12}}{x^{12}} - \frac{\sigma^6}{x^6}\right),$$

which comes from the widely used Lennard-Jones interatomic potential. We will use the velocity Verlet scheme to time-integrate the above second-order ODEs for L and R. You can use the accompanying script, `voting.py`, to simulate the ODEs above and play with the parameters m_1, m_2, s_1, s_2 and σ.

Our dynamical system, $\varphi : M \to M$, will be a non-Markovian map that time-integrates L and R with timestep 0.001. The map φ is the function `step` in the script restricted to the first two coordinates. That is, $[L,R] \to \varphi([L,R])$ is the numerical solution using the above scheme (defined in `step`) with initial condition $[L,R,v_L,v_R]$, for some fixed v_L, v_R,

after one timestep. Note that φ is a composition of the time-integrator with nonlinearities, $N_1 \circ N_2$, where $N_1(x) = x \bmod (m_1 - s_1)$ and $N_2 = x \bmod (m_2 + s_2)$, applied to the positions L and R. Thus, $M \equiv [m_1 - s_1, m_2 + s_2]^2$. Intuitively, when either player far exceeds the average positions of the public, they map back to regions of popular public positions. As in the code, set $m_1 = -1$, $m_2 = 1$, and $s_1 = s_2 = 1$.

I When $\sigma = 0$, prove that the vote shares, S_L and S_R are constant along orbits of φ. (1 point)

II When $\sigma = 0$, prove that there are multiple invariant ergodic measures (2 points).

III Write down a code snippet to differentiate S_L at timestep 1000 (or time 1) with respect to σ. (3 points) Plot this quantity and interpret your results for $0 \leq \sigma \leq 1$. (3 points)

IV Let the initial conditions be chosen uniformly on $M \times [0, 0]$. Explain how to use the derivative from Part III to find a σ that maximizes the probability of one of the players, say L, winning (the randomness is over the initial conditions for L and R.) (2 points)

V We will now write an *adjoint* version of the above derivative. Suppose we want to optimize all the parameters together, $m_i, s_i, i = 1, 2$ and σ such that the probability of L winning is maximized. Modify the code (and submit a pseudocode/code snippet) from Part IV so that adjoint vectors are time-integrated only once in order to differentiate with respect to all 5 parameters. (5 points)