Assignment 5 - PA.3

Due: Monday, November 13, 2023 10:00 pm (Eastern Standard Time)

Assignment description

The purpose of this assignment is to explore how complex problems, like prediction problems, model calibratio problems, control (guidance) problems, can be solved using least squares, nonlinear least squares, or constrained leas squares. This assignment is programming heavy.

Do not answer questions within `python` code. Any answer, including but not limited to derivations, numerical output from `python` code, plots, etc., must appear in a document (e.g., an MS Word document, a LaTeX document, etc.) tha is submitted in crowdmark.

You must use `python` and you must submit your `python` code. You may not use another programming language such as `matlab`, C++, VB, fortan, etc. All plotting must be done in `python`, and not in MS Excel, google sheets desmos `https://www.desmos.com/calculator`, etc.. Save your code as a pdf and upload it along with you handwritten/typed solutions.

Note, the points allotted to each question is not yet set, and will change upon grading.

Submit your assignment

After you have completed the assignment, please save, scan, or take photos of your work and upload your files to the questions below. Crowdmark accepts PDF, JPG, and PNG file formats.

Q1 (10 points)

Time left Hide
10 days, 8 hours

1. An auto-regressive (AR) model is a time-domain model of a form

$$y_{k+1} = \beta_\ell y_k + \beta_{\ell-1} y_{k-1} + \cdots + \beta_2 y_{k-\ell+2} + \beta_1 y_{k-\ell+1}, \quad (1$$

where $\ell > 0$ is the AR model memory. When ℓ is small the AR model "forgets the past quickly". When ℓ i large the AR model "is slow to forget the past". AR models are used for prediction in various contexts, be i economics or signal processing. An AR model can be "fit" to data using least squares in the following way Consider

$$y_{k+2} = \beta_2 y_{k+1} + \beta_1 y_k, \quad (2$$

where $\ell = 2$. Stacking the data up from $k = 0$ to $k = N$

$$\begin{bmatrix} y_1 & y_0 \\ y_2 & y_1 \\ \vdots & \vdots \\ y_{N-1} & y_{N-2} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix},$$

which is a standard $\mathbf{A}\mathbf{x} = \mathbf{b}$ problem involving a tall $\mathbf{A}$ matrix. The column matrix $\mathbf{b}$ does not necessarily lie i the column space of $\mathbf{A}$, meaning $\mathbf{b} \notin \mathcal{R}(\mathbf{A})$. As such, to solve for β_1 and β_2, simply solve the associated leas squares problem.

