First Name: ____________________ Last Name: ____________________

Student Number: ____________________

ECE 421F — Introduction to Machine Learning
MidTerm Examination

Wed Oct 16^{th}, 2019
4:10-6:00 p.m.

Instructor: Ashish Khisti

Circle your tutorial section:

- TUT0101 Thu 1-3
- TUT0102 Thu 4-6

Instructions

- Please read the following instructions carefully.
- You have 1 hour fifty minutes (1:50) to complete the exam.
- Please make sure that you have a complete exam booklet.
- Please answer *all* questions. Read each question carefully.
- The value of each question is indicated. Allocate your time wisely!
- No additional pages will be collected beyond this answer book. You may use the reverse side of each page if needed to show additional work.
- This examination is closed-book; One 8.5×11 aid-sheet is permitted. A non-programmable calculator is also allowed.
- Good luck!

1. (40 MARKS) Consider a multi-class linear classification problem where the data points are two dimensional, i.e., $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and the labels $y \in \{1, 2, 3\}$. Throughout this problem consider the data-set with following five points:

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), (\mathbf{x}_4, y_4), (\mathbf{x}_5, y_5)\}$$

where the input data-vectors are given by:

$$\mathbf{x}_1 = (-1, 0)^T, \quad \mathbf{x}_2 = (1, 0)^T, \quad \mathbf{x}_3 = (1, 1)^T, \quad \mathbf{x}_4 = (-1, 1)^T, \quad \mathbf{x}_5 = (0, 3)^T$$

and the associated labels are given by

$$y_1 = 1, \quad y_2 = 2, \quad y_3 = 2, \quad y_4 = 1, \quad y_5 = 3$$

Our aim is to find a linear classification rule that classifies this dataset.

10 marks (**a**) Suppose we implement the perceptron learning algorithm for binary classification that finds a perfect classifier separating the data points between the two sets: $\mathcal{S}_1 = \{(\mathbf{x}_1, y_1), (\mathbf{x}_4, y_4)\}$ and $\mathcal{S}_2 = \{(\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3)\}$.
Assume that the initial weight vector $\mathbf{w} = (0, 0, 0)^T$, that each point that falls on the boundary is treated as a mis-classified point and the algorithm visits the points in the following order:

$$\mathbf{x}_1 \to \mathbf{x}_2 \to \mathbf{x}_3 \to \mathbf{x}_4 \to \mathbf{x}_1 \to \mathbf{x}_2 \cdots$$

until it terminates. Show the output of the perceptron algorithm in each step and sketch the final decision boundary when the algorithm terminates. [**Important**: When applying the perceptron update, recall that you have to transform the data vectors to include the constant term i.e., $\mathbf{x}_1 = (-1, 0)^T$ must be transformed to $\tilde{\mathbf{x}}_1 = (1, -1, 0)^T$ etc.]

total/-

[continue part (a) here]

total/10

5 marks

(b) Find any linear classification rule that perfectly separates the data points between the two sets: $\mathcal{S}_{12} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), (\mathbf{x}_4, y_4)\}$ and $\mathcal{S}_3 = \{(\mathbf{x}_5, y_5)\}$. Draw your decision boundary and clearly mark the labels for all the decision regions. You need not use a perceptron algorithm to find the classification rule.

total/10

10 marks (**c**) Explain how to combine parts (a) and (b) to develop a classification rule that given any input $\mathbf{x} \in \mathbb{R}^2$ outputs a label $\hat{y} \in \{1, 2, 3\}$. Your classification rule must achieve perfect classification on the training set. Sketch your decision boundaries in $\mathbb{R}^2$ and show the labels associated with each decision region.

total/10

10 marks

(**d**) Suppose we wish to implement a multi-class logistic regression model for classifying the training set $\mathcal{D}$. Let $\Omega = \{\mathbf{w}(1), \mathbf{w}(2), \mathbf{w}(3)\}$ denote the model parameters of your logistic regression model where $\mathbf{w}(i) \in \mathbb{R}^3$ is the weight vector associated with class label $y = i$. Given an input data vector $\mathbf{x} = (x_1, x_2)^T$ the model outputs is a probability vector:

$$\hat{p}_\Omega(i|\mathbf{x}) = \frac{e^{[\mathbf{w}^T(i)\cdot\tilde{\mathbf{x}}]}}{\sum_{j=1}^{3} e^{[\mathbf{w}^T(j)\cdot\tilde{\mathbf{x}}]}}, \quad i = 1, 2, 3$$

where $\tilde{\mathbf{x}} = (x_0 = 1, x_1, x_2)^T \in \mathbb{R}^3$ is the augmented vector of $\mathbf{x}$ as discussed in class. We assume a standard log-loss function for the training error, i.e.,

$$E_{\text{in}}(\Omega) = \frac{1}{5}\sum_{n=1}^{5} e_n(\Omega), \qquad e_n(\Omega) = -\log \hat{p}_\Omega(y_n|\mathbf{x}_n)$$

