CS 725 Autumn 2024 | Midsem Exam

Instructor: Preethi Jyothi Date/Time: Sep 21, 2024, 6.30 pm to 8.30 pm

Total points: **25**

Name: _____

Roll number: _____

Instructions

- This is an open notes exam which should be completed individually.
- No form of collaboration or discussion is allowed.
- No laptops or cell phones are allowed.
- Write your name and roll number on the top of this page.
- This exam consists of 5 problems with sub-parts. The maximum possible score is 25.
- Write your answers legibly with a pen in the space provided on the exam sheet. (If necessary, use/ask for extra sheets for rough work. These extra sheets will not be graded.)
- Work efficiently. The questions are not sorted in any order of difficulty. Do a full pass, try and attempt the easier questions first.
- Good luck!

Question	Score
Mixed Bag	/8
Gradients of Linear Regression	/3
Perceptron Variants	/4
Logistic Regression	/4
Linear Predictors	/6
Total	/25

1 Mixed Bag (8 points)

(A) In a decision tree constructed using the information gain splitting criterion, the maximum information gain at every child node is guaranteed to be less than or equal to the maximum information gain at the parent node. True or False? Justify your answer. **[1 pts]**

Solution: False. Counterexample is a DT built on an XOR dataset.

> Give 0.25 points for correct answer (False) without justification.

(B) In an unregularized linear regression model, training error will not increase with the addition of a new feature. True or False? Justify your answer. **[1 pts]**

Solution: True. With a new feature, the model can at least fit the training data as well as before, if not better. So, training error can stay the same or decrease.

> Give 0.25 points for correct answer (True) without justification.

(C) Say you are given a set of m elements $\{u_1, u_2, \ldots, u_m\}$ and another set of n elements $\{v_1, v_2, \ldots, v_n\}$. You want to compute a matrix M of size $m \times n$ such that $M_{ij} = u_i + v_j$; here, M_{ij} refers to the element in the i^{th} row and j^{th} column of matrix M. How would you do this using NumPy broadcasting and no `for` loops? **[1 pts]**

Solution: Define a numpy array A of shape $m \times 1$, initialized to $u_1, \ldots, u_m$ and B of shape $1 \times n$ initialized to $v_1, \ldots, v_n$. Then, $M = A + B$.

(D) Let $x_1, \ldots, x_n$ be i.i.d. samples drawn from the following distribution:

$$P(x; \theta) = \theta x^{-\theta-1} \text{ where } \theta > 1, x \geq 1$$

What is the maximum likelihood estimator of θ? Show your work. **[1 pts]**

Solution:

$$LL(\theta) = n \log \theta + \sum_i (-\theta - 1) \log x_i$$

$$\frac{\partial LL}{\partial \theta} = 0 \Rightarrow \frac{n}{\theta} - \sum_i \log x_i = 0$$

$$\theta = \frac{n}{\sum_i \log x_i} \tag{1}$$

Since $\theta > 1$, $\theta_{\text{mle}} = \max(1, \frac{n}{\sum_i \log x_i})$.

> Give full points for the answer in eqn (1) as well.

(E) Which is more characteristic of decision tree classifiers – high bias or high variance? Briefly justify. **[1 pts]**

Solution: High variance. A small change in the data set can lead to changes in split points early during tree constructions, thus resulting in very different trees.

(F) For a given dataset $\mathcal{D}$, a project team considered all of $\mathcal{D}$ and reduced a large feature set down to a smaller set using feature selection. Next, they created train and test splits from $\mathcal{D}$. They estimated models on the train split with varying hyperparameters and reported the best test error that they obtained on the test split. Highlight two problems here with what the team did. **[1 pts]**

Solution: 1. Reporting the best test error is prone to overfitting. Should have used a dev/val set. 2. Feature selection on the full data will leak information from the test data into the model.

> $\frac{1}{2}$ point for stating each problem.

