Machine Learning

SEU — 2024 Spring

Assignment 3

58122204 Xie xing *Instructor: Xu-Ying Liu*

§ Support Vector Machine & Bayesian Classifiers §

Problem 1: Support Vector Machine

(1)

1. (a) Generalized Lagrangian function

$$L(\omega, b, \epsilon, \alpha, \mu) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N}\epsilon_i, - \sum_{i=1}^{N}\alpha_i(y_i(\omega^T x_i + b) - 1 + \epsilon) - \sum_{i=1}^{N}\mu_i\epsilon_i \tag{1.1}$$

, where $\alpha_i >= 0, \mu_i >= 0$. The dual problem of the original function is

$$\max_{\alpha\geq 0, \mu\geq 0} \min_{\omega,b,\epsilon} L(\omega, b, \epsilon, \alpha, \mu) \tag{1.2}$$

Find the partial derivative :

$$\nabla_w L(w, b, \epsilon, \alpha, \mu) = w - \sum_{i=1}^{N}\alpha_i y_i x_i = 0 \tag{1.3}$$

$$\nabla_b L(w, b, \epsilon, \alpha, \mu) = -\sum_{i=1}^{N}\alpha_i y_i = 0 \tag{1.4}$$

$$\nabla_{\epsilon_i} L(w, b, \epsilon, \alpha, \mu) = C - \alpha_i - \mu_i = 0 \tag{1.5}$$

Solutions have to:

$$\begin{cases} w = \sum_{i=1}^{N}\alpha_i y_i x_i \\ \sum_{i=1}^{N}\alpha_i y_i = 0 \\ C - \alpha_i - \mu_i = 0 \end{cases} \tag{1.6}$$

Bringing in $L(w, b, \epsilon, \alpha, \mu)$ gets:

$$\min_{\omega,b,\epsilon} L(w, b, \epsilon, \alpha, \mu) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\,(x_i \cdot x_j) + \sum_{i=1}^{N}\alpha_i \tag{1.7}$$

calcuate the maximum of $\min_{\omega,b,\epsilon} L(w, b, \epsilon, \alpha, \mu)$ on α, we get the dual problem:

$$\max_{\alpha} -\frac{1}{2}\sum_{i}^{N}\sum_{j}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{N}\alpha_i \tag{1.8}$$

$$s.t. \sum_{i}^{N}\alpha_i y_i = 0 \tag{1.9}$$

$$C - \alpha_i - \mu_i = 0 \tag{1.10}$$

$$\alpha_i \geq 0 \tag{1.11}$$

This assignment is due May 31, 2024 and the date of submission is May 31, 2024.

$$\mu_i \geq 0, i = 1, 2, ..., N \tag{1.12}$$

First of all, there is a partial derivative solution $\sum_{i=1}^{N} \alpha_i y_i = 0$,;
Secondly, the Lagrange multiplier is greater than or equal to 0, that is $\alpha, \mu >= 0$, when seeking partial derivatives, we get $C - \alpha_i - \mu_i = 0$;
Finally, comprehensively we get $0 \leq \alpha_i \leq C$.

So the dual problem is:

$$\max_{\alpha} -\frac{1}{2} \sum_{i}^{N} \sum_{j}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{N} \alpha_i \tag{1.13}$$

$$s.t. \sum_{i}^{N} \alpha_i y_i = 0 \tag{1.14}$$

$$0 \leq \alpha_i \leq C \tag{1.15}$$

2. (b) At the saddle point, the solutions of the primal and dual problems are the same, satisfying the following KKT conditions:

1. Primal feasibility condition:
$$y_i(w \cdot x_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad \forall i \tag{1.16}$$
2. Dual feasibility condition:
$$0 \leq \alpha_i \leq C, \quad \forall i \tag{1.17}$$
3. Complementary slackness condition:
$$\alpha_i [y_i(w \cdot x_i + b) - 1 + \epsilon_i] = 0, \quad \forall i \tag{1.18}$$
4. Slack variable condition:
$$\mu_i \epsilon_i = 0, \quad \forall i \tag{1.19}$$
5. Gradient condition:
$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{1.20}$$

So the primal problem and the dual problem are strongly dual, thus the dual problem is equivalent to the primal problem when at its saddle point.

