Written Assignment 3

Yushan Liu   Student ID: 2024214103

November 16, 2024

All the tasks in this WA is done entirely by MYSELF.
The source code for 3.4(b) can be found at: `Task3-4-b.ipynb`

3.1 VC Dimension

The concept class $C = \{c_{a,b} \mid a, b \in \mathbb{R}, a < b\}$ is defined as

$$c_{a,b}(x) = \begin{cases} 1, & x \in [a, b], \\ 0, & \text{otherwise.} \end{cases}$$

To proof VC$dim(C) = 2$, means that the concept class can distinguish all labelings for up to 2 points but cannot distinguish all labelings for 3 points.

First let's proof VC$dim(C) \geq 2$: Let $S = \{x_1, x_2\}$, where $x_1 < x_2$. We verify that the concept class C can distinguish all possible labelings of S. For the 4 possible labelings:

- **Labeling (0, 0)**: Neither point is covered. Choose $c_{a,b}$ such that $b < x_1$.
- **Labeling (1, 0)**: Only x_1 is covered. Choose $c_{a,b}$ such that $a \leq x_1 < b \leq x_2$.
- **Labeling (0, 1)**: Only x_2 is covered. Choose $c_{a,b}$ such that $a > x_1$ and $b \geq x_2$.
- **Labeling (1, 1)**: Both x_1 and x_2 are covered. Choose $c_{a,b}$ such that $a \leq x_1$ and $b \geq x_2$.

For each labeling, we can find a function $c_{a,b} \in C$ that assigns the labels correctly. Hence, C can distinguish all possible labelings of 2 points, so VCdim$(C) \geq 2$.

Then let's proof VC$dim(C) \leq 2$: Consider $S = \{x_1, x_2, x_3\}$, where $x_1 < x_2 < x_3$. We verify that C cannot distinguish all possible labelings of S. For example, consider the labeling $(1, 0, 1)$, where x_1 and x_3 are covered, but x_2 is not. Since every function $c_{a,b} \in C$ corresponds to a continuous interval $[a, b]$, it is impossible to construct an interval that includes x_1 and x_3 but excludes x_2. Thus, C cannot distinguish all possible labelings of 3 points, so VCdim$(C) \leq 2$.

Therefore, we have

$$\text{VCdim}(C) = 2$$

.

3.2 Rademacher Complexity

The Rademacher complexity of a function class F over a sample $S = \{x_1, x_2, \ldots, x_m\}$ is defined as:

$$\mathcal{R}_m(F) = \mathbb{E}_\sigma \left[\sup_{f \in F} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(x_i) \right],$$

where σ_i are independent Rademacher random variables ($\sigma_i \in \{-1, +1\}$).

(a)

For $g(x) = af(x) + b \in aF + b$, we compute:

$$\frac{1}{m} \sum_{i=1}^{m} \sigma_i g(x_i) = \frac{1}{m} \sum_{i=1}^{m} \sigma_i \big(af(x_i) + b\big) = a \cdot \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(x_i) + b \cdot \frac{1}{m} \sum_{i=1}^{m} \sigma_i.$$

Since σ_i are symmetric and independent, the expectation of the term involving b vanishes:

$$\mathbb{E}_\sigma\left[b \cdot \frac{1}{m}\sum_{i=1}^{m}\sigma_i\right] = 0.$$

Thus, the Rademacher complexity becomes:

$$\mathcal{R}_m(aF+b) = \mathbb{E}_\sigma\left[\sup_{f\in F}\frac{1}{m}\sum_{i=1}^{m}\sigma_i \cdot af(x_i)\right] = |a| \cdot \mathbb{E}_\sigma\left[\sup_{f\in F}\frac{1}{m}\sum_{i=1}^{m}\sigma_i f(x_i)\right].$$

Therefore, we have:

$$\mathcal{R}_m(aF+b) = |a|\mathcal{R}_m(F).$$

(b)

For $l_h(x,y)$, we can rewrite as:

$$l_h(x,y) = \frac{1-h(x)y}{2} = -\frac{1}{2}\cdot h(x)y + \frac{1}{2}.$$

By the result from part (a), the Rademacher complexity of a linearly transformed function class satisfies:

$$\mathcal{R}_m(\mathcal{L}(H)) = \left|-\frac{1}{2}\right|\mathcal{R}_m(H) = \frac{1}{2}\mathcal{R}_m(H).$$

