Homework 1 - Introduction to Machine Learning for Engineers

kipngeno koech - bkoech

January 31, 2025

1 Probability

1. Suppose W is a Gaussian random variable with distribution $N(\mu, \sigma^2)$ and U a uniform random variable over the interval $[a, b]$. Assuming that W and U are independent, what is the expected value $\mathbb{E}[Z]$ and variance $\text{Var}[Z]$ of $Z = 3W + 2U$?

   - **Expected value:**
   $$\mathbb{E}[Z] = \mathbb{E}[3W + 2U] = 3\mathbb{E}[W] + 2\mathbb{E}[U]$$
   Since W is Gaussian with mean μ and U is uniform over $[a, b]$ with mean $\frac{a+b}{2}$:
   $$\mathbb{E}[Z] = 3\mu + 2\left(\frac{a+b}{2}\right) = \mathbf{3}\boldsymbol{\mu} + (\mathbf{a} + \mathbf{b})$$

   - **Variance:** if a random variable X, is scaled by a constant a, then the variance of the scaled random variable is a^2 times the variance of the original random variable. Therefore:
   $$\text{Var}[Z] = \text{Var}[3W + 2U] = 3^2\text{Var}[W] + 2^2\text{Var}[U]$$
   Since W is Gaussian with variance σ^2 and U is uniform over $[a, b]$ with variance $\frac{(b-a)^2}{12}$:
   $$\text{Var}[Z] = 9\sigma^2 + 4\left(\frac{(b-a)^2}{12}\right) = 9\sigma^2 + \frac{(b-a)^2}{3}$$

2. Consider the following joint distribution between the random variable X, which takes values T or F, and the random variable Y, which takes values a, b, c, or d.

$P(X,Y)$	$Y = a$	$Y = b$	$Y = c$	$Y = d$
$X = T$	0.1	0.2	0.1	0.1
$X = F$	0.1	0.1	0.2	0.1

   (a) **Marginal distribution P_Y:**
   $$\begin{aligned}
   \Pr(Y = a) &= \Pr(X = T, Y = a) + \Pr(X = F, Y = a) = 0.1 + 0.1 = \mathbf{0.2}\\
   \Pr(Y = b) &= \Pr(X = T, Y = b) + \Pr(X = F, Y = b) = 0.2 + 0.1 = \mathbf{0.3}\\
   \Pr(Y = c) &= \Pr(X = T, Y = c) + \Pr(X = F, Y = c) = 0.1 + 0.2 = \mathbf{0.3}\\
   \Pr(Y = d) &= \Pr(X = T, Y = d) + \Pr(X = F, Y = d) = 0.1 + 0.1 = \mathbf{0.2}
   \end{aligned}$$

   (b) **Conditional probability $\Pr(X = T \mid Y \in \{b, c, d\})$:** since we are conditioning on Y being in the set $\{b, c, d\}$, we need to find the probability of $X = T$ and Y being in the set $\{b, c, d\}$ and divide it by the probability of Y being in the set $\{b, c, d\}$:
   this is the bayes theorem:
   $$\Pr(A \mid B) = \frac{\Pr(B \mid A)\Pr(A)}{\Pr(B)} = \frac{\Pr(A \cap B)}{\Pr(B)}$$
   $$\begin{aligned}
   \Pr(Y \in \{b, c, d\}) &= \Pr(Y = b) + \Pr(Y = c) + \Pr(Y = d) = 0.3 + 0.3 + 0.2 = 0.8\\
   \Pr(X = T \cap Y \in \{b, c, d\}) &= \Pr(X = T, Y = b) + \Pr(X = T, Y = c) + \Pr(X = T, Y = d) = 0.2 + 0.1 + 0.1 = 0.4\\
   \Pr(X = T \mid Y \in \{b, c, d\}) &= \frac{\Pr(X = T \cap Y \in \{b, c, d\})}{\Pr(Y \in \{b, c, d\})} = \frac{0.4}{0.8} = \mathbf{0.5}
   \end{aligned}$$

