HOMEWORK 5

>>Abhay Kumar<<
>>9081403157<<

Instructions: Although this is a programming homework, you only need to hand in a pdf answer file. There is no need to submit the latex source or any code. You can choose any programming language, as long as you implement the algorithm from scratch.
Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Please check Piazza for updates about the homework.

Linear Regression (100 pts total, 10 each)

The Wisconsin State Climatology Office keeps a record on the number of days Lake Mendota was covered by ice at http://www.aos.wisc.edu/~sco/lakes/Mendota-ice.html. Same for Lake Monona: http://www.aos.wisc.edu/~sco/lakes/Monona-ice.html. As with any real problems, the data is not as clean or as organized as one would like for machine learning. Curate two clean data sets for each lake, respectively, starting from 1855-56 and ending in 2018-19. Let x be the year: for 1855-56, $x = 1855$; for 2017-18, $x = 2017$; and so on. Let y be the ice days in that year: for Mendota and 1855-56, $y = 118$; for 2017-18, $y = 94$; and so on. Some years have multiple freeze thaw cycles such as 2001-02, that one should be $x = 2001, y = 21$.

1. Plot year vs. ice days for the two lakes as two curves in the same plot. Produce another plot for year vs. $y_{Monona} - y_{Mendota}$.

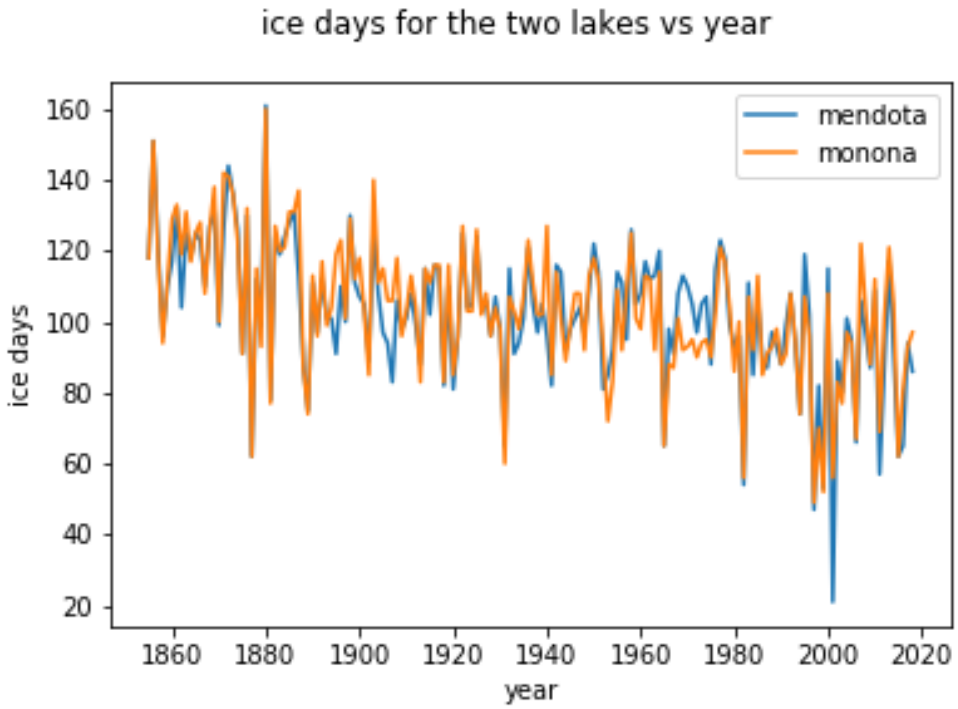

Fig 1: ice days for the two lakes vs year

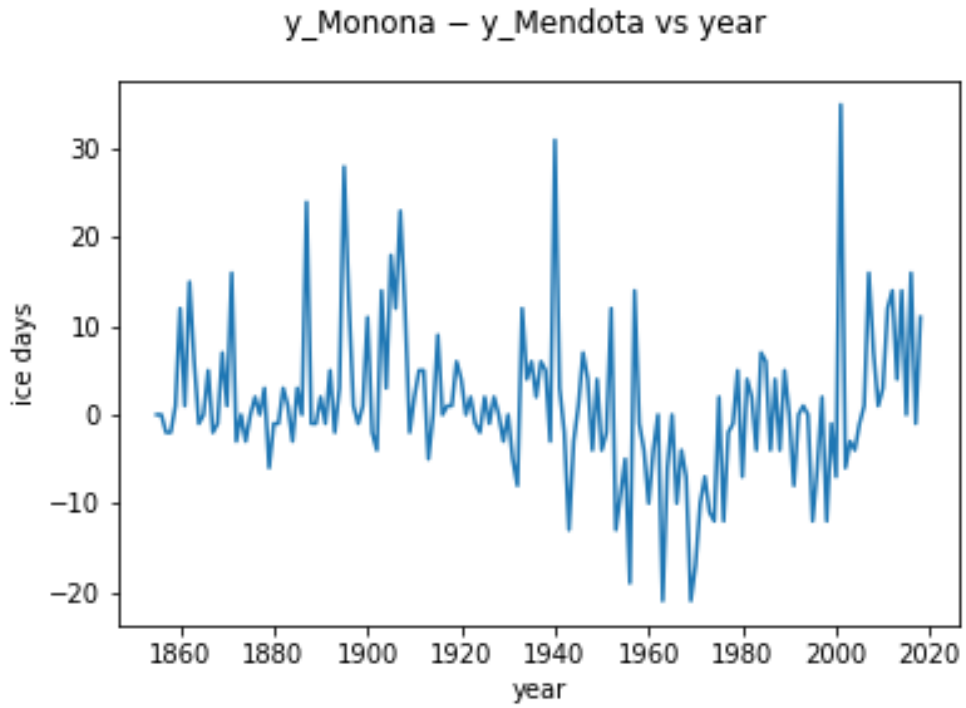

Fig 2: $y_{Monona} - y_{Mendota}$ vs year

