# DSC 210 Numerical Linear Algebra, Fall 2025

Homework Problems for Topic 4: *ML and Least Squares Problems*

Student Name (PID):

---

Write your solutions to the following problems by typing them in LaTeX. Unless otherwise noted by the problem's instructions, show your work and provide justification for your answer. Homework is due via Gradescope at **11th December 2025, 11:59 PM**.

**Late Policy**: If you submit your homework after the deadline we will apply a late penalty of 10% per day.

**Guidelines for Homework Related Questions:**

(a) As a general rule, we can help you understand the homework problems and explain the material from the corresponding lectures, but we cannot give you the entire solution.

(b) Regarding debugging programming questions: We ask you to do some debugging on your own first, including printing out intermediate values in your algorithms, trying a simpler version of the problem, etc.

(c) We will not be pre-grading the homework, i.e. we won't confirm if the answer you have is correct.

**AI Usage Policy:**

(a) Code: You may use LLMs to debug your code; however, you may not use LLMs to generate your entire code, and code must be reviewed and tested.

(b) Writing: You may use LLMs to correct grammar, style and latex issues; however, you may not use LLMs to generate entire solutions, sentences or paragraphs. All writing must be in your own voice.

**Academic Integrity Policy:**

The UC San Diego Academic Integrity Policy (formerly the Policy on Integrity of Scholarship) is effective as of September 25, 2023 and applies to any cases originating on or after September 25, 2023. The university expects both faculty and students to honor the policy. For students, this means that all academic work will be done by the individual to whom it's assigned, without unauthorized aid of any kind. If violations of academic integrity occur, the same Sanctioning Guidelines apply regardless of which policy was effective for that case.

For more information on how the policy is implemented, refer to the most current procedures. Remember: When in doubt about what constitutes appropriate collaboration or resource use, please ask TAs before proceeding. It's always better to clarify expectations than to risk an academic integrity violation. Academic integrity violations can have serious consequences for your academic record, and you will get zero grades.

You can access the Homework Template using the following link: `https://www.overleaf.com/read/vfhcmsppvskp`

**Question 1: Least Squares Regression (10 points)**

Consider a dataset with $m$ datapoints: $(t_i, y_i), i = 1, ..., m$. We want to minimize least squares by fitting an estimation function

$$\hat{y}(t) = at^2 + bt + c$$

on this data. This problem can be written in the form $\mathbf{Ax} = \mathbf{y}$, such that the least squares solution for $\mathbf{x}$ gives the coefficients $a, b, c$.

What are $\mathbf{A}$, $\mathbf{x}$ and $\mathbf{y}$ in terms of the variables above?

**Solution:**

**Question 2: Linear Regression (30 points)**

Use a linear regressor to fit the data, where $\hat{y}(t) = at + b$ with $a, b$ being scalars. Let $\mathbf{x} = \begin{bmatrix} a \\ b \end{bmatrix}$ contain the regressor coefficients. Recall that the linear algebraic formula for least squares gives $\mathbf{x} = (A^\top A)^{-1}A^\top \mathbf{y}$ with $A^\dagger = (A^\top A)^{-1}A^\top$ known as the pseudo-inverse of $A$.

(a) Implement 3 different ways to find the regressor coefficients using the `numpy` package and show that they agree on the random data generated in the notebook (make sure to calculate both $a$ and $b$):

    i Calculate $\mathbf{x}$ directly from the least squares formula.

    ii Use the function `np.linalg.pinv` to find the values of regressor coefficients $\mathbf{x}$.

    iii Solve the problem using the built-in `numpy` function: `np.linalg.lstsq`

(b) Draw a scatter plot between $\mathbf{T}$ and $\mathbf{y}$, and overlay it with the linear regression line.

**Solution:** We have provided a starter code to generate the datapoints. **Do not modify the provided code block -** google colab link.

(For all the following questions, make a local copy of the colab notebook. Write the solution in the copy and submit.)

**Question 3: Logarithmic Regression (30 points)**

(a) Write a function `my_func_fit(T,y)`, where **T** and **y** are column vectors of the same size containing experimental data. The function should return the values for $\alpha$ and $\beta$ which are the scalar parameters of the estimation function

$$\hat{y}(t) = \alpha.t^{\beta}$$

   *Hint*: Minimize least squares in log-space, i.e. $\min \sum_i (\log(\hat{y}_i) - \log(y_i))^2$.

