CSCI 5521: Machine Learning Fundamentals (Spring 2021)

Homework 1

(Due Tue, Feb. 16, 11:59 PM central)

1. **(25 points)** Consider doing least squares regression based on a training set $\mathcal{X} = \{(x^t, r^t)\}_{t=1}^N$ where $x^t \in \mathbb{R}$ is the feature and $r^t \in \mathbb{R}$ is the target value.

   (i) **(10 points)** Consider fitting a linear model of the form

   $$g_1(x) = w_1 x + w_0 \ ,$$

   with unknown parameters $w_1, w_0 \in \mathbb{R}$, which are selected so as to minimize the following empirical loss:

   $$E(w_1, w_0|\mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}(r^t - (w_1 x^t + w_0))^2 \ .$$

   Derive the optimal values of w_1, w_0. For full credit, you must clearly show all steps of the derivation.

   (ii) **(10 points)** Consider fitting a polynomial model of the form

   $$g_2(x) = v_2 x^{2021} + v_1 x + v_0 \ ,$$

   with unknown parameters $v_2, v_1, v_0 \in \mathbb{R}$, which are selected so as to minimize the following empirical loss:

   $$E(v_2, v_1, v_0|\mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}(r^t - (v_2 (x^t)^{2021} + v_1 x^t + v_0))^2 \ .$$

   Derive the optimal values of v_2, v_1, v_0. For full credit, you must clearly show all steps of the derivation.[1]

   (iii) **(5 points)** For a given training set $\mathcal{X}$, let (w_1^*, w_0^*) be the optimal values of (w_1, w_0) in (i) above, and let (v_2^*, v_1^*, v_0^*) be the optimal values of (v_2, v_1, v_0) in (ii) above. Professor Gopher claims that the following is true for any given $\mathcal{X}$:

   $$E(v_2^*, v_1^*, v_0^*|\mathcal{X}) \leq E(w_1^*, w_0^*|\mathcal{X})$$

   Is Professor Gopher's claim correct? Clearly explain your answer. (A correct answer with insufficient or incorrect explanation will not get any credit.)

[1] It is OK to leave the solution in terms of a linear system, say $A\mathbf{v} = \mathbf{b}$, where $A \in \mathbb{R}^{3\times 3}$, $\mathbf{b} \in \mathbb{R}^3$ are known, and $\mathbf{v} = [v_0 \ v_1 \ v_2]^\top \in \mathbb{R}^3$ is a vector of the unknown parameters. If you choose to do this, please also mention your preferred approach to solve such a linear system.

2. **(15 points)** Consider the following 4×4 matrix:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}.$$

   (i) **(5 points)** What are the values of $\text{tr}(A), \text{tr}(A^\top), \text{tr}(A^\top A)$, and $\text{tr}(AA^\top)$. You can use Python libraries like numpy for the computations.

   (ii) **(5 points)** From a geometric perspective, explain how $|A|$ (determinant of A) can be computed.

   (iii) **(5 points)** Are the rows of A linearly independent? Clearly explain your answer. (A correct answer with insufficient or incorrect explanation will not get any credit. You can use Python libraries to arrive at your answer. If you do that, clearly explain what you did and why. Note, there is a way to arrive at the answer without using Python libraries.)

3. **(30 points)** Let $\mathcal{X} = \{x^1, \ldots, x^N\}$ be a set of N samples drawn i.i.d. from an univariate distribution with density function $p(x|\theta)$, where θ is an unknown parameter. In general, θ will belong to a specified subset of $\mathbb{R}$, the set of real numbers. For the following choices of $p(x|\theta)$, derive the maximum likelihood estimate (MLE) of θ based on the samples $\mathcal{X}$:

   (a) **(10 points)** $p(x|\theta) = \frac{1}{\sqrt{2\pi}\theta} \exp\left(-\frac{(x-2)^2}{2\theta^2}\right), \theta > 0.$

   (b) **(10 points)** $p(x|\theta) = \frac{1}{\theta} \exp\left(-\frac{x}{\theta}\right), 0 \leq x < \infty, \theta > 0.$

   (c) **(10 points)** $p(x|\theta) = \frac{1}{\theta}\ , 0 \leq x \leq \theta, \theta > 0.$

   You must show all steps of your derivation. A correct answer without showing the steps will not receive any credit.

Programming assignment: The next problem involves programming. We will consider two datasets for this problem:

(a) `Boston`: The Boston housing dataset comes prepackaged with scikit-learn. The dataset has 506 data points, 13 features, and 1 target (response) variable. You can find more information about the dataset here: `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html`.

While the original dataset is for a regression problem, we will create two classification datasets for the homework. Note that you only need to work with the `target` t to create these classification dataset, the `data` X should not be changed.

First, load the dataset in with the following commands:

```
import sklearn as sk

X, t = sk.datasets.load_boston(return_X_y=True)
```

Then, create the two following data sets.

i. `Boston50`: Let τ_{50} be the median (50th percentile) over all t (response) values. Create a 2-class classification problem such that one class corresponds to label $r = 1$ if $t \geq \tau_{50}$ and the other class corresponds to label $r = 0$ if $t < \tau_{50}$. By construction, note that the class priors will be $p(r = 1) \approx \frac{1}{2}, p(r = 0) \approx \frac{1}{2}$.

ii. `Boston75:` Let τ_{75} be the 75th percentile over all t (response) values. Create a 2-class classification problem such that one class corresponds to label $r = 1$ if $t \geq \tau_{75}$ and the other class corresponds to label $r = 0$ if $t < \tau_{75}$. By construction, note that the class priors will be $p(r = 1) \approx \frac{1}{4}, p(r = 0) \approx \frac{3}{4}$.