2. Split the datasets: $x \leq 1970$ as training, and $x > 1970$ as test. (Comment: due to the temporal nature this is NOT an iid split. But we will work with it.) On the training set, compute the sample mean $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and the sample standard deviation $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$ for the two lakes, respectively.

   Monona Lake:
   $\bar{y}_Monona = 108.48275862068965$
   sample standard deviation $= 18.122521543826256$

   Mendota Lake:
   $\bar{y}_Mendota = 107.1896551724138$
   sample standard deviation $= 16.74666159754441$

3. Using training sets, train a linear regression model

$$\hat{y}_{Mendota} = \beta_0 + \beta_1 x + \beta_2 y_{Monona}$$

   to predict $y_{Mendota}$. Note: we are treating y_{Monona} as an observed feature. Do this by finding the closed-form MLE solution for $\beta = (\beta_0, \beta_1, \beta_2)^\top$ (no regularization):

$$\min_{\beta} \frac{1}{n}\sum_{i=1}^{n}(x_i^\top \beta - y_i)^2.$$

   Give the MLE formula in matrix form (define your matrices), then give the MLE value of $\beta_0, \beta_1, \beta_2$.

   Assuming $p(y_i \mid x_i, \beta) \sim N(x_i^T\beta, \sigma^2)$ i.e. $y_i = x_i^T\beta + error_i$ with $error \sim N(0, \sigma^2)$

$$p(y_i \mid x_i, \beta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - x_i^T\beta)^2}{2\sigma^2}}.$$

   MLE solution: To maximize the log-likelihood, differentiate w.r.t β,
   $\arg\max_\beta \sum_{i=1}^{n} log(p(y_i \mid x_i, \beta)) = \arg\max_\beta \sum_{i=1}^{n} -\frac{(y_i - x_i^T\beta)^2}{2\sigma^2}$
   $= \arg\min_\beta \sum_{i=1}^{n}(y_i - x_i^T\beta)^2$.

