COMP5328/COMP4328/COMP8328 Sample Final Exam 2025 Semester 2

Question 1 [10 pts]
1). Taylor's theorem is important for optimization methods like gradient descent and Newton's method. How does Taylor's theorem help approximate future loss based on current loss?
2). What are the key differences between gradient descent and Newton's methods for update loss?

Taylor's Theorem

Let $k \geq 1$ be an integer and let the function $f : \mathbb{R} \to \mathbb{R}$ be k times differentiable at the point $a \in \mathbb{R}$. Then there exists a function $h_k : \mathbb{R} \to \mathbb{R}$ such that

$$f(x) = f(a) + f'(a)(x-a) + \ldots + \frac{f^{(k)}(a)}{k!}(x-a)^k + h_k(x)(x-a)^k$$

and $\lim_{x \to a} h_k(x) = 0.$

[WRITE YOUR ANSWER HERE]
1). Taylor's theorem facilitates the prediction of future loss values based on current loss and gradients. Specifically, Taylor's theorem expresses the loss function L around a current point $\mathbf{x}$ as:

$$L(\mathbf{x} + \Delta\mathbf{x}) \approx L(\mathbf{x}) + \nabla L(\mathbf{x})^\top \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top H(\mathbf{x})\Delta\mathbf{x}.$$

Here, $\nabla L(\mathbf{x})$ is the gradient (first derivative), and $H(\mathbf{x})$ is the Hessian matrix (second derivative) of L at $\mathbf{x}$.
2). The key differences between gradient descent and Newton's method in updating the loss are based on their consideration of the Taylor series expansion:

Gradient Descent uses only the first derivative (gradient) of the loss function to guide the search for minima. The update rule is:

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} - \alpha \nabla L(\mathbf{x}_{\text{old}}),$$

where α is the learning rate.

Newton's Method: Uses both the first and second derivatives (gradient and Hessian). The update rule involves the inverse of the Hessian:

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} - [H(\mathbf{x}_{\text{old}})]^{-1} \nabla L(\mathbf{x}_{\text{old}}).$$

Question 2 [10 pts]

In the context of machine learning and signal processing, different norms are used to measure the sparsity of vectors. Sparsity in a vector implies that many of its elements are zero or near zero. Given the definitions of the L1, L2, and L3 norms for a vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ as follows:

- L1 norm ($\|\mathbf{x}\|_1$): Sum of the absolute values of the elements. $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$
- L2 norm ($\|\mathbf{x}\|_2$): Square root of the sum of the squared elements, also known as the Euclidean norm. $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$
- L3 norm ($\|\mathbf{x}\|_3$): Cube root of the sum of the cubed absolute values of the elements. $\|\mathbf{x}\|_3 = \left(\sum_{i=1}^{n} |x_i|^3\right)^{1/3}$

1). Which of these norms (L1, L2, L3) is best suited for use as a surrogate for measuring the sparsity of a vector, and which is the least suitable? Provide a detailed explanation for your answer, considering using figure to explain.

[WRITE YOUR ANSWER HERE]

1). The L0 norm is a theoretical measure of sparsity, as it directly counts the number of non-zero elements in a vector. This norm is ideal for sparsity measurement.

The provided figure illustrates the behavior of various norms as the exponent p approaches 0. As p decreases towards zero, the function representing $|a_j|^p$ approaches the behavior of the L0 norm, effectively approximating the count of non-zero components in the vector. This trend suggests that norms with smaller p values are better suited for sparsity measurement, as they more closely approximate the L0 norm.

Therefore, among the norms considered, the L1 norm (sum of absolute values) is superior for promoting and measuring sparsity than L2 norm, and L2 is superior than L3 norm.

Question 3 [10 pts]
Consider a binary classification task where a model predicts the probability of two classes, $C = \{0, 1\}$, based on input features. You are given two data points with both clean (true) class posterior probabilities and noisy (observed) class posterior probabilities. It is assumed that the transition matrix, which represents the noise in class labels, is instance-independent (i.e., the same across all data points).

- x_1: Clean class posterior probabilities: $P(Y = 0 \mid x_1) = 0.7$. $P(Y = 1 \mid x_1) = 0.3$ Noisy class posterior probabilities: $P(\tilde{Y} = 0 \mid x_1) = 0.5$, $P(\tilde{Y} = 1 \mid x_1) = 0.5$.
- Clean class posterior probabilities: $P(Y = 0 \mid x_2) = 0.4$, $P(Y = 1 \mid x_2) = 0.6$. Noisy class posterior probabilities: $P(\tilde{Y} = 0 \mid x_2) = 0.6$, $P(\tilde{Y} = 1 \mid x_2) = 0.4$.

Calculate the transition matrix with the provided information. Show your calucation step in detals.

[WRITE YOUR ANSWER HERE]

Given the transition matrix T has the form:

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

where each column sums to one, we derive:

$$a + c = 1,$$

$$b + d = 1.$$

Using Data from x_1 and x_2:

$$\begin{cases} 0.7a + 0.3b = 0.5 & (1) \\ 0.4a + 0.6b = 0.6 & (2) \\ a + c = 1 & (3) \\ b + d = 1 & (4) \end{cases}.$$

Multiply the first equation by 2 and subtract the second from the first to eliminate a:

$$a = 0.4.$$

Plug $a = 0.4$ into $7a + 3b = 5$:

$$2.8 + 3b = 5 \implies 3b = 2.2 \implies b = 0.7333.$$

Find c and d:

$$c = 1 - a = 1 - 0.4 = 0.6 d = 1 - b = 1 - 0.7333 = 0.2667.$$

The Transition Matrix T is then:

$$T = \begin{bmatrix} 0.4 & 0.7333 \\ 0.6 & 0.2667 \end{bmatrix}.$$

Question 4 [12 pts]

In the realm of domain adaptation, understanding and addressing covariate shift is crucial. Covariate shift occurs when the probability distributions of input data in the training (source) and testing (target) domains differ, specifically $p_s(X) \neq p_t(X)$, where $p_s(X)$ and $p_t(X)$ denote the probability densities of the source and target domains, respectively. Addressing this shift is essential for adapting models trained on the source domain to perform effectively in the target domain.

