Notes on Linear Least Squares Model, COMP24112

Tingting Mu TINGTINGMU@MANCHESTER.AC.UK
Department of Computer Science
University of Manchester
Manchester M13 9PL, UK

Editor: NA

1. Notations

In a supervised learning task, we are given N training samples. Each training sample is characterised by a total of d features. We store the feature values of these training samples in an $N \times d$ matrix, denoted by

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix}, \tag{1}$$

where x_{ij} denotes the ij-th element of this matrix. Usually, we use the simplified notation $\mathbf{X} = [x_{ij}]$ to denote this matrix, and use the d-dimensional column vector $\boldsymbol{x}_i$ to denote feature vector of the i-th training sample such that

$$\boldsymbol{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}. \tag{2}$$

As you can see, $\boldsymbol{x}_i$ contains elements from the i-row of the feature matrix $\mathbf{X}$.

In the ***single-output case***, each training sample is associated with one target output. The following column vector

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{3}$$

is used to store the output of all the training samples. Each element y_i corresponds to the single-variable output of the i-th training sample. In a regression task, the target output is a real-valued number ($y_i \in R$). In a binary classification task, the target output is often set as a binary integer, e.g., $y_i \in \{-1, +1\}$ or $y_i \in \{0, 1\}$.

In the ***multi-output case***, each training sample is associated with c different output variables. We use the $N \times c$ matrix $\mathbf{Y} = [y_{ij}]$ to store the output variables of all the training

samples:

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & y_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Nc} \end{bmatrix}. \tag{4}$$

We use the c-dimensional column vector

$$\boldsymbol{y}_i = \begin{bmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{ic} \end{bmatrix} \tag{5}$$

to store the c output variables of the i-th training sample.

2. Linear Model

In machine learning, building a linear model refers to employing a ***linear function*** to estimate a desired output. The general formulation of a linear function that takes n input variables is

$$f(x_1, x_2, \ldots, x_n) = a_0 + a_1 x_1 + a_2 x_2 + \cdots a_n x_n, \tag{6}$$

where $a_0, a_1, a_2 \ldots a_n$ are often referred to as the linear combination coefficients (weights), or linear model weights.

2.1 Single-output Case

We use one linear function to estimate the single output variable of a given sample based on its input features $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]^T$. The estimated output is given by

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots w_d x_d = w_0 + \sum_{i=1}^{d} w_i x_i = \boldsymbol{w}^T \tilde{\boldsymbol{x}}, \tag{7}$$

where the column vector $\boldsymbol{w} = [w_0, w_1, w_2, \ldots, w_d]^T$ stores the model weights. The modified notation

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \tag{8}$$

is introduced to simplify the writing of the linear model formulation, and it is called the expanded feature vector.

2.2 Multi-output Case

In this case, each target output is estimated using one linear function. We seek c different functions to predict the c output for a sample $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]^T$:

$$\begin{aligned}
\hat{y}_1 &= w_{01} + w_{11}x_1 + w_{21}x_2 + \cdots w_{d1}x_d = \boldsymbol{w}_1^T \tilde{\boldsymbol{x}}, && (9)\\
\hat{y}_2 &= w_{02} + w_{12}x_1 + w_{22}x_2 + \cdots w_{d2}x_d = \boldsymbol{w}_2^T \tilde{\boldsymbol{x}}, && (10)\\
&\vdots \\
\hat{y}_c &= w_{0c} + w_{1c}x_1 + w_{2c}x_2 + \cdots w_{dc}x_d = \boldsymbol{w}_c^T \tilde{\boldsymbol{x}}, && (11)
\end{aligned}$$

where the vector

$$\boldsymbol{w}_i = \begin{bmatrix} w_{0i} \\ w_{1i} \\ w_{2i} \\ \vdots \\ w_{di} \end{bmatrix} \tag{12}$$

stores the linear model weights for predicting the i-th target output. By collecting all the estimated output in a vector, a neat expression of the multi-output linear model can be obtained:

