STA 250 – Homework 0 (Self-Assessment), due: Never

Instructor: Dogyoon Song

Instructions: This assignment is solely for self-assessment and practice. It will *not* be collected or graded, nor will solutions be provided. It reviews selected topics from linear algebra, vector calculus/optimization, probability, basic learning theory, and Python coding. If any part feels especially challenging, please review relevant resources or consult the instructor.

Problem 1. Linear Algebra

(a) A set of matrices is said to be simultaneously diagonalizable if there exists a single invertible matrix P such that $P^{-1}MP$ is a diagonal matrix for every M in the set. Suppose A and B are $n \times n$ matrices. Under what conditions can they be simultaneously diagonalized? Give a brief justification.

(b) Let A be a symmetric $n \times n$ matrix.

(i) Show that A can be diagonalized by an orthonormal basis and that all its eigenvalues are real.

(ii) If A is also positive semidefinite (PSD), prove that its eigenvalues are nonnegative.

(c) Let A and B be symmetric positive semidefinite (PSD) matrices. Is AB necessarily PSD? If yes, provide a proof. If not, construct a counterexample and state conditions under which AB *is* PSD.

(d) Briefly state the definition of the SVD of an $m \times n$ matrix. How is it related to the eigen-decomposition of a symmetric matrix? Write down the Penrose-Moore pseudoinverse of a matrix M using its SVD.

Problem 2. Vector Calculus and Optimization

(a) Let $f(x) = \frac{1}{2}\|Ax - y\|_2^2$, where $A \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^m$. Compute $\nabla_x f(x)$ and $\nabla_x^2 f(x)$ explicitly in terms of A, x, and y.

(b) Consider the function $g : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$g(x_1, x_2) \;=\; x_1^4 + x_2^4 \;-\; 2\big(x_1^2 + x_2^2\big).$$

(i) Find $\nabla g(x_1, x_2)$ and determine *all* stationary points (where $\nabla g = 0$).

(ii) Classify each stationary point as a local minimum, local maximum, or saddle point. (Hint: consider the Hessian or another suitable test.)

(c) Let X be an $n \times n$ positive definite (PD) matrix. Define $h(X) = \log\det(X)$. Show that

$$\frac{\partial}{\partial X}\big[\log\det(X)\big] \;=\; \big(X^{-1}\big)^\top.$$

(You may assume X is symmetric PD, so X^{-1} is also symmetric.)

(d) Consider the function $u : \mathbb{R}^2 \to \mathbb{R}$ given by

$$u(x_1, x_2) \;=\; 4\,x_1^2 \;+\; x_2^2 \;-\; 6\,x_1 x_2 \;+\; 3\,x_2.$$

(i) Use the first-order necessary condition (i.e., set $\nabla u = 0$) to find all critical points.

(ii) For each critical point, classify it as a local minimum, local maximum, or saddle point. Justify your answer (e.g., by examining the Hessian).

(e) Use Lagrange multipliers to solve the following constrained optimization problem:

$$\min_{(x,y)\,\in\,\mathbb{R}^2} \quad x^2 + y^2 \quad \text{subject to} \quad x + y = 2.$$

(i) Set up the Lagrangian and write down the stationarity conditions.

(ii) Solve for (x, y) and the corresponding Lagrange multiplier.

(iii) Interpret the result geometrically (i.e., where on the line $x + y = 2$ do we minimize $x^2 + y^2$?).

Problem 3. Probability

(a) Recall Markov's inequality: if X is a nonnegative random variable, then for all $t > 0$,

$$\Pr\big(X \geq t\big) \leq \frac{\mathbb{E}[X]}{t}$$

(i) Prove Markov's inequality.

(ii) Recall Chebyshev's inequality: if X is a random variable that has mean μ and finite variance σ^2, then for any $t > 0$,

$$\Pr\big(\,|X - \mu| \geq t\big) \leq \frac{\sigma^2}{t^2}\,.$$

Give a short derivation of Chebyshev's inequality (for example, from Markov's inequality).

(b) A random variable X with mean μ is σ-sub-Gaussian if for all $t \in \mathbb{R}$,

$$\mathbb{E}\big[e^{\lambda(X-\mu)}\big] \leq \exp\Big(\frac{\sigma^2\,\lambda^2}{2}\Big).$$

We refer to σ^2 as the *variance proxy* of X.

(i) Show that for all $t > 0$,

$$\Pr\big(X - \mu > t\big) \leq \exp\Big(-\tfrac{t^2}{2\sigma^2}\Big) \quad \text{and} \quad \Pr\big(X - \mu < -t\big) \leq \exp\Big(-\tfrac{t^2}{2\sigma^2}\Big).$$

(ii) Let X be supported on a bounded interval $[a, b] \subset \mathbb{R}$. Is X necessarily sub-Gaussian? If so, provide an upper bound on its variance proxy in terms of a and b. If not, justify briefly.

(c) Let $X_1, \ldots, X_n$ be random variables where each X_i is σ_i-sub-Gaussian with mean μ_i. Define $Z = \sum_{i=1}^n X_i$.

(i) Show that Z is sub-Gaussian with mean $\sum_{i=1}^n \mu_i$. Give an upper bound on its variance proxy in terms of $\sigma_1, \ldots, \sigma_n$.

