Homework Assignment # 2

Due: Wednesday, October 12, 2016, 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 (including all the way to including 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 both the standard error over these multiple 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 parameters, $\lambda = 0.01$.

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

(d) [15 MARKS] Imagine that the dataset size continues to grow, which causes the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ for n samples and d features to become quite large. One option is to go back to subselecting features. In this question, you will implement a simple greedy algorithm for selecting a subset of features, often called forward greedy selection or matching pursuit (see for example the pseudocode in"On the Consistency of Feature Selection using Greedy Least Squares Regression", Zhang, 2009). The idea is to greedily add one new feature on each iteration, based on its correlation with the residual: the feature with the maximum absolute dot product with the residual. On each step, for current subset s of features and residual $\mathbf{R} = \mathbf{X}(:, s)\mathbf{w} - \mathbf{y}$, one adds the feature i with maximal $|\mathbf{X}(:, i)^\top \mathbf{R}|$. The coefficients are then recomputed for the current subset of features, and another feature considered. This iteration ends once the residual error is below some threshold, or if $\max_i |\mathbf{X}(:, i)^\top \mathbf{R}|$ is below some threshold. Implement `MPLinearRegression` and report error.

(e) [15 MARKS] Now imagine that you want to try a different feature selection method and you've heard 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). Discuss the impact of the choice of regularization parameter on the error of `LassoRegression`.

(f) [15 MARKS] Now imagine that your dataset has gotten even larger, to the point where dropping features is not enough. Instead of removing features, implement a stochastic optimization approach to obtaining the linear regression solution (see Section 4.5.3). Explain your implementation choices. Report the error.

(g) [15 MARKS] Implement batch gradient descent for linear regression. 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, and implement a reasonable approach to select the step-size for batch gradient descent. Report the error versus epochs, where one epoch involves processing the training set once. Report the error versus runtime.

Homework policies:

Your assignment will be submitted as a single pdf document and a zip file with code, on canvas. The questions must be typed; for example, in Latex, Microsoft Word, Lyx, etc. or must be written legibly and scanned. Images may be scanned and inserted into the document if it is too complicated to draw them properly. All code (if applicable) should be turned in when you submit your assignment. Use Matlab, Python, R, Java or C.

Policy for late submission assignments: Unless there are legitimate circumstances, late assignments will be accepted up to 5 days after the due date and graded using the following rule:

on time: your score 1

1 day late: your score 0.9

2 days late: your score 0.7

3 days late: your score 0.5

4 days late: your score 0.3

5 days late: your score 0.1

For example, this means that if you submit 3 days late and get 80 points for your answers, your total number of points will be $80 \times 0.5 = 40$ points.

All assignments are individual, except when collaboration is explicitly allowed. All the sources used for 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 Indiana University Code of Student Rights, Responsibilities, and Conduct.

Good luck!