CSE Ph.D. Qualifying Exam, Fall 2024

This is a **CLOSED BOOK** exam. No books or notes are allowed.

Data Analysis

Please answer three of the following four questions. All questions are graded on a scale of 10. If you answer all four, all answers will be graded and the three lowest scores will be used in computing your total. Show all your work and write in a readable way.

1. **Q1: Locally Weighted Linear Regression**

   Given $\{(x^{(i)}, y^{(i)}), \ldots, (x^{(m)}, y^{(m)})\}$ and corresponding positive weights $w^{(1)}, \ldots, w^{(m)}$, we consider the *Locally Weighted Linear Regression* problem where we minimize the following loss function:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} w^{(i)} \left(y^{(i)} - \theta^T x^{(i)}\right)^2. \tag{1}$$

   (1) [2 pts] When $w^{(i)} = 1, \forall i$, Eq. 1 to the regular linear regression with the loss function: $J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2$, which can be expressed in vector form as $J(\theta) = \|X\theta - \mathbf{y}\|_2^2$ where the transpose of each $x^{(i)}$ is a row in matrix X. Now generalize this vector form to Eq. 1 and show that $J(\theta)$ can also be written as

$$J(\theta) = (X\theta - \mathbf{y})^T W (X\theta - \mathbf{y}),$$

   for an appropriate diagonal matrix W, and clearly state what W is.

   (2) [2 pts] A common choice for $w^{(i)}$ is

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right),$$

   where τ is a bandwidth hyperparameter. Briefly describe how this loss function differs from regular linear regression and how the loss of individual data point is influenced by the weights.

   (3) [3 pts] Based on your understanding of overfitting and underfitting, explain how you would expect the training error and the validation error to vary when τ is very large and very small, respectively.

   (4) [3 pts] Suppose we have a training set of m independent examples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, but with differing variances in the $y^{(i)}$ observations, characterized by the probability distribution:

$$p\left(y^{(i)} \mid x^{(i)}; \theta\right) = \frac{1}{\sqrt{2\pi}\sigma^{(i)}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2(\sigma^{(i)})^2}\right).$$

   Prove that finding the maximum likelihood estimate of θ of this problem reduces to solving the locally weighted linear regression in Eq. 1. Also clearly state what the $w^{(i)}$ are in terms of the $\sigma^{(i)}$'s.

2. **Q2: Neural Networks**

(1) [2 pts] Given a weight matrix $\boldsymbol{W}$, an input vector $\boldsymbol{x}$, the hyperbolic tangent activation function $\tanh(z) = \left(e^z - e^{-z}\right) / \left(e^z + e^{-z}\right)$, we can construct a simple neural network $f(\boldsymbol{x}; \boldsymbol{W}) = \tanh\left(\boldsymbol{W}^\intercal \boldsymbol{x}\right)$. Let the model output be $\hat{y}$. Write down an expression for $\nabla_{\boldsymbol{W}} \hat{y}$.

(2) [3 pts] Given the following values of $\boldsymbol{W}$ and $\boldsymbol{x}$, calculate the network estimate $\hat{y}$ by doing the forward computation once. Let the ground-truth label y be 0 and the loss function be $\mathcal{L}(\hat{y}, y) = |\hat{y} - y|$, update the weights matrix once using the gradient descent rule: $\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \eta \nabla_{\boldsymbol{W}} \mathcal{L}$, where $\eta = 4$ is the learning rate.

$$\boldsymbol{W} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \qquad \boldsymbol{x} = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$$

Calculate $\hat{y}$ with the updated $\boldsymbol{W}$ and compare it to the previous estimate.

(3) [1 pts] L1 regularization on a neural network model's parameters $\boldsymbol{w}$ is defined as $\|\boldsymbol{w}\|_1 = \sum_i |w_i|$. Suppose the objective function is $\mathcal{L}\left(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}\right)$. First, write down the L1 regularized objective function $\hat{\mathcal{L}}$ with a coefficient parameter α. Then, using an element-wise sign function $sign(\cdot)$, write down the corresponding gradient of the regularized objective function $\hat{\mathcal{L}}$.

(4) [3 pts] To observe the effect of L1 regularization on the weights of the model, we can approximate the unregularized objective function by Taylor expansion and discarding high-order terms:

$$\tilde{\mathcal{L}} := \mathcal{L}(\boldsymbol{w}^*) + \frac{1}{2} \left(\boldsymbol{w} - \boldsymbol{w}^*\right)^\intercal \boldsymbol{H} \left(\boldsymbol{w} - \boldsymbol{w}^*\right),$$

where $\boldsymbol{w}^*$ is the optimal set of weights without the regularization term and $\boldsymbol{H}$ is the Hessian matrix. We assume that $\boldsymbol{H}$ is diagonal and its diagonal terms $H_{i,i}$ are positive.

