1. (**30 points**) Consider the following 2 sets of points in the plane:

$$A : \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad B : \begin{pmatrix} -1 \\ 2 \end{pmatrix}, \begin{pmatrix} -2 \\ 3 \end{pmatrix}$$

(a) What is the first principal component w_1 (use the unbiased estimation of covariance)? Draw the first principal component direction w_1 on the plot, anchored at the origin.

To find the first principal component, we must first find the sample mean and covariance matrix Σ of the corresponding zero-mean sample set:

$$M = \frac{1}{4}\left(\begin{pmatrix} 0 \\ 2 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ 2 \end{pmatrix} + \begin{pmatrix} -2 \\ 3 \end{pmatrix}\right) = \frac{1}{4}\begin{pmatrix} 0 + -1 + -1 + -2 \\ 2 + 1 + 2 + 3 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

$$\begin{aligned} \Sigma = cov(X) &= \frac{1}{n-1} X X^T \\ &= \frac{1}{4-1}\begin{pmatrix} 0-(-1) & -1-(-1) & -1-(-1) & -2-(-1) \\ 2-2 & 1-2 & 2-2 & 3-2 \end{pmatrix} X^T \\ &= \frac{1}{3}\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 0 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{3} & \frac{-1}{3} \\ \frac{-1}{3} & \frac{2}{3} \end{pmatrix} \end{aligned}$$

Next, find the eigenvalues and corresponding eigenvectors of Σ

$$\begin{aligned} \det(\Sigma - \lambda I) = \begin{vmatrix} \frac{2}{3} - \lambda & \frac{-1}{3} \\ \frac{-1}{3} & \frac{2}{3} - \lambda \end{vmatrix} &= \left(\frac{2}{3} - \lambda\right)\left(\frac{2}{3} - \lambda\right) - \left(\frac{-1}{3}\right)\left(\frac{-1}{3}\right) \\ &= \lambda^2 - \frac{4}{3}\lambda + \frac{1}{3} = (\lambda - \frac{1}{3})(\lambda - 1) = 0 \quad \iff \quad \lambda = 1 \texttt{ or } \lambda = \frac{1}{3} \end{aligned}$$

Since we are only interested in the first principle component, we only need to find the eigenvector for the largest eigenvalue $\lambda = 1$:

$$(\Sigma - \lambda I)\ v = (\Sigma - I)\ v = 0$$

$$\left(\begin{array}{cc|c} \frac{-1}{3} & \frac{-1}{3} & 0 \\ \frac{-1}{3} & \frac{-1}{3} & 0 \end{array}\right) \sim \left(\begin{array}{cc|c} \frac{-1}{3} & \frac{-1}{3} & 0 \\ 0 & 0 & 0 \end{array}\right)$$

$$\iff \quad \frac{-1}{3}v_1 + \frac{-1}{3}v_2 = 0 \quad \iff \quad v_1 = -v_2$$

$$\text{Hence, } v = v_2\begin{pmatrix} -1 \\ 1 \end{pmatrix} \text{ and } w_1 = \frac{v}{||v||} = \begin{pmatrix} \frac{-\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

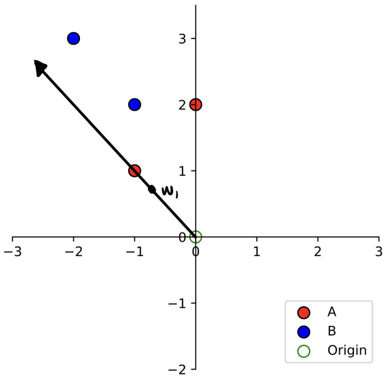

Figure 1: Provided plot with the first principal component direction w_1 drawn.

Continued on next page ⟶

(b) Consider a more general case (not specific to the aforementioned samples): PCA performs linear dimensionality reduction with $z^t = W^T x^t$, where $x^t \in \mathbb{R}^D$ is the original data for the t-th sample, $z^t \in \mathbb{R}^d$ is the low-dimensional projection ($d < D$), $W \in \mathbb{R}^{D\times d}$ is the PCA projection matrix (each column is a principal component). Professor HighLowHigh claims that we can reconstruct the original data with $v^t = W z^t$, so that $\forall_t\ v^t = x^t$. Is the claim correct? Explain your answer with necessary details (you can use formulations if it helps explain).

