Machine Learning for Business Analytics

Assignment 3

Due: November/ 25 23:59 (GMT+8)

Instruction

- There are two parts in this assignment. You only have to hand in Part 1 regarding the Julia Programming. **Part 2 will not be graded**; of notice is that similar questions may show up in midterm or final.
- Please hand in your Julia code with **a zip file, named hw3_Student_ID.zip (for instance, hw3_r10h41008.zip), containing 1) a PDF for the Julia code output and 2) all the csv files with their specified names.** Wrong format will not be graded.
- **Copying the assignment will result in zero points.**
- Late submission will be graded according to the following rule:
  your grade = original grade × (1 − 0.05*h*), if you submit *h* hours after the deadline.

Part 1 Programming Exercises (100%, CH6-CH11)

1. (40%) (Freg's lover) Freg is a frog, who is studied by ILFI (I-Love-Frog-Institute). According to the researchers in the institute, there are some characteristics about Freg's behaviour.
   (1) Freg can only jump on/nearby the trace $y(x) = \prod_{i=1}^{b} a_{2i-1}x - a_{2i}$, where ***a*** is a vector and the subscript indicates the location of that coefficient in vector ***a***. *b* is an integer, and $b \geq 3$.
   (2) Freg is a frog who is really good at doing approximation. When he was a child, he can do first-order Taylor approximation. And now, he can do second-order Taylor approximation.
   (3) Freg is a capricious frog. He changes the number of node **(**n_{now}**)** he lands every day. And you can predict the number of node he would land when he was a little frog by the formula $n_{now} = c + dn_{child}$. (If n_{child} is not an integer, then round it to integer.) He decides the points he lands by equally split the x-distance to n_{now} (n_{child} when he is a child) points and lands on those calculated x-value and y-value (calculated by Taylor expansion).
   (4) Freg has different starting *x*-value **(start_x)** and same ending *x*-value **(end_x)** on his journey every day.

   Since the ILFI was attacked by IHFI, they lost some precious trace plots about Freg. Use the following information and follow the instructions to define a function that outputs the plot ILFI lost. The features for each line are inside the parentheses; the requirements in the parentheses are (Name of the trace, the type of the plot, the color of the line, the width of the line).

(1) Plot the trace that Freg might jump. (Actual Trace, Line, Black, 3)

(2) Plot the trace that Freg will land now. (Current, Scatter+Line, Blue, 1.5)

(3) Plot the trace that Freg might land when he was a child. (Childhood, Scatter+Line, Green, 1.5)

(4) Add legend that contains the mapping relation between the color of the line and the name of the line.

(5) Name the plot as “Freg we love you!”, X-axis as “X value”, Y-axis as “Y value”

(6) Change the position of the legend to the place that won’t overlap with your line.

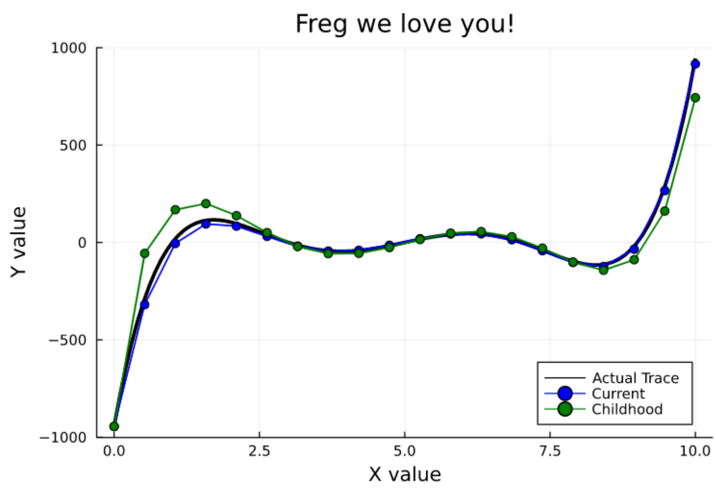
Example:

- Input:

  a = [1, 1, 1, 3, 1, 5, 1, 7, 1, 9], b = 5, c = 0, d = 1

  n_now = 20, start_x = 0, end_x = 10

- Output:



Test data:

(a) Input:

a = [1, 1, 1, 3, 1, 5, 1, 7, 1, 9], b = 5, c = 0, d = 1

n_now = 20, start_x = 0, end_x = 10

Output: Show your plot on the pdf.