(G) Consider the feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and the label vector $\mathbf{y} \in \mathbb{R}^n$, $n > d$. Let the ridge regression estimator with regularization coefficient $\lambda > 0$ be $\mathbf{w}_{\text{ridge}}$ and the unregularized/ordinary least squares estimator be $\mathbf{w}_{\text{ols}}$. Define a modified $\mathbf{X}'$ and $\mathbf{y}'$ such that $\mathbf{w}_{\text{ridge}}$ using $\mathbf{X}$ and $\mathbf{y}$ is equivalent to $\mathbf{w}_{\text{ols}}$ using $\mathbf{X}'$ and $\mathbf{y}'$. As additional notation, you can use $\mathbf{I}_d$ to refer to the identity matrix of size $d \times d$; $\mathbf{0}_d$ and $\mathbf{1}_d$ can be used to refer to an all-zeros vector and all-ones vector of size d, respectively. Show your work. **[2 pts]**

Solution:

$$\mathbf{w}_{\text{ols}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$
$$\mathbf{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

We want to find an $\mathbf{X}'$ and $\mathbf{y}'$ s.t. $(\mathbf{X}'^T\mathbf{X}')^{-1}\mathbf{X}'^T\mathbf{y}' = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$. Answer is:

$$\mathbf{y}' = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_d \end{bmatrix}, \mathbf{X}' = \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I}_d \end{bmatrix}$$

1 point each for correct $\mathbf{y}'$ and correct $\mathbf{X}'$.

2 Gradients of Linear Regression (3 points)

Consider the ridge loss for linear regression, $L_{\text{r}}(\mathbf{w}, \mathcal{D}) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \lambda||\mathbf{w}||^2$, where $\lambda > 0$ is the regularization hyperparameter and $\mathbf{y}$, $\mathbf{w}$ and $\mathbf{X}$ are vectors/matrices as defined in class (using the training dataset $\mathcal{D}$). For brevity, we will use L below to refer to $L_{\text{r}}(\mathbf{w}, \mathcal{D})$.

(A) What is the gradient of L, i.e. $\nabla_{\mathbf{w}} L$, in terms of $\mathbf{X}$, $\mathbf{y}$ and $\mathbf{w}$? You will get full points if you write down the answer using matrix notation without any justifications. **[1 pts]**

Solution: $\nabla_{\mathbf{w}} L = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w} + 2\lambda\mathbf{w}$

(B) We know that the gradient $\nabla_{\mathbf{x}} f$ of a function $f : \mathbb{R}^n \to \mathbb{R}$ is a vector of its first partial derivatives: $\nabla_{\mathbf{x}} f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$. The **Hessian matrix** $\nabla^2_{\mathbf{x}} f$ of f is a matrix of its second partial derivatives: $\nabla^2_{\mathbf{x}} f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$. What is $\nabla^2_{\mathbf{w}} L$, in terms of $\mathbf{X}$, $\mathbf{y}$ and $\mathbf{w}$? Is the Hessian matrix $\nabla^2_{\mathbf{w}} L$ invertible? Show your work, and write your answer using matrix notation. **[2 pts]**

Solution: $\nabla^2_{\mathbf{w}} L = 2\mathbf{X}^T\mathbf{X} + 2\lambda\mathbf{I}$.

$\nabla^2_{\mathbf{w}} L$ is invertible. $\nabla^2_{\mathbf{w}} L$ is positive definite $\Rightarrow$ for any non-zero $\mathbf{p}$, $\mathbf{p}^T\nabla^2_{\mathbf{w}} L\mathbf{p} = 2\mathbf{p}^T\mathbf{X}^T\mathbf{X}\mathbf{p} + 2\mathbf{p}^T\mathbf{p} = 2||\mathbf{X}\mathbf{p}||^2 + 2||\mathbf{p}||^2 > 0$. As we learned in class, any positive definite matrix is invertible.

1 point for correct answer for $\nabla^2_{\mathbf{w}} L$; give 0.75 points if they write 2λ instead of $2\lambda\mathbf{I}$.
1 point for showing that the Hessian is invertible.