2 Linear Algebra

1. Let $A_k \in \mathbb{R}^{n\times n}$ for $k = 1, \ldots, K$ such that $A_k = A_k^\top$, i.e., each A_k is a symmetric, n-dimensional square matrix. Suppose all A_k have the exact same set of eigenvectors $u_1, u_2, \ldots, u_n$ with the corresponding eigenvalues $\alpha_{k1}, \ldots, \alpha_{kn}$ for each A_k. Write down the eigenvectors and their corresponding eigenvalues for the following matrices:

   (a) $C = \sum_{k=1}^{K} A_k$

   $$\text{Eigenvectors: } \mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_n}$$

   $$\text{Eigenvalues: } \sum_{\mathbf{k=1}}^{\mathbf{K}} \alpha_{\mathbf{k1}}, \sum_{\mathbf{k=1}}^{\mathbf{K}} \alpha_{\mathbf{k2}}, \ldots, \sum_{\mathbf{k=1}}^{\mathbf{K}} \alpha_{\mathbf{kn}}$$

   (b) $D = A_i^{-1} A_j A_i$, where $i \neq j$ and $i, j \in \{1, 2, \ldots, K\}$. Here we assume A_i is invertible. A matrix A is similar to a matrix B if there exists an invertible matrix P such that $A = P^{-1}BP$. Similar matrices have the same eigenvalues. Therefore, the eigenvalues of D are the same as the eigenvalues of A_j.
   Since A_j has the same eigenvectors as A_i, the eigenvectors of D are the same as the eigenvectors of A_i.

   $$\text{Eigenvectors: } \mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_n}$$

   For the eigenvalues:

   $$\text{Eigenvalues: } \alpha_{\mathbf{j1}}, \alpha_{\mathbf{j2}}, \ldots, \alpha_{\mathbf{jn}}$$

2. Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ be given, and $\text{col}(A)$ be the column space of A. For a given value of m, under what conditions on b, $\text{col}(A)$, and $\text{rank}(A)$ will the equation $Ax = b$ have:
   The column space of a matrix A is the set of all possible linear combinations of the columns of A.:

   $$\text{col}(A) = \{A\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n\}$$

   The rank of a matrix is the dimension of the column space of the matrix.

   (a) **No solution:**

   The equation $Ax = b$ has no solution if $b \notin \text{col}(A)$. This means that b is not a linear combination of the columns of A.

   (b) **Exactly one solution:**

   The equation $Ax = b$ has exactly one solution if $b \in \text{col}(A)$ and $\text{rank}(A) = n$.

   (c) **Infinitely many solutions:**

   The equation $Ax = b$ has infinitely many solutions if $b \in \text{col}(A)$ and $\text{rank}(A) < n$.

3 Matrix Calculus

1. Find the first derivative of the following functions with respect to X. Before you attempt the questions below, you are encouraged to review the Matrix Calculus Wikipedia page. Note that in this problem, we follow the convention that the derivative of a scalar function $f(X)$ with respect to X should have the same dimension as X.

   (a) $f(X) = \mathrm{tr}(XX^\top)$, where $X \in \mathbb{R}^{n\times n}$ and tr is the trace of a square matrix.
   The derivative of the trace of a matrix is the matrix itself:

   $$\frac{\partial f}{\partial X} = \mathbf{2X}$$

   (b) $f(X) = a^\top X b$, where $X \in \mathbb{R}^{m\times n}$ and $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$.
   when you multiply a row vector by a matrix, the derivative is the row vector itself, here we have two row vectors, so the derivative is the outer product of the two row vectors. The first row vector is of dimension $1 \times m$ and the second row vector is of dimension $n \times 1$, so the derivative is of dimension $m \times n$, to get the resulant matrix, we need to adjust dimensions of the row vectors:

   $$\frac{\partial f}{\partial X} = \mathbf{ab}^\top$$

   (c) $f(X) = \|Xb\|^2$, where $X \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^n$.
   When we expand the Euclidean norm, we get:

   $$\|Xb\|^2 = (Xb)^\top(Xb) = b^\top X^\top X b$$

   $$\frac{\partial f}{\partial X} = \mathbf{2Xbb}^\top$$

4 MLE/MAP

1. Suppose $x \in \mathbb{R}$ is fixed and given. Assume $\lambda > 0$ is a scalar parameter, and $y \sim \text{Poisson}(\lambda)$, where $\text{Poisson}(\lambda)$ denotes the Poisson distribution with rate λ.