This claim is incorrect. Observe,

$$\begin{aligned} & z^t = W^T x^t \\ \iff \quad & W z^T = W W^T x^t \\ \iff \quad & W z^T = W W^T x^t \\ \iff \quad & v^t = W W^T x^T \qquad // \text{ by definition of } v^t \end{aligned}$$

Hence, $v^t = x^t \iff WW^T = I$. This, however, is not true in general when $d < D$. Consider, for example,

$$W = (w_1) = \begin{pmatrix} \frac{-\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

which is the first principal component from part (a) above ($d = 1 < D = 2$):

$$WW^T = \begin{pmatrix} \frac{-\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \frac{-\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{-1}{2} \\ \frac{-1}{2} & \frac{1}{2} \end{pmatrix} \neq I$$

If $d = D$, on the other hand, then the columns of W form an orthonormal basis for $\mathbb{R}^D$ (assuming that the principal components are of unit length), and so $WW^T = I = W^T W$ (i.e. W is an orthogonal matrix). This is equivalent to the scenario in which all D principal components are included in W. Intuitively, it makes sense that we cannot recover x^t from W when ($d < D$) since some information is necessarily lost in the projection to the lower dimensional vector space.

Continued on next page ⟶

(c) Compute the Within-Class Scatter matrix S_W and Between-Class Scatter matrix S_B.

To avoid name collisions, let classes A and B be renamed to class 1 and class 2 respectively. Observe,

$$m_1 = \frac{1}{2}\left(\begin{pmatrix}0\\2\end{pmatrix} + \begin{pmatrix}-1\\1\end{pmatrix}\right) = \begin{pmatrix}\frac{-1}{2}\\\frac{3}{2}\end{pmatrix}$$
$$m_2 = \frac{1}{2}\left(\begin{pmatrix}-1\\2\end{pmatrix} + \begin{pmatrix}-2\\3\end{pmatrix}\right) = \begin{pmatrix}\frac{-3}{2}\\\frac{5}{2}\end{pmatrix}$$

$$\begin{aligned}S_1 = (X_1 - m_1)(X_1 - m_1)^T &= \begin{pmatrix}0 - \frac{-1}{2} & -1 - \frac{-1}{2}\\ 2 - \frac{3}{2} & 1 - \frac{3}{2}\end{pmatrix}(X_1 - m_1)^T\\ &= \begin{pmatrix}\frac{1}{2} & \frac{-1}{2}\\ \frac{1}{2} & \frac{-1}{2}\end{pmatrix}\begin{pmatrix}\frac{1}{2} & \frac{1}{2}\\ \frac{-1}{2} & \frac{-1}{2}\end{pmatrix} = \begin{pmatrix}\frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & \frac{1}{2}\end{pmatrix}\end{aligned}$$

$$\begin{aligned}S_2 = (X_2 - m_2)(X_2 - m_2)^T &= \begin{pmatrix}-1 - \frac{-3}{2} & -2 - \frac{-3}{2}\\ 2 - \frac{5}{2} & 3 - \frac{5}{2}\end{pmatrix}(X_2 - m_2)^T\\ &= \begin{pmatrix}\frac{1}{2} & \frac{-1}{2}\\ \frac{-1}{2} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}\frac{1}{2} & \frac{-1}{2}\\ \frac{-1}{2} & \frac{1}{2}\end{pmatrix} = \begin{pmatrix}\frac{1}{2} & \frac{-1}{2}\\ \frac{-1}{2} & \frac{1}{2}\end{pmatrix}\end{aligned}$$

Using m_1, m_2, S_1, and S_2 we will now compute S_W and S_B:

$$S_W = S_1 + S_2 = \begin{pmatrix}\frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & \frac{1}{2}\end{pmatrix} + \begin{pmatrix}\frac{1}{2} & \frac{-1}{2}\\ \frac{-1}{2} & \frac{1}{2}\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

$$S_B = (m_1 - m_2)(m_1 - m_2)^T = \left(\begin{pmatrix}\frac{-1}{2}\\ \frac{3}{2}\end{pmatrix} - \begin{pmatrix}\frac{-3}{2}\\ \frac{5}{2}\end{pmatrix}\right)(m_1 - m_2)^T = \begin{pmatrix}1\\ -1\end{pmatrix}\begin{pmatrix}1 & -1\end{pmatrix} = \begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}$$

Continued on next page ⟶

(d) What is the Fisher projection direction w found by the Fisher Linear Discriminant Analysis (LDA)? Normalize w to have unit length, that is $||w||_2 = 1$, and draw such w on the plot, anchored at the origin.