3 Perceptron Variants (4 points)

Recall the original perceptron algorithm for binary classification. Given a training instance $(\mathbf{x}, y)$, $\mathbf{x} \in \mathbb{R}^d$, $y \in \{-1, 1\}$, the weight vector $\mathbf{w}$ is updated as:

$$\text{if } (y\mathbf{w}^T\mathbf{x} \leq 0) \text{ then } \mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$$

Below, we list weight update rules for two new variants of the original perceptron (with minor modifications). Assume that the initial weight vector is a zero vector.

(A) if $(y\mathbf{w}^T\mathbf{x} \leq 0)$ then $\mathbf{w} \leftarrow \mathbf{w} + ky\mathbf{x}$ where $k \in \mathbb{R}, k > 0$.

Let the weight vector after t iterations for the original perceptron and the variant above be $\mathbf{w}_t^{\text{orig}}$ and $\mathbf{w}_t^{\text{new}}$, respectively. Assume that after t iterations, the original perceptron algorithm and the variant above have made the same mistakes. Then, describe the relationship between $\mathbf{w}_t^{\text{orig}}$ and $\mathbf{w}_t^{\text{new}}$. Are they the same? If not, how do they differ? **[2 pts]**

Solution: They only differ in magnitude. $\mathbf{w}_t^{\text{new}} = k\mathbf{w}_t^{\text{orig}}$.

(B) if $(y\mathbf{w}^T\mathbf{x} \leq 0)$ then $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{x}$.

Is the modified perceptron with weight update rule (B) guaranteed to converge on linearly separable data? Justify your answer; a formal proof is not needed. **[2 pts]**

Solution: No, it is not guaranteed to converge. Consider a single-instance dataset with label -1. $\mathbf{w}$ will start from 0 and will take values $\mathbf{x}$, $2\mathbf{x}$, $3\mathbf{x}$ and so on. It will never be correct on this training example.

4 Logistic Regression (4 points)

Consider a softmax logistic regression classifier for multi-class classification with weight vectors $\mathbf{w} = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ for K classes. For an instance $\mathbf{x} \in \mathbb{R}^d$ and a label $y \in \{1, \ldots, K\}$, the probability is $P(y = k|\mathbf{x}; \mathbf{w}) = f(k, \mathbf{x}, \mathbf{w})$ where:

$$f(k, \mathbf{x}, \mathbf{w}) = \frac{\exp(\mathbf{w}_k^T \mathbf{x})}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{x})}$$

(A) Consider a fixed weight vector $\theta \in \mathbb{R}^d$. Say we subtract θ from each of the weight vectors $\mathbf{w}_k, k = 1 \ldots K$. Show that this operation does not change $P(y = k|\mathbf{x}; \mathbf{w})$.

Use the above result to define a new function $f'(k, \mathbf{x}, \mathbf{w}')$, where $\mathbf{w}' = \{\mathbf{w}'_1, \ldots, \mathbf{w}'_{K-1}\}, \mathbf{w}'_k \in \mathbb{R}^d$, such that $f'(k, \mathbf{x}, \mathbf{w}') = f(k, \mathbf{x}, \mathbf{w})$. You should define $\mathbf{w}'$ as a function of $\mathbf{w}$ (note that $\mathbf{w}'$ has one vector less than $\mathbf{w}$), and also define $f'(k, \mathbf{x}, \mathbf{w}')$ explicitly for each $k \in \{1, \ldots, K\}$. **[3 pts]**

Solution:

$$f(k, \mathbf{x}, \mathbf{w}) = \frac{\exp((\mathbf{w}_k - \theta)^T \mathbf{x})}{\sum_{j=1}^{K} \exp((\mathbf{w}_j - \theta)^T \mathbf{x})} = \frac{\exp(\mathbf{w}_k^T \mathbf{x}). \exp(-\theta^T \mathbf{x})}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{x}). \exp(-\theta^T \mathbf{x})} = \frac{\exp(\mathbf{w}_k^T \mathbf{x})}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{x})}$$