   (a) **Maximum likelihood estimation:**

   i. Write down the PMF (probability mass function) of $y \mid \lambda$.

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

   ii. Assume N independent observations $y_1, y_2, \ldots, y_N$ are drawn, where $y_n \sim \text{Poisson}(\lambda)$ for $n = 1, 2, \ldots, N$. Derive the joint PMF $P(y_1, \ldots, y_N \mid \lambda)$.

$$P(y_1, \ldots, y_N \mid \lambda) = \prod_{n=1}^{N} P(y_n \mid \lambda) = \prod_{n=1}^{N} \frac{\lambda^{y_n} e^{-\lambda}}{y_n!} = \frac{\lambda^{\sum_{n=1}^{N} y_n} e^{-N\lambda}}{\prod_{n=1}^{N} y_n!}$$

   iii. Write down the negative log-likelihood function of the joint PMF derived in the previous problem and simplify it into a form that can be minimized. (Hint: remove the terms that do not have λ).

$$\mathcal{L}(\lambda) = -\log P(y_1, \ldots, y_N \mid \lambda) = -\log\left(\frac{\lambda^{\sum_{n=1}^{N} y_n} e^{-N\lambda}}{\prod_{n=1}^{N} y_n!}\right)$$

$$\mathcal{L}(\lambda) = -\left(\sum_{n=1}^{N} y_n \log\lambda - N\lambda - \log\left(\prod_{n=1}^{N} y_n!\right)\right)$$

$$\mathcal{L}(\lambda) = -\sum_{n=1}^{N} y_n \log\lambda + N\lambda + \text{const}$$

   (b) **Maximum-a-posteriori (MAP) estimation:** Suppose $\lambda \sim \text{Gamma}(\alpha, \beta)$, with PMF:

$$P(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}, \quad \lambda > 0$$

   i. Write the joint distribution $P(y_1, \ldots, y_N, \lambda)$ where each $y_n \sim \text{Poisson}(\lambda)$ is drawn independently as above.

$$P(y_1, \ldots, y_N, \lambda) = P(y_1, \ldots, y_N \mid \lambda) P(\lambda)$$

$$P(y_1, \ldots, y_N, \lambda) = \left(\frac{\lambda^{\sum_{n=1}^{N} y_n} e^{-N\lambda}}{\prod_{n=1}^{N} y_n!}\right)\left(\frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}\right)$$

$$P(y_1, \ldots, y_N, \lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\lambda^{\sum_{n=1}^{N} y_n + \alpha - 1} e^{-(N+\beta)\lambda}}{\prod_{n=1}^{N} y_n!}$$

   ii. Write the negative logarithm of $P(y_1, \ldots, y_N, \lambda)$ and simplify it into a form that can be minimized to find the MAP estimate $\hat{\lambda}$. (Hint: remove the terms that do not have λ).

$$\mathcal{L}_{\text{MAP}}(\lambda) = -\log P(y_1, \ldots, y_N, \lambda) = -\log\left(\frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\lambda^{\sum_{n=1}^{N} y_n + \alpha - 1} e^{-(N+\beta)\lambda}}{\prod_{n=1}^{N} y_n!}\right)$$

$$\mathcal{L}_{\text{MAP}}(\lambda) = -\left(\sum_{n=1}^{N} y_n + \alpha - 1\right) \log\lambda + (N+\beta)\lambda + \text{const}$$

5 Linear Regression with Regularization

1. Consider a dataset with N samples (x_i, y_i), where:

$$y_i = x_i^\top w + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

and $w \in \mathbb{R}^d$ is the weight vector, ϵ_i is Gaussian noise, and $x_i \in \mathbb{R}^d$. To enforce smoothness in the weights w, we introduce a regularizer based on the differences between adjacent weights. This results in the following optimization problem:

$$\min_w L(w) = \|y - Xw\|_2^2 + \lambda\|w\|_2^2 + \mu\|Dw\|_2^2,$$

where $\lambda, \mu > 0$ are regularization parameters and $\|Dw\|_2^2 = \sum_{i=2}^{d-1}(2w_i - w_{i-1} - w_{i+1})^2$.

