ME8930: Data-driven Learning and Control of Dynamic Systems: Assignment 2

Assigned: 03/01/2024, Due: 04/01/2024

GitHub repo: https://github.com/clemson-dira/data_driven_learning_course

Problem 1.

Consider the one dimensional dynamical system of the form

$$\dot{x} = -x + x^3.$$

1. Perform the Koopman lifting of system using the basis functions $\{x^{2n+1}\}$ for $n = 0, 1, 2, \ldots, N$.
2. Write a MATLAB code to compute the finite dimensional approximation of the Koopman generator for any finite N.
3. Verify that the eigenfunction of the Koopman generator for this system is given by $\frac{x}{\sqrt{1-x^2}}$.
4. Compute the eigenfunction of the Koopman generator for changing N and compare it with the analytical expression of the eigenfunction for changing N.

Problem 2.

Consider the dynamics of a damped simple pendulum given by

$$\ddot{\theta} + b\dot{\theta} + \frac{g}{L}\sin\theta = 0.$$

Assume the pendulum has unit mass and unit length ($L = 1$) with damping coefficient $b = 2$.

1. Write the above system in a state space form and find the eigenvalues of this system around the stable equilibrium point at $(0, 0)$.
2. Load the training dataset from `training\pendulum_training.csv`. This dataset consists of 126 trajectories collected with a time span of 1 s and a sampling rate of 0.1 s for each trajectory. Visualize a few sample trajectories by plotting a phase portrait (θ vs $\dot{\theta}$).
3. Obtain the finite-dimensional approximation for the Koopman operator using Dynamic Mode Decomposition (DMD). Plot the eigenvalues of the operator in a unit circle. For this, you need to complete the code in `DMD\get_DMD.m` function and call it in the main file (`simple_pendulum.m`) for further analysis.

4. Lift the data using a monomial basis $\psi(\mathbf{x})$ and obtain the finite-dimensional approximation for the Koopman operator using Extended Dynamic Mode Decomposition (EDMD). Note that you have to append the original states $\mathbf{x}$ to obtain $\mathbf{\Psi} = [\mathbf{x}, \psi(\mathbf{x})]$ to obtain $\mathbf{C} = \begin{bmatrix} \mathbb{I}_{n\times n} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$.
Plot the eigenvalues of the operator in a unit circle. For this, you need to complete the `EDMD\get_EDMD.m` function and call it in the main file (`simple_pendulum.m`) for further analysis.

5. Load the validation dataset from `prediction/pendulum_validation.csv`. This dataset consists of 9 trajectories collected with a time span of 10 s and a sampling rate of 0.1 s for each trajectory. Visualize it by plotting a phase portrait.

6. Compare the prediction accuracy of the Koopman operator obtained using DMD and EDMD on the validation dataset using root mean square error (rmse) and % error.

   - DMD: use $\hat{\mathbf{x}}_{t+1} = \mathbf{K}\mathbf{x}_t$. Here $\hat{\mathbf{x}}$ is the one time-step prediction.
   - EDMD: use $\hat{\mathbf{z}}_{t+1} = \mathbf{K}\mathbf{z}_t$. Here $\hat{\mathbf{z}}_t$ is the is the one time-step prediction in the lifted state. To obtain the original state prediction, use $\hat{\mathbf{x}}_t = \mathbf{C}\hat{\mathbf{z}}_t$.

   For this, you need to complete the code in `prediction\eval_prediction.m` function to output `X_pred`, which contains trajectories obtained by stacking $\hat{\mathbf{x}}_t$ for $n = 20$ time-steps. Then, use the `prediction\rmse.m` function to evaluate rmse and % error.

Problem 3.

Consider the dynamics of a duffing oscillator given by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1 - \delta x_2 - x_1^3 \end{aligned}$$

where $\delta = 0.5$.

1. Load the training dataset from `training\duffing_training.csv`. This dataset consists of 100 trajectories obtained from initial conditions sampled using a uniform grid with the domain $[-2, 2]$. Each trajectory is collected with a time span of 1 s and a sampling rate of 0.1 s for each trajectory. Visualize a few sample trajectories by plotting a phase portrait (x_1 vs x_2).

2. Lift the data using a monomial basis $\psi(\mathbf{x})$ and obtain the finite-dimensional approximation for the Koopman operator using Extended Dynamic Mode Decomposition (EDMD). Plot the eigenvalues of the operator in a unit circle.

3. Load the validation dataset from `prediction\duffing_validation.csv`. This dataset consists of 9 trajectories collected with a time span of 10 s and a sampling rate of 0.1 s for each trajectory. Note that all trajectories lie within the basin of attraction of the equilibrium point $(1, 0)$. Visualize it by plotting a phase portrait.

4. Find the prediction accuracy of the Koopman operator obtained using EDMD on the validation dataset using root mean square error (rmse) and % error.

5. Now, load the training dataset from `training\duffing_multi_model_training.csv`. This dataset consists of 100 trajectories that lie within the domain of attraction of the equilibrium point $(1, 0)$. Each trajectory is collected with a time span of 1 s and a sampling rate of 0.1 s for each trajectory. Visualize a few sample trajectories by plotting a phase portrait (x_1 vs x_2).

6. Lift the data using a monomial basis $\psi(\mathbf{x})$ and obtain the finite-dimensional approximation for the Koopman operator using Extended Dynamic Mode Decomposition (EDMD). Plot the eigenvalues of the operator in a unit circle.

7. Compare the prediction accuracy of the two Koopman operators obtained on the same validation dataset using root mean square error (rmse) and % error.

Problem 4.

Consider a dynamical system of the form

$$\dot{x} = \begin{bmatrix} \frac{\left(7.5x_2^2+5.0\right)\left(x_1^3+x_1+\sin(x_2)\right)+\left(-x_1+x_2^3+2x_2\right)\cos(x_2)}{9x_1^2x_2^2+6x_1^2+3x_2^2+\cos(x_2)+2} \\ \\ \frac{2.5x_1^3+2.5x_1-\left(3x_1^2+1\right)\left(-x_1+x_2^3+2x_2\right)+2.5\sin(x_2)}{9x_1^2x_2^2+6x_1^2+3x_2^2+\cos(x_2)+2} \end{bmatrix}. \tag{1}$$

1. Verify that the principal eigenfunction associated with eigenvalues $\lambda_1 = -1$ and $\lambda_2 = 2.5$ are given by

$$\phi_{\lambda_1} = x_1 - 2x_2 - x_2^3, \quad \lambda_1 = -1 \text{ and}$$
$$\phi_{\lambda_2} = x_1 + \sin(x_2) + x_1^3, \quad \lambda_2 = 2.5.$$

2. You may use the `compute_eigfn.m` as the main file and the convex optimization code provided in `eigenfunctions/convex_opt.m` to approximate the principle eigenfunctions of this system and compare with part 1.

3. Compute the zero-level curves of the eigenfunctions to identify the stable and the unstable manifolds of the equilibrium point at the origin.

1 Notes for Assignment

3-4 problems. pick 2 from textbooks find if book has koopman for vanderpol and Lorrentz kind of systems
example 1:
- analytical example: construct system from known eig fun.
- hw will be to find Koopman and find eig fun of Koopman for such systems.

example 2:
- maybe inverted pendulum/ something we know works with DMD and EDMD
- some examples where we know how to tune things

example 3:
- data from different DOA, find Koopman within that DOA and see how good the prediction is
- num of koopman is same as num of eqb points
- verify the prediction accuracy using multi-model vs one model

example 4:
- Use convex optimization or path integral code to find eigenfunctions for analytical example