(b) `Digits`: The digits dataset comes prepackaged with scikit-learn. The dataset has 1797 data points, 64 features, and 10 classes corresponding to ten numbers $0, 1, \ldots, 9$. You can find more information about the dataset here: `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html`.

The `Boston50` and `Boston75` datasets will be used for binary (2-class) classification and the `Digits` dataset will be used for 10-class classification.

4. **(30 points)** We will consider three methods from the machine learning library `scikit-learn`: `LinearSVC`[2], `SVC`[3], and `LogisticRegression`[4]. Use the following parameters for the different methods mentioned:

   `LinearSVC: max_iter=2000`

   `SVC: gamma='scale', C=10`

   `LogisticRegression: penalty='l2', solver='lbfgs', multi_class='multinomial', max_iter=5000`

   Write code for `my_cross_val(method,`X`,`r`,`k`)` which performs k-fold cross-validation on the data (X, r) (X is a $N \times d$ matrix where the rows are the samples and columns are the features, and r is a N-dimensional vector of class labels) using `method`, and returns the error rate in each fold. Using `my_cross_val`, report the error rates in each fold as well as the mean and standard deviation of error rates across folds for the three methods: `LinearSVC`, `SVC`, and `LogisticRegression`, applied to the three classification datasets: `Boston50`, `Boston75`, and `Digits`.

   Please submit (a) **code** and (b) **summary of results** for `my_cross_val`:

   (a) **Code**: Submit the main file `my_cross_val.py` which contains the two functions `q4()` and `my_cross_val(method,`X`,`r`,`k`)`.

   `q4` function has no input and is used to prepare the datasets, and make calls to the function `my_cross_val(method,`X`,`r`,`k`)` to generate the results for each dataset and each method.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

`my_cross_val` function has **input**: (1) `method`, which specifies the (class) name of one of the three classification methods under consideration, (2) X,r, which is data for the 2-class or 10-class classification problem, (3) k, the number of folds for cross validation, and **output**: (1) the validation set error rates for each of the k folds.

Make sure the calls to `my_cross_val`(`method`,X,r,k) are made in the following order and add a print to the terminal before each call to show which method and dataset is being used:

1. `LinearSVC` with `Boston50`
2. `LinearSVC` with `Boston75`
3. `LinearSVC` with `Digits`
4. `SVC` with `Boston50`
5. `SVC` with `Boston75`
6. `SVC` with `Digits`
7. `LogisticRegression` with `Boston50`
8. `LogisticRegression` with `Boston75`
9. `LogisticRegression` with `Digits`

For example, the first call to `my_cross_val`(`method`,X,r,k) with $k = 10$ should result in the following output:

```
Error rates for LinearSVC with Boston50:
Fold 1:  ###
Fold 2:  ###
...
Fold 10:  ###
Mean:  ###
Standard Deviation:  ###
```

(b) **Summary of results**: For each dataset and each method, report the validation set error rates for each of the $k = 10$ folds, the mean error rate over the k folds, and the standard deviation of the error rates over the k folds. Make a table to present the results for each method and each dataset (9 tables in total). Include a column in the table for each fold, and add two columns at the end to show the overall mean error rate and standard deviation over the k folds. For example:

Error rates for LinearSVC with Boston50											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
#	#	#	#	#	#	#	#	#	#	#	#

Additional instructions: Code can only be written in Python 3.6+; no other programming languages will be accepted. One should be able to execute all programs from the Python command prompt or terminal. Please specify instructions on how to run your program in the README file.

Each function must take the inputs in the order specified in the problem and display the textual output via the terminal and plots/figures should be included in the report.

For each part, you can submit additional files/functions as needed. In your code, you can only use machine learning libraries such as those available from scikit-learn as specified in the problem description. You may use libraries for basic matrix computations and plotting such as numpy, pandas, and matplotlib. Put comments in your code so that one can follow the key parts and steps in your code.

Your code must be runnable on a CSE lab machine (e.g., csel-kh1260-01.cselabs.umn.edu). One option is to SSH into a machine. Learn about SSH at these links: `https://cseit.umn.edu/knowledge-help/learn-about-ssh`, `https://cseit.umn.edu/knowledge-help/choose-ssh-tool`, and `https://cseit.umn.edu/knowledge-help/remote-linux-applications-over-ssh`.

Instructions

Follow the rules strictly. If we cannot run your code, you will not get any credit.

- **Things to submit**

  1. hw1.pdf: A document which contains the solution to Problems 1, 2, 3, and 4 including the summary of results for problem 4. This document must be in PDF format (no word, photo, etc. is accepted). If you submit a scanned copy of a hand-written document, make sure the copy is clearly readable, otherwise no credit may be given.
  2. `my_cross_val.py`: Code for Problem 4.
  3. README.txt: README file that contains your name, student ID, email, instructions on how to run your code, any assumptions you are making, and any other necessary details.
  4. Any other files, except the data, which are necessary for your code (such as package dependencies like a requirements.txt or yml file).

Homework Policy. (1) You are encouraged to collaborate with your classmates on homework problems, but each person must write up the final solutions individually. You need to list in the README.txt which problems were a collaborative effort and with whom. (2) Regarding online resources, you should **not**:

- Google around for solutions to homework problems,
- Ask for help on online,
- Look up things/post on sites like Quora, StackExchange, etc.