CIVL 498A HW-2: MLE, MAP, ML pipeline.

Note: For hand calculation problems, you are supposed to write down your answers on a piece of paper and then submit the scanned version of the paper(s). Show your work by writing out derivations if there is any. For coding problems, you can either use Google Colab or Jupyter Notebook (on your own machine) to write and run the code. When submitting, submit only the ipynb file (you should use relative file directory for input files e.g., "./data.csv" in your code, so when I run the file on my local computer, files would load normally), and make sure when "run all" is clicked, all required results from the questions are shown (this is how I will grade your answer. If when I "run all" and your code crashes, it will be deemed as wrong).

1. **(Hand calculation)** Use the data in the following table:

Observations	x_1	x_2	x_3	**y**
1	0	0	1	5
2	1	1	e^2	22
3	3	2	e^2	54
4	2	0	e^3	16

1.1 Use the maximum likely hood method to find the hypothesis describing the relationship between y and the variables (x_1, x_2, x_3), plus another variable representing the intercept, described using the following equation:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 \ln(x_3) + \epsilon$$

Assume Gaussian distribution for the error term and the probability of the data, i.e., $\epsilon^{(i)} \sim N(0, \sigma^2), P(y^{(i)} | x^{(i)}; \theta) \sim N(\theta^T x^{(i)}, \sigma^2)$

1.1

Given assumptions...

$$f(\theta) = \log(L(\theta)) = m \cdot \log \frac{1}{\sqrt{2\pi}\sigma} + \sum_{i=1}^{m} \frac{-(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}$$

$$\arg\max f(\theta) = \arg\min \frac{1}{2} \sum_{i=1}^{m} \underbrace{(y^{(i)} - \theta^T x^{(i)})^2}_{= J(\theta)}$$

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 \ln(x_3)$$

$$\frac{dJ(\theta)}{d\theta_j} = \sum_{i=1}^{4} \left[\left(h_\theta(x)^{(i)} - y^{(i)} \right) \cdot x_j^{(i)} \right]$$

$$\begin{cases} h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \varepsilon \\ \text{Let } x_2 = x_2^2, \; x_3 = \ln(x_3) \end{cases}$$

$$\frac{dJ(\theta)}{d\theta_0} = (\theta_0 - 5)\times 1 + (\theta_0 + \theta_1 + \theta_2 + 2\theta_3 - 22)\times 1 + (\theta_0 + 3\theta_1 + 4\theta_2 + 2\theta_3 - 54)\times 1 + (\theta_0 + 2\theta_1 + 3\theta_3 - 16)\times 1 = 4\theta_0 + 6\theta_1 + 5\theta_2 + 7\theta_3 - 97$$

$$\frac{dJ(\theta)}{d\theta_1} = (\theta_0 - 5)\times 0 + (\theta_0 + \theta_1 + \theta_2 + 2\theta_3 - 22)\times 1 + (\theta_0 + 3\theta_1 + 4\theta_2 + 2\theta_3 - 54)\times 3 + (\theta_0 + 2\theta_1 + 3\theta_3 - 16)\times 2 = 6\theta_0 + 14\theta_1 + 13\theta_2 + 14\theta_3 - 216$$

$$\frac{dJ(\theta)}{d\theta_2} = (\theta_0 - 5)\times 0 + (\theta_0 + \theta_1 + \theta_2 + 2\theta_3 - 22)\times 1 + (\theta_0 + 3\theta_1 + 4\theta_2 + 2\theta_3 - 54)\times 4 + (\theta_0 + 2\theta_1 + 3\theta_3 - 16)\times 0 = 5\theta_0 + 13\theta_1 + 17\theta_2 + 10\theta_3 - 238$$

$$\frac{dJ(\theta)}{d\theta_3} = (\theta_0 - 5)\times 0 + (\theta_0 + \theta_1 + \theta_2 + 2\theta_3 - 22)\times 2 + (\theta_0 + 3\theta_1 + 4\theta_2 + 2\theta_3 - 54)\times 2 + (\theta_0 + 2\theta_1 + 3\theta_3 - 16)\times 3 = 7\theta_0 + 14\theta_1 + 10\theta_2 + 17\theta_3 - 200$$

$$\begin{bmatrix} 4 & 6 & 5 & 7 \\ 6 & 14 & 13 & 14 \\ 5 & 13 & 17 & 10 \\ 7 & 14 & 10 & 17 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} 97 \\ 216 \\ 238 \\ 200 \end{bmatrix}$$

$$\theta_0 = 5 \quad \theta_1 = 1 \quad \theta_2 = 10 \quad \theta_3 = 3$$

Hypothesis: $y = 5 + x_1 + 10x_2^2 + 3\ln(x_3) + \varepsilon$

2. **(Hand calculation)** Use the maximum a posteriori method to derive the cost function of LASSO regression. Assume the training examples follow Gaussian distribution i.e., $P(y^{(i)}|x^{(i)};\theta) \sim N(\theta^T x^{(i)}, \sigma^2)$, and the prior for thetas follow a Laplace prior i.e., $P(\theta_j) \sim Laplace(0,\gamma)$. Refer to the ridge regression example in the class notes as a reference.