1). Explain why covariate shift can be problematic when training models.

2). Describe how importance reweighting can be used to address covariate shift, including a detailed step-by-step calculation of the weights.

3). What assumptions underlie the use of importance reweighting for covariate shift?

[WRITE YOUR ANSWER HERE]

1). The primary issue arises because the distribution gap. Some feature X sampled in target domain can be s model's assumptions based on the training data do not hold for the test data, which affects the reliability and accuracy of the model's outcomes.

2). For covariate shift, we assume that the conditional distribution of Y given X is the same in both the target and source distributions ($p_t(Y \mid X) = p_s(Y \mid X)$). Therefore,

$$\beta(X, Y) = \frac{p_t(X, Y)}{p_s(X, Y)} = \frac{p_t(Y \mid X)p_t(X)}{p_s(Y \mid X)p_s(X)} = \frac{p_t(X)}{p_s(X)} = \beta(X).$$

By using importance reweighting, the expected risk under the target distribution using source samples is calculated as follows:

$$R^T(h) = E_{(X,Y)\sim p_s(X,Y)}[\beta(X)\ell(X, Y, h)] = E_{(X,Y)\sim p_s(X,Y)}\left[\frac{p_t(X)}{p_s(X)}\ell(X, Y, h)\right].$$

Its empirical approximation is:

$$R^T(h) \approx \frac{1}{n}\sum_{i=1}^{n}\frac{p_t(x_i^s)}{p_s(x_i^s)}\ell(x_i^s, y_i^s, h).$$

Now, we derive $\beta(X)$ using Bayes' rule:

$$\beta(X) = \frac{p_t(X)}{p_s(X)} = \frac{\frac{p(\text{target}|X)p(X)}{p(\text{target})}}{\frac{p(\text{source}|X)p(X)}{p(\text{source})}} = \frac{p(\text{target} \mid X)}{p(\text{source} \mid X)} \cdot \frac{p(\text{source})}{p(\text{target})}.$$

Note that $p_s(X)$ and $p_t(X)$ are shorthand for $p(X \mid \text{source})$ and $p(X \mid \text{target})$, respectively.

We can train a classifier to estimate the probability $p(\text{source} \mid X)$, and $p(\text{target} \mid X)$ is simply $1 - p(\text{source} \mid X)$.

After having estimated $p(\text{source} \mid X)$, we can make $\frac{p(\text{target}|X)}{p(\text{source}|X)}$ the estimated $\beta(X)$ and integrate these weights into the training process, typically by modifying the loss function to incorporate the weights directly. Note that as $\frac{p(\text{source})}{p(\text{target})}$ is a constant and the same for every instance, we can just ignore it.

3). It is crucial that any value of x that has a positive probability of occurring in the target distribution $p_t(x)$ also has a positive probability in the source distribution $p_s(x)$. This condition ensures that when we calculate weights using the formula $\beta(X) = \frac{p_t(X)}{p_s(X)}$, we do not end up with a situation where we divide by zero. If $p_s(X) = 0$ (meaning X never appears in the source data) for any X where $p_t(X) > 0$ (meaning X appears in the target data), the formula would result in an undefined division.

Question 5 [10 pts] Consider a loss function $\phi(z) = \log(1 + \exp(-z))$.
1). What is the key property of a classification-calibrated loss function? How to check if a loss function is classification-calibrated?
3). Is $\phi(z) = \log(1 + \exp(-z))$ a classification-calibrated loss function? Provide a detailed explanation and show the calculation steps in detail. Note that the derivative of $\exp(x)$ with respect to a variable x is $\exp(x)$ itself, and $\exp(x) > 0, \forall x$.
[WRITE YOUR ANSWER HERE]
1). The key property: minimizing a classification-calibrated surrogate loss function should lead to an optimal classifier as if we had directly minimized the 0-1 loss, which directly measures classification errors.
To determine if a loss function ϕ is classification-calibrated, ensure the following:

- Confirm Convexity: Verify that ϕ is convex over its domain.
- Evaluate Derivative at 0: Calculate the first derivative of $\phi(z)$ at zero. For ϕ to be classification-calibrated, it should satisfy $\phi'(0) < 0$.

2). Analysis of $\phi(z) = \log(1 + \exp(-z))$: Let's examine whether the logistic loss function $\phi(z)$ is classification-calibrated:
Confirm Convexity: The logistic loss function $\phi(z)$ is convex.
The first derivative of $\phi(z)$ is:

$$\phi'(z) = \frac{d}{dz}\left[\log(1 + \exp(-z))\right] = -\frac{\exp(-z)}{1 + \exp(-z)}.$$

Its second derivative:

$$\phi''(z) = \frac{d}{dz}\left[-\frac{\exp(-z)}{1 + \exp(-z)}\right] = \frac{\exp(-z)}{(1 + \exp(-z))^2} > 0 \quad \text{for all } z.$$

Since $\phi''(z) > 0$, $\phi(z)$ is convex.
Evaluating the first derivative of $\phi(z)$ at $z = 0$:

$$\phi'(0) = -\frac{\exp(0)}{1 + \exp(0)} < 0.$$

Since $\phi(z)$ is convex and $\phi'(0)$ is less than 0, this condition satisfies the requirement for classification calibration.