Homework Assignment # 2

Due: Thursday, October 24, 2019, 11:59 p.m.
Total marks: 100

Question 1. [25 MARKS]

Let $X_1, \ldots, X_n$ be i.i.d. Gaussian random variables, each having an unknown mean θ and known variance σ_0^2.

(a) [5 MARKS] Assume θ is itself selected from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ having a known mean μ and a known variance σ^2. What is the maximum a posteriori (MAP) estimate of θ?

(b) [10 MARKS] Assume θ is itself selected from a Laplace distribution $\mathcal{L}(\mu, b)$ having a known mean (location) μ and a known scale (diversity) b. Recall that the pdf for a Laplace distribution is

$$p(x) = \frac{1}{2b} \exp\left(\frac{-|x-\mu|}{b}\right)$$

For simplicity, assume $\mu = 0$. What is the maximum a posteriori estimate of θ? If you cannot find a closed form solution, explain how you would use an iterative approach to obtain the solution.

(c) [10 MARKS] Now assume that we have **multivariate** i.i.d. Gaussian random variables, $\mathbf{X}_1, \ldots, \mathbf{X}_n$ with each $\mathbf{X}_i \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}_0)$ for some unknown mean $\boldsymbol{\theta} \in \mathbb{R}^d$ and known $\boldsymbol{\Sigma}_0 = \mathbf{I} \in \mathbb{R}^{d \times d}$, where $\mathbf{I}$ is the identity matrix. Assume $\boldsymbol{\theta} \in \mathbb{R}^d$ is selected from a zero-mean multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \sigma^2\mathbf{I})$ and a known variance parameter σ^2 on the diagonal. What is the MAP estimate of $\boldsymbol{\theta}$?

Question 2. [75 MARKS]

In this question, you will implement variants of linear regression. We will be examining some of the practical aspects of implementing regression, including for a large number of features and samples. An initial script in python has been given to you, called `script_regression.py`, and associated python files. You will be running on a UCI dataset for CT slices[1], with 385 features and 53,500 samples. Baseline algorithms, including mean and random predictions, are used to serve as sanity checks. We should be able to outperform random predictions, and the mean value of the target in the training set.

(a) [5 MARKS] The main linear regression class is `FSLinearRegression`. The FS stands for FeatureSelect. The provided implementation has subselected features and then simply explicitly solved for $\mathbf{w} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$. Increase the number of selected features (up to all the features). What do you find? How can this be remedied?

(b) [5 MARKS] The current code averages the error over multiple training, test sets, subsampled from the data. Modify the code to additionally report the standard error over these multiple runs (i.e., the sample standard deviation divided by the square root of the number of runs).

(c) [5 MARKS] Now implement Ridge Regression, where a ridge regularizer $\lambda\|\mathbf{w}\|_2^2$ is added to the optimization. Run this algorithm on all the features. How does the result differ from (a)? Discuss the result in a couple of sentences, for one regularization parameter, $\lambda = 0.01$.

(d) [20 MARKS] Now imagine that you want to try a feature selection method and you've heard

[1] `https://archive.ics.uci.edu/ml/datasets/Relative+location+of+CT+slices+on+axial+axis`

all about this amazing and mysterious Lasso. Lasso can often be described as an algorithm, or otherwise as an objective with a least-squares loss and ℓ_1 regularizer. It is more suitably thought of as the objective, rather than an algorithm, as there are many algorithms to solve the Lasso. Implement an iterative solution approach that uses the soft thresholding operator (also called the shrinkage operator), described in the chapter on advanced optimization techniques.

(e) [20 MARKS] Implement a stochastic gradient descent approach to obtaining the linear regression solution (see the chapter on advanced optimization techniques). Report the error, for a step-size of 0.01 and 1000 epochs.

(f) [20 MARKS] Implement batch gradient descent for linear regression, using line search. Compare stochastic gradient descent to batch gradient descent, in terms of the number of times the entire training set is processed. Set the step-size to 0.01 for stochastic gradient descent. Report the error versus epochs, where one epoch involves processing the training set once. Report the error versus runtime.

Bonus (mandatory for 566). [20 MARKS]

Previously, you implemented several variants of linear regression using scalar learning rates and line search. While for small problems setting a scalar learning rate can be done through parameter studies, adjusting learning rates according to our training data on each training step would be beneficial. In this question you will be implementing two adaptive learning rate algorithms for stochastic gradient descent, which set learning rates for each feature based on the observed data. For both algorithms, report the error after 1000 iterations.

(a) [5 MARKS]

Momentum is one of the most simple stepsize adaptation methods which uses past gradient information to accelerate learning. Implement Momentum as an exponential moving average over gradients as done *Adam: A Method for Stochastic Optimization* [2].

(b) [5 MARKS]

Using the same reference as above, implement the *adaptive moment estimation* stepsize selection algorithm; also known as *Adam*.

(c) [10 MARKS]

Modify your implementation of Momentum and Adam to remove the initialization bias caused by having insufficient prior gradient information at the beginning of learning. Show that Momentum (without bias correction) is a biased estimate of the expected gradient.

Homework policies:

Your assignment should be submitted as a single pdf document and a zip file with code, on eClass. The questions must be typed (e.g., Latex) or must be written legibly and scanned. All code (if applicable) should be turned in when you submit your assignment.

Because assignments are more for learning, and less for evaluation, grading will be based on coarse bins. **The grading is atypical**. For grades between (1) 80-100, we round-up to 100; (2) 60-80, we round-up to 80; (3) 40-60, we round-up to 60; and (4) **0-40, we round down to 0**. The last bin is to discourage quickly throwing together some answers to get some marks. The goal for

[2] `https://arxiv.org/pdf/1412.6980.pdf`

the assignments is to help you learn the material, and completing less than 50% of the assignment is ineffective for learning.

Policy for late submission of assignments: Unless there are legitimate circumstances, grades for late assignments will be penalized by 1 full bin per day late (see above) and any assignment submitted more than 2 days late will receive a grade of 0. **Note:** You can still submit your assignment after 2 days (up to 5 days past the deadline) to receive feedback but you will not get credit for completing the assignment.

All assignments are individual, except when collaboration is explicitly allowed. All the sources used for the problem solution must be acknowledged, e.g. web sites, books, research papers, personal communication with people, etc. Academic honesty is taken seriously; for detailed information see the University of Alberta Code of Student Behaviour.

Good luck!