The Chinese University of Hong Kong, Shenzhen
DDA3020 2024-2025 Fall

Solutions to Assignment 3

Yinqi (Marcus) Zhong
marazhong1@gmail.com
+(86) 13710210206

December 7, 2024

1 Writing Problems

1.1 K-Means

1.1.1

Figure 1 : A.1
Figure 2 : B.1

1.1.2

(a) The new center for cluster 0 : (6.3,3.3)
(b) The new center for cluster 1: (5.3,4.7)
(c) 3 points for cluster 0: (5.5,3.1), (6.6,3.0), (6.8,3.8)
(d) 2 points for cluster 1: (5.1,4.8), (5.5,4.6)

1.1.3

(a) minimum possible value of objective : 0
There are two argmin function for this objective function. The outer argmin is trying to find the minimum K that minimize the distance between points and new centers. This case occurs when k = N, where each data point is assigned its own cluster center, and the distances between points and their assigned cluster centers are all zero, which returns the minmum value = 0 for the objective function.
(b) k = N = 100, reason stated in (a)

1.1.4

(a) For each datapoint, k possible clusters can be assigned to.
(b) Possible assignment numbers: 10^{1000} one assignment takes 0.01s, so the total seconds for brute force algorithm take to check all assignments would be :(/seconds)

$$0.01 * 10^{1000} = 10^{998}$$

1.2 PCA (Principal Component Analysis)

1.2.1 Proof of equivalance of two derivations

To prove

$$\max_{U,U^TU=I} \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \tilde{\mu}|^2 = \min_{U,U^TU=I} \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \tilde{\mathbf{x}}^{(n)}\|^2$$

Which equals to prove:

$$\frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \tilde{\mu}|^2 = -\frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \tilde{\mathbf{x}}^{(n)}\|^2 + constant$$

Echos with the given hint:

$$\frac{1}{N}\sum_{n=1}^{N} \|\tilde{\mathbf{x}}^{(n)} - \tilde{\mu}||^2 + \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \tilde{\mathbf{x}}^{(n)}\|^2 = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \mu||^2$$

By Pythagorean Theorem: we have

$$\frac{1}{N}\sum_{n=1}^{N} \|\tilde{\mathbf{x}}^{(n)} - \mu\|^2 + \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \tilde{\mathbf{x}}^{(n)}\|^2 = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \mu||^2$$

So now, we just need to prove $\mu = \tilde{\mu}$
Since

$$\tilde{x} = \mu + Uz \in R^D$$
$$z = U^T(x - \mu) \in R^K$$

where $U \in \mathbb{R}^{D\times K}$ as the projection matrix with columns $\{\mathbf{u}_k\}_{k=1}^K$
Using these two expressions, we can get:

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} \tilde{\mathbf{x}}^{(n)}$$

$$= \mu + \mathbf{U}\left(\frac{\mathbf{1}}{\mathbf{N}}\sum_{n=1}^{N} \mathbf{z}^{(n)}\right)$$

$$= \mu + \frac{\mathbf{1}}{\mathbf{N}}\mathbf{U}\mathbf{U}^{\mathbf{T}}\sum_{n=1}^{N}(\mathbf{x}^{(n)} - \mu) = \mu$$

So now we have prove $\mu = \tilde{\mu}$, so we have prove that problem (1) maximize the variance of the reconstructions and problem(2) minimize the the reconstruction error is equivalent.

1.2.2 Case: Reduce dimension to one

The variance of reconstructions can be reformulated according to $\tilde{\mathbf{x}} = \mu + \omega\mathbf{z}$ as :

$$\max_{\omega,\omega^T\omega=I} \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \tilde{\mu}|^2 = \max_{\omega,\omega^T\omega=I} \frac{1}{N}\sum_{n=1}^{N} \|\omega\mathbf{z}^{(n)}\|^2$$

$$\equiv \max_{\omega,\omega^T\omega=I} \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{z}^{(n)}\|^2$$

$$= \max_{\omega,\omega^T\omega=I} \frac{1}{N}\sum_{n=1}^{N} \|\omega^{\mathbf{T}}(x - \mu)\|^2$$

$$\equiv \max_{\omega,\omega^T\omega=I} \frac{1}{N}\sum_{n=1}^{N} \|\text{Trace}\left(\omega^T\left(\mathbf{x}^{(n)} - \mu\right)\left(\mathbf{x}^{(n)} - \mu\right)^T\omega\right)\|^2$$

By definition of the covariance matrix, we know the above equation can be simplified as:

$$\max_{\omega} \text{Trace}\left(\omega^T \Sigma \omega\right) = \sum_{k=1}^{K} \omega_k^T \Sigma \omega_k$$

$$\text{s.t. } \omega^T \omega = 1$$

Define the Lagrangian function as:

$$\mathcal{L}(\omega, \lambda) = \omega^T \Sigma \omega - \lambda(\omega^T \omega - 1)$$