set $\nabla_\beta \sum_{i=1}^{n}(y_i - x_i^T\beta)^2 = \vec{0}$

$\implies -\sum_{i=1}^{n} 2(y_i - x_i^\top \beta)\nabla_\beta(x_i^\top \beta) = \vec{0} \implies \sum_{i=1}^{n}(x_i x_i^\top \beta - y_i x_i) = \vec{0}$
$\implies \beta = (\sum_{i=1}^{n}(x_i x_i^\top))^{-1}(\sum_{i=1}^{n} y_i x_i)$ Closed form solution

where $x_i = \begin{bmatrix} 1 \\ \text{x} \\ y_{Monona} \end{bmatrix}$

The above can also be represented as-$\beta = (X^\top X)^{-1}X^\top Y$, where $X = \begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_i^\top \\ \vdots \\ x_n^\top \end{bmatrix}$ and $Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix}$

$\beta = (-64.1827663, 0.04122457, 0.85295064)^\top$

4. Using the MLE above, give the (1) mean squared error and (2) R^2 values on the Mendota test set. (You will need to use the Monona test data as observed features.)

   mean squared error = 124.2640948393
   R^2 = 0.7104900715

5. "Reset" to Q3, but this time use gradient descent to learn the β's. Recall our objective function is the mean squared error on the training set:

$$\frac{1}{n}\sum_{i=1}^{n}(x_i^\top \beta - y_i)^2.$$

   Derive the gradient.

$$\frac{\partial(\frac{1}{n}\sum_{i=1}^{n}(x_i^\top \beta - y_i)^2)}{\partial \beta} = \frac{2}{n}\sum_{i=1}^{n}(x_i^\top \beta - y_i)x_i$$

   where $x_i = [1,\ year,\ y_{Monona}]^\top$ and $\beta = [\beta_0, \beta_1, \beta_2]^\top$

   Gradient descent : $\beta_{t+1} = \beta_t - \eta\frac{2}{n}\sum_{i=1}^{n}(x_i^\top \beta - y_i)x_i$

6. Implement gradient descent. Initialize $\beta_0 = \beta_1 = \beta_2 = 0$. Use a fixed stepsize parameter $\eta = 0.1$ and print the first 10 iteration's objective function value. Tell us if further iterations make your gradient descent converge, and if yes when; compare the β's to the closed-form solution. Try other η values and tell us what happens. **Hint:** Update $\beta_0, \beta_1, \beta_2$ simultaneously in an iteration. Don't use a new β_0 to calculate β_1, and so on.

   For $\eta = 0.1$, first 10 iteration's objective function values:-
   6.18035081e+15
   3.33062663e+27
   1.79489386e+39
   9.67278633e+50
   5.21272023e+62
   2.80916494e+74
   1.51387515e+86
   8.15836035e+97
   4.39658735e+109

2.36934623e+121

For $\eta = 0.1$, β values for the first 10 iterations:-
[2.14379310e+01, 4.09649931e+04, 2.37904828e+03]$^\top$
[-1.57206880e+07, -3.00748788e+10, -1.70345107e+09]$^\top$
[1.15405873e+13, 2.20780291e+16, 1.25050386e+15]$^\top$
[-8.47196830e+18, -1.62075254e+22, -9.17997391e+20]$^\top$
[6.21928893e+24, 1.18979769e+28, 6.73903727e+26]$^\top$
[-4.56559249e+30, -8.73432875e+33, -4.94714078e+32]$^\top$
[3.35161061e+36, 6.41188826e+39, 3.63170597e+38]$^\top$
[-2.46042407e+42, -4.70698004e+45, -2.66604264e+44]$^\top$
[1.80620225e+48, 3.45540349e+51, 1.95714725e+50]$^\top$
[-1.32593669e+54, -2.53661864e+57, -1.43674573e+56]$^\top$

For $\eta = 0.1$, the gradient descent does not converge, For smaller values of η (say 10^{-7}, it converges with increasing number of iterations (more than 50000 iterations), but doesn't converge to the closed form solution.

7. As preprocessing, normalize your year and Monona features (but not $y_{Mendota}$). Then repeat Q6.

Normalized: For $\eta = 0.1$, first 10 iteration's objective function values:-
7545.3637323
4849.54746332
3126.82634003
2025.07151775
1319.92599258
868.28896791
578.8049941
393.10799322
273.88330853
197.26029159

For $\eta = 0.1$, the gradient descent does converge after approx 100 iterations, For smaller values of η, it converges with increased number of iterations (like $\eta = 0.01$ takes approx 1000 iterations), to the closed form solution.

closed form solution: $\beta = [107.18965517, 1.38040755, 15.39084445]^\top$

8. "Reset" to Q3 (no normalization, use closed-form solution), but train a linear regression model without using Monona:

$$\hat{y}_{Mendota} = \gamma_0 + \gamma_1 x.$$

   (a) Interpret the sign of γ_1.

   $\gamma_0 = 406.111060, \gamma_1 = -0.156298774$
   The sign of γ_1 is negative.

   (b) Some analysts claim that because β_1 the closed-form solution in Q3 is positive, fixing all other factors, as the years go by the number of Mendota ice days will increase, namely the model in Q3 indicates a cooling trend. Discuss this viewpoint, relate it to question 8(a).

From closed form solution: $\beta = (\beta_0, \beta_1, \beta_2)^\top = (-64.1827663, 0.04122457, 0.85295064)^\top$
we can notice that both β_1 and β_2 are positive but β_2 is much larger than β_1 ($\approx$ 20 times) showing that y_{Monona} has more weight in deciding the number of Mendota ice days. If we need to find the trend of the number of Mendota ice days over the years, our regression formulation should not consider any other features (which may be uncorrelated or irrelevant). Question 8(a) formulation $\hat{y}_{Mendota} = \gamma_0 + \gamma_1 x.$ does exactly that by discarding other features and considering only x features (i.e. the years).

9. Of course, Weka has linear regression. Reset to Q3. Save the training data in .arff format for Weka. Use classifiers / functions / LinearRegression. Choose "Use training set." Bring up Linear Regression options, set "ridge" to 0 so it does not regularize. Run it and tell us the model: it is in the output in the form of "β_1 * year + β_2 * Monona + β_0."

   setting "ridge" to 0
   $0.0412 * year + 0.853 * Monona - 64.1828$

10. Ridge regression.

    (a) Then set ridge to 1 and tell us the resulting Weka model.
    setting "ridge" to 1
    $0.0387 * year + 0.8436 * Monona - 58.3961$

    (b) Meanwhile, derive the closed-form solution in matrix form for the ridge regression problem:

$$\min_{\beta} \left(\frac{1}{n} \sum_{i=1}^{n} (x_i^\top \beta - y_i)^2 \right) + \lambda \|\beta\|_A^2$$

    where

$$\|\beta\|_A^2 := \beta^\top A \beta$$

    and

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

    This A matrix has the effect of NOT regularizing the bias β_0, which is standard practice in ridge regression. Note: Derive the closed-form solution, do not blindly copy lecture notes.

$$\nabla_\beta \left(\left(\frac{1}{n} \sum_{i=1}^{n} (x_i^\top \beta - y_i)^2 \right) + \lambda \beta^\top A \beta \right) = 0$$

$$\frac{2}{n} \sum_{i=1}^{n} x_i (x_i^\top \beta - y_i) + 2\lambda A \beta = 0$$

$$\sum_{i=1}^{n} (\frac{2}{n} x_i x_i^\top + 2\lambda A) \beta = \frac{2}{n} \sum_{i=1}^{n} x_i y_i)$$

$$\beta = \left(\sum_{i=1}^{n} (x_i x_i^\top) + n\lambda A \right)^{-1} \left(\sum_{i=1}^{n} x_i y_i \right)$$

$$\beta = \left(X^\top X + n\lambda A \right)^{-1} \left(X^\top Y \right)$$

    (c) Let $\lambda = 1$ and tell us the value of β from your ridge regression model.

    β = [-6.23294723e+01, 4.04390872e-02, 8.49714502e-01]$^\top$