Write down an analytical form for each weight term w_i in $\boldsymbol{w}$ when the gradient is zero. Note that the gradient of $|x|$ at $x = 0$ can take any value between $[-1, 1]$. Hint: there are three different conditions based on the values of w_i^*.

(5) [1 pts] Given your answer to the previous question, explain how L1 regularization will affect large and small weights.

3. **Q3: Reinforcement Learning**

Alice is asked to conduct reinforcement learning from human feedback (RLHF) on large language models(LLMs), which is denoted as $\pi_{ref}(y|x)$ with x as the prompt and y as the responded text. Please help Alice to complete the task.

(1) [4 points] The first step of RLHF is to learn a reward model. However, there is no directly reward labeled, but only pairwise comparison provided by human, *i.e.*,

given two answers y_1 and y_2 to the same prompt x, a human labeler will distinguish which one is better and which is worse, denoted as y_+ and y_-, respectively. Alice exploits the Bradley-Terry (BT) model to capture the pairwise comparison with an implicit reward $r_\theta(x, y)$, where θ denote the parameters of the model, *i.e.*,

$$\mathbb{P}(y_+ \succ y_- | x, y_+, y_-) = \frac{\exp(r_\theta(x, y_+))}{\exp(r_\theta(x, y_+)) + \exp(r_\theta(x, y_-))} \tag{2}$$

With such data collected $\mathcal{D} = \{(y_+, y_-, x)_{i=1}^n\}$, please help Alice to design the MLE of the BT model to learn θ.

(2) [4 points] With the learned reward model, Alice would like to update the LLM $\pi_\omega(y|x)$ through policy optimization, where ω is the parameter, *i.e.*,

$$\max_{\pi_\omega} \ \ell(\omega) = \mathbb{E}_{x\sim\mathcal{D}} \left[\mathbb{E}_{y\sim\pi_\omega(y|x)} \left[r_\theta(x, y) \right] - KL(\pi_\omega || \pi_{ref}) \right]. \tag{3}$$

Please help Alice to derive the policy gradient w.r.t. ω, *i.e.*, $\nabla_\omega \ell(\omega)$.

(3) [2 points] Alice was told she needs to repeat the step 1 and 2 iteratively. Could you please explain the reason to Alice?

4. **Q4: Linear Regression with Laplacian Noise**

You must have seen how least-squares regression is motivated by maximum likelihood estimation if we think our data obeys a linear relationship but has added noise that is normally distributed. However what if the noise is better modeled by the Laplace distribution?

Let $\epsilon \sim \text{Laplace}(\mu, \beta)$ indicate a random variable ϵ drawn from a univariate Laplace distribution with mean μ and scale parameter β. The PDF of this distribution is

$$f(\epsilon; \mu, \beta) = \frac{1}{2\beta} \exp\left(-\frac{|\epsilon - \mu|}{\beta} \right).$$

Following the usual notation, the input is an $n \times d$ data matrix X and a vector y such that y_i is the label for sample point X_i, where $X_i^\top$ is the ith row of X. To keep things simple, we will do linear regression through the origin (no bias term α), so the regression function is $h(x) = w \cdot x$. Our model is that each label y_i comes from a linear relationship perturbed by Laplacian noise,

$$y_i \sim \text{Laplace}(w \cdot X_i, \beta),$$

where $w \in \mathbb{R}^d$ is the true linear relationship. We will use maximum likelihood estimation to try to estimate w.

(a) (2 pts) Write the likelihood function $L(w; X, y)$ for the parameter w, given the fixed data X and y.

(b) (1 pt) Write the log likelihood function $\ell(w; X, y)$ for the parameter w, given the fixed data X and y, in as simple a form as you can. (Make sure your logarithms have the correct base.)

(c) (2 pts) What is the simplest cost function we can minimize that gives us the same value of w as maximizing the likelihood?

(d) (2 pts) How is the cost function you just derived different from standard least-squares regression? Is it more or less sensitive to outliers? Why?

(e) (3 pts) Write the batch gradient descent rule for minimizing your cost function, using η for the step size (aka learning rate). You may omit training points whose losses have undefined gradients. Hint: Recall that $\frac{d}{d\alpha}|\alpha|$ is 1 for $\alpha > 0$, -1 for $\alpha < 0$, and undefined for $\alpha = 0$.