Now, compute the gradient of $\mathcal{L}(\omega, \lambda)$ with respect to ω:

$$\nabla_\omega \mathcal{L}(w, \lambda) = 2\Sigma w - 2\lambda w$$

Set this equal to zero to find the critical points:

$$\Sigma \omega = \lambda \omega$$

Which means: ω must be an eigenvector of Σ, and the corresponding eigenvalue is λ.
Using SVD Decomposition, we know

$$\Sigma = Q \Lambda_D Q^T = \sum_{i=1}^{D} \lambda_i q_i q_i^T$$

where $Q = [q_1, \ldots, q_D] \in \mathbb{R}^{D \times D}$ with q_i being the eigenvector corresponding to the i-th largest eigenvalue λ_i, and $\Lambda_D = \text{diag}(\lambda_1, \ldots, \lambda_D)$ with $\lambda_1 \geq \cdots \geq \lambda_D$.
Substitute the equation to the objective function, we have

$$\sum_{k=1}^{K} \sum_{i=1}^{1} \lambda_i (\omega_k^T q_i)(q_i^T u_k) = \sum_{t \in T} \lambda_t.$$

T is the set of top-K eigenvalues, here K is limited to 1, so it's ovious that to optimize the objective function, λ should be the largest eigenvalue.
And ω should be the eigenvector associated with the largest eigenvalue of covariance matrix.

2 Coding Report

2.1 PCA

2.1.1 Split training set and test set:

Training set: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135]
Test set: [36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150]

2.1.2 SVD Decomposition

```
Q2.1.2 SVD decomposition:
Mean vector: [5.89047619 3.05809524 3.80571429 1.18666667]

Covariance matrix:
[[ 0.71019501 -0.05001814  1.30319728  0.50244444]
 [-0.05001814  0.20719637 -0.3829034  -0.14170159]
 [ 1.30319728 -0.3829034   3.19901497  1.28779048]
 [ 0.50244444 -0.14170159  1.28779048  0.55791746]]

Eigenvalues (vector): [4.30938932 0.2677887  0.07822324 0.01892255]

Eigenvectors (matrix):
[[ 0.3612084  -0.67668874 -0.5437191   0.34057361]
 [-0.09667266 -0.71059308  0.61616041 -0.32566581]
 [ 0.86000139  0.16246182  0.06924822 -0.47875719]
 [ 0.34724702  0.10371103  0.56561522  0.74077186]]
```

Figure 1: SVD

2.1.3 Project onto 1-dimensional subspace and reconstruct

```
Q2.1.3 Project onto 1-dimensional subspace and reconstruct:

Projection matrix W:
[[ 0.3612084 ]
 [-0.09667266]
 [ 0.86000139]
 [ 0.34724702]]

Shape of X train mapped: (105, 1), Shape of X train reconstructed: (105, 4)
Variance train: 4.309389319247093
Reconstruction loss train: 0.36493449027671576

Shape of X test mapped: (45, 1), Shape of X test reconstructed: (45, 4)
Variance test: 3.907084095695329
Reconstruction loss test: 0.2997891222678479
```

Figure 2: 1D Projection

2.1.4 Project onto 2-dimensional subspace and reconstruct:

```
Q2.1.4 Project onto 2-dimensional subspace and reconstruct:

Projection matrix W:
[[ 0.3612084  -0.67668874]
 [-0.09667266 -0.71059308]
 [ 0.86000139  0.16246182]
 [ 0.34724702  0.10371103]]

Shape of X train mapped: (105, 2), Shape of X train reconstruct: (105, 4)
Variance train: 4.577178016423884
Reconstruction loss train: 0.09714579309992513

Shape of X test mapped: (45, 2), Shape of X test reconstruct: (45, 4)
Variance test: 4.078378031615012
Reconstruction loss test: 0.11952572036446978
```

Figure 3: 2D Projection

2.1.5 Project onto 3-dimensional subspace and reconstruct:

```
Q2.1.5 Project onto 3-dimensional subspace and reconstruct:

Projection matrix W:
[[ 0.3612084  -0.67668874 -0.5437191 ]
 [-0.09667266 -0.71059308  0.61616041]
 [ 0.86000139  0.16246182  0.06924822]
 [ 0.34724702  0.10371103  0.56561522]]

Shape of X train mapped: (105, 3), Shape of X train reconstruct: (105, 4)
Variance train: 4.655401260550289
Reconstruction loss train: 0.018922548973519974

Shape of X test mapped: (45, 3), Shape of X test reconstruct: (45, 4)
Variance test: 4.1541473257389425
Reconstruction loss test: 0.03587183464503327
```