(2)

1. (a) The corresponding mapping function is:

$$\phi(x) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2) \tag{1.21}$$

2. (b) we use the code as follows:

```
import numpy as np
import matplotlib.pyplot as plt

# Create XOR data
np.random.seed(0)
X_xor = np.random.randn(40, 2)
y_xor = np.logical_xor(X_xor[:, 0] > 0, X_xor[:, 1] > 0)
y_xor = np.where(y_xor, 1, -1)

# Plot original XOR data
plt.scatter(X_xor[y_xor == 1, 0], X_xor[y_xor == 1, 1], color='g', marker='x',
 ↪ label='1')
```

```
plt.scatter(X_xor[y_xor == -1, 0], X_xor[y_xor == -1, 1], color='b', marker='o',
 ↪ label='-1')
plt.legend()  # Display legend
plt.title('Original XOR Data')
plt.show()

# Mapping function (x)
def phi(x):
    return np.array([x[0]**2, np.sqrt(2)*x[0]*x[1], x[1]**2])

# Apply mapping to the data
X_mapped = np.array([phi(x) for x in X_xor])

# Plot mapped data
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_mapped[y_xor == 1, 0], X_mapped[y_xor == 1, 1], X_mapped[y_xor == 1,
 ↪ 2], color='g', marker='x', label='1')
ax.scatter(X_mapped[y_xor == -1, 0], X_mapped[y_xor == -1, 1], X_mapped[y_xor ==
 ↪ -1, 2], color='b', marker='o', label='-1')
ax.set_xlabel('x1^2')
ax.set_ylabel('sqrt(2)x1x2')
ax.set_zlabel('x2^2')
plt.legend()  # Display legend
plt.title('Mapped XOR Data')
plt.show()
```

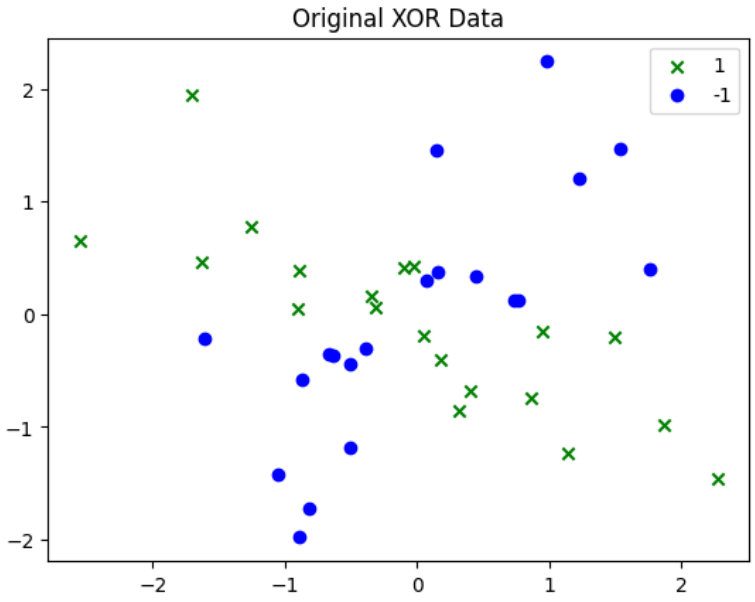

Figure 1: Original XOR Data

Figure 1 and figure 2 shows the data points in the transformed 3D space using the mapping function $\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$. In this space, the XOR data points should be linearly separable.

3. (c) The decision boundary in the figure (original feature space) is shown in Figure 3.
The decision boundary in the figure (Regenerative Kernel Hilbert Space) is shown in Figure 4.

