AMSC808N/CMSC828V Maria Cameron

Final exam. Due Monday, December 19, 2022, 11:59 PM.

Contents

This final exam is project-style.

Resources: *You can use any internet resources, textbooks, and course materials. You are not allowed to collaborate with your classmates or use anybody's help.*

Programming: *You can use any suitable language. A high-level language (e.g. Matlab or Python) is preferable. You are allowed to use built-in functions and available libraries for graph algorithms.*

Submission: *You should upload on ELMS a single pdf file with your codes linked to it. For example, you can publish Matlab's code or make a Jupiter notebook and link them to your pdf. Use latex or any other suitable text editor. I will subtract 10% of the maximal score if the file is hand-written.*

1 Project 1. Multiple discriminant analysis

Multiple discriminant analysis (MDA) is a linear dimensional reduction method aiming at projecting data from different categories to a low-dimensional space so that the images of data from different categories are separated as much as possible. See lecture notes `4-DimReduction.pdf`, Section 5.

Consider the following test system called LJ7 in 2D. Seven two-dimensional particles interact according to the Lennard-Jones pair potential. The particles are also placed in a rather small box in order to prevent them from going far apart from each other. Their dynamics are governed by the overdamped Langevin equation. This system has four distinct metastable states corresponding to four potential energy minima shown in Fig. 1: hexagon, trapezoid, capped parallelogram 1, and capped parallelogram 2.

This system is 14-dimensional: there are 7 particles and each of them is described by two coordinates (x, y). One standard way to reduce the dimension of this system is by means of physically motivated collective variables $\mu_2(x_1, y_1, \ldots, x_7, y_7)$ and $\mu_3(x_1, y_1, \ldots, x_7, y_7)$ defined in a rather complicated way (if you want to look up the precise definitions of μ_2 and μ_3, see Section 4.2.1 on page 18 in arXiv:2108.08979). Importantly, the functions μ_2 and μ_3 are invariant with respect to translations and orthogonal transformations of the space (x, y), and permutations of particles, and the mapping of $(x_1, y_1, \ldots, x_7, y_7)$ to the (μ_2, μ_3) space separates the four metastable states. The free energy[1] with respect to μ_2 and μ_3 is shown in Fig. 1.

The goal of this project is to construct another set of two collective variables for this system using MDA. Specifically, the goal is to design a set of collective variables that is

- (A) invariant with respect to translations and orthogonal transformations of the space (x, y), and permutations of particles,
- (B) and that separates the four metastable states as much as possible.

[1] Courtesy of Luke Evans, AMSC graduate student.

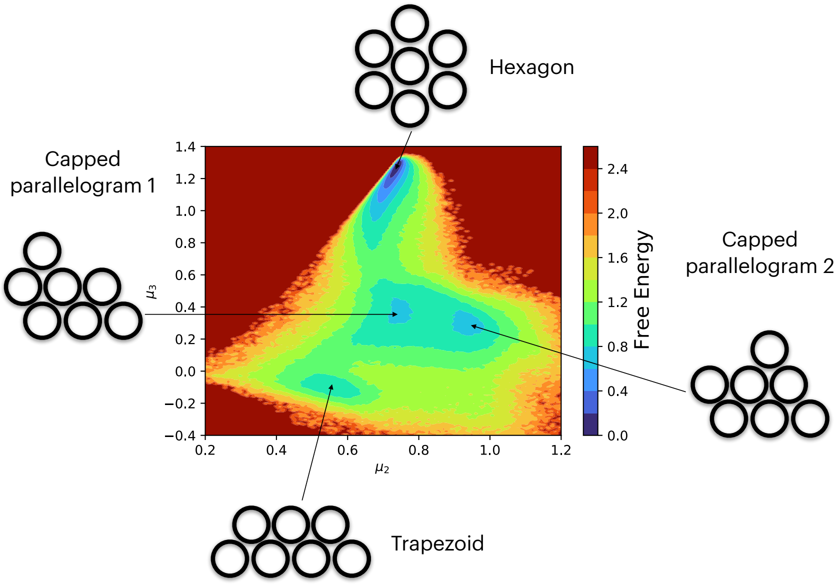

Figure 1: Four metastable states of the system consisting of 7 2D Lennard-Jones particles governed by overdamped Langevin dynamics.

