Fall 2022 CS165B: Introduction to Machine Learning – Homework 2
Due: **Sunday, Oct 30th, 11:59 pm PST**

Note: Please upload your written part as a PDF, and please upload your code as a separate file in (".py" or ".ipynb") form. Both files should be uploaded to Gauchospace.

1. (10 points) We are given a set of data points $x_1, x_2, \ldots, x_n$ that are i.i.d. drawn from the density function:

$$f(x|\sigma) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right), -\infty < x < \infty, \sigma > 0$$

   Find the maximum likelihood estimate of σ.

2. (10 points) Recall the objective function for linear regression can be expressed as

$$E(\mathbf{w}) = \frac{1}{N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2.$$

   Minimizing this function with respect to $\mathbf{w}$ leads to the optimal $\mathbf{w}^*$ as $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. This solution holds only when $\mathbf{X}^T\mathbf{X}$ is nonsingular. To overcome this problem, the following objective function is commonly minimized instead:

$$E_2(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda\|\mathbf{w}\|^2,$$

   where $\lambda > 0$ is a user-specified parameter. Please do the following:

   (a) (5 points) Derive the optimal $\mathbf{w}^*$ that minimizes $E_2(\mathbf{w})$.

   (b) (5 points) Explain how this new objective function can overcome the singularity problem of $\mathbf{X}^T\mathbf{X}$. [Hint: Think about positive definite and positive semi-definite definitions.]

3. (20 points) Consider the hat matrix $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$, where $\mathbf{X}$ is an N by $d+1$ matrix, and $\mathbf{X}^T\mathbf{X}$ is invertible.

   (a) (5 points) Show that $\mathbf{H}$ is symmetric.

   (b) (5 points) Show that $\mathbf{H}^K = \mathbf{H}$ for any positive integer K.

   (c) (5 points) If $\mathbf{I}$ is the identity matrix of size N, show that $(\mathbf{I} - \mathbf{H})^K = \mathbf{I} - \mathbf{H}$ for any positive integer K.

   (d) (5 points) Show that $\text{trace}(\mathbf{H}) = d+1$, where the trace is the sum of diagonal elements. [Hint: $\text{trace}(\mathbf{AB}) = \text{trace}(\mathbf{BA})$]

4. (20 points) Implement a gradient solver and a stochastic gradient solver for (a) least squares regression (lsqr) and (b) ridge regression (least squares with ℓ_2-norm regularization). For both lsqr and ridge, compare the result models from the two optimizers (gradient descent and stochastic gradient descent) and the closed-form solutions (where we set gradient to zero and solved the equation), using the ℓ_2-norm distance to measure the difference between the two solutions.
   Deliverable: (1) Code: Your implementation of gradient solver, stochastic gradient solver, and closed form solver. Your code then generates a random dataset, and your code compute solutions using 3 solvers. Finally, your code compare weights and report ℓ_2-norm distance

for each pair. Repeat this whole process with and without regularization. (2) PDF: Result of the ℓ_2-norm distance of weights between 3 comparisons: GD vs SGD, GD vs closed-form, and SGD vs closed-form. Repeat this whole process with and without regularization.
Hyper-parameters and Dataset: The dataset should be in a proper size. We recommend 1,000 data points with 50 number of features (dimensions) for this question. For regularization parameters, we suggest $1e-6$. Notice these are not hard requirements.

5. (10 points) Use your randomized datasets to compute the objective functions (lsqr and ridge, respectively) in iterations, plot the converagence graph (x-axis: iteration number, y-axis: objective function). Note that when computing the objective function, $E(w(t))$, you should use all the data points, for both gradient descent and stochastic gradient descent.
Deliverable: (1) Code: Using previous implemented solvers and datasets to generate the curves (with and without regularization; gradient descent and stochastic gradient descent). (2) PDF: The result as curves.

6. (10 points) Previously, stochastic gradient descent we introduced uses 1 data point in each iteration to estimate the gradient. In fact, the stochastic gradient descent can use a batch of data points to estimate/compute the gradient to accelerate the convergence speed. Implement batched version gradient descent for ridge regression. For this question, create a random dataset of 5000 data points and 1000 dimensions. Plot the convergence graph of stochastic gradient descent with batch size of $[1, 10, 100]$.
Deliverable: (1) Code: Stochastic gradient descent with different batch sizes. (2) PDF: 3 Convergence graphs with different batch sizes.

7. (20 points) Now, we want to study on how the training data influence generalization performance using closed-form solution.

   (a) (10 points) For this question, generating 1,000 random data points with 100 dimensions. Use 100 data points as test dataset. Among the remaining 900 data points, using different number of them (e.g. $[50, 100, ..., 900]$) to compute the closed form solution, and compute the MSE error on the 100 test data points. Plot the number of training examples used versus the test error.

   (b) (10 points) For this question, generating 1,000 random data points with different dimensions (e.g. $[100, 150, ..., 450]$). Use 100 data points as test dataset, and use the remaining 900 data points to compute the closed form solution. Compute the corresponding MSE error on the 100 test data points. Plot the number of dimensions versus the test error.

8. (10 points) Finally, we want to study on the computational complexity of the closed-form solution. For this question, generating 1,000 data points. The value of the dimensions varies in $[100, 200, ..., 2,000]$. For each possible dimension, use your previous code to calculate the closed-form solution of linear regression, and record the time needed for computation. Plot the number of dimensions versus the computational time.