(a) Find $D \in \mathbb{R}^{(d-2)\times d}$ such that $\|Dw\|_2^2 = \sum_{i=2}^{d-1}(2w_i - w_{i-1} - w_{i+1})^2$. Write D explicitly. Note that $\|\cdot\|_2$ denotes the usual ℓ_2, or Euclidean, norm.
we are tasked with finding a matrix D such that $\|Dw\|_2^2 = \sum_{i=2}^{d-1}(2w_i - w_{i-1} - w_{i+1})^2$.
if weight vector $w = [w_1, w_2, \ldots, w_d]^T$, then :

$$Dw = \begin{bmatrix} 2w_2 - w_1 - w_3 \\ 2w_3 - w_2 - w_4 \\ \vdots \\ 2w_{d-1} - w_{d-2} - w_d \end{bmatrix}$$

From the above, we can see that:

$$D = \begin{bmatrix} 0 & 2 & -1 & 0 & \ldots & 0 \\ 0 & -1 & 2 & -1 & \ldots & 0 \\ 0 & 0 & -1 & 2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & -1 & 2 & 1 \\ 0 & \ldots & 0 & 0 & -1 & 2 \end{bmatrix}_{(d-2)\times d}$$

In the first row, the first element is 0, the second element is 2, and the third element is -1. The rest of the elements are 0 because we only care about the relationship between adjancent weights w_i, w_{i-1}, w_{i+1}. The same applies to the rest of the rows.

(b) Derive the closed-form solution for w^*, the minimizer of $L(w)$.
this equation can be divided into three parts:
Residual sum of errors:

$$\text{RSS loss} = \|y - Xw\|_2^2 = (y - Xw)^\top(y - Xw) = y^\top y - 2w^\top X^\top y + w^\top X^\top Xw$$

N/B:scalars are symmetric, so $w^\top X^\top y = y^\top Xw$.
Let us find the derivative of the RSS loss w.r.t w:

$$\frac{\partial \text{RSS loss}}{\partial w} = -2X^\top y + 2X^\top Xw = 0$$

L2 regularization:

$$\text{L2 loss} = \lambda\|w\|_2^2 = \lambda w^\top w$$

to find the derivative of the L2 loss w.r.t w:

$$\frac{\partial \text{L2 loss}}{\partial w} = 2\lambda w$$

Smooth regularization:

$$\text{Smooth loss} = \mu\|Dw\|_2^2 = \mu w^\top D^\top Dw = \mu w^\top D^\top Dw$$

to find the derivative of the smooth loss w.r.t w:

$$\frac{\partial \text{Smooth loss}}{\partial w} = 2\mu D^\top Dw$$

Combining the three derivatives:

$$-2X^\top y + 2X^\top Xw + 2\lambda w + 2\mu D^\top Dw = 0$$
$$X^\top Xw + \lambda w + \mu D^\top Dw = X^\top y$$
$$(X^\top X + \lambda I + \mu D^\top D)w = X^\top y$$
$$w^* = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I} + \mu\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{X}^\top\mathbf{y}$$

(c) Denote the minimizer of problem (1) by w^*. Consider the same problem without the "smooth" regularization:

$$\min_w L_{\text{ridge}}(w) = \|y - Xw\|_2^2 + \lambda\|w\|_2^2.$$

Derive the closed-form solution for w^*_{ridge}, the minimizer of $L_{\text{ridge}}(w)$.
The loss function is:

$$L_{\text{ridge}}(w) = \|y - Xw\|_2^2 + \lambda\|w\|_2^2$$

The derivative of the loss function w.r.t w is:

$$\frac{\partial L_{\text{ridge}}(w)}{\partial w} = -2X^\top y + 2X^\top Xw + 2\lambda w = 0$$

$$X^\top Xw + \lambda w = X^\top y$$

$$(X^\top X + \lambda I)w = X^\top y$$

$$w^*_{\text{ridge}} = \mathbf{(X^\top X + \lambda I)^{-1} X^\top y}$$

Let us compare the two solutions:

$$w^* = \mathbf{(X^\top X + \lambda I + \mu D^\top D)^{-1} X^\top y}$$

$$w^*_{\text{ridge}} = \mathbf{(X^\top X + \lambda I)^{-1} X^\top y}$$

We need to prove that $w^* \leq w^*_{\text{ridge}}$ by showing that $(X^\top X + \lambda I + \mu D^\top D)^{-1} \preceq (X^\top X + \lambda I)^{-1}$.:

$$(X^\top X + \lambda I + \mu D^\top D)^{-1} - (X^\top X + \lambda I)^{-1} = (X^\top X + \lambda I)^{-1} \mu D^\top D (X^\top X + \lambda I)^{-1}$$