To satisfy (A) and (B), one can first map the data points $(x_1, y_1, \dots, x_7, y_7)$ by a set of functions $(z_1, \dots, z_m)$ each of which satisfies (A), and then apply MDA to the data matrix Z to achieve (B).

1. Let S_b and S_w be the between-class and within-class scatter matrices. Assume that the within-class scatter matrix S_w is nonsingular. Reduce the problem of finding the maximizer of the functional
$$J(w) = \frac{w^\top S_b w}{w^\top S_w w}$$
to a generalized eigenvalue problem. Then use the Cholesky decomposition of S_w, $S_w = LL^\top$, to reduce the generalized eigenvalue problem to a symmetric eigenvalue problem of the form $Ay = \lambda y$.

2. Let Z be a data matrix of size $n \times d$. Write out the formula for the projection of the matrix Z onto the two-dimensional space in terms of the found solution to the eigenvalue problem and the matrix L.

3. I ran a long trajectory of 10^7 steps of Euler-Maruyama with time step $5 \cdot 10^{-5}$ and recorded the following data at every 1000th step:
   - The (μ_2, μ_3) coordinates of each data point are saved to the file `TrajectoryCV_data.csv`, $10^4 \times 2$ array. This array allows us to visualize the recorded points in the (μ_2, μ_3)-space – see black dots in Fig. 2.
   - The set of the following functions $z_k(x_1, y_1, \dots, x_7, y_7)$ satisfying requirement (A). Let Δ be the matrix of distances squared:
$$\Delta_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2, \quad 1 \le i, j \le 7.$$

Then we define the vector z with components k as

$$z_k = \sum_i \sum_j e^{-\frac{\Delta_{ij}}{2\sigma_k^2}}, \quad \sigma_k = 1 + 0.1k, \quad 0 \le k \le 19.$$

The $10^4 \times 20$ matrix Z with rows z is saved to the file `Trajectory_data.csv`.

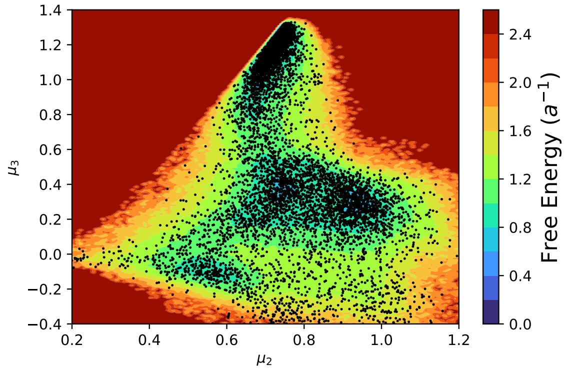

Figure 2: The data points shown in the coordinates (μ_2, μ_3).

Select training data for MDA. You can use any criterion to select data from the vicinity of the four metastable states (for example, surround them with circles or squares). If your S_w matrix turns out to be singular, regularize it by adding a small multiple of the identity matrix to it.

Find the two directions w_1 and w_2 and the corresponding eigenvalues. Print them out. Project the full set of Z-data on them, i.e., obtain two $10^4 \times 1$ vectors $u_1 = Zw_1$ and $u_2 = Zw_2$. Linearly rescale u_1 and u_2 onto the interval $[0, 1]$. Visualize u_1 and u_2 as follows. In the first figure, plot the level sets of the free energy using the contour plot and plot the points from the file `TrajectoryCV_data.csv` over them coloring the points according to the values of u_1. In the second figure, do the same, but color the points according to the values of u_2.