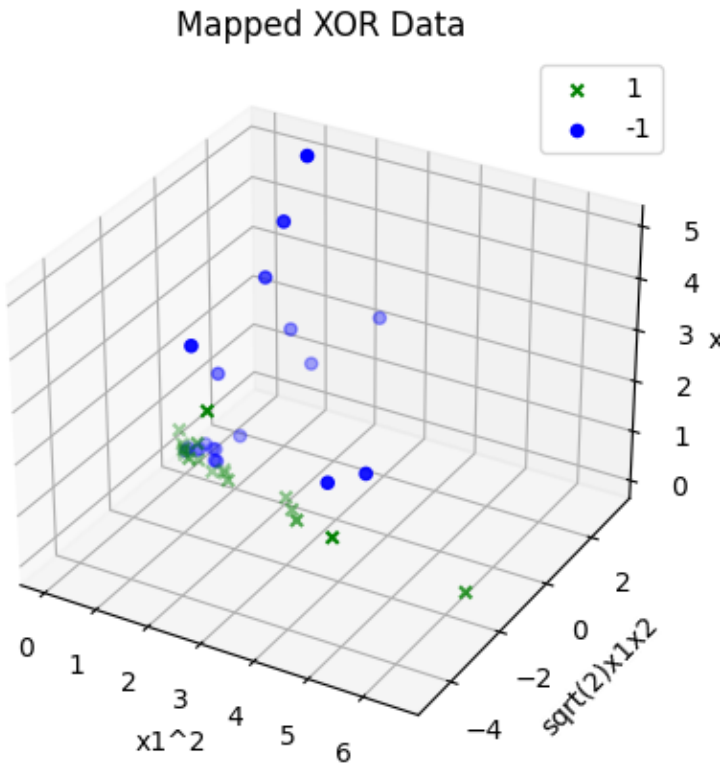

Figure 2: Mapped XOR Data

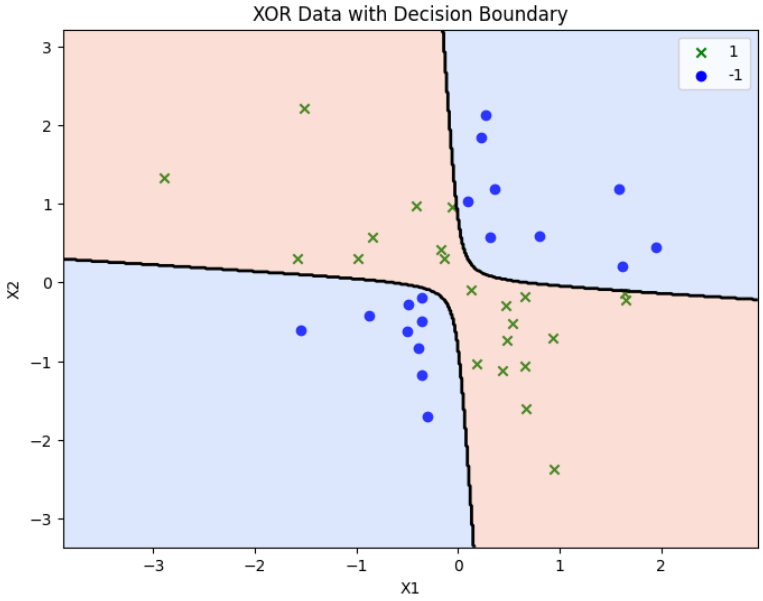

Figure 3: XOR Data with Decision Boundary

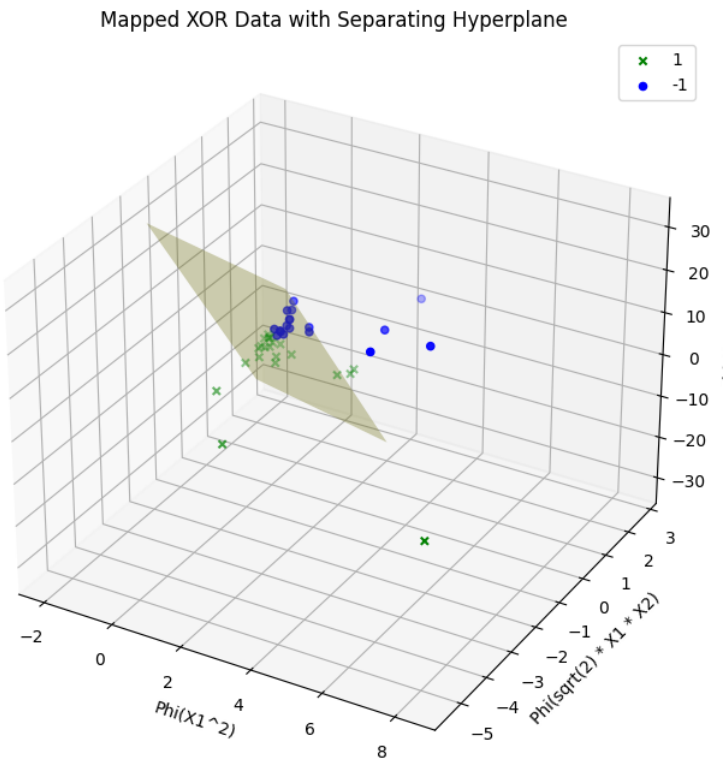

Figure 4: Mapped XOR Data with Separating Hyperplane

Problem 2: Naïve Bayes Classifier

(1)

$$P(w = A|y = +1) = \frac{10+1}{46+4} = \frac{11}{50} \tag{2.1}$$

$$P(w = B|y = +1) = \frac{5+1}{46+4} = \frac{6}{50} \tag{2.2}$$

$$P(w = C|y = +1) = \frac{18+1}{46+4} = \frac{19}{50} \tag{2.3}$$

$$P(w = D|y = +1) = \frac{13+1}{46+4} = \frac{14}{50} \tag{2.4}$$

$$P(w = A|y = -1) = \frac{7+1}{46+4} = \frac{8}{50} \tag{2.5}$$

$$P(w = B|y = -1) = \frac{11+1}{46+4} = \frac{12}{50} \tag{2.6}$$

$$P(w = C|y = -1) = \frac{9+1}{46+4} = \frac{10}{50} \tag{2.7}$$

$$P(w = D|y = -1) = \frac{19+1}{46+4} = \frac{20}{50} \tag{2.8}$$

(2) Given $A = 3, B = 2, C = 1, D = 2$:
Calculate the posterior probability for each class y using the Naïve Bayes formula.

$$P(y = +1) = P(y = -1) = \frac{1}{2} \tag{2.9}$$

For $y = +1$:

$$P(\mathbf{x}|y=+1) = P(A=3|y=+1)P(B=2|y=+1)P(C=1|y=+1)P(D=2|y=+1) \tag{2.10}$$

$$= \left(\frac{11}{50}\right)^3 \left(\frac{6}{50}\right)^2 \left(\frac{19}{50}\right)^1 \left(\frac{14}{50}\right)^2 \tag{2.11}$$

