# ECE 133B HW4

Lawrence Liu

June 1, 2023

## Problem 1

### (a)

If we represent the group that column $a_j$ is assigned to as a one hot vector $C_j$, and we have the a matrix $B$ with column $i$ corresponding to the $i$th centroid then we have:

$$J^{\text{clust}} = \frac{1}{n}\sum_{j=1}^{n} ||a_j - BC_j||^2$$

When we want to minimize, we can ignore the constant $\frac{1}{n}$ and we have that: we want to minimize:

$$\sum_{j=1}^{n} ||a_j - BC_j||^2$$

We can represnt this in matrix from with $C$ being a $k \times n$ matrix with one hot encoding for the columns. We can do this since if $C$ has columns that are one hot encodings, $BC$ effectively selects the column of $B$ based on the index of the one hot encoding of $C$. Therefore we can see that the minimization problem is equivalent to:

$$\min_{C,B} ||A - BC||_F^2$$

With the constraint that $C$ is a one hot encoding matrix.

## (b)

We have that

$$\begin{aligned}||A - BC||_F^2 &= \mathrm{tr}((A - BC)^T(A - BC)) \\ &= \mathrm{tr}(A^TA - C^TB^TA - A^TBC + C^TB^TBC) \\ &= \mathrm{tr}(A^TA) - \mathrm{tr}(C^TB^TA) - \mathrm{tr}(A^TBC) + \mathrm{tr}(C^TB^TBC) \\ &= \mathrm{tr}(A^TA) - 2\,\mathrm{tr}(A^TBC) + \mathrm{tr}(C^TB^TBC)\end{aligned}$$

If we take the derivative we get that we want:

$$\begin{aligned}\frac{\partial}{\partial C}\left(-2\,\mathrm{tr}(A^TBC) + \mathrm{tr}(C^TB^TBC)\right) &= 0 \\ -2B^TA + 2B^TBC &= 0 \\ B^TBC &= B^TA\end{aligned}$$

Because $B$ is not necissarily a full rank matrix, we cannot simply calculate the Moore-Penrose inverse. Rather we must SVD decompose $B = U\Sigma V^T$ and then we have that:

$$C = V\Sigma^{-1}U^TA$$

Where $\Sigma^{-1}$ simply denotes taking the reciprocal of the diagonal entries of $\Sigma$ and then taking the transpose of the resulting matrix, this takes $k$ operations.

We have that $U^T$ is of size $m \times m$ and $A$ is of size $m \times n$, and $\Sigma^{-1}$ is of size $k \times m$, and $V$ is of size $k \times k$. Therefore we have that the multiplication $U^TA$ takes $2nm^2$ operations. Since the diagonal of $\Sigma^{-1}$ is the only nonzero part of $\Sigma^{-1}$ we have that the multiplication $\Sigma^{-1}U^TA$ would simply takes only $km$ operations. Finally we have that the multiplication $V\Sigma^{-1}U^TA$ takes $2nk^2$ operations.

We also have that the SVD decomposition takes on the order of $mk^2$ operations. Therefore we have that the complexity of the algorithm is $\boxed{O(nm^2)}$ assuming that $k < m$ and $k < n$.

### (c)

We may want to use the rank-k optimization if we prioritize the speed over the accuracy of the clustering. Also with the rank-k optimization we can get weights that could be translated to some kind of "confidence" or "probability" of the clustering.

## Problem 2

### (a)

Let $A = U_1\Sigma_1 V_1^T$ and $B = U_2\Sigma_2 V_2^T$ be the SVDs of $A$ and $B$ respectively. Then we have that:

$$AA^T = U_1\Sigma_1^2 U_1^T$$

$$BB^T = U_2\Sigma_2^2 U_2^T$$

Since $V_2^T V_2 = I$ and $V_1^T V_1 = I$. We can see that these are the eigen-decompositions of $AA^T$ and $BB^T$ respectively. Therefore we can see that the eigenvalues of $AA^T$ and $BB^T$ are $\Sigma_1^2$ and $\Sigma_2^2$ respectively. Likewise the eigenvectors of $AA^T$ and $BB^T$ are $U_1$ and $U_2$ respectively. Therefore we can see that if

$$AA^T = BB^T$$

Then we have that $A$ and $B$ must have the same singular values and the same left singular vectors.

### (b)

We have that

$$A^T B = V_1\Sigma^T U^T U\Sigma V_2^T$$

$$A^T B = V_1\Sigma^T \Sigma V_2^T$$

We can see that

$$QH = V_1 V_2^T V_2 \Sigma^T \Sigma V_2^T$$

$$QH = V_1 \Sigma^T \Sigma V_2^T$$

So we have

$$A^T B = QH$$

### (c)

$$\begin{aligned} AQ &= U\Sigma V_1^T V_1 V_2^T \\ AQ &= U\Sigma V_2^T \\ AQ &= B \end{aligned}$$

## Problem 3

### (a)

The laplacian $L$ of the graph would be:

$$L = \begin{bmatrix} (n-1) & -1 & \dots & -1 \\ -1 & (n-1) & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & (n-1) \end{bmatrix}$$

### (b)

We have that

$$L1 = 0$$

Therefore 0 is an eignevalue of $L$, and since the rank of $L$ is $n-1$, we have that 0 is an eigenvalue of $L$ with multiplicity 1.
We can decompose $L$ as

$$L = (n-1)I - A$$

Where is a matrix with all ones except for the main diagonal. We can see that

$$A + I = 11^T$$

Therefore we have that $A$ has eigenvalue of $-1$ with multiplicity of $n-1$, since $11^T$ has rank of 1 and therefore has nullity of $n-1$. Therefore we have that $L$ has eigenvalue of $n$ with multiplicity of $n-1$.