Figure 4: 3D Projection

2.1.6 Plotting

Dimension - Variance:

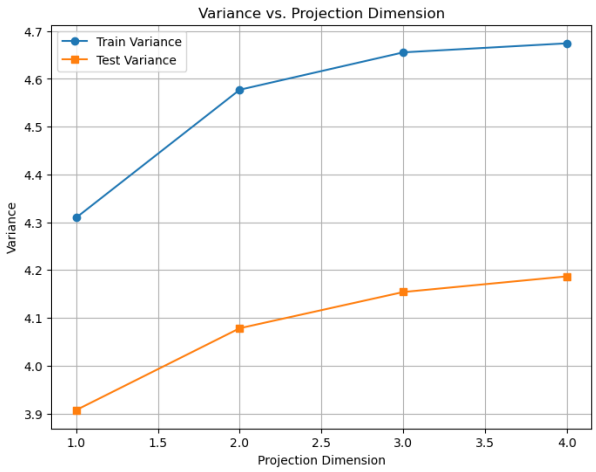

Figure 5: Dimension - Variance

As the projection dimension increases from 1 to 4, the variance captured by the principal components increases. At dimension 4, 100% of the variance is captured, indicating that all original data variance is retained.

Dimension - Reconstruction loss:

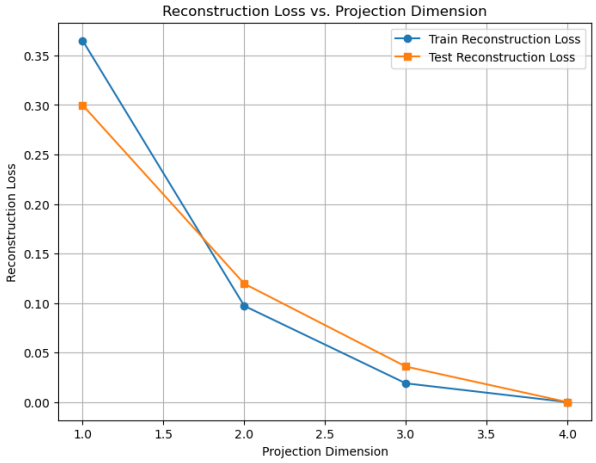

Figure 6: Dimension - Loss

The reconstruction loss decreases as the projection dimension increases. With more principal components, the reconstructed data becomes closer to the original data, resulting in lower reconstruction loss. At dimension 4, the reconstruction loss is zero or near-zero, indicating perfect reconstruction, but not practical in reality (no dimension reduction).

2.2 K-Means, Mixture Models and EM Algorithms

2.2.1 Data Pre-processing

As can be seen, there are 8 columns in the original abalone.data dataset, and their names are unassigned. Each column contains 4177 non null data. The column names corresponding to its column order can be found in .names file as : ['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight', 'Viscera weight', 'Shell weight', 'Rings']

One hot encoding is implemented to replace the categorical data into boolean type for further transformation, since string type is not suitable for calculation.

Min - Max normalization is implemented to scale the numerical variables to guarantee the accuracy of prediction.
To better visualize the numerical data distribution, I choose bar chart to plot the mean value for each numerical column. And I plot their histograms and box plots to compare their frequencies.

