HY673 - Spring 2026 Problem Set #1

Yannis Pantazis (25/02/2026)

Instructions

- **Due date:** Wednesday March 11th, 2026
- Submission via e-mail to the class account: hy673@csd.uoc.gr
- Provide one file with the written solutions.
- Provide one folder with code.
  - The name of each file in the folder should indicate the respective exercise.
  - Your code should run on colab.
- All assignments in this course are individual, not group, assignments. You may freely discuss homework assignments with your fellow classmates. The final solutions, however, must be written entirely on your own. This includes programming assignments.
- You are allowed to use generative AI tools (e.g., ChatGPT, Gemini) for homework assignments only for grammatical corrections, unless explicitly permitted otherwise. To maintain academic integrity, students must disclose any use of AI-generated material.

Problem 1 (Change of Variable and Inverse Transform – 20 points)**.** *Let $X \sim Exp(1)$ be an exponential random variable with probability density function $f_X(x) = e^{-x}$ for $x \geq 0$. Define the transformed random variable $Y = -\log(X)$.*

(a) Analytical Derivation. *Calculate the probability density function (PDF), $f_Y(y)$, using the change of variable formula. Specify the support of the resulting distribution.*

(b) Numerical Validation via Transformation. *Generate a dataset $\{y_i = -\log(x_i)\}_{i=1}^n$ where $\{x_i\}_{i=1}^n$ are independent samples drawn from $Exp(1)$.*

- *Compute and plot the normalized histograms for $n \in \{10^3, 10^4, 10^5\}$.*
- *Overlay the analytical PDF $f_Y(y)$ derived in part (a) on the same figure.*
- *Describe the convergence behavior as n increases.*

(c) Inverse Transform Sampling. *Repeat the visualization from part (b), but generate the dataset* $\{y_i\}_{i=1}^n$ *using the Inverse Transform method:*

$$y_i = F_Y^{-1}(u_i), \quad u_i \sim \mathcal{U}(0,1)$$

where $F_Y(y) = \int_{-\infty}^{y} f_Y(z)\,dz$ *is the cumulative distribution function (CDF). You may derive* F_Y^{-1} *analytically or utilize the* `sympy.integrate` *function in Python to determine the CDF and its inverse numerically.*

Problem 2 (Gaussian Linear Combinations – 20 points)**.** *Let* $X = [X_1, X_2, X_3]^T$ *be a random vector distributed as* $X \sim \mathcal{N}(\mu, \Sigma)$*, where* $\mu \in \mathbb{R}^3$ *and* $\Sigma \in \mathbb{R}^{3\times 3}$*.*

Define the transformation $Y = [Y_1, Y_2]^T$ *by:*

$$Y_1 = X_1 + X_2$$
$$Y_2 = X_2 + X_3$$

(a) Theoretical Derivation. *Determine the joint distribution of* Y*. Specifically, derive the expressions for the mean vector* μ_Y *and the covariance matrix* Σ_Y *as functions of the original parameters* μ *and* Σ*.*

(b) Numerical Validation. *Verify your results from part (a) using a Monte Carlo approach. Generate* n *samples from the 3-dimensional distribution* $\mathcal{N}(\mu, \Sigma)$ *for* $\mu = [0, 1, -1.5]^T$ *and*

$$\Sigma = \begin{bmatrix} 1 & 0.5 & -0.8 \\ 0.5 & 0.9 & -0.7 \\ -0.8 & -0.7 & 1.1 \end{bmatrix},$$

and compute the corresponding samples for Y*.*

Perform this estimation for the following sample sizes:

$$n \in \{10^3, 2\times 10^3, 10^4, 2\times 10^4, 10^5\}$$

For each n*, report the estimated sample mean vector* $\hat{\mu}_Y$ *and sample covariance matrix* $\hat{\Sigma}_Y$*. Comment on the convergence of these estimates to your theoretical results.*

Problem 3 (Maximum Likelihood Estimation for the Poisson Distribution – 20 points)**.** *Let* $X_1, X_2, \ldots, X_n$ *be a sequence of independent and identically distributed (i.i.d.) random variables following a Poisson distribution with unknown parameter* $\lambda > 0$*, such that:*

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}, \quad k \in \{0, 1, 2, \ldots\}$$

(a) Data Simulation. *Generate a synthetic dataset of size* $n = 1000$ *using a true parameter value* $\lambda^* = 3$*. Plot the empirical frequency distribution (histogram) of the sampled data and compare it visually with the theoretical Poisson PMF.*

(b) The Log-Likelihood Function. *Given the observed sample* $\mathbf{x} = \{x_1, \ldots, x_n\}$*, write down the joint likelihood function* $L(\lambda; \mathbf{x})$*.*