Since $D^\top D$ is positive semi-definite, then $(X^\top X + \lambda I + \mu D^\top D)^{-1} \preceq (X^\top X + \lambda I)^{-1}$.
this shows that:

$$w^* \leq w^*_{\text{ridge}}$$

(d) Briefly (in 1-3 sentences) explain how smooth regularization affects the bias and variance of the learned model w.

Smooth regularization increases the bias of the learned model by penalizing large differences between adjacent weights, leading to smoother weight vectors. However, it reduces the variance by preventing overfitting to the noise in the data. This is the trade-off between bias and variance in machine learning models, where increasing smoothness reduces variance but increases bias.

6 Online Update

1. In the standard linear regression model, we consider a model for which the observed response variable y is the prediction $x^\top w$ perturbed by noise, namely

$$y = w^\top x + \epsilon$$

where ϵ is a Gaussian random variable with mean 0 and variance σ^2 and $x, w \in \mathbb{R}^d$. Suppose our training dataset contains N observations of d-dimensional features. We denote the corresponding feature matrix by $X \in \mathbb{R}^{N\times d}$, and the labels associated with each observation are denoted by $y \in \mathbb{R}^N$. We have shown in the class that the MLE (maximum likelihood estimate) of w is given by

$$\hat{w} = (X^\top X)^{-1} X^\top y.$$

Suppose that we get a new observation $(\tilde{x}, \tilde{y})$ and would like to compute the updated MLE $\hat{w}_{\text{new}}$ after adding this observation to our dataset. Now $X_{\text{new}} = \begin{pmatrix} X \\ \tilde{x}^\top \end{pmatrix} \in \mathbb{R}^{(N+1)\times d}$ and $y_{\text{new}} = \begin{pmatrix} y \\ \tilde{y} \end{pmatrix} \in \mathbb{R}^{N+1}$. If we simply apply $\hat{w}_{\text{new}} = (X_{\text{new}}^\top X_{\text{new}})^{-1} X_{\text{new}}^\top y_{\text{new}}$, we will need to solve the normal equations for $\hat{w}_{\text{new}}$ again, which is inefficient. In this problem, we will derive a better algorithm for updating the parameters. The main idea is that the inverse of $X_{\text{new}}^\top X_{\text{new}}$, which is a low rank correction of $X^\top X$, can be computed by doing a low rank correction to the inverse of $X^\top X$.

Note that $X_{\text{new}}^\top X_{\text{new}} = X^\top X + \tilde{x}\tilde{x}^\top$. Using the Sherman–Morrison–Woodbury formula, we can show that

$$(X_{\text{new}}^\top X_{\text{new}})^{-1} = (X^\top X)^{-1} - \frac{(X^\top X)^{-1}\tilde{x}\tilde{x}^\top (X^\top X)^{-1}}{1 + \tilde{x}^\top (X^\top X)^{-1}\tilde{x}}.$$

Since $X_{\text{new}}^\top y_{\text{new}} = X^\top y + \tilde{x}\tilde{y}$, we have

$$\hat{w}_{\text{new}} = (X_{\text{new}}^\top X_{\text{new}})^{-1} X_{\text{new}}^\top y_{\text{new}} = \left((X^\top X)^{-1} - \frac{(X^\top X)^{-1}\tilde{x}\tilde{x}^\top (X^\top X)^{-1}}{1 + \tilde{x}^\top (X^\top X)^{-1}\tilde{x}} \right)(X^\top y + \tilde{x}\tilde{y}).$$