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_c \end{bmatrix} = \begin{bmatrix} w_{01} + w_{11}x_1 + w_{21}x_2 + \cdots w_{d1}x_d \\ w_{02} + w_{12}x_1 + w_{22}x_2 + \cdots w_{d2}x_d \\ \vdots \\ w_{0c} + w_{1c}x_1 + w_{2c}x_2 + \cdots w_{dc}x_d \end{bmatrix} = \begin{bmatrix} w_{01} & w_{11} & w_{21} & \ldots & w_{d1} \\ w_{02} & w_{12} & w_{22} & \ldots & w_{d2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{0c} & w_{1c} & w_{2c} & \ldots & w_{dc} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \mathbf{W}^T \tilde{\boldsymbol{x}}, \tag{13}$$

where the $(d+1) \times c$ matrix

$$\mathbf{W} = \begin{bmatrix} w_{01} & w_{02} & \cdots & w_{0c} \\ w_{11} & w_{12} & \cdots & w_{1c} \\ w_{21} & w_{22} & \cdots & w_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ w_{d1} & w_{d2} & \cdots & w_{dc} \end{bmatrix}. \tag{14}$$

stores all the model weights.

3. Least Squares

Training a linear model refers to the process of finding the optimal values of the model weights, by utilising information provided by the training samples. The least squares approach refers to the method of finding the optimal model weights by minimising the sum-of-squares error loss.

3.1 Sum-of-squares Error

We use $\tilde{\boldsymbol{x}}_i = [1, x_{i1}, x_{i2}, \ldots x_{id}]^T$ do denote the expanded feature vector for the i-th training sample. The sum-of-squares error loss is computed as the sum of the squared differences between the true target outputs and their estimation:

- In the single-output case, the error loss computed using N training samples is given as

$$O(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{N}\left(\left(w_0 + \sum_{k=1}^{d} w_k x_{ik}\right) - y_i\right)^2 = \frac{1}{2}\sum_{i=1}^{N}\left(\boldsymbol{w}^T\tilde{\boldsymbol{x}}_i - y_i\right)^2 \tag{15}$$

- In the multi-output case, each sample is associated with multiple output variables (e.g., $y_{i1}, y_{i2}, \ldots, y_{ic}$ for the i-th training sample). The error loss is computed by examining the squared difference over each target output of each training sample, resulting in the following sum:

$$O(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{c}(\hat{y}_{ij} - y_{ij})^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{c}\left(\left(w_{0j} + \sum_{k=1}^{d} w_{kj} x_{ik}\right) - y_{ij}\right)^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{c}\left(\boldsymbol{w}_j^T\tilde{\boldsymbol{x}}_i - y_{ij}\right)^2. \tag{16}$$

3.2 Normal Equations

We use the following notation to denote an expanded feature matrix of N training samples:

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix}. \tag{17}$$

We re-express the error loss.

- Single-output case: The sum-of-squares error loss in Eq. (15) can be expressed as

$$O(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(\boldsymbol{w}^T\tilde{\boldsymbol{x}}_i - y_i\right)^2 = \frac{1}{2}\left\|\tilde{\mathbf{X}}\boldsymbol{w} - \boldsymbol{y}\right\|_2^2. \tag{18}$$

- Multi-output case: The sum-of-squares error loss in Eq. (16) can be expressed as

$$O(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{c}\left(\boldsymbol{w}_j^T\tilde{\boldsymbol{x}}_i - y_{ij}\right)^2 = \frac{1}{2}\sum_{i=1}^{N}\left\|\mathbf{W}^T\tilde{\boldsymbol{x}}_i - \boldsymbol{y}_i\right\|_2^2 = \frac{1}{2}\left\|\tilde{\mathbf{X}}\mathbf{W} - \mathbf{Y}\right\|_F^2. \tag{19}$$

You can check that the above equations hold using the definition of the l_2-norm and the Frobenius norm in Section 1.2 of the maths notes.

We know that the minimal error of zero is achieved when $\tilde{\mathbf{X}}\boldsymbol{w} = \boldsymbol{y}$ and $\tilde{\mathbf{X}}\mathbf{W} = \mathbf{Y}$. This turns the problem to solving linear equations. When $N > d$, these are overdetermined systems, it has a unique solution when $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ is invertible. When $N \leq d$, these are underdetermined systems. We explain how to compute the solutions below.