(b) Input:

a = [1/2, 3/2, 1/3, -1, 1/2, 1, 1/2, -1, 2/5, 2, 2, -1], b = 6, c = 7, d = 2

n_now = 20, start_x = -3.5, end_x = 5.3

Output: Show your plot on the pdf.

(c) Input:

a = [1.73, 0.8, 0.253, -0.347, 0.88, 1.24, 2.384, -10.83, 0.537, 3.456, 0.923, 6.34],

b = 6, c = 0, d = 1, n_now = 10, start_x = -4.5, end_x = 6.5

Output: Show your plot on the pdf.

(d) Input:

a = [1.73, 0.8, 0.253, -0.347, 0.88, 1.24, 2.384, -10.83, 0.537, 3.456, 0.923, 6.34],

b = 6, c = 0, d = 1, n_now = 20, start_x = -4.5, end_x = 6.5

Output: Show your plot on the pdf.

(e) Compare the result of (c) and (d), First, state your opinion about their difference, which one did the better job on approximation? And why the difference(s) take(s) place? Second, compare the result of first order Taylor approximation and second order Taylor approximation, which one did a better job? Why? Third, observe the result of (a)(b)(c)(d), Taylor approximation performs a bad approximation when your function looks bumpy or smooth?

2. (40%) (A Michelin inspector, Anny) Anny is a Michelin inspector, who take responsibility for trying different food all over the world. Anny has her own unique taste. She thinks that there are three classes of food, "Mouth-watering" (class 1, MW), "I-feel-it-on-mouth" (class 2, IFIOM) and "Not-even-food" (class 3, NEF) food. From her point of view, the taste, smell and her own preference to the food matters. She records the scores for each taste. With those scores, Anny can easily classify the food she tasted by a self-trained mathematical classifier. Since Anny holds a master's degree in Statistics from NTU, she wants to apply a non-linear classifier, called Bayes classifier. The logic of the Bayes classifier is as following:

(1) The distance function is defined as $d_i(X_k) = \ln\big(p(w_i)\big) + \frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln\big(\det(C_i{}^{-1})\big) - \frac{1}{2}[(X_k - \theta_i)^T C_i^{-1}(X_k - \theta_i)]$, which gives you the distance between your data and i^{th} cluster's center.

Notation:

- *X* is the score matrix recorded by Anny, X_k is the scores for dish *k*.
- $p(w_i)$ is the proportion that Anny thinks of how many dishes should be classified into class *i* before she carries out the classification.
- C_i, θ_i are the covariance matrix and mean vector for class *i*, *n* is the number of classes (i.e. n=3 here)
- $\det(C_i)$ can be attained by the function "det(.)" in LinearAlgebra package.

(2) If $d_i(X_k) < d_j(X_k)$, then we classify dish *k* to class *i*. If $d_i(X_k) = d_j(X_k)$, then classify dish *k* to the class with larger index (i.e. if *i* > *j*, then classify dish *k* to

class *i*).

Part 1. Read the files then carry out the classification with given parameters, then output your result to a csv file.

Example:

- Input:

$$\theta_1 = \begin{bmatrix} 80 \\ 90 \\ 100 \end{bmatrix}, \theta_2 = \begin{bmatrix} 50 \\ 60 \\ 70 \end{bmatrix}, \theta_3 = \begin{bmatrix} 20 \\ 30 \\ 40 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 125 & -75 & 45 \\ -75 & 310 & 90 \\ 45 & 90 & 440 \end{bmatrix}, C_2 = \begin{bmatrix} 300 & 125 & -400 \\ 125 & 750 & -375 \\ -400 & -375 & 900 \end{bmatrix}, C_3 = \begin{bmatrix} 360 & 125 & 190 \\ 125 & 250 & 80 \\ 190 & 80 & 160 \end{bmatrix}$$

$$\mathrm{p}(w_1) = 0.3, \mathrm{p}(w_2) = 0.25, \mathrm{p}(w_3) = 0.45$$

  For the score data, see the file “**Anny_M_score.csv**”

- Output:

  See the file “**Anny_classify_result.csv**”

Test Data:

