CS 583: Assignment 1

Due Date: 11:59PM, February 24, 2022

Please submit a PDF file containing the first four questions and .ipynb or .py file containing your code to canvas.

Q1 (6%). $\mathbf{x} = [5, -3, -1, 2]^T$ is a 4-dimensional vector. Calculate the following values:

1. the squared ℓ_2-norm of $\mathbf{x}$:

   $\|\mathbf{x}\|_2^2 =$

2. the ℓ_1-norm of $\mathbf{x}$:

   $\|\mathbf{x}\|_1 =$

3. the inner product of $\mathbf{x}$ and a, where $\mathbf{a} = [4, -2, 6, -1]^T$:

   $\mathbf{a}^T\mathbf{x} =$

Q2 (4%). The matrix $\mathbf{A} \in \mathbb{R}^{2\times 3}$ and vector $\mathbf{b} \in \mathbb{R}^3$ are defined in the following:

$$\mathbf{A} = \begin{bmatrix} 6 & 1 & -2 \\ -5 & 7 & 9 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} -4 \\ 5 \\ 2 \end{bmatrix}.$$

Caculate the following values:

1. the matrix-vector product:

   $\mathbf{Ab} =$

2. the matrix-matrix product:

   $\mathbf{AA}^T =$

Q3 (6%). Let $\mathbf{x} = [x_1, x_2, x_3]$ and $y = \frac{x_1^2}{2} + \log_e x_2 - \frac{x_1}{x_3}$. Calculate the following:

$\frac{dy}{d\mathbf{x}}$ at $\mathbf{x} = [9, 1, \frac{1}{2}]$:

Hint: the answer should be a three dimensional vector.

Q4 (4%). $\mathbf{X}$ is an $n \times d$ matrix, $\mathbf{y}$ is an $n \times 1$ vector, and $\mathbf{w}$ is an $d \times 1$ vector. Let $f(\mathbf{w}) = \|\mathbf{Xw} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$. Calculate the following:

$\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} =$

Q5 (80%). In this problem, you will need to write python codes and build logistic regression classifiers on breast cancer dataset to predict whether the cancer is malignant or benign on the patients. The dataset contains 569 samples and 30 features. You may refer to `https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)` for more information about the dataset.

Your implementations should include:

- Load data, clean data and partition them into training and testing data.
- Build logistic regression and L2-regularized logistic regression models.
- Implement three gradient descent algorithms for each model: Batch Gradient Descent (GD), Mini-Batch Gradient Descent (MB-SGD) and Stochastic Gradient Descent (SGD).
- Compare the loss curve of three gradient descent algorithms (GD/MB-SGD/SGD).
- Compare logistic regression and regularized version in terms of training and testing error.
- Try to tune different parameters (regularization parameter, learning rate, etc.) to see their effects.

You may find more detailed procedure in the IPython notebook file. You could use sklearn or any other packages to load and process the data, but you **can not** directly use the package to train the model.