



**STOR 512: OPTIMIZATION FOR ML AND NN**  
**DEPARTMENT OF STATISTICS AND OPERATIONS RESEARCH**  
----- **SPRING 2026** -----

INSTRUCTOR: QUOC TRAN-DINH

---

**HOMEWORK 2: MATH TOOLS REVIEW AND LINEAR LEAST-SQUARES**

**Note:** Please do not distribute this homework without instructor's permission.

**Question 1. (20 points):** Given  $\alpha \in \mathbb{R}$ , consider the following matrices:

$$A = \begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & \alpha \\ \alpha & \alpha^2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & \alpha \\ 0 & \alpha^2 & \alpha \\ \alpha & \alpha & 1 + \alpha^2 \end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix} 1 & \alpha & 0 \\ \alpha & 1 & \alpha \\ 0 & \alpha & 1 \end{bmatrix}.$$

- For each matrix, find all the values of  $\alpha$  such that such a matrix is positive semidefinite, respectively, positive definite.
- **Extra questions: (These questions will not be graded.)** For matrix  $B$ , can you find a matrix  $U$  such that  $B = UU^\top$ ? For matrix  $C$ , when  $\alpha = 1$ , can you find a matrix  $U$  such that  $C = UU^\top$ ?

**Question 2. (20 points):** Analytically calculate the gradient and Hessian of the the following function:

$$f(x) := \sum_{i=1}^d p_i \log \left( \frac{e^{x_i}}{\sum_{i=1}^d e^{x_i}} \right),$$

where  $x \in \mathbb{R}^d$  is a variable, and  $p \in \mathbb{R}^d$  is a given vector of parameter such that  $\sum_{i=1}^d p_i = 1$  and  $p_i \geq 0$  for all  $i$ . Implement two functions in Python to evaluate this gradient and Hessian, respectively for any given input vectors  $x$  and  $p$ . Provide an example for  $p = (0.2, 0.3, 0.1, 0.4)^\top$  and  $x = (1, 2, -1, 4)^\top$  to test your function.

**Question 3. (20 points):** Prove that the following one-variable functions are convex.

- $\varphi(t) = \sqrt{t^2 + 1}$  on  $\mathbb{R}$ ;
- $\psi(t) = -\sqrt{1 - t^2}$  on  $(-1, 1)$ .

Now, assume that  $\ell$  is one of the above two functions  $\varphi$  and  $\psi$ . We consider the following function

$$f(x) = \sum_{i=1}^n \ell(a_i^\top x + b_i),$$

where  $n \geq 1$ ,  $a_i \in \mathbb{R}^d$  and  $b_i \in \mathbb{R}$  are given for all  $i = 1, \dots, n$ . Prove that  $f$  is convex. Compute the gradient of  $f$  (write down the mathematical form of this gradient for each case  $\ell = \varphi$  and  $\ell = \psi$ ).

**Question 4. (20 points):** The following dataset is drawn from a quadratic model:  $C(x) = 1500 + 20x + 0.05x^2 + \epsilon$ , where  $x$  represents the amount of products,  $C(x)$  represents the cost of producing  $x$  products, and  $\epsilon$  is a Gaussian noise of zero mean and variance  $\sigma^2$ , where  $\sigma = 0.1$ .

Products	600	650	700	750	800	850	900	950	1000
$C(x)$ [\$]	31499.99	35624.92	40000.10	44624.98	49499.92	54625.13	59999.97	65624.94	71500.07

In practice, we do not know the model above, but only know some observed data (as in the above table, where "Products" is  $x$ ). Using this dataset to form a linear regression model and estimating its coefficient vector  $\beta$ . Solve the linear regression models by three different methods:

- using directly the normal equation
- using the Cholesky decomposition, and
- using the `scikit-learn` package.

Provide the details (math derivations, code, results, and explanation) of each case. Make a new prediction for 4 different values of  $x$  as  $\hat{x} \in \{105, 120, 200, 250\}$ . Plot the results of your experiments using `matplotlib`.

**Question 5. (20 points):** The following least-squares regression is slightly different from the standard one by incorporating a weight  $w_i$  for each sample  $x^{(i)}$ :

$$\min_{\beta \in \mathbb{R}^{d+1}} \left\{ \mathcal{L}(\beta) := \frac{1}{2} \sum_{i=1}^n w_i (\beta_0 + \beta_1 x_1^{(i)} + \cdots + \beta_d x_d^{(i)} - y_i)^2 \right\}, \quad (1)$$

where  $x^{(i)} \in \mathbb{R}^d$  and  $y_i \in \mathbb{R}$  are given input data for  $i = 1, \dots, n$ , and  $w_i > 0$  are given weights for  $i = 1, \dots, n$ .

(a) Solve this optimization problem directly using its Fermat's rule for the case  $d = 2$  and  $n = 3$ , where

$$w = \begin{pmatrix} 1 \\ 2 \\ 1.5 \end{pmatrix}, \quad X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ (x^{(3)})^T \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ -2 & 1 \\ 3 & 4 \end{bmatrix}, \quad \text{and} \quad y = \begin{pmatrix} 15 \\ 12 \\ 20 \end{pmatrix}.$$

(b) Reformulate (1) into a standard least-squares problem using the change of variables technique:

$$\min_{\hat{\beta} \in \mathbb{R}^{d+1}} \left\{ \hat{\mathcal{L}}(\hat{\beta}) := \frac{1}{2} \sum_{i=1}^n (\hat{\beta}_0 + \hat{\beta}_1 \hat{x}_1^{(i)} + \cdots + \hat{\beta}_d \hat{x}_d^{(i)} - \hat{y}_i)^2 \right\}, \quad (2)$$

where  $\hat{\beta}$  is a new parameter vector. How to recover the optimal parameter vector  $\beta^*$  of (1) from the optimal parameter vector  $\hat{\beta}^*$  of (2)?

(c) Generate a dataset  $\{(x^{(i)}, y_i)\}_{i=1}^n$ , where  $x^{(i)}$  is a random vector generated by Gaussian distribution of mean  $\mu = 5$  and variance  $\sigma^2$  with  $\sigma = 0.2$  for all  $i = 1, \dots, n$ , with  $n = 10$ . (You can use `X = 5 + 0.2 * np.random.randn(n, d)` to generate  $X$ ). The response  $y_i$  is generated by the following linear model:

$$y_i = 1 + \sum_{j=1}^d (j+1)x_j^{(i)} + \epsilon,$$

where  $\epsilon$  is a Gaussian noise of zero mean and variance  $\sigma^2$  with  $\sigma = 0.5$ , and  $d = 8$ . Solve the least-squares problem (1) associated with this dataset for two cases:

- Case 1:  $w_i = 1$  for  $i = 1, \dots, n$ .
- Case 2:  $w$  is a random vector generated uniformly between  $(0, 1)$  (e.g., `w = np.random.rand(n, 1)`).