(a) Input:

$$\theta_1 = \begin{bmatrix} 80 \\ 90 \\ 100 \end{bmatrix}, \theta_2 = \begin{bmatrix} 50 \\ 60 \\ 70 \end{bmatrix}, \theta_3 = \begin{bmatrix} 20 \\ 30 \\ 40 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 125 & -75 & 45 \\ -75 & 310 & 90 \\ 45 & 90 & 440 \end{bmatrix}, C_2 = \begin{bmatrix} 300 & 125 & -400 \\ 125 & 750 & -375 \\ -400 & -375 & 900 \end{bmatrix}, C_3 = \begin{bmatrix} 360 & 125 & 190 \\ 125 & 250 & 80 \\ 190 & 80 & 160 \end{bmatrix}$$

$$\mathrm{p}(w_1) = 0.3, \mathrm{p}(w_2) = 0.25, \mathrm{p}(w_3) = 0.45$$

For the data, read the file “**Anny_M_score1.csv**”

Output: Write your result into a file named “**Anny_classify_result1.csv**”

(b) Input:

$$\theta_1 = \begin{bmatrix} 87.96 \\ 61.85 \\ 118.42 \end{bmatrix}, \theta_2 = \begin{bmatrix} 127.69 \\ 116.18 \\ 80.31 \end{bmatrix}, \theta_3 = \begin{bmatrix} 74.90 \\ 49.92 \\ 92.98 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 66.65 & 62.86 & 5.78 \\ 62.86 & 77.46 & -8.41 \\ 5.78 & -8.41 & 140.11 \end{bmatrix}, C_2 = \begin{bmatrix} 161.54 & 53.49 & 39.35 \\ 53.49 & 177.16 & 64.00 \\ 39.35 & 64.00 & 159.26 \end{bmatrix}, C_3 = \begin{bmatrix} 29.93 & 27.92 & 12.57 \\ 27.92 & 35.09 & 1.90 \\ 12.57 & 1.90 & 137.73 \end{bmatrix}$$

$$\mathrm{p}(w_1) = 0.2, \mathrm{p}(w_2) = 0.5, \mathrm{p}(w_3) = 0.3$$

For the data, read the file “**Anny_M_score2.csv**”

Output: Write your result into a file named “**Anny_classify_result2.csv**”

***Tutorial for how to read csv file and export matrix to csv file:

(1) Download the packages “Tables”, “CSV” and “DelimitedFiles”

(2) If you want to export a matrix to a csv file, you can use function “write” in CSV package.

CSV.write(**your export directory and file name**, Tables.table(**matrix name**), writeheader = false)

For example:

My export directory is **C:/Users/acer/Desktop/MLBA/Anny_classify1.csv**

The black part tells Julia the place you want to store your file; the red part tells Julia the file name and the file type. Be aware of “/”!

(3) If you want to read a csv file and create a matrix with the data, you can use function “readdlm” in DelimitedFiles.

readdlm(**the location of your file and its name**, ',', Float64)

Part 2. Since Anny received death threat when she classifies a dish as “Not-even-food food”, she decides to remove “Not-even-food food” from the class. As a result, her new classification model only considered the taste score and her own preference. Do the classification with the score data and the coefficients in each test data. Follow the instructions to create the plot.

(1) Plot the classification result, use different color to represent different class.

(2) Plot the boundary for the classification, and conceal its label.

(3) Adjust the line width to 3 and change its color to black.

(4) Show the legend which tell us the mapping relationship between the class and its representative color.

Hint: Boundary is the line that $d_i(X) = d_j(X)$. But it’s difficult to obtain the exact solution directly from this equation, you may consider to use trial-and-error method (grid search, google it!) and set a tolerance to find the solution efficiently. (i.e. when $|d_i(X) - d_j(X)| < \varepsilon$, then we consider X as an acceptable point for the boundary.)

Example:

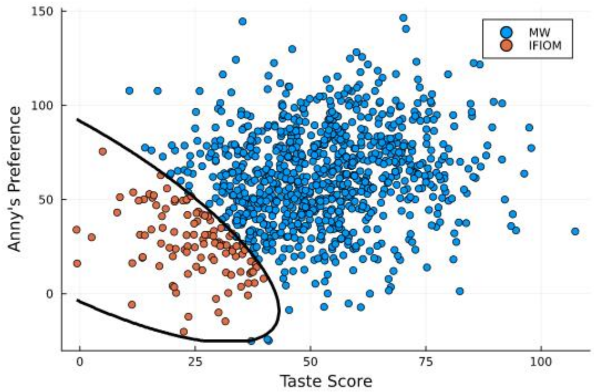
- Input:

$$\theta_1 = \begin{bmatrix} 20 \\ 30 \end{bmatrix}, \theta_2 = \begin{bmatrix} 50 \\ 60 \end{bmatrix}, C_1 = \begin{bmatrix} 125 & -75 \\ -75 & 310 \end{bmatrix}, C_2 = \begin{bmatrix} 300 & 125 \\ 125 & 750 \end{bmatrix}$$

$$p(w_1) = 0.3, \qquad p(w_2) = 0.7$$

Using “**death_threat.csv**” as your data

- Output:



Test Data:

(a) Input:

$$\theta_1 = \begin{bmatrix} 20 \\ 30 \end{bmatrix}, \theta_2 = \begin{bmatrix} 50 \\ 60 \end{bmatrix}, C_1 = \begin{bmatrix} 125 & -75 \\ -75 & 310 \end{bmatrix}, C_2 = \begin{bmatrix} 300 & 125 \\ 125 & 750 \end{bmatrix}$$

$$p(w_1) = 0.3, \qquad p(w_2) = 0.7$$

Use the file "**death_threat1.csv**" as your data

Output: Show the plot on your pdf.

(b) Input:

$$\theta_1 = \begin{bmatrix} 80 \\ 120 \end{bmatrix}, \theta_2 = \begin{bmatrix} 140 \\ 150 \end{bmatrix}, C_1 = \begin{bmatrix} 1225 & -525 \\ -525 & 400 \end{bmatrix}, C_2 = \begin{bmatrix} 900 & 390 \\ 390 & 400 \end{bmatrix}$$

$$p(w_1) = 0.4, \qquad p(w_2) = 0.6$$

Use the file "**death_threat2.csv**" as your data

Output: Show the plot on your pdf.

(c) Observe the classification result of (a) and (b), do you think this classifier a good classifier? Why? If not, propose an ideal classification method from your own opinion

3. (20%) (Metaverse tourists, Hsin and Eric) Hsin and Eric have great interest on Metaverse. They couldn't wait to experience what it would be on the Metaverse. So they went to Facebook's (META, now) headquarter to ask a try. And, surprisingly, they accepted their asks. However, they encountered an evil wizard/witch on the Metaverse. The evil wizard/witch casted a spell on Eric and the world, which sent Eric to a different place (x', y', z') and different time zone (t'), which is recorded by a different coordinate system and time system. The evil wizard/witch left a matrix about how s/he did the transformation and a vector about Eric's location (i.e. c' =(x', y', z', t'), a column vector) on the new system. The transformation laws are as following:

(1) The wizard/witch do Gram-Schmidt algorithm to the matrix

(2) Use the new basis (q in your textbook) s/he found to construct new coordinate system

(3) Sent Eric to the place according to the new coordinate system and record the mapping coordinate on the new coordinate system (Hint. The basis for the old coordinate system $\{\begin{bmatrix}1\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\}$).

Please write a function to help Hsin learn about the basis of new coordinate system and find Eric!

Some important instructions:

1. Round your result to 4 digits (use function "round"), so the grading system can recognize your answer.
2. Put your result on one matrix. (i.e.$[\mathrm{q1} \;\; \mathrm{q2} \;\; \mathrm{q3} \;\; \mathrm{q4} \;\; c']$)

Example:

- Input: See the file "**Metaverse_input.csv**"
- Output: See the file "**Metaverse_output.csv**"

Test Data:

(a) Input: Use the file "**Metaverse_input1.csv**" as your data
Output: Write your file to "**Metaverse_output1.csv**"

(b) Input: Use the file "**Metaverse_input2.csv**" as your data
Output: Write your file to "**Metaverse_output2.csv**"

(c) Input: Use the file "**Metaverse_input3.csv**" as your data
Output: Write your file to "**Metaverse_output3.csv**"

(d) Input: Use the file "**Metaverse_input4.csv**" as your data
Output: Write your file to "**Metaverse_output4.csv**"

(e) Input: Use the file "**Metaverse_input5.csv**" as your data
Output: Write your file to "**Metaverse_output5.csv**"

*** The sign of the basis may variate by applying different functions. The grading system allows both representations (i.e. If **v** is one of the basis for the world, both **v** and –**v** will be recognized as correct outputs.)

Part 2 Mathematical Exercises (0%)

Chapter 7 Exercises

1. *3-D rotation*. Let x and y be 3-vectors representing positions in 3-D. Suppose that the vector y is obtained by rotating the vector x about the vertical axis (i.e., e_3) by 45° (counterclockwise, i.e., from e1 toward e_2). Find the 3×3 matrix A for which $y = Ax$. *Hint*. Determine the three columns of A by finding the result of the transformation on the unit vectors e_1, e_2, e_3.