(ii) Given $\delta \in (0, 1)$, provide an interval I (as tight as possible) such that

$$\Pr\Big(Z \in I\Big) \geq 1 - \delta.$$

Briefly explain how you derived this bound. How would the interval differ if you did *not* assume sub-Gaussianity (e.g., using only Chebyshev's inequality)?

(iii) If the X_i are independent, can you obtain a tighter (smaller) upper bound on the variance proxy of Z? Does this lead to a narrower interval? Explain briefly.

Problem 4. Learning Theory Basics

Consider a regression problem with data points $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Define the L_2 empirical risk

$$R_2(w) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - \langle w,\, x_i\rangle\big)^2.$$

(a) Show that if $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n\times d}$ has full column rank ($d \le n$ and rank d), the minimizer $w_2^* \in \arg\min_w(R_2(w))$ satisfies

$$w_2^* = \big(X^\top X\big)^{-1} X^\top y.$$

What happens if $X^\top X$ is not invertible (e.g., $d > n$ or columns of X are linearly dependent)?

(b) Comment on the geometric or algebraic interpretation of the solution w_2^*. What would it mean when $X^\top X$ is singular?

(c) From a learning-theoretic perspective, under what circumstances can small empirical risk (training error) imply small true risk (test error)? Briefly discuss how model complexity and sample size affect this relationship.

(d) Now consider a different empirical risk function (L_1 empirical risk),

$$R_1(w) = \frac{1}{n}\sum_{i=1}^{n}\big|y_i - \langle w,\, x_i\rangle\big|.$$

How can you characterize $w_1^* \in \arg\min_w(R_2(w))$? That is, what are necessary conditions for minimizing the L_1 empirical risk? Can you comment on any geometric or algebraic interpretation of w_1^*?

Problem 5. Python Coding

If you are unfamiliar with deep learning software packages, please begin with the PyTorch tutorials on the course website. You may also use LLMs like ChatGPT for your learning purposes (e.g., to generate examples and debug your code), but you remain responsible for any errors in your code or plots.

(a) Consider a training dataset $S = \{(x_i, y_i)\}_{i=1}^n$ drawn i.i.d. from the following distribution: $X \in \mathbb{R}^d$ is sampled from an equal-weight mixture of two Gaussians, $\frac{1}{2}\mathrm{N}(\mu, I_d) + \frac{1}{2}\mathrm{N}(-\mu, I_d)$, and $y \in \{\pm 1\}$ indicates which mixture component X is from. Specifically, one can generate (x_i, y_i) as follows:

- $y \sim \mathrm{Unif}\{\pm 1\}$,
- $z \sim \mathrm{N}(0, I_d)$,
- $x = y\,\mu + z$.

Provide Python code to create such a dataset S given n, d, and μ. Print the shapes of X and y to confirm correctness, and visualize a 2D projection (choose two coordinates if $d > 2$) color-coded by y for an intuitive view.

(b) We consider a 2-layer neural network with parameters $a \in \mathbb{R}^m$ and $W = \begin{bmatrix} w_1 & \cdots & w_m \end{bmatrix}^\top \in \mathbb{R}^{m\times d}$:

$$f(x; W, a) \;=\; \sum_{j=1}^{m} a_j\, \phi\big(\langle w_j, x\rangle\big),$$

where the activation function is ReLU, $\phi(q) = \max\{0, q\}$. Provide PyTorch code (or your own Python, though PyTorch is recommended) that instantiates a two-layer network class allowing the user to specify the number of neurons m. (Hint: you should be using linear layers, and `nn.Sequential` or `nn.ModuleList` can simplify building multi-layer nets.)

(c) Define the empirical risk over the training data via the margin $y_i \cdot f(x_i; W, a)$:

$$\hat{L}(W, a) \;=\; \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i \cdot f(x_i; W, a)\big),$$

where $\ell(q) = \log\big(1 + e^{-q}\big)$ is the binary cross-entropy loss. Let $\theta = (W, a)$ and perform gradient descent with fixed learning rate η:

$$\theta_{t+1} \;=\; \theta_t \;-\; \eta \, \nabla \hat{L}(\theta_t),$$

starting from some initial θ_0. Provide PyTorch code that:

(i) Trains the two-layer network (using default PyTorch initialization) with full-batch gradient descent, logging both training loss (and optionally classification accuracy) at each iteration.

(ii) Reports the final loss and classification accuracy on a validation set.

The code should allow user-specified n_{train}, n_{valid}, d, m, and learning rate η.

(d) Plot the training results in two different settings. In both, let $\mu = d^{\alpha} \cdot v$ with $\alpha = 0.25$, where v is sampled uniformly from the unit sphere, and set $\eta = 0.001$:

(i) $d = 1000$, $n = 100$,

(ii) $d = 100$, $n = 1000$.

For each setting, generate two plots with two lines each: one plot showing training vs. validation loss, and the other showing training vs. validation accuracy.

(e) If time permits, vary the data model parameters (d, n, α), learning rate η or hidden size m. How does it affect training stability, speed, and final accuracy? Provide brief comments or additional plots.