Machine Learning **(Due: 10th June)**

Assignment #5 (Dimensionality Reduction and Clustering)

Instructor: Beilun Wang *Name:* Haorui Li, *ID:* 61518407

Problem Description:

Problem 1: Principle component analysis

Consider n training sample points $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \in \mathbb{R}^p$ such that the sample mean of $\boldsymbol{x}$ is zero: $\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i = \boldsymbol{0}$.

(1) PCA: Maximizing the variance

Consider projection vector $\boldsymbol{u} \in \mathbb{R}^p$, where $\|\boldsymbol{u}\|_2^2 = 1$. We would like to maximize the sample variance $\tilde{V}[\boldsymbol{u}^\top \boldsymbol{x}]$, which is the sample variance of $\boldsymbol{x}$ projected onto $\boldsymbol{u}$. The sample variance of n samples of a variable z is $\tilde{V}[z] = \frac{1}{n}\sum_{i=1}^n z_i^2$, where $\frac{1}{n}\sum_{i=1}^n z_i = 0$. We can write an optimization problem to maximize the sample variance

$$\max_{\boldsymbol{u}} \tilde{V}[\boldsymbol{u}^\top \boldsymbol{x}] \tag{1.1}$$

Reformulate maximum problem (1.1) to the following optimization problem, and define the $p \times p$ matrix $\boldsymbol{\Sigma}$:

$$\max_{\boldsymbol{u}} \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} \tag{1.2}$$

(2) PCA: Minimizing the reconstruction error

Consider the same variables as those defined in question (1). Instead of maximizing the projected variance, we now seek to minimize the reconstruction error. In other words, we would like to minimize the difference between $\boldsymbol{x}$ and the reconstructed $\boldsymbol{u}\boldsymbol{u}^\top \boldsymbol{x}$. Note that $\boldsymbol{u}\boldsymbol{u}^\top \boldsymbol{x}$ first projects $\boldsymbol{x}$ onto $\boldsymbol{u}$, and then projects this scalar back into the p-dimensional space along the $\boldsymbol{u}$ axis. Minimizing the reconstruction error can be written as the following optimization problem.

$$\min_{\boldsymbol{u}} \frac{1}{n}\sum_{i=1}^n \|\boldsymbol{x_i} - \boldsymbol{u}\boldsymbol{u}^\top \boldsymbol{x_i}\|_2^2 \tag{1.3}$$

Reformulate minimum problem (1.3) to the following optimization problem with the same $\boldsymbol{\Sigma}$ as in the question (1).

$$\max_{\boldsymbol{u}} \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} \tag{1.4}$$

(3) Kernel PCA

Kernel PCA is a non-linear dimensionality reduction where a principle vector $\boldsymbol{u}_j$ is computed as a linear combination of training sample points in the feature space

$$\boldsymbol{u}_j = \sum_{i=1}^n \alpha_{ij} \phi(\boldsymbol{x}_i).$$

Computing the principle component of a new point $\boldsymbol{x}$ can then be done using kernel evaluations

$$z_j(\boldsymbol{x}) = \langle \boldsymbol{u}_j, \phi(\boldsymbol{x}) \rangle = \sum_{i=1}^n \alpha_{ij} \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) \rangle = \sum_{i=1}^n \alpha_{ij} k(\boldsymbol{x}_i, \boldsymbol{x}).$$

You will show that kernel PCA can be represented by a neural network. First we define a *kernel node*. A kernel node with a vector $\boldsymbol{w}_i$ of incoming weights and an input vector $\boldsymbol{x}$ computes the output $y = k(\boldsymbol{x}, \boldsymbol{w}_i)$ (The kernel function is already known). Show that, given the training sample points $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, there exits a network with a single hidden layer consisting of several kernel nodes and the output of the network is the kernel principle components $z_1(\boldsymbol{x}), ..., z_k(\boldsymbol{x})$ for a given input $\boldsymbol{x}$.

Specify the number of nodes in the input, output and hidden layers, and the weights of the edges in terms of $\alpha_{ij}, \boldsymbol{x}_1, ..., \boldsymbol{x}_n$.

Problem 2: Clustering

Given data set $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^p$, we want to partition them into k clusters $c_1, c_2, \cdots, c_k$.

(1) Under the assumption of k-means clustering, the loss function could be written as

$$\min_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in c_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2 \tag{1.5}$$

where $\boldsymbol{\mu}_i = \frac{1}{|c_i|} \sum_{\boldsymbol{x} \in c_i} \boldsymbol{x}$ is the mean vector of each cluster. Here $|c_i|$ denotes the number of points in one cluster. Prove that the loss (1.5) is equivalent to the maximum problem below:

$$\max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \sum_{\boldsymbol{x},\boldsymbol{y} \in c_i} \boldsymbol{x}^\top \boldsymbol{y} \tag{1.6}$$

(**Hint**: prove by expansion.)

(2) Further show that the problem (1.6) is equivalent to the loss below:

$$\min_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \sum_{\boldsymbol{x},\boldsymbol{y} \in c_i} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{1.7}$$

which represents the weighted within-class total distance.

(3) Reformulate problem (1.6) to the matrix form:

$$\max_{c_1,c_2,\cdots,c_k} \operatorname{tr}(\boldsymbol{H}^\top \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{H}) \tag{1.8}$$

where each entry $\boldsymbol{H}_{ij}$ in $\boldsymbol{H} \in \mathbb{R}^{n \times k}$ is subject to

$$\boldsymbol{H}_{ij} = \begin{cases} \frac{1}{\sqrt{|c_j|}}, & \boldsymbol{x}_i \in c_j \\ 0, & \text{otherwise.} \end{cases} \tag{1.9}$$

and $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)^\top$.