3.2.1 Case $N > d$

Assume you have more rows than columns in $\tilde{\mathbf{X}}$ ($N > d$) and $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ is invertible, your optimal weights of the linear model are computed by solving normal equations:

- For the single-output case, the normal equation for computing the optimal weight vector $\boldsymbol{w}^*$ is given as

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{w}^* = \tilde{\mathbf{X}}^T\boldsymbol{y}. \tag{20}$$

- For the multi-output case, the normal equation for computing the optimal weight matrix $\mathbf{W}^*$ possesses a similar form:

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{W}^* = \tilde{\mathbf{X}}^T\mathbf{Y}. \tag{21}$$

This gives the solution below:

- For the single-output case,

$$\boldsymbol{w}^* = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T\boldsymbol{y}. \tag{22}$$

- For the multi-output case,

$$\mathbf{W}^* = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{Y}. \tag{23}$$

3.2.2 Case $N \leq d$

When you have more columns than rows in $\tilde{\mathbf{X}}$ ($N \leq d$) and when $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ is invertible, your optimal weights are computed by

- For the single-output case,

$$\boldsymbol{w}^* = \tilde{\mathbf{X}}^T\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}\boldsymbol{y}. \tag{24}$$

- For the multi-output case,

$$\mathbf{W}^* = \tilde{\mathbf{X}}^T\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}\mathbf{Y}. \tag{25}$$

3.2.3 Pseudo-inverse Formula

The quantity $\left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T$ in the case of $N > d$ and $\tilde{\mathbf{X}}^T\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}$ in the case of $N \leq d$ is called the Moore-Penrose pseudo-inverse of the matrix $\tilde{\mathbf{X}}$. We use the notation $\tilde{\mathbf{X}}^\dagger$ to denote the pseudo inverse of $\tilde{\mathbf{X}}$. Regardless of the row size and column size, you can compute the optimal weights of the linear model simply by

- For the single-output case,

$$\boldsymbol{w}^* = \tilde{\mathbf{X}}^\dagger\boldsymbol{y}. \tag{26}$$

- For the multi-output case,

$$\mathbf{W}^* = \tilde{\mathbf{X}}^\dagger\mathbf{Y}. \tag{27}$$

To implement the formula, you can seek help from existing linear algebra libraries on computing pseudo-inverse of a given matrix, e.g., "numpy.linalg.pinv" in Python.

4. Regularised Least Squares model

The regularised least squares model finds its model weights by minimising the following modified error loss:

$$O(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2 + \frac{1}{2}\lambda\left(w_0^2 + \sum_{i=1}^{d} w_i^2\right) \tag{28}$$

for the single-output case, and

$$O(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{c}\left(\hat{y}_{ij} - y_{ij}\right)^2 + \frac{1}{2}\lambda\sum_{j=1}^{c}\left(w_{0j}^2 + \sum_{i=1}^{d} w_{ij}^2\right) \tag{29}$$

for the multi-output case. Here, $\lambda > 0$ is the regularisation parameter. The normal equations for the regularised least squares model are given as

$$\boldsymbol{w}^* = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I}\right)^{-1}\tilde{\mathbf{X}}^T\boldsymbol{y} \text{ (single-output)}, \tag{30}$$

$$\mathbf{W}^* = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{Y} \text{ (multi-output)}. \tag{31}$$

These are derived by setting the gradient of $O(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ to zero, and by setting the gradient of $O(\mathbf{W})$ with respect to $\mathbf{W}$ to zero.

5. How to derive results in Section 3? (Optional Reading)

This section is optional reading materials for students who are interested.

5.1 Least Squares with $N > d$

Assuming $N > d$ and $\left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}$ is invertible, the overdetermined system has a unique solution. The normal equations are derived by setting the gradient of the error loss as zero.