(c) Analytical Derivation of the MLE. *Derive the Maximum Likelihood Estimator (MLE),* $\hat{\lambda}_{MLE}$*, by finding the stationary point of the log-likelihood function. Verify the second-order condition to confirm that this point is indeed a maximum.*

(d) Numerical Verification. *Using the data simulated in part (a), numerically maximize the log-likelihood function using an optimization routine (e.g.,* `scipy.optimize` *or* `optim()`*). Verify that the numerical estimate matches your analytical result.*

(e) Asymptotic Normality. *Recall that under standard regularity conditions,* $\sqrt{n}(\hat{\lambda}_{MLE} - \lambda^*) \xrightarrow{d} \mathcal{N}(0, I(\lambda^*)^{-1})$*, where* $I(\lambda)$ *is the Fisher Information.*

1. *Calculate the Fisher Information* $I(\lambda) = -E\left[\frac{\partial^2}{\partial \lambda^2} \ln f(X; \lambda)\right]$*.*
2. *Perform a Monte Carlo study: Repeat the estimation process* $B = 1000$ *times. Plot the distribution of the normalized errors and overlay the theoretical Gaussian density to verify the asymptotic normality.*

Problem 4 (Exploring the Double Descent Phenomenon – 30 points)**.** *This problem investigates the non-monotonic relationship between model capacity and generalization error, identifying the "interpolation threshold" where classical statistical intuition fails.*

(a) Dataset and Protocol. *Generate a synthetic dataset* $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ *where* $y = \sin(w^\top x) + \varepsilon$*. Here,* $x \in \mathbb{R}^d$ *is the input vector, and* $w \in \mathbb{R}^d$ *is a fixed weight vector typically sampled from a unit sphere or* $\mathcal{N}(0, \frac{1}{d} I)$*. The noise term is* $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$*. For* $d \in \{10, 50, 100\}$*, fix a small training set size* n *and a much larger independent test set. Standardize all inputs. To ensure statistical reliability, all subsequent plots should display the mean and* ± 1 *standard deviation across at least 5 random seeds.*

(b) Capacity Sweep and Training Dynamics. *Train a one hidden layer MLP,* $\hat{y} = W_2 \phi(W_1 x + b_1) + b_2$*, using MSE loss. Vary the hidden width* $m \in \{2^1, 2^2, \ldots, 2^{11}\}$ *and calculate the total parameter count* $p(m)$ *for each. For three distinct regimes: underparameterized* ($p \ll n$)*, critical* ($p \approx n$)*, and overparameterized* ($p \gg n$)*, plot the training loss vs. epochs to observe how convergence speed changes near the interpolation threshold.*

(c) The Double Descent Curve. *Plot the final training and test errors against the model width* m *(or the ratio* p/n*) using a logarithmic x-axis. Identify the* ***interpolation threshold****: the point where training error first vanishes. Observe the "peak" in test error at this threshold and the subsequent improvement in generalization as* m *continues to increase.*

(d) Sensitivity to Noise and Regularization. *Analyze how the test error peak is influenced by external factors. First, repeat the sweep for varying noise levels* $\sigma_\varepsilon \in \{0, 0.1, 0.5, 1.0\}$ *and overlay the test error curves on a single plot. Second, introduce weight decay (i.e., L2 regularization) and describe its effect on the magnitude of the generalization gap near the interpolation threshold.*

Problem 5 (The Great Deblurring Cook-off – 20 points)**.** *In this quiz, you act as . Your mission as a forensic imaging specialist is to recover clear signals from five "corrupted" images using state-of-the-art classical and AI tools. Generative AI tools can be freely used.*

(a) Crime Scene Investigation: Blur Characterization. *Before restoring, you must diagnose. Identify the "fingerprint" of the blur: is it **Gaussian** (smooth/hazy), **Motion** (directional streaks), or **Defocus** (circular "bokeh" artifacts)? Briefly explain which of these is most likely to suffer from "ringing" during restoration.*

(b) The Restoration Toolkit. *Apply at least **three** different "chefs' techniques" to each image. You may choose from:*

- *Classical: Wiener Filtering or Richardson-Lucy Deconvolution.*
- *Modern: A Generative Deep Learning model trained for Deblurring.*
- *Tool: A dedicated (and free) AI tool for Deblurring.*

Record the processing time for each. Does the quality of AI justify the computational "cost"?

(c) Forensic Evaluation. *For real-world images where no "ground truth" exists, provide a "Blind Subjective Score" (1–10). Look specifically for:*

- ***Artifacts:** Are there "halos" or "ripples" (ringing) around high-contrast edges?*
- ***Hallucinations:** Did the AI model invent details that weren't there?*

(d) The "Verdict" Report. *Summarize your findings in a concise "Forensic Brief". Include a side-by-side comparison figure of your best recovery vs. the original blur. Which tool would you trust for a license plate recognition task, and why?*