Threrfore, we have:

$$2\mathcal{R}_m(H) = \mathcal{R}_m(\mathcal{L}(H)).$$

3.3 K-means

The objective of the k-means clustering problem is to partition the data into k clusters $C_1,\dots,C_k$ such that the within-cluster sum of squares is minimized:

$$\min_C \sum_{j=1}^{k}\sum_{x\in C_j}\|x-\mu_j\|^2,$$

where μ_j is the mean (center) of the j-th cluster:

$$\mu_j = \frac{1}{|C_j|}\sum_{x\in C_j} x.$$

(a)

We can expand the squared deviation term $\|x-\mu_j\|^2$ for each cluster C_j as follows:

$$\sum_{j=1}^{k}\sum_{x\in C_j}\|x-\mu_j\|^2 = \sum_{j=1}^{k}\left(\sum_{x\in C_j}\|x\|^2 - 2\sum_{x\in C_j}x^\top\mu_j + |C_j|\|\mu_j\|^2\right)$$

Consider the second term in Eq:

$$-2\sum_{x\in C_j} x^\top \mu_j = -2\sum_{x\in C_j} x^\top \left(\frac{1}{|C_j|}\sum_{x'\in C_j} x'\right) = \frac{-2}{|C_j|}\sum_{x\in C_j}\sum_{x'\in C_j} x^\top x'$$

Consider the third term in Eq:

$$|C_j|\|\mu_j\|^2 = |C_j|\left\|\frac{1}{|C_j|}\sum_{x'\in C_j} x'\right\|^2 = \frac{1}{|C_j|}\sum_{x'\in C_j}\|x'\|^2$$

We can rewrite the objective function as:

$$\begin{aligned}
&\sum_{j=1}^{k}\left(\sum_{x\in C_j}\|x\|^2 - 2\sum_{x\in C_j} x^\top \mu_j + |C_j|\|\mu_j\|^2\right) \\
=&\sum_{j=1}^{k}\left(\frac{1}{C_j}\sum_{x\in C_j}\sum_{x'\in C_j}\|x\|^2 - 2\frac{1}{|C_j|}\sum_{x,x'\in C_j} x^\top x' + \frac{1}{|C_j|}\sum_{x'\in C_j}\|x'\|^2\right) \\
=&\sum_{j=1}^{k}\frac{1}{C_j}\sum_{x\in C_j}\sum_{x'\in C_j}\left(\|x\|^2 - 2x^\top x' + \|x'\|^2\right) = \sum_{j=1}^{k}\frac{1}{C_j}\sum_{x\in C_j}\sum_{x'\in C_j}\|x-x'\|^2
\end{aligned}$$

Because the calculation $\sum_{x\in C_j}\sum_{x'\in C_j}\|x-x'\|^2$ actually calculates the squared distance for every pair of points in cluster C_j. Since this is a double summation, each pair of points (x, x') is counted twice, so we can write the objective function as:

$$\sum_{j=1}^{k}\frac{1}{|C_j|}\sum_{x\in C_j}\sum_{x'\in C_j}\|x-x'\|^2 = \sum_{j=1}^{k}\frac{1}{2|C_j|}\sum_{x,x'\in C_j}\|x-x'\|^2$$

Therefore, we can know that:

$$\sum_{j=1}^{k}\sum_{x\in C_j}\|x-\mu_j\|^2 = \sum_{j=1}^{k}\frac{1}{2|C_j|}\sum_{x,x'\in C_j}\|x-x'\|^2$$

which shows that the k-means clustering problem is equivalent to minimizing the pairwise squared deviation between points in the same cluster.

(b)

We begin with a simplification as follows:

$$S \triangleq \sum_{i=1}^{k}\sum_{j=1}^{k}|C_i||C_j|\|\mu_i-\mu_j\|^2 = \sum_{i=1}^{k}\sum_{j=1}^{k}|C_i||C_j|\left(\|\mu_i\|^2 - 2\mu_i^\top\mu_j + \|\mu_j\|^2\right)$$

Before the proof, we define some notations as follows:

- $m = \sum_{j=1}^{k}|C_j|$ is the total number of data points.
- $\bar{x} = \frac{1}{m}\sum_{x\in X} x$ is the overall mean of the data.