5.1.1 Single-output Case

Expand the error loss Eq. (18) as below:

$$\begin{aligned}
O(\boldsymbol{w}) &= \frac{1}{2}\left\|\tilde{\mathbf{X}}\boldsymbol{w} - \boldsymbol{y}\right\|_2^2 \\
&= \frac{1}{2}\left(\tilde{\mathbf{X}}\boldsymbol{w} - \boldsymbol{y}\right)^T\left(\tilde{\mathbf{X}}\boldsymbol{w} - \boldsymbol{y}\right) \\
&= \frac{1}{2}\left(\boldsymbol{w}^T\tilde{\mathbf{X}}^T - \boldsymbol{y}^T\right)\left(\tilde{\mathbf{X}}\boldsymbol{w} - \boldsymbol{y}\right) \\
&= \frac{1}{2}\left(\boldsymbol{w}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{w} - \boldsymbol{y}^T\tilde{\mathbf{X}}\boldsymbol{w} - \boldsymbol{w}^T\tilde{\mathbf{X}}^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y}\right) \\
&= \frac{1}{2}\left(\boldsymbol{w}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{w} - 2\boldsymbol{y}^T\tilde{\mathbf{X}}\boldsymbol{w} + \boldsymbol{y}^T\boldsymbol{y}\right).
\end{aligned} \tag{32}$$

It contains three terms: $\boldsymbol{w}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{w}$ is a quadratic function of $\boldsymbol{w}$, $2\boldsymbol{y}^T\tilde{\mathbf{X}}\boldsymbol{w}$ is a linear function of $\boldsymbol{w}$, and $\boldsymbol{y}^T\boldsymbol{y}$ is a constant term. Utilising the gradient formulations for linear and

quadratic functions (see Section 3 in the maths notes), it is straightforward to derive the gradient of $O(\boldsymbol{w})$ with respect to $\boldsymbol{w}$:

$$\nabla_{\boldsymbol{w}} O(\boldsymbol{w}) = \frac{1}{2}\left(\left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^T \boldsymbol{w} + \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{w} - \left(2\boldsymbol{y}^T\tilde{\mathbf{X}}\right)^T\right) = \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{w} - \tilde{\mathbf{X}}^T\boldsymbol{y}. \tag{33}$$

When the minimum of $O(\boldsymbol{w})$ is reached, its gradient has to be equal to zero: $\nabla_{\boldsymbol{w}} O(\boldsymbol{w}) = 0$. Therefore

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{w}^* = \tilde{\mathbf{X}}^T\boldsymbol{y}, \tag{34}$$

which gives $\boldsymbol{w}^* = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T\boldsymbol{y}$.

5.1.2 Multi-output Case

Expand the error loss Eq. (19) as below:

$$\begin{aligned}
O(\mathbf{W}) &= \frac{1}{2}\left\|\tilde{\mathbf{X}}\mathbf{W} - \mathbf{Y}\right\|_F^2 \\
&= \frac{1}{2}\mathrm{tr}\left[\left(\tilde{\mathbf{X}}\mathbf{W} - \mathbf{Y}\right)^T\left(\tilde{\mathbf{X}}\mathbf{W} - \mathbf{Y}\right)\right] \\
&= \frac{1}{2}\mathrm{tr}\left[\left(\mathbf{W}^T\tilde{\mathbf{X}}^T - \mathbf{Y}^T\right)\left(\tilde{\mathbf{X}}\mathbf{W} - \mathbf{Y}\right)\right] \\
&= \frac{1}{2}\mathrm{tr}\left(\mathbf{W}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{W} - \mathbf{Y}^T\tilde{\mathbf{X}}\mathbf{W} - \mathbf{W}^T\tilde{\mathbf{X}}^T\mathbf{Y} + \mathbf{Y}^T\mathbf{Y}\right) \\
&= \frac{1}{2}\left(\mathrm{tr}\left(\mathbf{W}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{W}\right) - \mathrm{tr}\left(\mathbf{Y}^T\tilde{\mathbf{X}}\mathbf{W}\right) - \mathrm{tr}\left(\mathbf{W}^T\tilde{\mathbf{X}}^T\mathbf{Y}\right) + \mathrm{tr}\left(\mathbf{Y}^T\mathbf{Y}\right)\right).
\end{aligned} \tag{35}$$

Based on the trace property as shown in Eq. (15) of the maths notes, it has

$$\mathrm{tr}\left(\mathbf{W}^T\tilde{\mathbf{X}}^T\mathbf{Y}\right) = \mathrm{tr}\left[\left(\mathbf{W}^T\tilde{\mathbf{X}}^T\mathbf{Y}\right)^T\right] = \mathrm{tr}\left(\mathbf{Y}^T\tilde{\mathbf{X}}\mathbf{W}\right). \tag{36}$$

