Indian Institute of Technology Jodhpur
Machine Learning I: Fractal 2
Practice Problems

1. Let $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \in \mathbb{R}^{d \times n}$ be a matrix such that its columns are orthonormal to each other, i.e., $\mathbf{a}_i^\top \mathbf{a}_j = 1$ if $i = j$ and $\mathbf{a}_i^\top \mathbf{a}_j = 0$ if $i \neq j$. Then, show that $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$.

2. Determine the gradient of the function $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{x}^\top \mathbf{b}$. Here, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{x} \in \mathbb{R}^{n \times 1}$, and $\mathbf{b} \in \mathbb{R}^{n \times 1}$.

3. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a matrix and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ be two vectors. Then, show that $\mathbf{x}^\top \mathbf{A}\mathbf{y} = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i y_j a_{ij}$.

4. The trace of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is defined as $\text{Trace}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$. Let $\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_n \end{bmatrix} \in \mathbf{R}^{n \times n}$ be a matrix. Then, show that $\text{Trace}(\mathbf{B}^\top \mathbf{A}\mathbf{B}) = \sum_{i=1}^{n} \mathbf{b}_i^\top \mathbf{A}\mathbf{b}_i$.

5. Let $f : \mathbb{R}^{n \times n} \to \mathbb{R}$ be matrix valued scalar function. For example, $f(\mathbf{A}) = \text{trace}(\mathbf{A}) = a_{11} + a_{22}$ for the matrices $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \in \mathbb{R}^{2 \times 2}$. The gradient of the function f with respect the matrix $\mathbf{A}$ is defined as
$$\frac{df}{d\mathbf{A}} = \begin{bmatrix} \frac{df}{da_{11}} & \frac{df}{da_{12}} \\ \frac{df}{da_{21}} & \frac{df}{da_{22}} \end{bmatrix}.$$
Now, find the gradient of the following functions. Here, $\mathbf{B} \in \mathbb{R}^{n \times n}$.
   (a) $f(\mathbf{A}) = \text{trace}(\mathbf{A}^\top \mathbf{A})$
   (b) $f(\mathbf{A}) = \text{trace}(\mathbf{A}^\top \mathbf{B}\mathbf{A})$
   (c) $f(\mathbf{A}) = \text{trace}(\mathbf{B}^\top \mathbf{A})$

6. Consider the clustering problem, where given a set of elements $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$, we partition $\mathcal{X}$ into groups $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$ of similar elements. Now consider the k-means algorithm which solve this problem. In k-means, we used $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k$ as respective group representatives (centers). Then, we minimized the error $\sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{C}_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2$ to find the optimal centers. We found that the optimal groups are obtained using $\boldsymbol{\mu}_i = \frac{\sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x}}{|\mathcal{C}_i|}$. Now, consider a different variant of the k-means cost function. Let $\mathbf{C} \in \{0, 1\}^{m \times k}$ be a binary matrix such that $\mathbf{C}_{j,i} = 1$ if the j^{th} point $\mathbf{x}_j$ belongs to the i^{th} cluster C_i and $\mathbf{C}_{j,i} = 0$ otherwise. Show, that
$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{C}_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2 = \sum_{i=1}^{k} \sum_{j=1}^{m} \mathbf{C}_{j,i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|_2^2.$$
Furthermore, find the optimal center $\boldsymbol{\mu}_i$ by minimizing the cost $\sum_{i=1}^{k} \sum_{j=1}^{m} \mathbf{C}_{j,i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|_2^2$.

7. Let $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1, \dots, \mathbf{x}_m\end{bmatrix} \in \mathbb{R}^{d \times m}$ be the input data points. We can construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ on these point where vertex v_i in $\mathcal{G}$ represents the point $\mathbf{x}_i$. The weight $\mathbf{W}_{i,j}$ of the edge between vertices v_i and v_j is defined as $\mathbf{W}_{i,j} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}}$. Consider the adjacency matrix $\mathbf{W}$, degree matrix $\mathbf{D}$, and the Laplacian matrix $\mathbf{L}$ as defined in the class. In spectral clustering we try to minimize the cost $\sum_{i=1}^{k} \frac{1}{|\mathcal{C}_i|} \sum_{r \in \mathcal{C}_i} \sum_{s \notin \mathcal{C}_i} \mathbf{W}_{r,s}$. We defined the cluster assignment matrix $\mathbf{H} \in \mathbb{R}^{m \times k}$ such that $\mathbf{H}_{j,i} = \frac{1}{\sqrt{|\mathcal{C}_j|}}$ if the j^{th} point $\mathbf{x}_j$ belongs to the i^{th} cluster $\mathcal{C}_i$ and $\mathbf{H}_{j,i} = 0$ otherwise. Then, we saw that minimizing the cost $\sum_{i=1}^{k} \frac{1}{|\mathcal{C}_i|} \sum_{r \in \mathcal{C}_i} \sum_{s \notin \mathcal{C}_i} \mathbf{W}_{r,s}$ gives the same clustering if we minimize the cost $\text{Trace}(\mathbf{H}^\top \mathbf{L} \mathbf{H})$ such that $\mathbf{H}^\top \mathbf{H} = \mathbf{I}$. Now, consider that we want to minimize the cost $\sum_{i=1}^{k} \frac{1}{\sum_{v \in \mathcal{C}_i} \mathbf{D}_{v,v}} \sum_{r \in \mathcal{C}_i} \sum_{s \notin \mathcal{C}_i} \mathbf{W}_{r,s}$. Design a similar approach to optimize this cost function.

8. Let $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1, \dots, \mathbf{x}_m\end{bmatrix} \in \mathbb{R}^{d \times m}$ be the input data points. Let $\mathbf{X}\mathbf{X}^\top \mathbf{u}_i = \lambda_i \mathbf{u}_i, \forall i \in \{1, 2, \dots, d\}$ be the EVD of the matrix $\mathbf{X}\mathbf{X}^\top$. Here, we assume that the eigenvalues are such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$. Now, consider the PCA algorithm where our goal is to reduce the dimensionality of the input data points from d to k, where $k << d$. We used a compression matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$ to reduce the dimensionality and used a recovery matrix $\mathbf{U} \in \mathbb{R}^{d \times k}$ to recover the original input point and minimize the reconstruction error $\sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{U}\mathbf{W}\mathbf{x}_i\|_2^2$ to find the optimal $\mathbf{W}$ and $\mathbf{U}$. Remember that optimal $\mathbf{W} = \mathbf{U}^\top$ and $\mathbf{U} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_k\end{bmatrix}$. The time required to find the eigenvalue decomposition of a $d \times d$ matrix is of the order $O(d^3)$. Therefore, if d is very large and $m << d$, then finding the optimal $(\mathbf{W}, \mathbf{U})$ solution becomes an intractable problem. Design a better algorithm for finding the optimal $(\mathbf{W}, \mathbf{U})$. [**Hint**: can we use the EVD of $\mathbf{X}^\top \mathbf{X}$ instead of $\mathbf{X}\mathbf{X}^\top$?]