For the first and the third term, we have:

$$\sum_{i=1}^{k}\sum_{j=1}^{k}|C_i||C_j|\|\mu_i\|^2 = \left(\sum_{i=1}^{k}|C_i|\|\mu_i\|^2\right)\left(\sum_{j=1}^{k}|C_j|\right)$$
$$= m\sum_{i=1}^{k}|C_i|\|\mu_i\|^2 = m\sum_{j=1}^{k}|C_j|\|\mu_j\|^2$$

For the second term, we have:

$$-2\sum_{i=1}^{k}\sum_{j=1}^{k}|C_i||C_j|\mu_i^\top\mu_j = -2\sum_{i=1}^{k}|C_i|\mu_i^\top\left(\sum_{j=1}^{k}|C_j|\mu_j\right)$$

Since we have:

$$\sum_{j=1}^{k}|C_j|\mu_j = \sum_{j=1}^{k}\sum_{x\in C_j}x = \sum_{x\in X}x = m\bar{x}.$$

Therefore, we can rewrite the second term as:

$$-2\left(\sum_{i=1}^{k}|C_i|\mu_i^\top\right)m\bar{x} = -2m\bar{x}^\top m\bar{x} = -2m^2\|\bar{x}\|^2.$$

Combine the results above, we have:

$$S = m\sum_{i=1}^{k}|C_i|\|\mu_i\|^2 - 2m^2\|\bar{x}\|^2 + m\sum_{j=1}^{k}|C_j|\|\mu_j\|^2 = 2m\sum_{j=1}^{k}|C_j|\|\mu_j\|^2 - 2m^2\|\bar{x}\|^2$$

Now we define the **Total Sum of Squares, TSS** as:

$$\text{TSS} = \sum_{x\in X}\|x-\bar{x}\|^2$$

The TSS of the data can be devided into two parts:

- **Within-cluster Sum of Squares, WSS**: $\sum_{j=1}^{k}\sum_{x\in C_j}\|x-\mu_j\|^2$
- **Between-cluster Sum of Squares, BSS**: $\sum_{j=1}^{k}|C_j|\|\mu_j-\bar{x}\|^2$

We can easily know that:

$$\text{TSS} = \text{WSS} + \text{BSS}$$

Now we find the relationship between S and BSS:
First, we simplify the BSS as follows:

$$\text{BSS} = \sum_{j=1}^{k}|C_j|\left(\|\mu_j\|^2 - 2\mu_j^\top\bar{x} + \|\bar{x}\|^2\right)$$

$$= \sum_{j=1}^{k} |C_j| \|\mu_j\|^2 - 2 \left(\sum_{j=1}^{k} |C_j| \mu_j^\top \right) \bar{x} + \left(\sum_{j=1}^{k} |C_j| \right) \|\bar{x}\|^2$$

as we have proved that $\sum_{j=1}^{k} |C_j| \mu_j = m\bar{x}$, we can know that:

$$\text{BSS} = \sum_{j=1}^{k} |C_j| \|\mu_j\|^2 - 2m\|\bar{x}\|^2 + m\|\bar{x}\|^2 = \sum_{j=1}^{k} |C_j| \|\mu_j\|^2 - m\|\bar{x}\|^2$$

We can rewriting as:

$$\sum_{j=1}^{k} |C_j| \|\mu_j\|^2 = \text{BSS} + m\|\bar{x}\|^2.$$

So, we can know that:

$$S = 2m \left(\text{BSS} + m\|\bar{x}\|^2 \right) - 2m^2\|\bar{x}\|^2 = 2m \cdot \text{BSS} + 2m^2\|\bar{x}\|^2 - 2m^2\|\bar{x}\|^2 = 2m \cdot \text{BSS}$$

This means that maximizing S is equivalent to maximizing the between-cluster sum of squares (BSS), since m is a constant (the total number of data points).