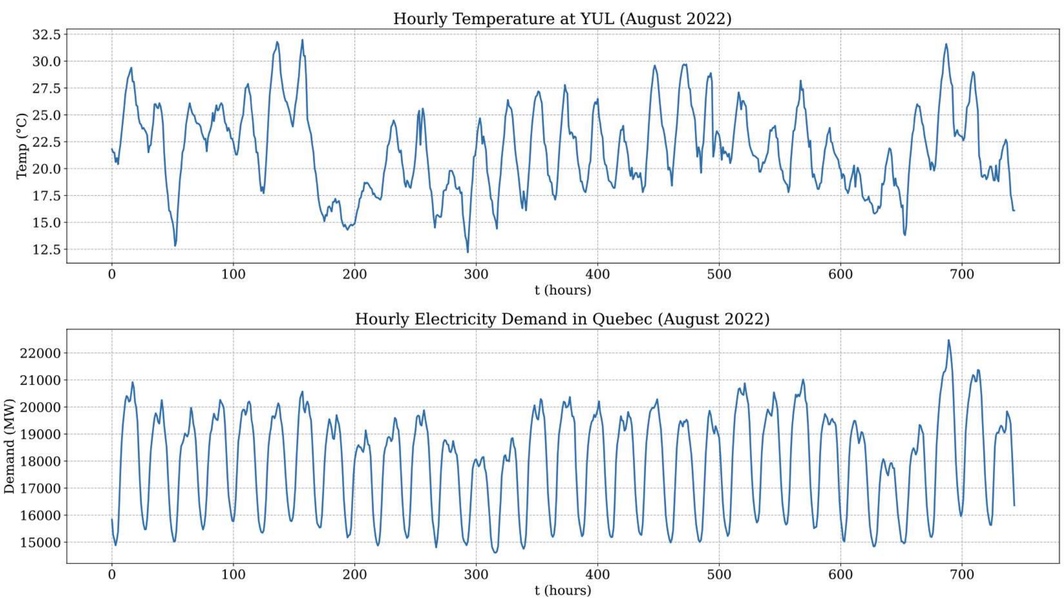


Figure 1: (Top) Hourly temperature data recorded at YUL, 2022-08-01 to 2022-08-31. (Bottom) Hourly electricit demand in Québec, 2022-08-01 to 2022-08-31.

Table 1: The mean and standard deviations of the absolute error e_k and relative error $e_{k,\mathrm{rel}}$.

(a) Hourly temperature prediction.

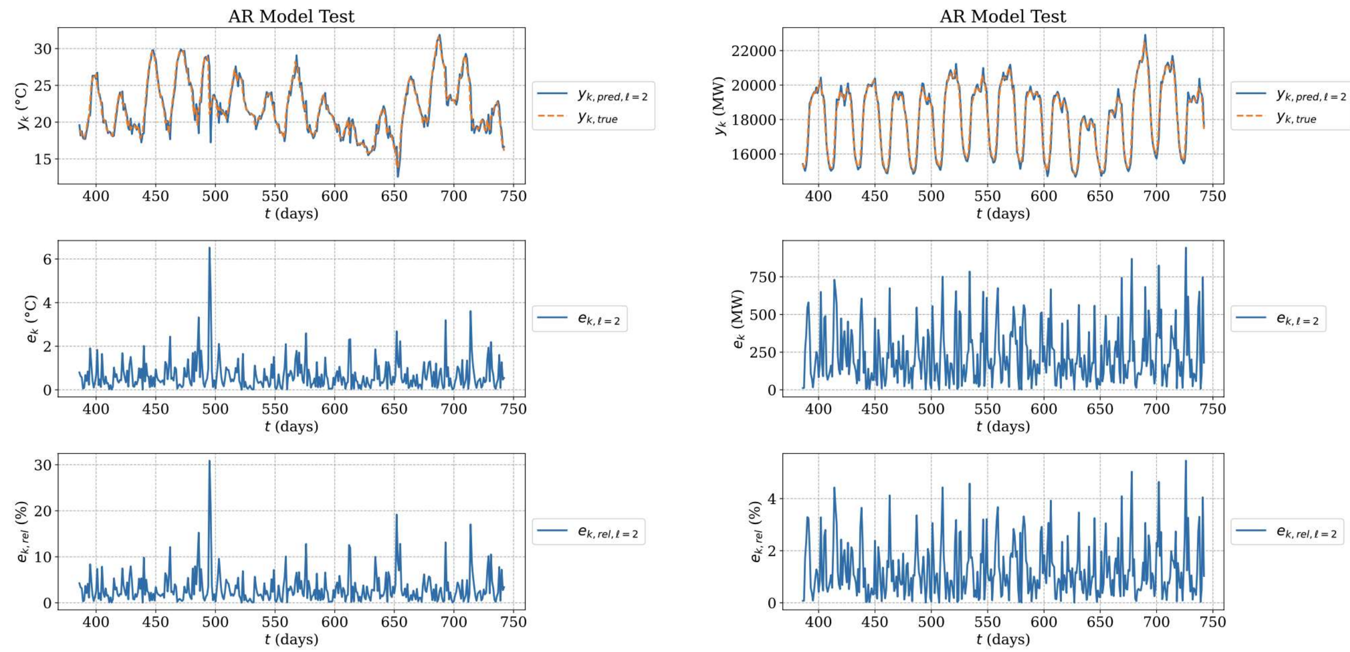
	e_k (°C)	$e_{k,\mathrm{rel}}$ (%)
mean	0.6587	3.0824
standard deviation	0.6905	3.3512

(b) Hourly electricity demand prediction.

	e_k (MW)	$e_{k,\mathrm{rel}}$ (%)
mean	226.61	1.2593
standard deviation	187.51	1.049

The Government of Canada provides historical climate data, such as temperature, dew point, relative humid ity, wind speed, etc., for various locations across Canada.[1] Additionally, Hydro-Québec pr electricity demand data for the province of Québec.[2] Electricity demand is defined as the r consumption, measured in the unit of megawatts (MW).

[1] https://climate.weather.gc.ca/historical_data/search_historic_data_e.html
[2] https://www.hydroquebec.com/documents-data/open-data/history-electricity-demand-quebec/



(a) Hourly temperature prediction.

(b) Hourly electricity demand prediction.

Figure 2: Prediction, error, and relative error for $\ell = 2$, 2022-08-17 to 2022-08-31.

Figure 1 shows the hourly temperature data recorded at Montréal-Pierre Elliott Trudeau International Airpor (YUL) and hourly electricity demand data in Québec for the month of August 2022. The AR model can be usec to predict the data (be it temperature or electricity demand) 1 hour ahead given the data from the previous N hours, which is done by finding β_i's in (1), given $y_1, y_2, \ldots, y_N$.

Two "individual" AR models of the form given in (2) that use $\ell = 2$ are fit to the temperature data and the electricity demand data, respectively, from the 16 days of August 2022, then tested on the last 15 days of Augus 2022. The temperature AR model is able to predict the temperature 1 hour ahead with a mean absolute error o about $\approx 0.66\,°\mathrm{C}$, and the electricity demand AR model is able to predict the electricity demand 1 hour aheac with a mean absolute error of $\approx 227\,\mathrm{MW}$. See Figure 2 and Table 1 for details.

In this question, you will fit a "coupled" AR model using the temperature and electricity demand data to predic the temperature and electricity demand 1 hour in the future given previous temperature and electricity demanc data.

a) Let $\mathbf{y}_k = \begin{bmatrix} T_k & P_k \end{bmatrix}^\mathsf{T}$ with temperature T_k (°C), and hourly electricity demand P_k (MW) at hour k. Giver the fixed AR model memory $\ell = 2$, the multivariate form of AR model in Equation (1) is

$$T_{k+1} = \alpha_2 T_k + \alpha_1 T_{k-1} + \beta_2 P_k + \beta_1 P_{k-1}, \tag{3}$$
$$P_{k+1} = \gamma_2 T_k + \gamma_1 T_{k-1} + \mu_2 P_k + \mu_1 P_{k-1}. \tag{4}$$

Equations (3) and (4) can be explicitly written out at each time step $k = 1, \ldots, N$, and can be written ir the form $\mathbf{y}_{\ell+k}^\mathsf{T} = \mathbf{a}_k^\mathsf{T}\mathbf{X}$ where $\mathbf{X}$ is a 4×2 matrix full of the unknown parameters α_i, β_i, γ_i, μ_i, $i = 1, 2$ Write out the matrices $\mathbf{X}$ and $\mathbf{a}_k^\mathsf{T}$. With your $\mathbf{X}$ and $\mathbf{a}_k^\mathsf{T}$, write out the $\mathbf{A}\mathbf{X} = \mathbf{B}$ problem. Clearly indicate the dimension of $\mathbf{A}$, $\mathbf{A}$

b) The problem $\mathbf{A}\mathbf{X} = \mathbf{B}$ cannot be solved exactly because the columns of $\mathbf{B}$ do not lie in $\mathcal{R}(\mathbf{A})$. Instead, ar approximation solution of $\mathbf{X}$ is sought. Let $\mathbf{B} = \mathbf{P} + \mathbf{R}$ where $\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 \end{bmatrix}$, $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}$ and $\mathbf{r}_i \in \mathcal{N}(\mathbf{A}^\mathsf{T})$, $i = 1, 2$. Find the least squares solution to $\mathbf{A}\mathbf{X} = \mathbf{B}$ by exploiting the fact th of $\mathbf{R}$ lies in the null space of $\mathbf{A}^\mathsf{T}$.