This suggests we can set θ to any $\mathbf{w}_k, k \in \{1, \ldots, K\}$. Let $\theta = \mathbf{w}_K$. Then,

$$\mathbf{w}' = \{\mathbf{w}_1 - \mathbf{w}_K, \mathbf{w}_2 - \mathbf{w}_K, \ldots, \mathbf{w}_{K-1} - \mathbf{w}_K\}$$

. And,

$$f'(k, \mathbf{x}, \mathbf{w}) = \begin{cases} \frac{\exp((\mathbf{w}_k - \mathbf{w}_K)^T \mathbf{x})}{\sum_{j=1}^{K} \exp((\mathbf{w}_j - \mathbf{w}_K)^T \mathbf{x})} & \text{if } k = 1, \ldots, K-1 \\ 1 - \sum_{j=1}^{K-1} f(j, \mathbf{x}, \mathbf{w}) & \text{if } k = K \end{cases}$$

1 point for showing that subtracting θ from each $\mathbf{w}_k$ does not change $P(y = k|\mathbf{x}; \mathbf{w})$.
1 point for defining $\mathbf{w}'$ and 1 point for defining $f'(k, \mathbf{x}, \mathbf{w})$.

(B) The softmax logistic regression classifier minimizes the cross-entropy loss function, $\mathcal{L}_{\text{ce}}$. We learned in class that the cross-entropy loss function is convex. Is $\mathbf{w} = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ that minimizes $\mathcal{L}_{\text{ce}}$ unique or not? Briefly justify your answer. **[1 pts]**

Solution: From part (A), we know that if $\mathcal{L}_{\text{ce}}(\mathbf{w})$ is minimized by some $\mathbf{w}_1, \ldots, \mathbf{w}_K$, then it is also minimized by $\mathbf{w}_1 - \theta, \ldots, \mathbf{w}_K - \theta$. Hence, the minimizer of $\mathcal{L}_{\text{ce}}(\mathbf{w})$ is not unique.

5 Linear Predictors (6 points)

In the following problems, $X = \mathbb{R}$ and $Y = \{+1, -1\}$ and we are given data points $(\mathbf{x}, \mathbf{y}) \in X \times Y$, where $\mathbf{y} = f(\mathbf{x})$ for some f. Show that this dataset is linearly separable w.r.t. a transformed feature space $\Phi(\mathbf{x})$, when f and Φ are as specified below. That is, show that $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{d} \alpha_i \phi_i(\mathbf{x}))$, where $\Phi(x) = (\phi_1(x), \ldots, \phi_d(x)) \in \mathbb{R}^d$ (assume $\text{sign}(0) = +1$).

(A) Suppose $X = \mathbb{R}$, f is given by

$$f(x) = \begin{cases} -1 & \text{if } x \in (a, b) \\ +1 & \text{otherwise.} \end{cases}$$

and $\Phi(x) = (1, x, x^2)$. Show that f is linearly separable with respect to Φ. **[2 pts]**

Solution:

$$f(\mathbf{x}) = \text{sign}((x - a)(x - b)) = \text{sign}(ab - (a + b)x + x^2)$$

(B) Suppose $X = \mathbb{R}$, f is given by

$$f(x) = \begin{cases} -1 & \text{if } x < a \\ (-1)^{\lfloor x \rfloor} & \text{if } x \in [a, b) \\ +1 & \text{if } x \geq b \end{cases}$$

and $\Phi(x) = (1, x, (-1)^{\lfloor x \rfloor})$. ($\lfloor x \rfloor$ denotes the floor function that produces as output the greatest integer less than or equal to x.) Show that f is linearly separable with respect to Φ. **[4 pts]**

Solution: Let α, β be such that the line $\alpha x + \beta$ equals -1 at $x = a$ and equals 1 at $x = b$. Solving for α, β, we get:

$$\alpha = \frac{2}{(b-a)} \quad , \quad \beta = \frac{(a+b)}{(a-b)}$$

Then, $f(x) = \text{sign}(\beta + \alpha x + (-1)^{\lfloor x \rfloor})$.

Any partial/reasonable path towards the solution can be given up to 2 points.