Q1

All these methods (PCA,CCA, Random Projection) work for a specific K. Ideally, we want K as small as possible. So, we should find a way to penalize space usage or encourage space savings (compared to the uncompressed original version) for dimensionality reduction.

PCA PCA's objective is to maximize the variance (minimize the reconstruction error) of the projections for a given K. We need to have a trade-off in the selection of K between the maximization of variance and space savings. The easiest way to accomplish this is to do the following:

$$\mathop{\mathbf{argmax}}_{K\in[1,d]} Variance[Projection] + Space[Original] - Space[Projection]$$

Notice that this approach is not scale-free. If we scale the numbers by a factor of 2 the Variance is going to increase but the Space Savings are going to remain the same (same K). In order to account for this, we need to use a normalized version:

$$\mathop{\mathbf{argmax}}_{K\in[1,d]} \frac{Variance[Projection]}{Variance[Original]} + \frac{Space[Original] - Space[Projection]}{Space[Original]}$$

Finally, we can make this more generic by using a variable $\alpha \in (0,1)$ such that:

$$\mathop{\mathbf{argmax}}_{K\in[1,d]} \alpha\frac{Variance[Projection]}{Variance[Original]} + (1-\alpha)\frac{Space[Original] - Space[Projection]}{Space[Original]}$$

For PCA (according to slides of Lecture 3), this becomes as follows:

$$\mathop{\mathbf{argmax}}_{K\in[1,d]} \frac{\alpha\sum_{i=1}^{K}\lambda_i}{\sum_{i=1}^{d}\lambda_i} + (1-\alpha)\frac{d-K}{d}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ are the eigenvalues of the covariance matrix and d is the original number of dimensions.

Thus, when you obtain the eigenvalues in a sorted order, you can start from λ_1 and you can stop at λ_K when the objective has stopped increasing (Intermediate step in the existing algorithm).

Notice, that if you assign $\alpha = 0$, you completely tune for size efficiency and K would always be 0. If you set $\alpha = 1$, you completely tune for variance maximization and K would always be d. If you assign $\alpha = 0.5$ then the heuristic will set K up to the point where an eigenvalue contributed more to maximizing the variance than the average eigenvalue contribution for this case, which seems more sensible. Finally, increasing α from this point means that you care less about the space and more about the accuracy of the results, while decreasing it means that you care more about space efficiency and less about accuracy (it depends in the application).

CCA Similarly with PCA we would try to have the following objective (according to slide of Lecture 4):

$$\mathop{\mathbf{argmax}}_{K\in[1,d]} \frac{\alpha\sum_{i=1}^{K}\lambda_i^2}{\sum_{i=1}^{d_1+d_2}\lambda_i^2} + (1-\alpha)\frac{d_1+d_2-K}{d_1+d_2}$$

The trade-off here is between maximizing the correlation between two views vs maximizing the space savings. Similarly with PCA, when you obtain the eigenvalues in a sorted order, you can start from λ_1 and you can stop at λ_K when the objective has stopped increasing (Intermediate step in the existing algorithm). Again, $\alpha = 0.5$ seems like a sensible choice.

RP The purpose of Random Projection is to maintain the distances between two points in the compressed version similar to the corresponding ones in the original version. Thus (according to slides of Lecture 5), we know that for any $\epsilon > 0$, if $K = \frac{\log(n/\delta)}{\epsilon^2}$, with probability $1 - \delta$, for all pairs of data points $i, j \in \{1, \dot{;}n\}$ we have:

$$(1-\epsilon)\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2 \leq \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \leq (1+\epsilon)\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$$

With similar strategy as before we should try to do the following:

$$\underset{K\in[1,d]}{\mathbf{argmin}}\ \alpha \frac{\epsilon^2}{\epsilon[K=1]^2} + (1-\alpha)\,\frac{Space[Projection]}{Space[Original]}$$

which leads to:

$$\underset{K\in[1,d]}{\mathbf{argmin}}\ \alpha \frac{1}{K} + (1-\alpha)\,\frac{K}{d}$$

For $\alpha = 0.5$, this happens when $K = \sqrt{d}$.

Q2

We have:

$$
\begin{aligned}
M_2(C_1, C_2, \ldots, C_k) =& \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} \|\boldsymbol{x}_s - \boldsymbol{x}_t\|_2^2 \\
=& \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} \|\boldsymbol{x}_s - \boldsymbol{r}_j + \boldsymbol{r}_j - \boldsymbol{x}_t\|_2^2 \\
=& \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} \|(\boldsymbol{x}_s - \boldsymbol{r}_j) - (\boldsymbol{x}_t - \boldsymbol{r}_j)\|_2^2 \\
=& \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} [(\boldsymbol{x}_s - \boldsymbol{r}_j) - (\boldsymbol{x}_t - \boldsymbol{r}_j)] \cdot [(\boldsymbol{x}_s - \boldsymbol{r}_j) - (\boldsymbol{x}_t - \boldsymbol{r}_j)] \\
=& \sum_{j=1}^{K} \frac{1}{|C_j|} \left[\sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} \|\boldsymbol{x}_s - \boldsymbol{r}_j\|_2^2 + \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} \|\boldsymbol{x}_t - \boldsymbol{r}_j\|_2^2 - 2 \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} (\boldsymbol{x}_s - \boldsymbol{r}_j) \cdot (\boldsymbol{x}_t - \boldsymbol{r}_j) \right] \\
=& \sum_{j=1}^{K} \frac{1}{|C_j|} \left[\sum_{\boldsymbol{x}_s \in C_j} |C_j| \|\boldsymbol{x}_s - \boldsymbol{r}_j\|_2^2 + |C_j| \sum_{\boldsymbol{x}_t \in C_j} \|\boldsymbol{x}_t - \boldsymbol{r}_j\|_2^2 - 2 \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} (\boldsymbol{x}_s - \boldsymbol{r}_j) \cdot (\boldsymbol{x}_t - \boldsymbol{r}_j) \right] \\
=& \sum_{j=1}^{K} \frac{1}{|C_j|} \left[|C_j| \sum_{\boldsymbol{x}_s \in C_j} \|\boldsymbol{x}_s - \boldsymbol{r}_j\|_2^2 + |C_j| \sum_{\boldsymbol{x}_t \in C_j} \|\boldsymbol{x}_t - \boldsymbol{r}_j\|_2^2 - 2 \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} (\boldsymbol{x}_s - \boldsymbol{r}_j) \cdot (\boldsymbol{x}_t - \boldsymbol{r}_j) \right] \\
=& \sum_{j=1}^{K} \frac{1}{|C_j|} \left[2|C_j| \sum_{\boldsymbol{x}_t \in C_j} \|\boldsymbol{x}_t - \boldsymbol{r}_j\|_2^2 - 2 \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} (\boldsymbol{x}_s - \boldsymbol{r}_j) \cdot (\boldsymbol{x}_t - \boldsymbol{r}_j) \right] \\
=& 2 \sum_{j=1}^{K} \sum_{\boldsymbol{x}_t \in C_j} \|\boldsymbol{x}_t - \boldsymbol{r}_j\|_2^2 - 2 \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} (\boldsymbol{x}_s - \boldsymbol{r}_j) \cdot (\boldsymbol{x}_t - \boldsymbol{r}_j) \\
=& 2M_1(C_1, C_2, \ldots, C_k) - 2 \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} (\boldsymbol{x}_s - \boldsymbol{r}_j) \cdot (\boldsymbol{x}_t - \boldsymbol{r}_j) \\
=& 2M_1(C_1, C_2, \ldots, C_k) - 2 \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{\boldsymbol{x}_s \in C_j} \sum_{\boldsymbol{x}_t \in C_j} (\boldsymbol{x}_s \cdot \boldsymbol{x}_t - \boldsymbol{x}_s \cdot \boldsymbol{r}_j - \boldsymbol{r}_j \cdot \boldsymbol{x}_t + \boldsymbol{r}_j \cdot \boldsymbol{r}_j) \\
=& 2M_1(C_1, C_2, \ldots, C_k) - 2 \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{\boldsymbol{x}_s \in C_j} \left(\boldsymbol{x}_s \cdot \sum_{\boldsymbol{x}_t \in C_j} \boldsymbol{x}_t - |C_j| \boldsymbol{x}_s \cdot \boldsymbol{r}_j - \boldsymbol{r}_j \cdot \sum_{\boldsymbol{x}_t \in C_j} \boldsymbol{x}_t + |C_j| \boldsymbol{r}_j \cdot \boldsymbol{r}_j \right) \\
=& 2M_1(C_1, C_2, \ldots, C_k) - 2 \sum_{j=1}^{K} \sum_{\boldsymbol{x}_s \in C_j} \left(\boldsymbol{x}_s \cdot \frac{1}{|C_j|} \sum_{\boldsymbol{x}_t \in C_j} \boldsymbol{x}_t - \boldsymbol{x}_s \cdot \boldsymbol{r}_j - \boldsymbol{r}_j \cdot \frac{1}{|C_j|} \sum_{\boldsymbol{x}_t \in C_j} \boldsymbol{x}_t + \boldsymbol{r}_j \cdot \boldsymbol{r}_j \right) \\
=& 2M_1(C_1, C_2, \ldots, C_k) - 2 \sum_{j=1}^{K} \sum_{\boldsymbol{x}_s \in C_j} (\boldsymbol{x}_s \cdot \boldsymbol{r}_j - \boldsymbol{x}_s \cdot \boldsymbol{r}_j - \boldsymbol{r}_j \cdot \boldsymbol{r}_j + \boldsymbol{r}_j \cdot \boldsymbol{r}_j) \\
=& 2M_1(C_1, C_2, \ldots, C_k)
\end{aligned}
$$

which finishes our case.

Q3

Before we go into the proof we should do the following observation:

$$\forall c_1, c_2, \ldots, c_n \,.\, |\{(i,j) \in E \mid c(i) = c(j)\}| + |\{(i,j) \in E \mid c(i) \neq c(j)\}| = |E| \tag{1}$$

Now we have:

$$\begin{aligned}
\underset{c_1,c_2,\ldots,c_n\in[1,K]}{\textbf{argmin}} |\{(i,j) \in E \mid c(i) \neq c(j)\}| &\overset{(1)}{=} \underset{c_1,c_2,\ldots,c_n\in[1,K]}{\textbf{argmin}} |E| - |\{(i,j) \in E \mid c(i) = c(j)\}| \\
&= \underset{c_1,c_2,\ldots,c_n\in[1,K]}{\textbf{argmin}} -|\{(i,j) \in E \mid c(i) = c(j)\}| \\
&= \underset{c_1,c_2,\ldots,c_n\in[1,K]}{\textbf{argmax}} |\{(i,j) \in E \mid c(i) = c(j)\}|
\end{aligned}$$

This finishes our proof. The second equality holds from the fact that E is given and $|E|$ is a constant independent of the clustering. The third equality suggests that minimizing a negative quantity is equivalent to maximizing its absolute value.

We can find an approximate solution by utilizing the Unnormalized Spectral Clustering Algorithm (see slides of Lecture 11).

Another approach is to find the min-cut in the graph (polynomial time), then find the min-cut in the resulting clusters (run min-cut for each one of them and select the cut that is minimum) and proceed like this until you reach K clusters. This greedy approach is not optimal (approximate solution again).