Assuming that we select $\mathbf{w}(1) = (1,0,0)^T$, $\mathbf{w}(2) = (0,1,0)^T$ and $\mathbf{w}(3) = (0,0,1)^T$ numerically evaluate $\nabla_{\mathbf{w}(j)}\{e_1(\Omega)\}$ for $j = 1, 2, 3$.

total/-

[continue part (d) here]

total/10

5 marks

(e) For the problem in part (d), find the output of one-step update of the stochastic gradient descent (SGD) algorithm when the selected training example is $n = 1$, and ϵ is selected as the learning rate.

total/5

2. (40 MARKS) Consider a linear regression model where the training set is specified by

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\},$$

with $\mathbf{x}_i \in \mathbb{R}^{d+1}$ and $y_i \in \mathbb{R}$. We assume that each data vector is in the augmented dimension i.e., $\mathbf{x}_i = (x_{i,0} = 1, x_{i,1}, \ldots x_{i,d})$. We aim to find a weight vector $\mathbf{w} \in \mathbb{R}^{d+1}$ that aims to minimized a **weighted** squared error loss function

$$E_{\text{in}}^{\mathbf{\Lambda}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \lambda_i \cdot (\mathbf{w}^T\mathbf{x}_i - y_i)^2,$$

where $\mathbf{\Lambda} = (\lambda_1, \ldots, \lambda_N)^T$ is a pre-specified vector of **non-negative** constants that determine the importance of each sample.

Suppose that $\mathbf{w}^\star$ minimizes the weighted squared error loss i.e.,

$$\mathbf{w}^\star = \arg\min_{\mathbf{w} \in \mathbb{R}^{d+1}} E_{\text{in}}^{\mathbf{\Lambda}}(\mathbf{w}). \tag{1}$$

10 marks

(a) The optimal solution $\mathbf{w}^\star$ can be expressed as a solution to the following expression: $\mathbf{M} \cdot \mathbf{w}^\star = \mathbf{U} \cdot \mathbf{y}$ where $\mathbf{M}$ and $\mathbf{U}$ are matrices of dimension $(d+1) \times (d+1)$ and $(d+1) \times N$ respectively and $\mathbf{y} = [y_1, \ldots, y_N]^T$ is the observation vector. Provide an expression for $\mathbf{M}$ and $\mathbf{U}$ in terms of the following matrices:

$$\mathcal{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} \in \mathbb{R}^{N \times (d+1)} \qquad \mathcal{L} = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \ldots & 0 \\ 0 & \sqrt{\lambda_2} & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & \sqrt{\lambda_N} \end{bmatrix} \in \mathbb{R}^{N \times N}.$$

Please note that $\mathcal{L}$ is a diagonal matrix where the j−th diagonal entry is $\sqrt{\lambda_j}$.

total/-

[continue part (a) here]

total/10

10 marks (b) Provide an expression for $\nabla_{\mathbf{w}}\left(E_{\text{in}}^{\mathbf{\Lambda}}(\mathbf{w})\right)$ and use it to provide a (full) gradient descent algorithm for numerically computing the optimal solution $\mathbf{w}^{\star}$. Assume that a constant learning rate of ϵ is used in the algorithm.

total/10

5 marks (**c**) Using gradient analysis in part (b), provide a stochastic gradient descent (SGD) algorithm for numerically computing the optimal solution $\mathbf{w}^\star$. Assume that a constant learning rate of ϵ is used in the algorithm.

5 marks (**d**) List any two advantages of using the SGD algorithm over the solution in part (a)

total/10

10 marks

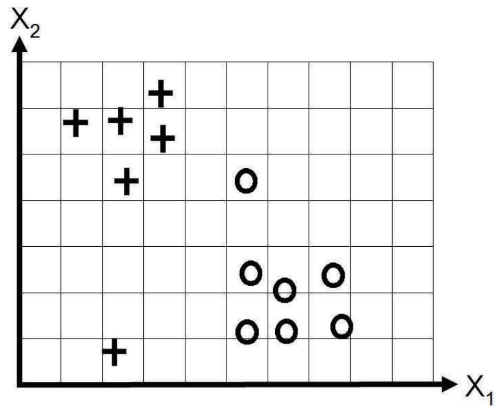
(e) Suppose we wish to find $\mathbf{w}_\beta^\star$ minimizes the following:

$$\mathbf{w}_\beta^\star = \arg\min_{\mathbf{w}\in\mathbb{R}^{d+1}} \left\{ E_{\text{in}}^{\mathbf{\Lambda}}(\mathbf{w}) + \beta \cdot \|\mathbf{w}\|^2 \right\}, \tag{2}$$

where $E_{\text{in}}^{\mathbf{\Lambda}}(\mathbf{w})$ is the weighted squared error loss as in part (a), $\|\mathbf{w}\|^2$ is the squared Eucledian norm of $\mathbf{w}$ and $\beta > 0$ is the regularization constant. Provide an analytical closed-form expression for $\mathbf{w}_\beta^\star$ in terms of $\mathcal{X}$, $\mathcal{L}$, $\mathbf{y}$, the identity matrix, and the constant β.

total/10

20 marks **3**. Consider a binary linear classification problem where $\mathbf{x} \in \mathbb{R}^2$ and $y \in \{-1, +1\}$. We illustrate the training dataset below. The '+' label refers to $y = +1$ and the 'o' label refers to $y = -1$. We would like to construct a classifier $h_{\mathbf{w}}(\mathbf{x}) = \text{sign}(w_0 + w_1 x_1 + w_2 x_2)$ where $\text{sign}(\cdot)$ is the *sign* function as discussed in class.



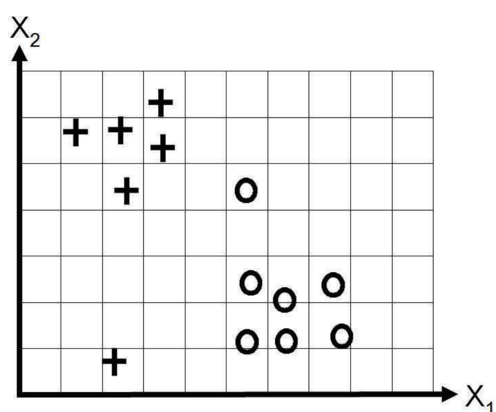
In the figure above, the adjacent vertical (and horizontal) lines are 1 unit apart from each other. Assume that the training points are above are $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ (with $N = 13$). We consider the classification loss

$$L(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(y_i \neq h_{\mathbf{w}}(\mathbf{x}_i))$$

where $\mathbb{I}$ denotes the indicator function.

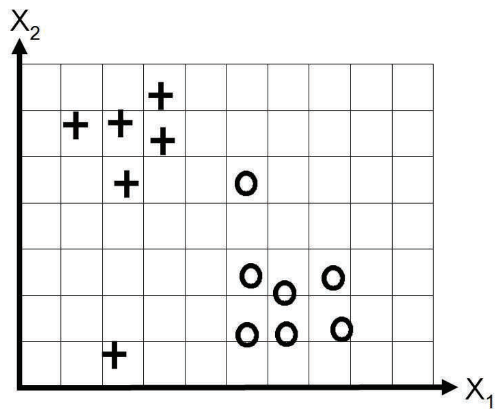
2 marks (**a**) Draw a decision boundary in the figure above that achieves zero training error.

6 marks (**b**) Suppose that we attempt to minimize the following loss function over $\mathbf{w}$: $J(\mathbf{w}) = L(\mathbf{w}) + \lambda w_0^2$, where $\lambda = 10^7$ is a huge constant. Sketch a possible decision boundary in the figure below. How many points are mis-classified.

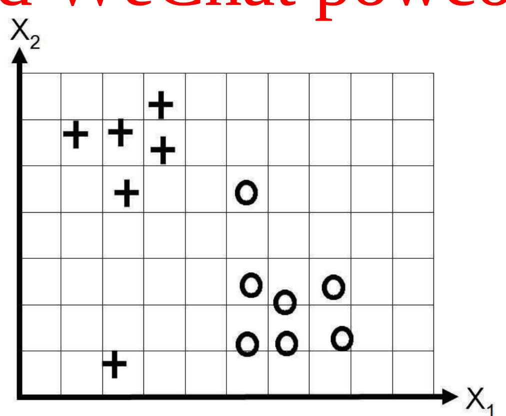


total/8

6 marks (c) Suppose that we attempt to minimize the following loss function over $\mathbf{w}$: $J(\mathbf{w}) = L(\mathbf{w}) + \lambda w_1^2$, where $\lambda = 10^7$ is a huge constant. Sketch a possible decision boundary in the figure below. How many points are mis-classified.



6 marks (d) Suppose that we attempt to minimize the following loss function over $\mathbf{w}$: $J(\mathbf{w}) = L(\mathbf{w}) + \lambda w_2^2$, where $\lambda = 10^7$ is a huge constant. Sketch a possible decision boundary in the figure below. How many points are mis-classified.



total/12