Since the TSS only depends on the data, means that TSS is contant. So minimizing the within-cluster sum of squares (WSS) is equivalent to maximizing the between-cluster sum of squares (BSS).

$$\arg\min_C \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2 \quad \Longleftrightarrow \quad \arg\max_C \sum_{i=1}^{k} \sum_{j=1}^{k} |C_i||C_j| \|\mu_i - \mu_j\|^2.$$

3.4 Spectral Clustering

(a)

We have known that for subgraphs A and B, the Ratiocut and Ncut are defined as:

$$\text{RatioCut}(A, B) = \frac{\text{Cut}(A, B)}{|A|} + \frac{\text{Cut}(A, B)}{|B|},$$

$$\text{NCut}(A, B) = \frac{\text{Cut}(A, B)}{\text{vol}(A)} + \frac{\text{Cut}(A, B)}{\text{vol}(B)},$$

Under the Partition1,

- Subgraph $A = \{1, 2, 3\}$,
- Subgraph $B = \{4, 5, 6, 7\}$.

So the cut value is $\text{Cut}(\{1, 2, 3\}, \{4, 5, 6, 7\}) = 1$.
The Ratiocut is:

$$\frac{1}{|A|} + \frac{1}{|B|} = \frac{1}{3} + \frac{1}{4} = \frac{7}{12}$$

- $\text{vol}(A) = 2 + 2 + 3 = 7$,

- $\text{vol}(B) = 2 + 3 + 1 + 1 = 7$.

So the Ncut is:

$$\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} = \frac{1}{7} + \frac{1}{7} = \frac{2}{7}$$

Under the Partition2,

- Subgraph $A = \{1, 2, 3, 4\}$,
- Subgraph $B = \{5, 6, 7\}$.

So the cut value is $\text{Cut}(\{1, 2, 3, 4\}, \{5, 6, 7\}) = 1$.
The Ratiocut is:

$$\frac{1}{|A|} + \frac{1}{|B|} = \frac{1}{4} + \frac{1}{3} = \frac{7}{12}$$

- $\text{vol}(A) = 2 + 2 + 3 + 2 = 9$,
- $\text{vol}(B) = 3 + 1 + 1 = 5$.

So the Ncut is:

$$\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} = \frac{1}{9} + \frac{1}{5} = \frac{14}{45}$$

For RatioCut, both partitions yield the same value ($\frac{7}{12}$). However, for NCut, the partition along edge (4,5) yields a smaller value ($\frac{14}{45}$), indicating a better partition.

(b)

Step 1: D and A

The Adjacency Matrix A is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The Degree Matrix D is:

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 2: Compute L

The Laplacian Matrix L is defined as $L = D - A$

Step 3: Compute L_{norm}

The normalized Laplacian matrix is given by:

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2}.$$

Step 4: Compute Eigenvalues and Eigenvectors

Perform eigen decomposition on L_{norm} to obtain its eigenvalues and eigenvectors.

Step 5: Partition by Using the Fiedler Vector

The eigenvector corresponding to the second smallest eigenvalue (known as the Fiedler vector) is used to partition the graph:

- Assign nodes with positive values in the Fiedler vector to one cluster.
- Assign nodes with negative values to the other cluster.

We can use the Python lib `numpy` and `scipy.linalg.eigh` to compute the results below. The code and the output are shown by `Task-3-4-b.ipynb`.

We can know that :

Eigenvalues:

$$\left[1.11 \times 10^{-15}, \quad 0.149, \quad 0.871, \quad 1.0, \quad 1.5, \quad 1.537, \quad 1.943\right]$$

Eigenvectors:

$$\begin{bmatrix} -0.378 & -0.407 & 0.325 & 0.0 & 0.707 & 0.286 & -0.063 \\ -0.378 & -0.407 & 0.325 & -0.0 & -0.707 & 0.286 & -0.063 \\ -0.463 & -0.350 & -0.295 & 0.0 & 0.0 & -0.726 & 0.222 \\ -0.378 & 0.086 & -0.743 & -0.0 & 0.0 & 0.383 & -0.388 \\ -0.463 & 0.529 & 0.060 & -0.0 & 0.0 & 0.222 & 0.673 \\ -0.267 & 0.359 & 0.270 & -0.707 & 0.0 & -0.239 & -0.412 \\ -0.267 & 0.359 & 0.270 & 0.707 & 0.0 & -0.239 & -0.412 \end{bmatrix}$$

So the Fiedler vector is:

$$\begin{bmatrix} -0.407 \\ -0.407 \\ -0.350 \\ 0.086 \\ 0.529 \\ 0.359 \\ 0.359 \end{bmatrix}$$

By using the Fiedler vector, we can partition the graph into two clusters:

- Cluster 1: $\{1, 2, 3, 4\}$
- Cluster 2: $\{5, 6, 7\}$