Machine Learning HW3

TA: 吳京軒 (handwriting) justinwu880520@gmail.com
蘇拉傑 (programming) surajdengale@gapp.nthu.edu.tw

Deadline: 2023/05/22 (MON) 23:59

Grading Policy:

1. In the handwriting assignment, please submit the pdffile. (HW3_studentID_Handwriting.pdf)
2. In the programming assignment, the code and report (HW3_studentID_Programming.pdf) should be compressed into a ZIP file and uploaded to the eeclass website. Also, please write a README file to explain how to run your code and describe related characteristics used in your report. The report format is not limited.
3. You are required to finish this homework with Python 3. Moreover, built-in machine learning libraries or functions, e.g., *sklearn.linear_model*, are NOT allowed to be used. But you can use dimension reduction functions such as *sklearn.decomposition.PCA*.
4. Discussions are encouraged, but plagiarism is strictly prohibited (changing variable names, etc.). You can use any open source you want with a clear reference mentioned in your report. If there is any plagiarism, you will get 0 in this homework.

Submission:

Please follow the following format and naming rules when submitting files.

1. HW3_student_id_Handwriting.pdf
2. HW3_student_id.zip

```
        | --- HW3_studentID_Programming.pdf
        | --- README.txt
        | --- HW3.py (only *.py)
```

- You need to upload both HW3_studentID_Handwriting.pdf and HW3_studentID.zip to eeclass website.

1) **Handwriting (30%)**

1) Consider a Gaussian process regression model in which the target variable t has dimensionality D. Write down the conditional distribution of t_{N+1} for a test input vector x_{N+1}, given a training set of input vectors $x_1, \ldots, x_N$ and corresponding target observations $t_1, \ldots, t_N$.

2) Drive the result

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{X}, \boldsymbol{\alpha}, \beta) &= \ln \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C}) \\ &= -\frac{1}{2}\left\{N\ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}\right\}\end{aligned}$$

For the marginal likelihood function in the regression RVM, by performing the Gaussian integral over w in

$$p(\mathbf{t}|\mathbf{X}, \boldsymbol{\alpha}, \beta) = \int p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta)p(\mathbf{w}|\boldsymbol{\alpha})\,\mathrm{d}\mathbf{w}.$$

Using the technique of completing the square in the exponential.

3) Show that there are $2^{M(M-1)/2}$ distinct indirected graphs over a set of M distinct random variables. Draw the 8 possibilities of the case of M = 3.

4) In Section 7.2.1 we used direct maximization of the marginal likelihood function to derive the re-estimation equations

$$\begin{aligned}\alpha_i^{\text{new}} &= \frac{\gamma_i}{m_i^2} \\ (\beta^{\text{new}})^{-1} &= \frac{\|\mathbf{t} - \boldsymbol{\Phi}\mathbf{m}\|^2}{N - \sum_i \gamma_i}\end{aligned}$$

for finding values of the hyperparamters α and β for the regression RVM. Similarly, in Section 9.3.4 we used the EM algorithm to maximize the same marginal likelihood, giving the re-estimation equations

$$\begin{aligned}\alpha_i^{\text{new}} &= \frac{1}{m_i^2 + \Sigma_{ii}} \\ (\beta^{\text{new}})^{-1} &= \frac{\|\mathbf{t} - \boldsymbol{\Phi}\mathbf{m}_N\|^2 + \beta^{-1}\sum_i \gamma_i}{N}\end{aligned}$$

Show that these two sets of re-estimation equations are formally equivalent.

2) Programming (70%)

In this problem, you need to apply the Maximum Likelihood (ML) and Bayesian linear regression methods to train a linear model in order to predict the calories burnt during exercise.

Data

Contains two *.csv files, [exercise.csv] and [calories.csv]. More detailed descriptions are given below:
The exercise.csv have 15000 pieces of data in total (X)
The calories.csv have 15000 pieces of data in total (Y)
You have to merge them and split them into 70:10:20 for training, validation, and testing, respectively.

Problem

Please employ the linear model to predict calories burnt with respect to exercise in the testing set.

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^{P+2} w_j \phi_j(\mathbf{x})$$

There should be at least two functions BLR() and MLR() in your hw3.py.

In data evaluation, main() will call BLR() and MLR() to calculate the predicted calories burnt.

1) *(20 points for implementation + 10 points for MSE Screenshot)* Please use Least Squares i.e. Maximum Likelihood Estimation (see Q.3) to train the model. Then, use your trained linear model to predict the burnt calories and compute the mean squared error for each data in testing_set.
2) (20 points for implementation + 10 points for *MSE Screenshot*) Please use Bayesian Linear Regression to estimate w. Then, use your estimated parameter to predict the burnt calories and compute the mean squared error for each data in Validation_set.

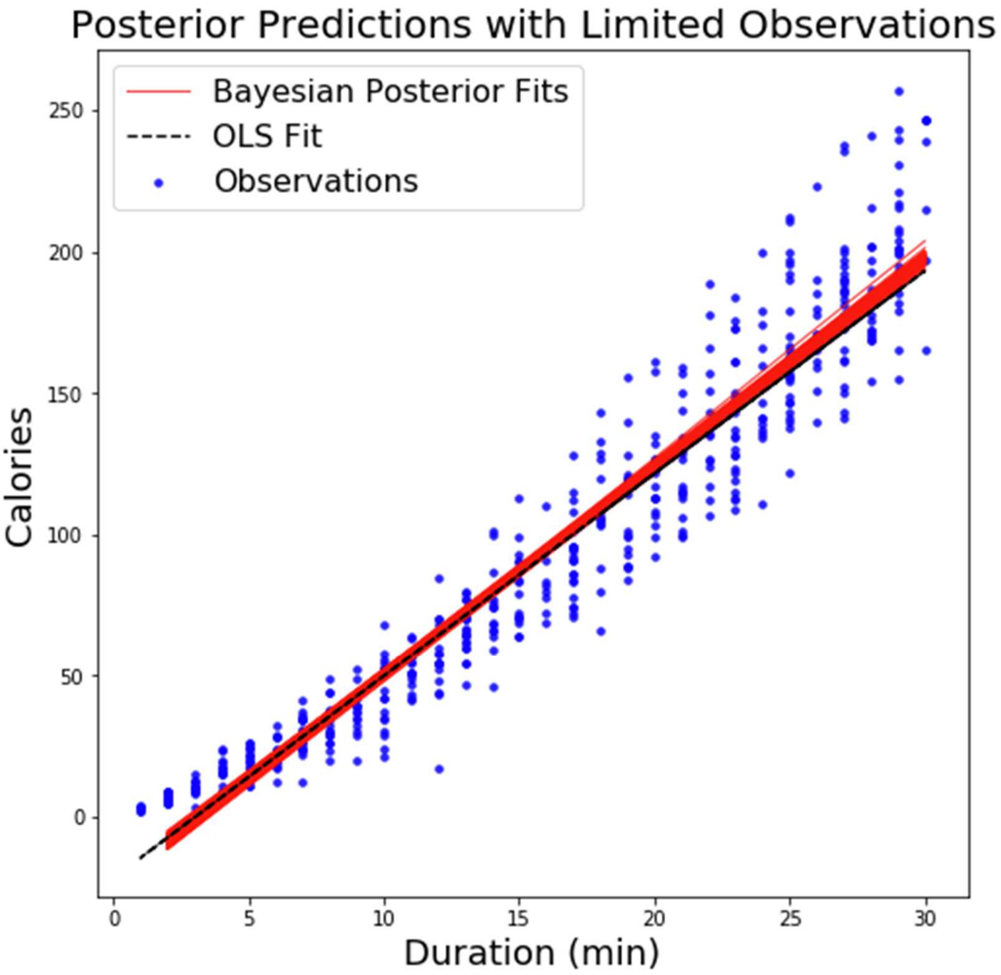
[***Note*** *: While doing Bayesian Regression you have to consider some posteriors first with limited observations and then again with all observations.*

Note: *We expect students to understand the concept that, changing the posteriors can change your regression models distribution. The general concept is, if you train the model with more new data, the parameters try to approximate to the set values.*]

Mean squared error: $\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$

3) (10 points) Please discuss the difference between Maximum Likelihood and Bayesian Linear Regression. Plot the best fit lines for both models. Best fit lines for Bayesian Linear Regression means that you have to plot the intercept and slope. [***Note*** *: Consider the distribution of error terms to be normally distributed. Therefore, in this case Ordinary Least Square is same as Maximum Likelihood Estimation. So, you just have to apply* ***OLS****.*]

The expected graph might look something like this –



Bonus Hint – Try to predict a single datapoint with both BLR and OLS or MLR and plot the prediction to see how there is a difference between BLR and OLS.

4) (10 points) Please implement any regression model you want to get the best possible MSE. *use of built-in libraries is allowed only for this question.