2. *Sum property of convolution*. Show that for any vectors a and b, we have $\mathbf{1}^T (a * b) = (\mathbf{1}^T a)(\mathbf{1}^T b)$. In words: The sum of the coefficients of the convolution of two vectors is the product of the sums of the coefficients of the vectors. *Hint*. If the vector a represents the coefficients of a polynomial p, $\mathbf{1}^T a = p(1)$.

Chapter 8 Exercises

1. *Interpolation of polynomial values and derivatives*. The 5-vector c represents the coefficients of a quartic polynomial $p(x) = c_1 + c_2x + c_3x^2 + c_4x^3 + c_5x^4$. Express the conditions

$$p(0) = 0, p'(0) = 0, p'(1) = 1, p'(1) = 0,$$

as a set of linear equations of the form $Ac = b$. Is the system of equations underdetermined, over-determined, or square?

2. *Affine combinations of solutions of linear equations*. Consider the set of m linear equations in n variables $Ax = b$, where A is an $m \times n$ matrix, b is an m-vector, and x is the n-vector of variables. Suppose that the n-vectors $z_1, \ldots, z_k$ are solutions of this set of equations, i.e., satisfy $Az_i = b$. Show that if the coefficients $\alpha_1, \ldots, \alpha_k$ satisfy $\alpha_1 + \ldots + \alpha_k = 1$, then the affine combination

$$w = \alpha_1 z_1 + \ldots + \alpha_k z_k$$

is a solution of the linear equations, i.e., satisfies $Aw = b$. In words: Any affine combination of solutions of a set of linear equations is also a solution of the equations.

Chapter 9 Exercises

1. Reducing a Markov model to a linear dynamical system. Consider the 2-Markov model

$$x_{t+1} = A_1 x_t + A_2 x_{t-1}, \qquad t = 2, 3, \ldots,$$

where x_t is an n-vector. Define $z_t = (x_t, x_{t-1})$. Show that z_t satisfies the linear dynamical system equation $z_{t+1} = B z_t$, for $t = 2, 3, \ldots$, where B is a $(2n) \times (2n)$ matrix. This idea can be used to express any K-Markov model as a linear dynamical system, with state $(x_t, \ldots, x_{t-K+1})$.

2. *Recursive averaging*. Suppose that $u_1, u_2, \ldots$ is a sequence of n-vectors. Let $x_1 = 0$,

and for t = 2, 3, ..., let x_t be the average of $u_1, \ldots, u_{t-1}$, i.e., $x_t = (u_1 + \cdots + u_{t-1}) / (t - 1)$. Express this as a linear dynamical system with input, i.e., $x_{t+1} = A_t x_t + B_t x_t$, $t = 1, 2, \ldots$ (with initial state $x_1 = 0$). *Remark*. This can be used to compute the average of an extremely large collection of vectors, by accessing them one-by-one.

3. *Complexity of linear dynamical system simulation*. Consider the time-invariant linear dynamical system with n-vector state x_t and m-vector input u_t, and dynamics $x_{t+1} = Ax_t + Bu_t$, $t = 1, 2, \ldots$. You are given the matrices A and B, the initial state x_t, and the inputs $u_t, \ldots, u_{T-1}$. What is the complexity of carrying out a simulation, i.e., computing $x_2, x_3, \ldots, x_T$? About how long would it take to carry out a simulation with $n = 15$, $m = 5$, and $T = 10^5$, using a 1 Gflop/s computer?

Chapter 10 Exercises

1. *Multiplication by a diagonal matrix.* Suppose that A is an $m \times n$ matrix, D is a diagonal matrix, and $B = DA$. Describe B in terms of A and the entries of D. You can refer to the rows or columns or entries of A.

2. *Norm of matrix product*. Suppose A is an $m \times p$ matrix and B is a $p \times n$ matrix. Show that $\|AB\| \leq \|A\|\|B\|$, i.e., the (matrix) norm of the matrix product is no more than the product of the norms of the matrices. *Hint*. Let $a_1^T, \ldots, a_m^T$ be the rows of A, and $b_1, \ldots, b_n$ be the columns of B. Then

$$\|AB\|^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(a_i^T b_j\right)^2.$$

   Now use the Cauchy–Schwarz inequality.

3. *Gram matrix and QR factorization*. Suppose the matrix A has linearly independent columns and QR factorization $A = QR$. What is the relationship between the Gram matrix of A and the Gram matrix of R? What can you say about the angles between the columns of A and the angles between the columns of R?