MTH 4320/5320 Homework 2

Deadline: Wednesday, September 2

Instructions

- Preferably, create 1 notebook file. Or, create 1 PDF and submit code files separately.

- You document should be similar to my notes for Weeks 1-2 in GitHub, containing three parts:

    1. Typed work or pictures of your written work for problem 1
    2. Text explanations for problems 2-4
    3. Well-commented code for problems 2-4

- Submit your notebook or PDF + code files (+ any other files, if needed) in Canvas.

Problems

1. Find an exact matrix expression for β minimizing the ridge regression loss function,

$$\min_{\beta} L(\lambda, \beta) = \min_{\beta} \left(\|X\beta - y\|_2^2 + \lambda \|\beta\|_2^2 \right),$$

   where $\lambda \in \mathbb{R}$, $X \in \mathbb{R}^{n \times d}$, and $y \in \mathbb{R}^n$ are constant. [5 points]

2. Write a gradient descent function with momentum. Then, fit a linear least squares model to predict house prices from other data to the whole Mount Pleasant Real Estate dataset available at

   http://www.hawkeslearning.com/Statistics/dis/datasets.html

   using two approaches: (1) gradient descent method and (2) gradient descent method *with momentum* and compare the speed for a few combinations of the learning rate and momentum parameter. Compare the speed[1] and accuracy.

   [*Hint*. You need to preprocess the data to remove non-numerical data and convert yes/no variables to binary.] [10 points]

3. Write a class using gradient descent with momentum for linear regression using an elastic net loss function. [10 points]

4. Download the Weather in Szeged 2006-2016 dataset available at

   https://www.kaggle.com/budincsevity/szeged-weather

   Tune the hyperparameters to get the best fit you can to predict the temperatures from other data in the set. (You do *not* have to use all of the variables.)

   Randomly split the dataset into 60%/20%/20% training/validation/testing sets. When tuning hyperparameters, test on the validation set. At the end, use the test set.[2] [5 points]

Bonus: Whoever achieves the lowest loss on the test set gets +5 points.

---

[1] The time.time() function at https://docs.python.org/3/library/time.html can be used.

[2] Use **random_state = 1** in **train_test_split** or set the random seed to 1 in any language before splitting data.