Answer:

Problem 1: Principle component analysis

(1) Because:

$$\begin{aligned}\tilde{V}\left[\boldsymbol{u}^T\boldsymbol{x}\right] &= \frac{1}{n}\sum_{i=1}^{n}\left[\boldsymbol{u}^T\boldsymbol{x}_i - \sum_{j=1}^{n}\boldsymbol{u}^T\boldsymbol{x}_j\right]^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\boldsymbol{u}^T\boldsymbol{x}_i\boldsymbol{x}_i^T\boldsymbol{u} - 2\boldsymbol{u}^T\boldsymbol{x}_i\sum_{j=1}^{n}\boldsymbol{x}_j^T\boldsymbol{u} + \left(\sum_{j=1}^{n}\boldsymbol{u}^T\boldsymbol{x}_j\right)^2\right] \\ &= \frac{1}{n}\boldsymbol{u}^T\sum_{i=1}^{n}\left[\boldsymbol{x}_i\boldsymbol{x}_i^T - 2\boldsymbol{x}_i\sum_{j=1}^{n}\boldsymbol{x}_j^T + \sum_{j=1}^{n}\sum_{k=1}^{n}\boldsymbol{x}_j\boldsymbol{x}_k^T\right]\boldsymbol{u} \\ &= \frac{1}{n}\boldsymbol{u}^T\left[\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T + (n-2)\sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_j^T\right]\boldsymbol{u}\end{aligned}$$

And$\sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_j^T = \boldsymbol{O}$
So we canreformulate maximum problem (1.1) to the optimization problem:

$$\max_{\boldsymbol{u}}\ \boldsymbol{u}^\top\boldsymbol{\Sigma}\boldsymbol{u} \tag{2.10}$$

$$s.t.\ u = x^T v \tag{2.11}$$

$$v^T v = 1 \tag{2.12}$$

and the *sum* matrix is:

$$\sum = \frac{1}{N}\sum_{i=1}^{N}x_i x_i^T$$

(2)

$$\begin{aligned}&\min_{\boldsymbol{u}}\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\left\|x_i - uu^T x_i\right\|_2^2 \\ &= min_{\boldsymbol{u}}\tfrac{1}{n}\textstyle\sum_{i=1}^{n}(x_i - uu^T x_i)(x_i - uu^T x_i)^T \\ &= \min_{\boldsymbol{u}}\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\left(x_i^T x_i - 2x_i^T uu^T x_i + x_i^T uu^T x_i\right)\end{aligned}$$

Due to the x is coefficient and the problem can be rewriten as:

$$\begin{aligned}&\min_{\boldsymbol{u}}\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\left\|x_i - uu^T x_i\right\|_2^2 \\ &= \max\textstyle\sum_{i=1}^{n}x_i^T uu^T x_i \\ &= \max_{\boldsymbol{u}} u^{\boldsymbol{T}}\boldsymbol{\Sigma}u\end{aligned}$$

(3) Fully-connected network:
Input: x node
Hidden layers: n kernel nodes
Output : k nodes and everyone connected to all the hidden layers nodes. The weight between the j_{th} output node and the i_{th} hidden layers node i_s hidden layers node is α_{ij}

Problem 2: Clustering

(1) Expansion formula we can get:

$$\begin{aligned}
&\min_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \sum_{\boldsymbol{x}\in c_i} \|x-\mu_i\|^2 \\
&= \min_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \sum_{\boldsymbol{x}\in c_i} (x-u_i)^T (x-u_i) \\
&= \min_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \sum_{\boldsymbol{x}\in c_i} x^T x - 2x^T u_i + u_i^T u_i \\
&= \max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \sum_{\boldsymbol{x}\in c_i} x^T u_i \\
&= \max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \left(\sum_{\boldsymbol{x}\in c_i} x^T\right) \left(\sum_{x\in c_i} x\right) \\
&= \max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \sum_{\boldsymbol{x},\boldsymbol{y}\in c_i} x^T y
\end{aligned}$$

(2)

Just do the reverse operation:

$$\begin{aligned}
&\max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \sum_{\boldsymbol{x},\boldsymbol{y}\in c_i} x^T y \\
&= min_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \sum_{\boldsymbol{x},\boldsymbol{y}\in c_i} x^T x - 2x^T y + y^T y \\
&= \min_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \sum_{\boldsymbol{x},\boldsymbol{y}\in c_i} \|\boldsymbol{x}-\boldsymbol{y}\|^2
\end{aligned}$$

(3)

Like (2):

$$\begin{aligned}
&\max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \sum_{\boldsymbol{x},\boldsymbol{y}\in c_i} \boldsymbol{x}^T \boldsymbol{y} \\
&= \max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \frac{1}{|c_i|} \left(\sum_{\boldsymbol{x}\in c_i} x^T\right) \left(\sum_{x\in c_i} x\right) \\
&= \max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \left(\sum_{\boldsymbol{x}\in c_i} \frac{1}{\sqrt{|c_i|}} x^T\right) \left(\sum_{x\in c_i} \frac{1}{\sqrt{|c_i|}} x\right) \\
&= \max_{c_1,c_2,\cdots,c_k} \sum_{i=1}^{k} \left(X^T h_i\right)^T \left(X^T h_i\right) \\
&= \max_{c_1,c_2,\cdots,c_k} \operatorname{tr}\left(\boldsymbol{H}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{H}\right)
\end{aligned}$$