**Problem 1: Concept Review**

1. Cross validation is a technique we’ve learnt in class that is used to assess the performance and generalizability of a model by training and evaluating it on multiple subsets of the dataset. The goal is to check how well the model performs on data it hasn’t seen during training, helping to identify potential issues like overfitting. We can use cross validation to validate clustering results. First split the dataset into k folds and perform clustering on k-1 folds (training data) and validate the results on the remaining fold (test data). Repeat this process k times, each time using a different fold as the testing set. For each iteration, evaluate the clustering performance. Calculate the average performance across all iterations to get a more robust measure of the clustering algorithm’s performance. If clusters are consistently identified in multiple iterations, it adds confidence to the validity of the clustering.
2. *(i)* There isn’t enough information to determine that. Looking at an example, if d(1,4)=2, d(1,5)=3, d(2,4)=1, d(2,5)=3, d(3,4)=4 and d(3,5)=1, then the single linkage dissimilarity between clusters {1,2,3} and {4,5} would be 1 and the complete linkage dissimilarity between clusters {1,2,3} and {4,5} would be 4. Then with single linkage, it would fuse at a height of 1 while complete linkage would fuse at a height of 4. However, if all inter-observations distances are 2, then the single and complete linkage dissimilarities between clusters {1,2,3} and {4,5} are 2.

*(ii)* These would fuse at the same height. If d(5,6)=2, the single and complete linkage dissimilarities between clusters {5} and {6} would be 2 and will fuse at the height of 2 for both single and complete linkage.

1. *(i)* Code in RScript.

PVE: 0.62006039, 0.24744129, 0.08914080, 0.04335752

*(ii)* Code in RScript.

PVE: 0.62006039, 0.24744129, 0.08914080, 0.04335752

1. *(i)* (XTX)-1 = [(UDVT)T(UDVT)]-1 = [(VDTUT)(UDVT)]-1 = [VD2VT]-1 = VD-2VT

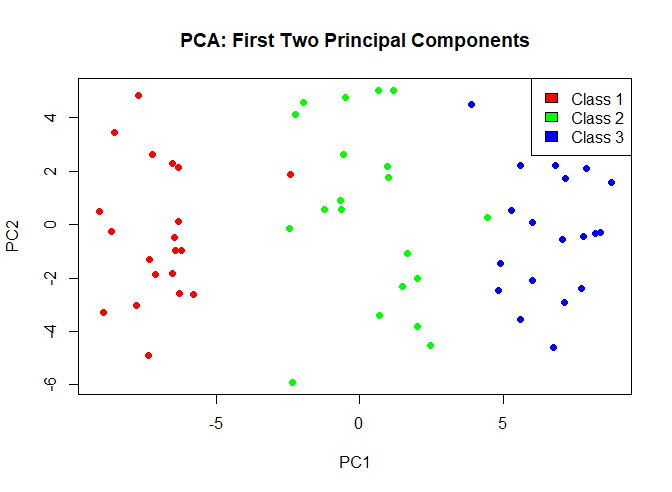
*(ii)* X(XTX)-1XT = UDVT((UDVT)TUDVT)-1(UDVT)T = UDVT(VDDVT)-1(UDVT)T = UDVT(VT)-1D-1V-TUT = UDVTVTD-1V-TUT = UDD-1UT = UUT

*(iii)* B̂ = (XTX)-1XTY = (VDUTUDVT)-1VDUTY = (VD2VT)-1VDUTY = VD-2VTVDUTY = VD-2DUTY = VUTY

*(iv)* The SVD allows for a more numerically stable and efficient computation of B̂ compared to direct matrix inversion, especially for large datasets. The decomposition provides a structured representation of the original matrix X, allowing for a more efficient computation of the inverse of XTX as VD-2VT. This decomposition-based approach leverages the orthogonality and diagonal structure of U, V, D, leading to optimized matrix operations and enhanced computational scalability. The lm() function’s utilization of matrix decomposition contributes to its efficiency in handing large datasets, making it a scalable and robust tool for linear regression analysis.

**Problem 2: Simulations for Unsupervised Learning**

* 1. Code in RScript.
  2. PVE: 0.2789121
  3. PCA identifies directions in the data, called principal components, along which the variability is maximized. While it’s true that each of the 50 features in our dataset comes from a normal distribution with a mean shift, PCA is not concerned with the individual features but rather with capturing the overall variability in the data. In our case, although all features are equally interesting, the principal components are combinations of these features that capture the most variation. It isn’t about singling out specific features as “interesting” but about finding the most informative directions in the entire feature space. The first few principal components often account for a significant proportion of the total variance because they capture the dominant patterns in the data, allowing for dimensionality reduction without losing much information. So, even though all features are equally valid, the variance is not evenly distributed among them, and PCA efficiently captures the most significant sources of variability.
  4. Plot of first two principal component score vectors:



* 1. K-means clustering with K=3.