For $y = -1$:

$$P(\mathbf{x}|y=-1) = P(A=3|y=-1)P(B=2|y=-1)P(C=1|y=-1)P(D=2|y=-1) \tag{2.12}$$

$$= \left(\frac{8}{50}\right)^3 \left(\frac{12}{50}\right)^2 \left(\frac{10}{50}\right)^1 \left(\frac{20}{50}\right)^2 \tag{2.13}$$

The Naïve Bayes decision rule is to choose the class with the higher posterior probability. Thus:

$$\hat{y} = \arg\max_y \left(P(y) \cdot P(\mathbf{x}|y)\right) \tag{2.14}$$

For $y = +1$:

$$P(\mathbf{x}|y=+1)P(y=+1) = \frac{1}{2} \cdot \left(\frac{11}{50}\right)^3 \left(\frac{6}{50}\right)^2 \left(\frac{19}{50}\right)^1 \left(\frac{14}{50}\right)^2 \tag{2.15}$$

For $y = -1$:

$$P(\mathbf{x}|y=-1)P(y=-1) = \frac{1}{2} \cdot \left(\frac{8}{50}\right)^3 \left(\frac{12}{50}\right)^2 \left(\frac{10}{50}\right)^1 \left(\frac{20}{50}\right)^2 \tag{2.16}$$

Compare the two quantities:

$$\left(\frac{11}{50}\right)^3 \left(\frac{6}{50}\right)^2 \left(\frac{19}{50}\right)^1 \left(\frac{14}{50}\right)^2 < \left(\frac{8}{50}\right)^3 \left(\frac{12}{50}\right)^2 \left(\frac{10}{50}\right)^1 \left(\frac{20}{50}\right)^2 \tag{2.17}$$

So $\hat{y} = -1$, namely the label of the new sample where $A = 3, B = 2, C = 1, D = 2$ is predicted to be -1.