Recall Laplace distribution follows the equation: $P(x) \sim Laplace(\mu,\gamma) = \frac{1}{2\gamma} e^{(\frac{-|x-\mu|}{\gamma})}$

$$\text{Arg}_\theta \max \; P(\theta|\vec{y}) = \arg_\theta \max \frac{P(\vec{y}|\theta)\cdot P(\theta)}{P(\vec{y})}$$

$$\arg_\theta \max \left[P(\vec{y}|\theta)\cdot P(\theta)\right] = \text{argmax}\left[\log P(\vec{y}|\theta) + \log P(\theta)\right]$$

① $y^{(i)} \sim N(\theta^T x^{(i)}, \sigma^2)$

$$P(\vec{y}|\theta) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\cdot\sigma} e^{\left(\frac{-(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right)}$$

$$\log P(\vec{y}|\theta) = m\log\left(\frac{1}{\sqrt{2\pi}\cdot\sigma}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(y^{(i)}-\theta^T x^{(i)}\right)^2$$

② $\theta_j \sim \text{Laplace}(0,\gamma)$

$$P(\theta) = \prod_{j=1}^{n} \frac{1}{2\gamma} e^{\left(\frac{-|\theta_j - 0|}{\gamma}\right)}$$

$$\log P(\theta) = n\log\frac{1}{2\gamma} - \sum_{j=1}^{n}\frac{|\theta_j|}{\gamma}$$

M.A.P → $\text{argmax}\left[\log\left(P(\vec{y}\mid\theta)\right)+\log\left(P(\theta)\right)\right]$

$$= \text{argmax}\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(y^{(i)}-\theta^T x^{(i)}\right)^2-\frac{1}{\gamma}\sum_{j=1}^{n}|\theta|\right]$$

$$= \text{argmin}\left[\frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(y^{(i)}-\theta^T x^{(i)}\right)^2+\frac{1}{\gamma}\sum_{j=1}^{n}|\theta_j|\right]$$

$$= \text{arg min}\left[\frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)}-\theta^T x^{(i)}\right)^2+\frac{\sigma^2}{\gamma}\sum_{j=1}^{n}|\theta_j|\right]$$

$$= \text{arg min}\left[\underbrace{\frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)}-\theta^T x^{(i)}\right)^2}_{J(\theta)}+\lambda\sum_{j=1}^{n}|\theta_j|\right]$$

$$= \text{arg min}\left[J(\theta)+\underbrace{\lambda\sum_{j=1}^{n}|\theta_j|}_{\text{Lasso L1 norm}}\right]$$

3. **(Coding Problem)** Use the dataset named "housing.csv" for the following questions. The data pertains to the houses in California based on the 1990 census data. There are 10 columns in the dataset.

- longitude
- latitude
- housingMedianAge: Median age of a house within a block; a lower number is a newer building
- totalRooms: Total number of rooms within a block
- totalBedrooms: Total number of bedrooms within a block
- population: Total number of people residing within a block
- households: Total number of households for a block
- medianIncome: Median income for households within a block of houses (measured in tens of thousands of USD)
- medianHouseValue: Median house value for households within a block (measured in USD)
- oceanProximity: Location of the house w.r.t ocean/sea

3.1. Read in the housing data as a dataframe, and perform data exploration, including: showing the first 10 rows of the data using head(), showing the data type of columns using info(), showing stats of the numerical columns using describe(). Draw histograms of all numerical columns to show the distributions of the data.

3.2. For data visualization, create a scatter plot of the data using *longitude* as the x axis and *latitude* as the y axis, and *population/100* as the size of the scatter points, and *median_house_value* as the color of the scatter points with higher values in warmer colors and lower values in colder colors (use "jet" as the color map). Show the color map bar on the right-hand side of the figure.

3.3. Compute the correlation matrix for all numerical columns of the dataset, and show the ranking of correlation of each column with the *median_house_value* from larger to smaller values. Use the scatter_matrix method from pandas.plot to plot the top three ranked columns with respect to *median_house_value.*

3.4. Fill the missing values in *total_bedrooms* using median values of the column, and then create three more columns as *rooms_per_household (i.e., total_rooms/households), bedrooms_per_room (i.e., total_bedrooms/total_rooms), and population_per_household (i.e., polulation/households)*.

3.5. Transform the ocean_proximity column from strings into numerical values using OneHotEncoder from sklearn.preprocessing, and then add encoded columns to the dataframe and drop the original ocean_proximity column. Now your dataframe should have 17 columns (i.e., original 10 + 3 columns added in 3.4, and 5 columns added in 3.5, and 1 column dropped in 3.5).

3.6. Perform train/test split using the train_test_split method in sklearn.model_selection (20% test, 80% train, random_state as 42). Use *median_house_value* as target, and other columns as input to fit LinearRegression, Ridge, and SVR as models, all with default parameters and report the performance of the models by showing the testing errors using RMSE.