$$w = c \cdot S_W^{-1}(m_1 - m_2) = c \cdot I \left(\begin{pmatrix} \frac{-1}{2} \\ \frac{3}{2} \end{pmatrix} - \begin{pmatrix} \frac{-3}{2} \\ \frac{5}{2} \end{pmatrix} \right) = c \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

To make w be of unit length, let $c = \frac{-\sqrt{2}}{2}$. Hence,

$$w = \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

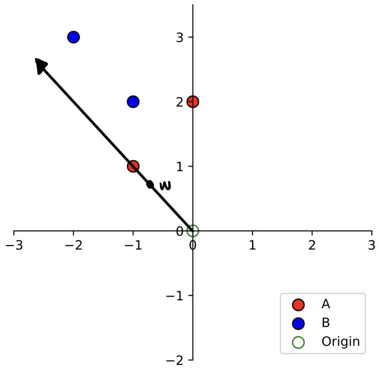

Figure 2: Provided plot with the Fisher projection direction w found by LDA drawn

2. (**30 points**) Given the following data points in 1D: $x_1 = 1,\ x_2 = 4,\ x_3 = 5,\ x_4 = 6,\ x_5 = 7,\ x_6 = 8,\ x_7 = 10,\ x_8 = 12,\ x_9 = 14$, perform k-means clustering algorithm for $k = 3$.

   (a) Start from initial cluster centers $c_1 = 0,\ c_2 = 5,\ c_3 = 10$. Show your steps for all iterations: (1) the cluster assignments $y_1,\ ...\ ,\ y_9$; (2) the updated cluster centers at the end of that iteration.

Let $y_i = k$ reflect that data point x_i is assignment to cluser k which has a center of c_k. Note that the assignment step of each iteration is nothing more than finding the class whose cluster center is nearest to the point in question.

****Iteration 1****

Assignment:

$$\begin{aligned} &y_1 = 1, \\ &y_2 = y_3 = y_4 = y_5 = 2, \\ &y_6 = y_7 = y_8 = y_9 = 3 \end{aligned}$$

Updated cluster centers:

$$\begin{aligned} c_1 &= \frac{x_1}{1} = \frac{1}{1} = 1 \\ c_2 &= \frac{x_2 + x_3 + x_4 + x_5}{4} = \frac{4 + 5 + 6 + 7}{4} = 5.5 \\ c_3 &= \frac{x_6 + x_7 + x_8 + x_9}{4} = \frac{8 + 10 + 12 + 14}{4} = 11 \end{aligned}$$

****Iteration 2****

Assignment:

$$\begin{aligned} &y_1 = 1, \\ &y_2 = y_3 = y_4 = y_5 = y_6 = 2, \\ &y_7 = y_8 = y_9 = 3 \end{aligned}$$

Updated cluster centers:

$$\begin{aligned} c_1 &= \frac{x_1}{1} = \frac{1}{1} = 1 \\ c_2 &= \frac{x_2 + x_3 + x_4 + x_5 + x_6}{5} = \frac{4 + 5 + 6 + 7 + 8}{5} = 6 \\ c_3 &= \frac{x_7 + x_8 + x_9}{3} = \frac{10 + 12 + 14}{3} = 12 \end{aligned}$$

****Iteration 3****

Assignment:

$$\begin{aligned} &y_1 = 1, \\ &y_2 = y_3 = y_4 = y_5 = y_6 = 2, \\ &y_7 = y_8 = y_9 = 3 \end{aligned}$$

Hence, no assignments have changed from iteration 2, meaning that the cluster centers are also unchanged:

$$c_1 = 1,\ c_2 = 6,\ c_3 = 12$$

Continued on next page ⟶

(b) How many iterations does it take for k-means algorithm to converge (i.e., number of iterations includes all iterations you perform to find convergence)? What is the reconstruction error (i.e., distortion measure J, equation 9.1 of the Bishop's textbook) at the end of that iteration?