Therefore,

$$O(\mathbf{W}) = \frac{1}{2}\left(\mathrm{tr}\left(\mathbf{W}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{W}\right) - 2\mathrm{tr}\left(\mathbf{W}^T\tilde{\mathbf{X}}^T\mathbf{Y}\right) + \mathrm{tr}\left(\mathbf{Y}^T\mathbf{Y}\right)\right). \tag{37}$$

We can use the following readily given trace derivative rules to compute the gradient:

$$\frac{\partial \mathrm{tr}\left(\mathbf{Z}^T\mathbf{A}\right)}{\partial \mathbf{Z}} = \mathbf{A}, \tag{38}$$

$$\frac{\partial \mathrm{tr}\left(\mathbf{Z}^T\mathbf{B}\mathbf{Z}\right)}{\partial \mathbf{Z}} = \mathbf{B}\mathbf{Z} + \mathbf{B}^T\mathbf{Z}. \tag{39}$$

In our case, we have $\mathbf{Z} \leftarrow \mathbf{W}$. We also have $\mathbf{B} \leftarrow \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ for the first term in $O(\mathbf{W})$, and $\mathbf{A} \leftarrow \tilde{\mathbf{X}}^T\mathbf{Y}$ for the second term in $O(\mathbf{W})$. Therefore the gradient of $O(\mathbf{W})$ with respect to $\mathbf{W}$ is given by

$$\nabla_{\mathbf{W}} O(\mathbf{W}) = \frac{1}{2}\left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{W} + \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^T\mathbf{W} - 2\tilde{\mathbf{X}}^T\mathbf{Y}\right) = \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{W} - \tilde{\mathbf{X}}^T\mathbf{Y}. \tag{40}$$

Setting the gradient to zero $\nabla_{\mathbf{W}} O(\mathbf{W}) = 0$, we have

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{W}^* = \tilde{\mathbf{X}}^T\mathbf{Y}, \tag{41}$$

which gives $\mathbf{W}^* = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{Y}$.

5.2 Least Squares with $N \leq d$

When $N \leq d$ and $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}$ is invertible, the underdetermined system has infinitely many solutions. We attempt to find that single solution that has the smallest l_2 norm for $\boldsymbol{w}$ and the smallest Frobenius norm for $\mathbf{W}$, which results in the following constrained optimisation:

- Single-output case:

$$\begin{aligned} \min_{\boldsymbol{w} \in R^{d+1}} \ & \|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \ & \tilde{\mathbf{X}}\boldsymbol{w} = \boldsymbol{y}. \end{aligned} \tag{42}$$

- Multi-output case:

$$\begin{aligned} \min_{\mathbf{W} \in R^{(d+1)\times c}} \ & \|\mathbf{W}\|_F^2 \\ \text{s.t.} \ & \tilde{\mathbf{X}}\mathbf{W} = \mathbf{Y} \end{aligned} \tag{43}$$

Assuming $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ is invertible, their solutions are the ones in Eqs. (24) and (25).

5.3 Regularised Least Squares model

The modified error loss of the regularised least squares model in Eq. (28) can be re-written in the matrix form as below:

$$O(\boldsymbol{w}) = \frac{1}{2}\left\|\tilde{\mathbf{X}}\boldsymbol{w} - \boldsymbol{y}\right\|_2^2 + \frac{1}{2}\lambda\|\boldsymbol{w}\|_2^2. \tag{44}$$

For the multi-output case, the modified error loss in Eq. (29) can be re-written as

$$O(\mathbf{W}) = \frac{1}{2}\left\|\tilde{\mathbf{X}}\mathbf{W} - \mathbf{Y}\right\|_F^2 + \frac{1}{2}\lambda\left\|\mathbf{W}\right\|_F^2. \tag{45}$$

Based on these, the gradient of $O(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ and the gradient of $O(\mathbf{W})$ with respect to $\mathbf{W}$ can be derived by following a similar procedure as explained in Section 5.1. You can give it a go as a practice.