In order to plot the level sets of the free energy, you will need the data file `LJ7FreeEnergy.csv`, a 401×201 array with grid data for the free energy in (μ_2, μ_3). For the data in this array: $0.2 \le \mu_2 \le 1.2$, $-0.5 \le \mu_3 \le 1.7$.

2 Project 2. Analysis of a real-world network

You will need to investigate a real-world network: the PGP web of trust network (2009) available at https://icon.colorado.edu/#!/networks. Ref.: M. Boguna, R. Pastor-Satorras, A. Diaz-Guilera, and A. Arenas, Physical Review E, vol. 70, 056122 (2004).

This network is undirected and unweighted.

The number of vertices is 10680.

The number of edges is 24316.

The indices of vertices start with 1.

For your convenience, I created a CSV file `PGPedges.csv` containing the list of edges. This is a 24316×2 array.

1. Use the depth-first search algorithm to verify that the PGP network is connected.

2. Find the degree distribution for this network. Plot p_k versus k in log-log scale. You will see something like Fig. 4.22 (top right) in Barabasi, "Network Science", chapter 4. Then use log-binning (like Fig. 4.22 (bottom left)) and plot the result in the same figure. For doing log-binning, use bins b_0 containing only nodes of degree 1, b_1 containing nodes of degrees 2 and 3, ..., b_n containing nodes of degrees $2^n \le k < 2^{n+1}$, etc. The last bin $n_{\max}$ should be such that $n_{\max}$ is the largest integer such that $2^{n_{\max}+1}$ does not exceed the largest degree in the network. The nth point in log-binning has coordinates (the mean degree in the bin b_n, the mean degree probability in bin b_n):

$$\left(\langle k\rangle_n = \frac{\sum_{k=2^n}^{2^{n+1}-1} k p_k}{\sum_{k=2^n}^{2^{n+1}-1} p_k}, \langle p_k\rangle_n = \frac{\sum_{k=2^n}^{2^{n+1}-1} p_k}{2^n}\right). \tag{1}$$

   Approximate the log-binning data for the degree distribution with the power-law with exponential cut-off:

$$p_k = Ce^{-\alpha k}k^{-\tau}, \tag{2}$$

   where C, α, and τ are positive constants that you need to find by solving a linear least squares problem. To set it up, take logs of both sides of (2). Plot the approximation to the degree distribution that you find in the same figure as in the previous item. Include legend.

3. Find the average shortest-path length in the actual network. The use of a built-in or a library function for this task is preferable. Now imagine a random graph that has degree distribution (2) with parameters that you have found. For brevity, we will refer to it as *the random graph with the same degree distribution.* Estimate the average shortest-path length in this random graph. *Hint: the paper by Newman, Strogatz and Watts should be very helpful.*

4. Find the clustering coefficient C for the actual network defined as

$$C = \frac{\#(\text{closed paths of lengths 2})}{\#(\text{paths of length 2})}. \tag{3}$$

   Now find the clustering coefficient for the random graph with the same degree distribution. Proceed as follows. Let v be an arbitrary node of degree 2 or more. Randomly pick two of its first neighbors i and j. Let their *excess degrees* be k_i and k_j. The probability that there is a link between i and j is $\frac{k_ik_j}{2m}$ where $2m = n\langle k\rangle$ is twice the expected number of edges, and n is the number of nodes. Then the clustering coefficient is equal to the expectation for $\frac{k_ik_j}{2m}$ taken with respect to the joint excess degree distribution for i and j. Due to the independence of excess degree distributions for i and j in the random graph, the joint probability mass function is $q_{k_i}q_{k_j}$. Calculate this expectation and show that it is equal to

$$C_{random} = \frac{1}{n}\frac{\left[\langle k^2\rangle - \langle k\rangle\right]^2}{\langle k\rangle^3}. \tag{4}$$

5. Comment on the relationships between the shortest-path length and clustering coefficient for the actual network and the random graph with the same degree distribution. Try to explain the discrepancies between them.