Three iterations were needed for convergence of the k-means algorithm in part (a). The reconstruction error is:

$$\begin{aligned}
J &= \sum_{n=1}^{9}\sum_{k=1}^{3} r_{nk}||x_n - c_k||^2 \\
&= |x_1 - c_1|^2 + |x_2 - c_2|^2 + |x_3 - c_2|^2 + |x_4 - c_2|^2 + |x_5 - c_2|^2 + |x_6 - c_2|^2 + |x_7 - c_3|^2 + |x_8 - c_3|^2 + |x_9 - c_3|^2 \\
&= |1 - 1|^2 + |4 - 6|^2 + |5 - 6|^2 + |6 - 6|^2 + |7 - 6|^2 + |8 - 6|^2 + |10 - 12|^2 + |12 - 12|^2 + |14 - 12|^2 \\
&= 0 + 4 + 1 + 0 + 1 + 4 + 4 + 0 + 4 = 18
\end{aligned}$$

(c) Repeat the above steps with initial cluster centers $c_1 = 2,\ c_2 = 7,\ c_3 = 12$.

Iteration 1

Assignment:

$$\begin{aligned}
&y_1 = y_2 = 1, \\
&y_3 = y_4 = y_5 = y_6 = 2, \\
&y_7 = y_8 = y_9 = 3
\end{aligned}$$

Updated cluster centers:

$$\begin{aligned}
c_1 &= \frac{x_1 + x_2}{2} = \frac{1 + 4}{2} = 2.5 \\
c_2 &= \frac{x_2 + x_3 + x_4 + x_5}{4} = \frac{5 + 6 + 7 + 8}{4} = 6.5 \\
c_3 &= \frac{x_7 + x_8 + x_9}{3} = \frac{10 + 12 + 14}{3} = 12
\end{aligned}$$

Iteration 2

Assignment:

$$\begin{aligned}
&y_1 = y_2 = 1, \\
&y_3 = y_4 = y_5 = y_6 = 2, \\
&y_7 = y_8 = y_9 = 3
\end{aligned}$$

Hence, no assignments have changed from iteration 1, meaning that the cluster centers are also unchanged:

$$c_1 = 2.5,\ c_2 = 6.5,\ c_3 = 12$$

(d) How many iterations does it take for k-means algorithm to converge in this case? What is the reconstruction error at the end of that iteration?

Only two iterations were needed for convergence of the k-means algorithm in part (c). The reconstruction error is:

$$\begin{aligned}
J &= \sum_{n=1}^{9}\sum_{k=1}^{3} r_{nk}||x_n - c_k||^2 \\
&= |x_1 - c_1|^2 + |x_2 - c_1|^2 + |x_3 - c_2|^2 + |x_4 - c_2|^2 + |x_5 - c_2|^2 + |x_6 - c_2|^2 + |x_7 - c_3|^2 + |x_8 - c_3|^2 + |x_9 - c_3|^2 \\
&= |1 - 2|^2 + |4 - 2|^2 + |5 - 6.5|^2 + |6 - 6.5|^2 + |7 - 6.5|^2 + |8 - 6.5|^2 + |10 - 12|^2 + |12 - 12|^2 + |14 - 12|^2 \\
&= 1 + 4 + 1.5^2 + 0.5^2 + 1.5^2 + 1.5^2 + 4 + 0 + 4 = 20
\end{aligned}$$

Continued on next page ⟶

(e) Comparing (a) with (c), which solution is better? Why?

Note that the final iteration reconstruction error of $L = 18$ for (a) is smaller than the final iteration reconstruction error of $L = 20$ for (c). Since the objective of the k-means clustering algorithm is to minimize this reconstruction error (i.e. to minimize the within cluster sum of squares), we can conclude that (a) is the better solution between the two.

3. (**40 points**) In this programming exercise you will implement k-means and Principal Component Analysis algorithms:

(a) k-means ($K = 8$): Report the number of iterations for convergence and the classification accuracy on the test samples. Plot the history of reconstruction errors. Is the plot shape following what you expect?

Using raw data converged in 27 iterations (1.30 seconds) with a classification accuracy of 0.94 on the test samples. The shape of the plot shown in figure 3 is as expected. Specifically, the reconstruction error is expected to decrease with each iteration as the sum of squared distances to the nearest cluster center is reduced at each iteration of the EM process, eventually converging to some local minimum.