Hint! You will have to do some math similar to what's in the "Linear Least Squares – Derivation of the Solution" slides.

c) A starter `python` code and `csv` files of the temperature and electricity demand data are provided. You wi modify the starter code to *fit* a "coupled" AR model to the temperature and electricity demand data for firs 16 days of August 2022. You will then *test* the AR model on the last 15 days of August 2022. It is ver important to always fit the AR model and test the AR model using different datasets.

Modify the functions `form_A_B`, `fit`, and `predict`. In particular, write code in the `form_A_B` functio to form the matrices **A** and **B** given the fixed $\ell = 2$, using the data from the first 16 days in August 2022.

Then, write code in the `fit` function to find the approximate solution **X**, from your result in Q 1.b. You ma use the `python` function `linalg.solve()` to solve for **X**.

Finally, write code in the `predict` function to predict T_{k+1} and P_{k+1} given the AR model parameters α_i's β_i's, γ_i's, μ_i's, and T_k, T_{k-1}, P_k, P_{k-1}.

Once you've correctly modified the functions `form_A_B`, `fit`, and `predict`, a "coupled" AR model ca be fit using the data from the first 16 days in August 2022, and tested using the data from the last 15 days c August 2022.

Make sure to submit your `python` code.

Note! Do not change the start and end hours of training and testing, nor the AR model memory ℓ. Jus modify the code in the required places.

d) This question and the following questions require using the plots and metrics to assess if the "coupled" AI model does a good job of predicting $\mathbf{y}_{k+1}$ given $\mathbf{y}_{k-1}$ and $\mathbf{y}_k$. All testing is done using data from last 15 day in August 2022.

Let the absolute value of error be $e_{T,k} = \left|T_k - \hat{T}_k\right|$ and $e_{P,k} = \left|P_k - \hat{P}_k\right|$ for temperature and electricit demand, respectively, where T_k and P_k are true temperature and electricity demand respectively, and $\hat{T}$ and $\hat{P}_k$ are the predicted values of T_k and P_k. Let the relative error be $e_{T,k,\text{rel}} = \left|\frac{T_k - \hat{T}_k}{T_k}\right| \times 100\%$ an $e_{P,k,\text{rel}} = \left|\frac{P_k - \hat{P}_k}{P_k}\right| \times 100\%$ for temperature and electricity demand, respectively.

Plot the following and submit the plots. Plot the predicted values $\hat{T}_k$ and $\hat{P}_k$ and the true values T_k and P versus time. Plot the absolute error $e_{T,k}$ and $e_{P,k}$ versus time. Plot the relative error $e_{T,k,\text{rel}}$ and $e_{P,k,\text{r}}$ versus time.

Note! These plots are automatically generated for you in the sample code! Just make sure to submit them!

e) Compute the mean of $e_{T,k}$, e_{P_k}, and standard deviation of $e_{T,k}$, e_{P_k}, and submit the values. Compute th mean of $e_{T,k,\text{rel}}$, $e_{P_k,\text{rel}}$, and standard deviation of $e_{T,k,\text{rel}}$, $e_{P_k,\text{rel}}$, and submit the values.

Note! You may use `python`'s `np.mean()` and `np.std()` functions.

f) Regarding the performance of the "coupled" AR model in comparison to the "individual" AR models, answe the following questions. Refer to Figure 2 and Table 1 for the "individual" AR model results.

   i) Compare the plots you generated with Figure 2. Do the plots look similar?

   ii) Compute the percent difference of each metric (i.e., the metrics are mean of $e_{T,k}$, e_{P_k}, standard deviatio of $e_{T,k}$, e_{P_k}, mean of $e_{T,k,\text{rel}}$, $e_{P_k,\text{rel}}$, and standard deviation of $e_{T,k,\text{rel}}$, $e_{P_k,\text{rel}}$) between the values yo computed and the results from Table 1. Submit the values. Comment on the percent difference value and how they indicate which model performs better, roughly equally as good, or perhaps even worse.