(a) Derive the formula for $\hat{w}_{\text{new}}$ using the Sherman–Morrison formula. Your formula should express $\hat{w}_{\text{new}}$ explicitly in terms of $\hat{w}$. The $\hat{w_{new}}$ can be expressed as:

$$\hat{w}_{\text{new}} = \left((X^\top X)^{-1} - \frac{(X^\top X)^{-1}\tilde{x}\tilde{x}^\top (X^\top X)^{-1}}{1 + \tilde{x}^\top (X^\top X)^{-1}\tilde{x}} \right)(X^\top y + \tilde{x}\tilde{y})$$

Expanding the equation:

$$\hat{w}_{\text{new}} = (X^\top X)^{-1}(X^\top y + \tilde{x}\tilde{y}) - \frac{(X^\top X)^{-1}\tilde{x}\tilde{x}^\top (X^\top X)^{-1}(X^\top y + \tilde{x}\tilde{y})}{1 + \tilde{x}^\top (X^\top X)^{-1}\tilde{x}}$$

$$\hat{w}_{\text{new}} = (X^\top X)^{-1}X^\top y + (X^\top X)^{-1}\tilde{x}\tilde{y} - \frac{(X^\top X)^{-1}\tilde{x}\tilde{x}^\top (X^\top X)^{-1}X^\top y + (X^\top X)^{-1}\tilde{x}\tilde{x}^\top (X^\top X)^{-1}\tilde{x}\tilde{y}}{1 + \tilde{x}^\top (X^\top X)^{-1}\tilde{x}}$$

$$\hat{w}_{\text{new}} = \hat{w} + (X^\top X)^{-1}\tilde{x}(\tilde{y} - \tilde{x}^\top \hat{w}) - \frac{(X^\top X)^{-1}\tilde{x}\tilde{x}^\top \hat{w} + (X^\top X)^{-1}\tilde{x}\tilde{x}^\top \tilde{x}(\tilde{y} - \tilde{x}^\top \hat{w})}{1 + \tilde{x}^\top (X^\top X)^{-1}\tilde{x}}$$

Algorithm 1 Online update

```
1: Receive observation $(\tilde{x}, \tilde{y})$.
2: Set $\xi \leftarrow K\tilde{x}/(1 + \tilde{x}^\top K\tilde{x})$.
3: Set $\alpha \leftarrow \alpha + \xi(\tilde{y} - \tilde{x}^\top \alpha)$.
4: Set $K \leftarrow K - \xi\tilde{x}^\top K$.
```

(b) Assume that n new data points arrive in a sequential order and we would like to update $\hat{w}$ every time a new data point arrives. Consider the following two methods:

i. First, initialize $\hat{w}$ using the original N data points. Then run Algorithm 1 once for each new data point.

ii. Directly compute $\hat{w}_{\text{new}} = (X_{\text{new}}^\top X_{\text{new}})^{-1} X_{\text{new}}^\top y_{\text{new}}$ every time a new data point arrives.

Compare the total computation complexity (in terms of number of multiplications) of methods (a) and (b) by showing the computational complexity of initialization plus the computational complexity of n updates. Your answers should be in terms of N, d, and n. Which method is more efficient?

Solution:

- Method (a):
  - Initialization: $\mathcal{O}(Nd^2 + d^3)$ initialization involves computing $(X^\top X)^{-1}$ and $X^\top y$. where $X \in \mathbb{R}^{N\times d}$ and $y \in \mathbb{R}^N$. so:
  $$X^\top X = \mathcal{O}(Nd^2), \quad X^\top y = \mathcal{O}(Nd)$$
  You also need to invert $X^\top X$ which is $\mathcal{O}(d^3)$.
  To multiply $(X^\top X)^{-1}$ by $X^\top y$ is $\mathcal{O}(Nd^2)$.
  - Each update: $\mathcal{O}(d^2 + d^3)$ for n-th update
  - Total for n updates: $\mathcal{O}(Nd^2 + d^3 + nd^2 + nd^3) = \mathcal{O}(Nd^2 + nd^2 + d^3 + nd^3)$
- Method (b):
  - Each update: $\mathcal{O}((N+k)d^2 + d^3)$ for k-th update
  - Total for n updates: $\mathcal{O}(\sum_{k=1}^{n}((N+k)d^2 + d^3)) = \mathcal{O}(nNd^2 + nd^3 + \frac{n(n+1)}{2}d^2) = \mathcal{O}(nNd^2 + nd^3 + n^2d^2)$

Method (a) is more efficient.