Problem 3: Gaussian Bayesian Classifier

(1) The Bayes optimal classifier that minimizes the misclassification error rate classifies a sample $\mathbf{x}$ to the class y with the highest posterior probability $P(y = k|\mathbf{x})$. This classifier is defined as:

$$\hat{y}(\mathbf{x}) = \arg\max_{k \in \{1,2,\ldots,K\}} P(y=k|\mathbf{x}) \tag{3.1}$$

Using Bayes' theorem, this can be rewritten as:

$$\hat{y}(\mathbf{x}) = \arg\max_{k \in \{1,2,\ldots,K\}} \frac{P(\mathbf{x}|y=k)P(y=k)}{P(\mathbf{x})} \tag{3.2}$$

Since $P(\mathbf{x})$ is the same for all classes, it simplifies to:

$$\hat{y}(\mathbf{x}) = \arg\max_{k \in \{1,2,\ldots,K\}} P(\mathbf{x}|y=k)P(y=k) \tag{3.3}$$

(2) Given that the samples in the k-th class are i.i.d. and follow a normal distribution $\mathcal{N}(\mu_k, \Sigma)$, the probability density function is:

$$p(\mathbf{x}|y=k) = \frac{1}{(2\pi)^{d/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^\top \Sigma^{-1}(\mathbf{x}-\mu_k)\right) \tag{3.4}$$

The prior probability for class k is $\pi_k = P(y=k)$. The Bayes optimal classifier is:

$$\hat{y}(\mathbf{x}) = \arg\max_{k\in\{1,2,\ldots,K\}} p(\mathbf{x}|y=k)\pi_k \tag{3.5}$$

Substituting the probability density function:

$$\hat{y}(\mathbf{x}) = \arg\max_{k\in\{1,2,\ldots,K\}} \frac{1}{(2\pi)^{d/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^\top \Sigma^{-1}(\mathbf{x}-\mu_k)\right)\pi_k \tag{3.6}$$

Simplifying, we get:

$$\hat{y}(\mathbf{x}) = \arg\max_{k\in\{1,2,\ldots,K\}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^\top \Sigma^{-1}(\mathbf{x}-\mu_k)\right)\pi_k \tag{3.7}$$

Since the exponential function is monotonically increasing, this can be further simplified to:

$$\hat{y}(\mathbf{x}) = \arg\max_{k\in\{1,2,\ldots,K\}} \left(-\frac{1}{2}(\mathbf{x}-\mu_k)^\top \Sigma^{-1}(\mathbf{x}-\mu_k) + \log\pi_k\right) \tag{3.8}$$

(3) For the binary classification problem, assume the samples in each class are i.i.d. and follow normal distributions $\mathcal{N}(\mu_0, \Sigma)$ and $\mathcal{N}(\mu_1, \Sigma)$ with equal prior probabilities $\pi_0 = \pi_1$.

The decision rule for LDA is to assign $\mathbf{x}$ to class 0 if the following discriminant function is greater than zero:

$$(\mathbf{x} - \frac{\mu_0+\mu_1}{2})^\top \Sigma^{-1}(\mu_1-\mu_0) > 0 \tag{3.9}$$

This can be derived as follows:

Given the discriminant functions for the two classes, we assign $\mathbf{x}$ to class 0 if:

$$p(\mathbf{x}|y=0)\pi_0 > p(\mathbf{x}|y=1)\pi_1 \tag{3.10}$$

Since $\pi_0 = \pi_1$, it simplifies to:

$$p(\mathbf{x}|y=0) > p(\mathbf{x}|y=1) \tag{3.11}$$

Taking the logarithm of both sides:

$$\log p(\mathbf{x}|y=0) > \log p(\mathbf{x}|y=1) \tag{3.12}$$

Substituting the normal distribution PDFs:

$$-\frac{1}{2}(\mathbf{x}-\mu_0)^\top \Sigma^{-1}(\mathbf{x}-\mu_0) > -\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1) \tag{3.13}$$

Simplifying this inequality, we get:

$$(\mathbf{x}-\mu_0)^\top \Sigma^{-1}(\mathbf{x}-\mu_0) < (\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1) \tag{3.14}$$

This can be rewritten as:

$$2\mathbf{x}^\top \Sigma^{-1}(\mu_1-\mu_0) > \mu_1^\top \Sigma^{-1}\mu_1 - \mu_0^\top \Sigma^{-1}\mu_0 \tag{3.15}$$

Finally, the decision rule for LDA is:

$$\mathbf{x}^\top \Sigma^{-1}(\mu_1-\mu_0) > \frac{1}{2}(\mu_1^\top \Sigma^{-1}\mu_1 - \mu_0^\top \Sigma^{-1}\mu_0) \tag{3.16}$$

Given $w = \Sigma^{-1}(\mu_1-\mu_0)$:

$$\mathbf{x}^\top w > \frac{1}{2}(\mu_1^\top \Sigma^{-1}\mu_1 - \mu_0^\top \Sigma^{-1}\mu_0) \tag{3.17}$$

This shows that the LDA decision rule corresponds to the Bayes optimal classifier under the given assumptions.

Problem 4: MLE and Linear Regression

(1) Given that $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, the probability density function of y_i given X_i is:

$$p(y_i|X_i, f) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_i - f(X_i))^2}{2\sigma_i^2}\right) \tag{4.1}$$

The likelihood of all observations y given X and f is the product of the individual densities:

$$L(y|X, f) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_i - f(X_i))^2}{2\sigma_i^2}\right) \tag{4.2}$$

The log-likelihood function is:

$$\log L(y|X, f) = \sum_{i=1}^{n} \left(-\frac{1}{2}\log(2\pi\sigma_i^2) - \frac{(y_i - f(X_i))^2}{2\sigma_i^2}\right) \tag{4.3}$$

Maximizing the log-likelihood is equivalent to minimizing the negative log-likelihood:

$$\text{Minimize: } \sum_{i=1}^{n} \left(\frac{(y_i - f(X_i))^2}{2\sigma_i^2}\right) \tag{4.4}$$

(2) Assuming a linear relationship $f(X_i) = w \cdot X_i$, where w is a p-dimensional vector of weights, we substitute this into our objective function:

$$\text{Minimize: } \sum_{i=1}^{n} \frac{(y_i - w \cdot X_i)^2}{2\sigma_i^2} \tag{4.5}$$

To express this in matrix notation, let's define:

1. y as the n-dimensional vector of labels.
2. X as the $n \times p$ design matrix.
3. w as the p-dimensional vector of weights.
4. Σ as a diagonal matrix with σ_i^2 on the diagonal.

The cost function can be written as:

$$J(w) = \frac{1}{2}(y - Xw)^T \Sigma^{-1} (y - Xw) \tag{4.6}$$

Here, Σ is the diagonal matrix of noise variances σ_i^2.

(3) To find the optimal w, we take the gradient of $J(w)$ with respect to w and set it to zero:

$$\nabla_w J(w) = -X^T \Sigma^{-1} (y - Xw) \tag{4.7}$$

Setting the gradient to zero gives:

$$X^T \Sigma^{-1} (y - Xw) = 0 \tag{4.8}$$

Solving for w:

$$X^T \Sigma^{-1} X w = X^T \Sigma^{-1} y \tag{4.9}$$

Thus, the solution to the optimization problem is:

$$w = (X^T \Sigma^{-1} X)^{-1} X^T \Sigma^{-1} y \tag{4.10}$$