   Hint! Percent difference is $|\hat{x} - \tilde{x}|/|\tilde{x}| \times 100$ where $\tilde{x}$ is either the "true" x value or the between $\tilde{x}$ and $\hat{x}$.

0_coupled_ar_sc.py
2022-demande-electricite-quebec.csv
en_climate_hourly_QC_7025251_08-2022_P1H.csv

iii) Summarize your findings. Does the "coupled" AR model do a better job of predicting the temperatur and electricity demand than the "individual" AR models? Why or why not? In which circumstance would you use the "coupled" model over the "individual" models?

Q2 (10 points)

Time left Hide
10 days, 8 hours

2. Hot-wire anemometry is a fluid velocity measurement technique most commonly used in the investigation of turbulent flows. For reference, Figure 3 shows pictures of the constituent components of a hot-wire anemometer: the electronics (Figure 3a) and the sensor (Figure 3b). In order to interpret the output signal of the anemometer, the device must be calibrated to establish a relationship between the input (fluid velocity) and the output (anemometer voltage). The calibration is performed by fitting a calibration curve to known input-output data. The anemometer calibration equation most commonly used in practice is a modified form of a heat transfer relation for the hot-wire sensor. The calibration equation is a power law of the form

$$y(v) = a + bv^c, \tag{5}$$

where $y(v)$ is the measured output voltage squared, v is the measured fluid velocity, and a, b, c are the calibration constants.

(a) Hot-wire anemometer electronics.

(b) Hot-wire sensor viewed with magnification.

Figure 3: Hot-wire anemometer system.

a) Using each data pair (v_i, y_i) and (5), it follows that

$$y_1 = a + bv_1^c \tag{6}$$
$$y_2 = a + bv_2^c \tag{7}$$
$$\vdots$$
$$y_n = a + bv_n^c. \tag{8}$$

Equations (6) through (8) lead to the nonlinear least squares problem

$$\underset{\mathbf{x}}{\text{minimize}}\ J(\mathbf{x}) = \tfrac{1}{2} \left\| \mathbf{b} - \mathbf{f}(\mathbf{x}) \right\|_2^2 , \tag{9}$$

which can be solved for $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} a & b & c \end{bmatrix}^\mathsf{T}$. Using (6) through (8), explicitly write out the $\mathbf{b}$ and $\mathbf{f}(\mathbf{x})$ matrices.

b) Let $\mathbf{A}$ be defined as the Jacobian of $\mathbf{f}(\cdot)$ such that

$$\mathbf{A} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}. \tag{10}$$

Identify the Jacobian matrix $\mathbf{A}$. What are the dimensions of the matrix $\mathbf{A}$?

Hint! Evaluate the Jacobian for the ith row of $\mathbf{f}(\mathbf{x})$ such that

$$\frac{\partial f_i(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_i(\mathbf{x})}{\partial x_1} & \frac{\partial f_i(\mathbf{x})}{\partial x_2} & \frac{\partial f_i(\mathbf{x})}{\partial x_3} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_i(\mathbf{x})}{\partial a} & \frac{\partial f_i(\mathbf{x})}{\partial b} & \frac{\partial f_i(\mathbf{x})}{\partial c} \end{bmatrix},$$

c) Calibration data is provided in `calibrationData.csv`. Using the provided sample code, solve the non linear least squares problem posed in equation (9). What do the calibration constants $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} a & b & c \end{bmatrix}^\mathsf{T}$ converge to? How many iterations does it take for the calibration constants to converge?

Note! You may use `np.linalg.solve` when solving for $\delta\mathbf{x}$.

d) Plot the calibration data alongside the calibration curve defined by equation (5) with the calibration constant found in Q. 2.c. Additionally, compute the root mean square error (RMSE). Comment on the calibratio curve fit. Does it represent the calibration data well?

Note! The plots and RMSE are automatically generated in the sample code. Just make sure to submit an comment on them.

calibrationData.csv

hotWireCalibration_sc.py

Q3 (10 points)

3_control_find_max_vel_sc.py

Time left Hide
10 days, 8 hours