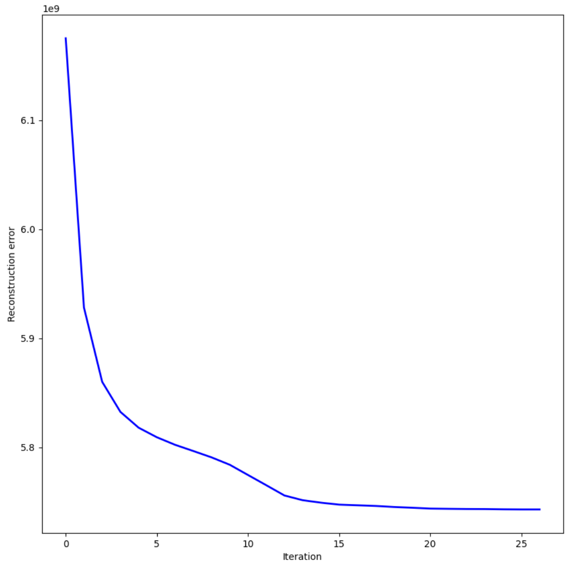

Figure 3: k-means ($K = 8$) reconstruction errors for each iteration while fitting to the raw training data.

Continued on next page ⟶

(b) k-means ($K = 8$) with PCA capturing $> 90\%$: How many dimenstions are necessary in this case? Does PCA help clustering? Explain. (Hint: Consider both the classification accuracy and the runtime of the algorithm.)

To capture $> 90\%$ of the variance in the training data set, the first 73 principal components are used to project the data into in 73 dimensions. The k-means fitting step converged in 26 iterations (0.79 seconds) when using the 73 dimensional data. The resulting classification accuracy was 0.94 on the test set. These results clearly show that PCA helped to improve the k-means clustering algorithm in this case. In particular, there was a signficant runtime speedup ($\approx 1.65\times$) observed when using the low dimensional (PCA) data when compared with the 784 dimensional raw data, without any observed change in the final classification accuracy.

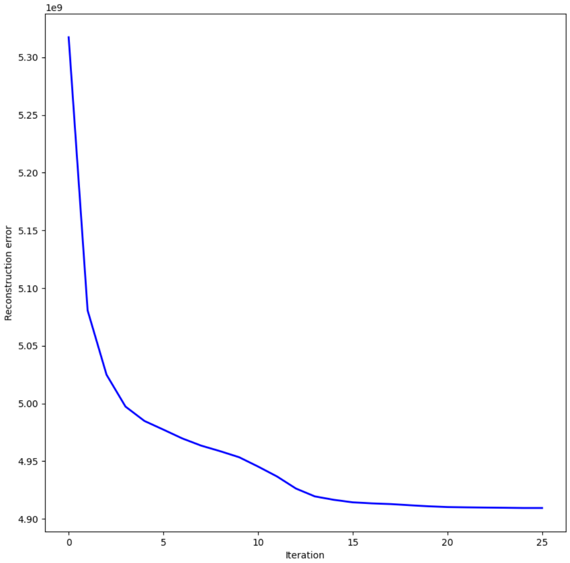

Figure 4: k-means ($K = 8$) reconstruction errors for each iteration while fitting to the 73 dimensional (PCA) training data.

Continued on next page ⟶

(c) k-means ($K = 8$) using only first principal component: Are the results better than in (b)? Explain.

With the data projected into 1 dimension with PCA, the k-means fitting step converged in 33 iterations (0.86 seconds), and resulted in a classification accuracy of 0.74. While these results demonstrate a simlar runtime speedup to that observed in part (b), the classification accuracy has significantly degraded when compared to the 784 dimensional raw data version. The inferiority of the 1 dimensional data projection makes sense as too much of the original data variance is lost when only a single principal component is used.

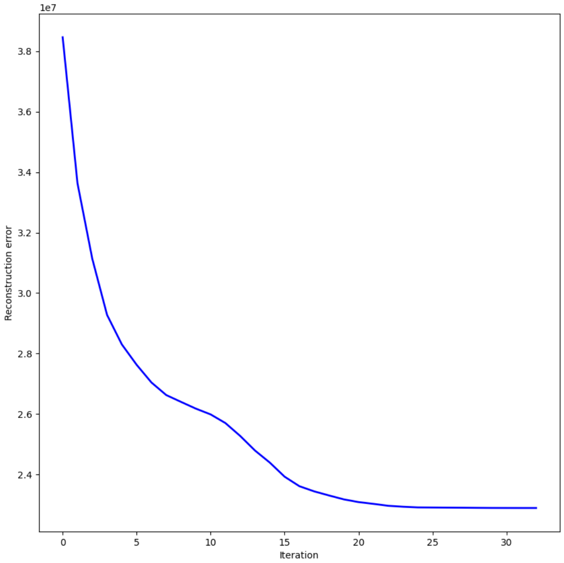

Figure 5: k-means ($K = 8$) reconstruction errors for each iteration while fitting to the 1 dimensional (PCA) training data.