10-315 Intro to Machine Learning HW4

INSTRUCTIONS

- **Due: Monday, 24 February 2020 at 11:59 PM EDT.**
- **Format:** Complete this pdf with your work and answers. Whether you edit the latex source, use a pdf annotator, or hand write / scan, make sure that your answers (tex'ed, typed, or handwritten) are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.
- **How to submit:** Submit a pdf with your answers on Gradescope. Log in and click on our class 10-315, click on the appropriate *Written* assignment, and upload your pdf containing your answers. Don't forget to submit the associated *Programming* component on Gradescope if there is any programming required.
- **Policy:** See the course website for homework policies and Academic Integrity.

Name	
Andrew ID	
Hours to complete (both written and programming)?	

For staff use only

Q1	Q2	Q3	Q4	Q5	Total
/14	/26	/20	/20	/10	/90

Q1. [14 pts] MLE

Consider the following distribution with parameters k and α:

$$p(x \mid k, \alpha) = \begin{cases} \frac{\alpha k^{\alpha}}{x^{\alpha+1}} & x \in [k, \infty) \\ 0 & \text{otherwise} \end{cases}$$

We also have that $k \in (0, \infty)$ and $\alpha \in (0, \infty)$.

This distribution is often used for modelling the distribution of wealth in a society, fitting the trend that a large portion of wealth is held by a small fraction of the population. This is due to its nature as a skewed, heavy-tailed distribution.

Suppose you have a dataset $\mathcal{D}$ which contains N i.i.d samples x_1,x_2,...,x_N drawn from the above distribution.

(a) [6 pts] Derive the log-likelihood $\ell(k, \alpha; \mathcal{D})$.

$\ell(k, \alpha; \mathcal{D})$**:**

(b) [6 pts] Give the MLE for the parameter α.

$\hat{\alpha}_{MLE}$**:**

(c) [2 pts] Next, give the MLE for the parameter k.

Hint: You may be tempted to set k to infinity, but when $k = \infty$, what happens to $p(x \mid k, \alpha)$?

$\hat{k}_{MLE}$**:**

Q2. [26 pts] Priors and Regularization

Remember our probabilistic linear regression set-up, in which we assume the relationship between response $y \in \mathbb{R}$ and an input $\mathbf{x} \in \mathbb{R}^M$ is linear:

$y = \mathbf{w}^T\mathbf{x} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$

Recall that the pdf of a Gaussian random variable $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$ is given by:

$p(\epsilon \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(\epsilon-\mu)^2}$

Also, the pdf of a Laplacian random variable $z \sim \mathcal{L}(\mu, b)$ is given by:

$p(z \mid \mu, b) = \frac{1}{2b} e^{-\frac{1}{b}(|z-\mu|)}$

With this in mind, suppose we have a dataset of N i.i.d. samples $\mathcal{D} = \{(y^{(n)}, \mathbf{x}^{(n)})\}_{n=1}^N$

(a) [6 pts] Assume that each weight $\mathbf{w}_i, i \in [1, M]$ is drawn from $\mathcal{L}(0, b)$. Derive the prior on the weights $p(\mathbf{w})$.

$p(\mathbf{w})$**:**

(b) [6 pts]

With your answer from part a, derive the conditional likelihood times the prior, $p(\mathbf{y} \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})$, which is proportional to the posterior $p(\mathbf{w} \mid \mathcal{D})$.

$p(\mathbf{y} \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})$**:**

(c) [7 pts] Derive the log of the conditional likelihood times the prior, $\log\left(p(\mathbf{y} \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})\right)$.

$\log\left(p(\mathbf{y} \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})\right)$**:**

(d) [7 pts] Argue that $\text{argmin}_{\mathbf{w}}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_1$ is equivalent to finding $\mathbf{w}$ by minimizing the negative log of the posterior. Specifically, write λ in terms of the log posterior parameters, b and σ^2.

Answer:

λ**:**

Q3. [20 pts] Regularization and Augmentation

So far, we have seen ridge regression as the minimization the sum of the MSE and $\lambda||\mathbf{w}||_2^2$, where λ is a adjustable parameter.
Now, we introduce another form of ridge regression. In fact, ridge regression estimates can be obtained by ordinary least squares regression on an augmented data set.
To express the idea of data-set augmentation, it helps to introduce Block Matrices. A Block Matrix is a matrix that is defined using smaller matrices, called blocks. They have the following general forms:
A horizontal block matrix: $\mathbf{C} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}$, where $\mathbf{A}$ has dimension $M \times N_1$, $\mathbf{B}$ has dimension $M \times N_2$, and $\mathbf{C}$ has dimension $M \times (N_1 + N_2)$.
A vertical block matrix: $\mathbf{C} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$, where $\mathbf{A}$ has dimension $M_1 \times N$, $\mathbf{B}$ has dimension $M_2 \times N$, and $\mathbf{C}$ has dimension $(M_1 + M_2) \times N$.
Now, suppose in a Ridge Regression setup, we have a dataset $\mathcal{D} = \{(y^{(i)}, \mathbf{x}^{(i)})\}_{i=1}^N$, with each $\mathbf{x} \in \mathbb{R}^M$, weight vector $\mathbf{w} \in \mathbb{R}^M$, and we wish to minimize

$$\sum_{i=1}^{N} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2 + \lambda ||\mathbf{w}||_2^2$$

Call the design matrix $\mathbf{X}$ and the response vector $\mathbf{y}$. Augmenting the data set refers to adding new samples to the existing collection of samples, resulting in a new design matrix $\tilde{\mathbf{X}}$ and new target vector $\tilde{\mathbf{y}}$.
We wish to find $\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{A} \end{bmatrix}$ and $\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{b} \end{bmatrix}$ s.t. solving the ordinary linear regression problem for the augmented data set $\tilde{\mathbf{X}}, \tilde{\mathbf{y}}$ yields the sample weight vector $\mathbf{w}$ as solving the Ridge Regression problem for $\mathbf{X}, \mathbf{y}$.