(b) Test your code on the generated sample dataset and report the coefficients. A starter code to generate the logarithmic dataset is provided in the notebook.

(c) Plot a graph between **T** vs **y**, and overlay it with the regression line.

**Note:** You are only allowed to use **numpy** library functions.

**Solution:** Provide the solution in the notebook google colab.

**Question 4: Functional Regression (30 points)**

(a) Write a function `my_lin_regression(f, T, y)`, where $f$ is a list containing function objects to pre-defined basis functions and, **T** and **y** are arrays containing noisy data. Assume that **T** and **y** are the same size, i.e, $T^{(i)} \in \mathbb{R}, y^{(i)} \in \mathbb{R}$.

Return an array $\beta$ which represent the coefficients of the solved problem. We are solving for the $\beta$ which contains the coefficients in the regressor

$$\hat{y}(t) = \beta_0 + \beta_1 f_1(t) + \cdots + \beta_n f_n(t)$$

with $f_i$ being the pre-defined basis functions.

(b) Write a function `regression_plot(f,T,y,beta)` which plots a graph between **T** and **y** , and overlays it with the regression line.

Run the provided test scenarios provided in the notebook. First one uses $f = [\sin, \cos]$ and second $f = [\exp]$.

**Note:** You are only allowed to use **numpy** library functions.

**Solution:** Provide the solution in the notebook google colab.

**(BONUS)Question 5: Ridge Regression (10 points)**

Let $A \in \mathbb{R}^{n \times d}$ and $y \in \mathbb{R}^n$, where $n > d$. Assume that $A$ is rank-deficient, i.e. $\text{rank}(A) = r < d$. For $\lambda > 0$, consider the ridge regression problem

$$L_\lambda(w) = \|Aw - y\|_2^2 + \lambda \|w\|_2^2, \quad w \in \mathbb{R}^d.$$

(a) **Gradient and Hessian.**

    i. Compute the gradient $\nabla L_\lambda(w)$.

    ii. Compute the Hessian $H_\lambda = \nabla^2 L_\lambda(w)$.

(b) **Closed-form ridge solution via normal equations.** Show that any minimizer $w_\lambda^\star$ of $L_\lambda$ satisfies the linear system

$$(A^\top A + \lambda I_d) \, w_\lambda^\star = A^\top y,$$

and hence

$$w_\lambda^\star = (A^\top A + \lambda I_d)^{-1} A^\top y.$$

(c) **SVD and spectrum of the Hessian.** Let the singular value decomposition (SVD) of $A$ be

$$A = U \Sigma V^\top,$$

where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{d \times r}$ have orthonormal columns, and

$$\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_r), \quad \sigma_1 \geq \cdots \geq \sigma_r > 0.$$

    i. Express $A^\top A$ and $H_\lambda$ in terms of $U$, $\Sigma$, and $V$.

    ii. Show that the eigenvalues of $H_\lambda$ are

$$2(\sigma_1^2 + \lambda), \ldots, 2(\sigma_r^2 + \lambda), \underbrace{2\lambda, \ldots, 2\lambda}_{d-r \text{ times}}.$$

(d) **Ridge solution in the SVD basis and relation to the pseudo-inverse.**

    i. Using the SVD of $A$, derive an explicit formula for $w_\lambda^\star$ of the form

$$w_\lambda^\star = \sum_{i=1}^{r} \alpha_i(\lambda) \, v_i,$$

    where $v_i$ are the columns of $V$, and identify the coefficients $\alpha_i(\lambda)$ in terms of $\sigma_i$ and $U^\top y$.

    ii. Show that, as $\lambda \to 0^+$,

$$w_\lambda^\star \to w_0^\star := A^+ y,$$

    the minimum-norm least squares solution. Here $A^+$ is the pseudo-inverse of $A$.

(e) Verify the solutions with the code template provided in the notebook.

**Note:** You are only allowed to use `numpy` library functions.

**Solution:** Typewrite the solution for parts (a) to (d) in latex and code the solution to (e) in the notebook google colab.