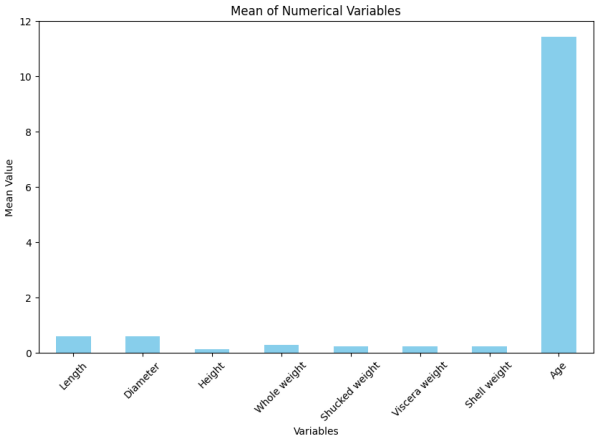

Figure 7: Bar Chart of Numerical Variables

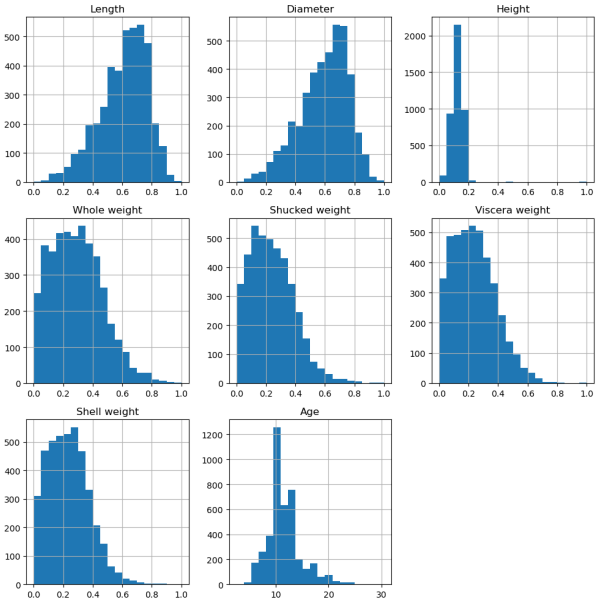

Figure 8: Histograms

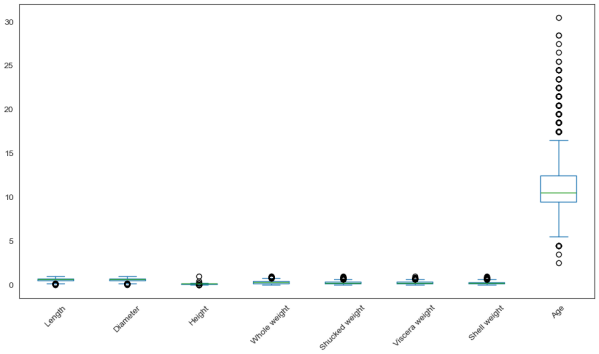

Figure 9: Box Plot of Numerical Variables

2.2.2 K-Means Clustering(using sklearn)

The Silhouette Coefficient for 5 clusters K-Mean: 0.5470483921709525

2.2.3 GMM and EM Algorithms

i) Using 20 iterations GMM on 2 Gaussians on ['Whole weights]: Mixture: Gaussian(0.392093, 0.172422), Gaussian(0.207296, 0.121346), mix=0.463)
ii) Plot of fitted gmm

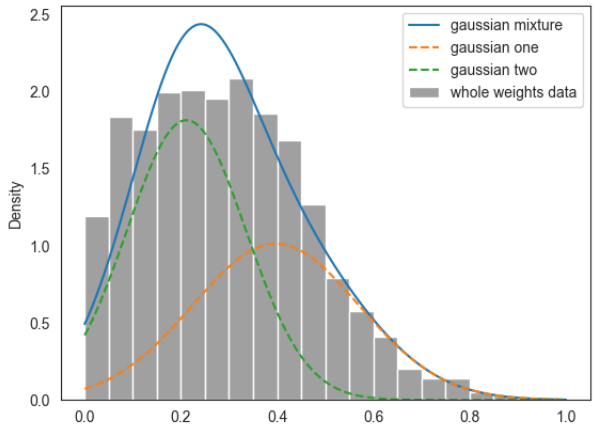

Figure 10: Fitted GMM using reference

iii) Using sklearn package: Mixture: Gaussian(0.168589, 0.096994), Gaussian(0.412434, 0.144819), mix=0.491)

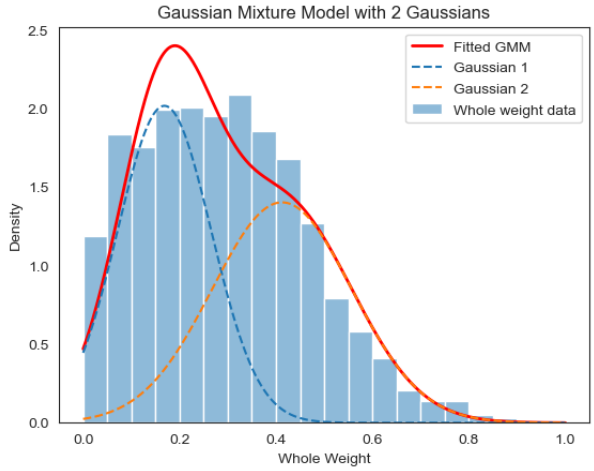

Figure 11: Fitted GMM using sklearn

iv) Fit using all predictors, the result can be shown in the code file (Too long to display in the report, not necessary.
v) The silhouette coefficient for GMM modelled in iii) is 0.5804321592203657.
The silhouette coefficient for K Means modelled in 2.2.2) is 0.5470483921709526.
Based on the silhouette coefficients, I prefer the Gaussian Mixture Model (GMM) over the K-Means clustering. The GMM achieved a higher silhouette score of 0.5804 compared to 0.5470 for K-Means, indicating that GMM provides better-defined and more cohesive clusters for the 'Whole weight' data.