(a) [10 pts] Propose a choice for $\mathbf{A}$ and $\mathbf{b}$. *Hint:* consider the closed-form solution of Ridge Regression problem.

Answer:

(b) [10 pts] For your choice of $\mathbf{A}$ and $\mathbf{b}$ above, prove that they satisfy the desired condition: that is, prove that solution of regular linear regression problem with data set $\tilde{\mathbf{X}}$, $\tilde{\mathbf{y}}$ is indeed the solution of the Ridge Regression problem on the original data set $\mathbf{X}, \mathbf{y}$.
Recall the closed-form solution of Ridge Regression: $\hat{\beta}_{Ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$.
Let $\hat{\beta}_{Aug} = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{y}}$. Show that $\hat{\beta}_{Ridge} = \hat{\beta}_{Aug}$.

Answer:

Q4. [20 pts] Regularization in Logistic Regression

Let $\mathcal{D} = \{(y^{(n)}, \mathbf{x}^{(n)})\}_{n=1}^{N}$ be a training set, with each $\mathbf{x}^{(n)} \in \mathbb{R}^M$ and each $y^{(n)} \in \{0, 1\}$. The objective function ℓ_2-regularized logistic regression is the negative log likelihood times the negative log of the prior:

$$J(\mathbf{w}) = -\sum_{i=1}^{N}(y^{(n)} \log(\mu^{(n)}) + (1 - y^{(n)}) \log(1 - \mu^{(n)})) + \lambda \frac{1}{2}||\mathbf{w}||_2^2 \tag{1}$$

where $\mu^{(n)} = 1/(1 + \exp(-\mathbf{w}^T \mathbf{x}^{(n)}))$, and $\lambda \geq 0$ is the regularization parameter. In this problem, you will use both gradient descent and Newtons method to minimize this objective function on a small training set. You'll derive the math in this written question and then code up the optimizations in the programming component.

Heres the setup: We have four data points ($\in \mathbb{R}^2$), two of class 1, and two of class 0. Here is the data (you may want to draw this on paper to see what the data looks like):

$$\mathbf{X} = \begin{bmatrix} 0 & 3 \\ 1 & 3 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Here, $\mathbf{X}$ is the design matrix and each row $\mathbf{X}_n^T = \mathbf{x}^{(n)}$ is a data point. Note that despite the data not being separable by a boundary through the origin, we have NOT included a bias term. Thus, our weight vector will simply be two-dimensional.

For the written and programming questions, you may use the fact that the Newton method follows the following update rule:

$$\mathbf{w}^{(i+1)} \leftarrow \mathbf{w}^{(i)} - \mathbf{H}(\mathbf{w}^{(i)})^{-1} \nabla_{\mathbf{w}} J(\mathbf{w}^{(i)})$$

where

$$\mathbf{H} = \frac{\partial^2 J(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^T} = \frac{\partial}{\partial \mathbf{w}} \left(\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right)^T$$

(a) [10 pts] Derive the gradient of the objective function. Your answer should be a simple matrix-vector expression. Do NOT write your answer in terms of the individual elements of the gradient vector.

$\nabla_{\mathbf{w}} J(w)$**:**

(b) [10 pts] State the Hessian of the objective function. Again, your answer should be a simple matrix-vector expression. You may use the notation $diag(a_i b_i)$ to represent a diagonal matrix where the i-th diagonal entry is the i-th entry in vector $\mathbf{a}$ times the i-th entry in vector $\mathbf{b}$.

H(w):

Q5. [10 pts] Programming

(a) Programming Q1

(i) [2 pts] Include the plots of the objective function for four different values of lambda. Running `python3.6 autograder.py -q Q1` should have auto generated these images and saved it to the `figures` directory within your code directory.

Plot $\lambda = 0$**:**	**Plot** $\lambda = 0.1$**:**
Plot $\lambda = 1$**:**	**Plot** $\lambda = 10$**:**

(ii) [1 pt] What should be the labels for the horizontal and vertical axes?

Labels:

(iii) [1 pt] Describe the effect of different lambda values on the objective function.

Explain:

(b) Programming Q2

(i) [2 pts] Include the plot of the gradient descent for three different learning rates. Running `python3.6 autograder.py -q Q2` should have auto generated this image and saved it to the `figures` directory within your code directory.

Plot:

(ii) [1 pt] Describe the effect of learning rate on gradient descent convergence for this problem.

Explain:

(c) Programming Q3

(i) [2 pts] Include the plots of the Newton's method convergence for two different values of lambda. Running `python3.6 autograder.py -q Q3` should have auto generated these images and saved it to the `figures` directory within your code directory.

Plot $\lambda = 0.1$**:**	**Plot** $\lambda = 2$**:**

(ii) [1 pt] Describe the effect of different lambda values on the Newton's method convergence for this problem.

Explain: