Homework 3

Yixuan Li
CSE 151A: Machine Learning and Algorithms

May 14, 2025

1 Conceptual and Mathematical Problems

1. unit vector $u = \begin{bmatrix} \frac{1}{\sqrt{14}} \\ \frac{2}{\sqrt{14}} \\ \frac{3}{\sqrt{14}} \end{bmatrix}$

2. $u_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$, $u_2 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

3. $w = \begin{bmatrix} 2 & -1 & 6 \end{bmatrix}$

4. A: 10×30. B: 30×20.

5. (a) $X : n \times d$
   (b) : $n \times n$
   (c) (i, j) entry of XX^T: $\sum_{a=1}^{d} x_a^{(i)} x_a^{(j)}$

6. $x^T x = 35$. $xx^T = \begin{bmatrix} 1 & 3 & 5 \\ 3 & 9 & 15 \\ 5 & 15 & 25 \end{bmatrix}$

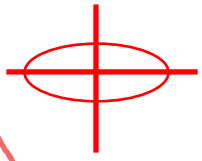
7. $M = \begin{bmatrix} 3 & 1 & -2 \\ 1 & 0 & 0 \\ -2 & 0 & 6 \end{bmatrix}$

8. (a) $|A| = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 = 40320$
   (b) $A^{-1} = diag(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8})$

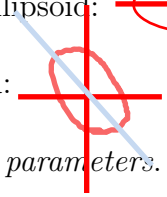
9. (a) $UU^T = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}_{d \times d} = I_{d \times d}$
   (b) $U^{-1} = U^T$

10. Since A is singular, $|A| = z - 6 = 0$, $z = 6$.

11. *Gaussian contours.*
    (a) The shape is an axis-aligned ellipsoid:

    (b)The shape is a rotated ellipsoid:

12. *Qualitative appraisal of Gaussian parameters.*
    (a) $q < 0,\ p > 0,\ r > 0$.
    (b) $q = 0,\ p \geq 0,\ r \geq 0$.
    (c) $q^2 = pr,\ p > 0,\ r > 0$.
    (d) $q = 0,\ r = 0,\ p \geq 0$.

13. (a) Linear.
    (b) Axis-aligned Ellipsoid.
    (c) Spherical.

2 Programming Problems

14.

(a) I used the data in the `x_validation` and `y_validation` data set to test the accuracy of prediction under different c. They are split from the `train_data` and `train_labels` both with the data size of 10000. And the multivariate distribution model (with the parameter `pi`, `mu`, `cov`) is fitted from the `x_train` data. The code of selecting the best value of c is as follows:

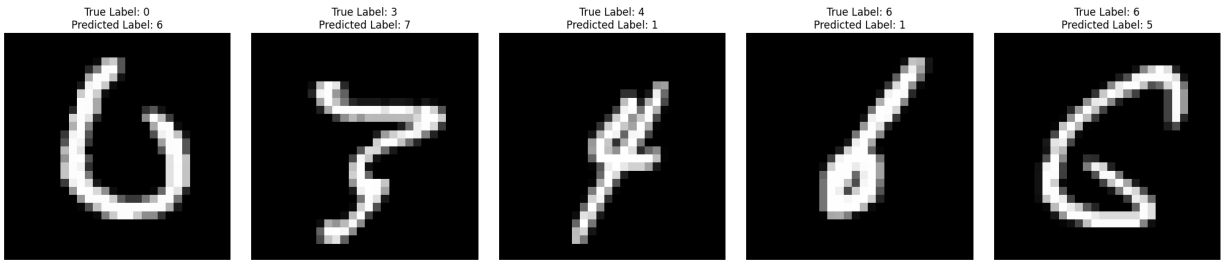
```
    candidate_c = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 0.1, 1, 5, 10, 20, 50, 80
                                        , 100, 500, 1000, 5000, 10000,
                                        100000]
    best_c = None
    best_acc = -1

    # Test all the candidate c
    for c in candidate_c:
        mu_tmp = np.zeros((k, d))
        sigma_tmp = np.zeros((k, d, d))
        pi_tmp = np.zeros(k)

        # For each class, calculate prior distribution on x_train
        for j in range(k):
            xj = x_train[y_train == j]
            pi_tmp[j] = len(xj) / len(x_train)
            mu_tmp[j] = np.mean(xj, axis=0)
            centered = xj - mu_tmp[j]
            cov = centered.T @ centered / len(xj)
            cov += c * np.eye(d)
            sigma_tmp[j] = cov
```

```
        # Fit the model and get the multivariate distribution
        score = np.zeros((len(y_validation), k))

        for label in range(k):
            try:
                rv = multivariate_normal(mean=mu_tmp[label], cov=sigma_tmp
                                                        [label],
                                                        allow_singular=True)
            except np.linalg.LinAlgError:
                print(f"Warning: Class {label} covariance matrix is
                                                        singular at c = {c}")
                continue
            # Obtain the prob distribution of each label in y_validatuon
            for i in range(len(y_validation)):
                score[i, label] = np.log(pi_tmp[label]) + rv.logpdf(
                                                        x_validation[i, :])

        # Use validation data set to test the accuracy of different c
        predictions = np.argmax(score, axis=1)
        acc = accuracy_score(y_validation, predictions)
        print(f"Validation accuracy for c={c:.4g}: {acc:.4f}")

        if acc > best_acc:
            best_acc = acc
            best_c = c
            print(f"New best c found: {best_c}")
```

(b) The best value of c is 5000, which is chosen from the `candidate_c` = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 0.1, 1, 5, 10, 20, 50, 80, 100, 500, 1000, 5000, 10000, 100000]. It has the accuracy of 94.9% on the validation data set.

(c) The error rate on my MNIST test set is 4.34%.

(d) The following five misclassified test digits examples (randomly selected) with their true label and predicted label are listed below.



(a) True:0, Pred:6 (b) True:3, Pred:7 (c) True:4, Pred:1 (d) True:6, Pred:1 (e) True:6, Pred:5

Figure 1: Problem 14 (d) Five misclassified test digits examples