A black background with white text

Description automatically generated

For true class 1, 19 observations are assigned to cluster 1. 1 observation is assigned to cluster 2. 0 observations assigned to cluster 3.

For true class 2, 2 observations assigned to cluster 1. 17 observations assigned to cluster 2. 1 observation assigned to cluster 3.

For true class 3, 0 observations assigned to cluster 1. 1 observation assigned to cluster 2. 19 observations assigned to cluster 3.

The diagonal elements represent correct assignments (e.g., 19 observations from true class 1 are correctly assigned to cluster 1). Off-diagonal elements indicate misclassifications (e.g., 2 observations from true class 2 are assigned to cluster 1).

* 1. K-means clustering with K=2

A black screen with white text

Description automatically generated

For true class 1, all 20 observations are assigned to cluster 1.

For true class 2, 9 observations assigned to cluster 1, 11 observations are assigned to cluster 2.

For true class 3, all 20 observations are assigned to cluster 2.

* 1. K-means clustering with K=4

A black background with white text

Description automatically generated

For true class 1, 19 observations assigned to cluster 2, 1 observation assigned to cluster 3.

For true class 2, 1 observation assigned to cluster 1, 2 observations assigned to cluster 2, 12 observations assigned to cluster 3, 5 observations assigned to cluster 4.

For true class 3, all 20 observations are assigned to cluster 1.

The results suggest that with K=4, K-means attempts to find additional substructures in the data, resulting in more clusters.

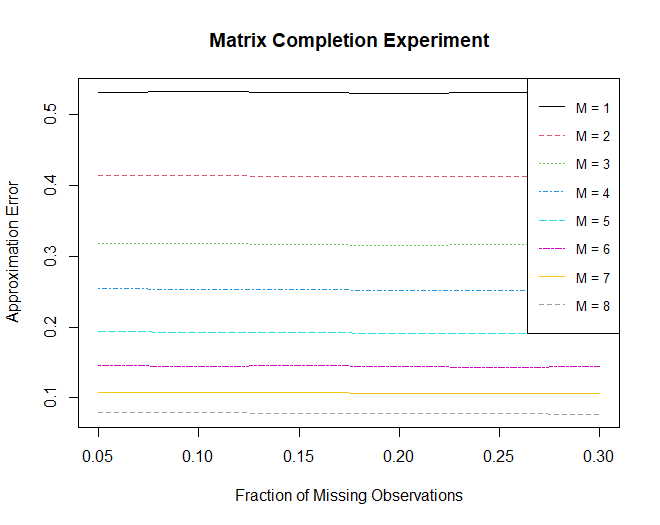
* 1. K-means clustering with K=3

A black background with white text

Description automatically generated

This table shows a different clustering pattern compared to using the raw data. In this case, the clusters appear to be more balanced, with each true class distributed across multiple clusters. True class 1 has 1 observation in cluster 1 and the majority in cluster 2. True class 2 shows a split between clusters 1, 2, and 3, indicating a more nuanced assignment compared to the raw data clustering. True class 3 is mostly assigned to clusters 1 and 3. This suggests that utilizing the first two principal component score vectors as input for K-means captures a different aspect of the data’s variability compared to using the raw data directly. The choice of representation can impact the clustering results, emphasizing different dimensions of the data’s structure.

**Problem 3: Matrix Completion**



The plot suggests that using a higher value of M in the matrix completion algorithm leads to a more accurate approximation of the missing values. In the context of SVD, increasing the value of M allows the algorithm to capture more information from the original matrix. When we set M=8, the algorithm can better fill in the missing values, resulting in a lower approximation error.

**Problem 4: Hierarchical Clustering and Classification**

1. Results depend on the type of linkage used. Left to right: Complete, Single, Average

A diagram of a cluster

Description automatically generated A diagram of a cluster

Description automatically generated

A diagram of a cluster

Description automatically generated

1. While logistic regression, LDA and QDA can be applied, it is important to be aware of the assumptions and potential challenges.

Logistic regression: Assumes that the relationship between the independent variable (gene measurements) and the log-odds of the dependent variable (disease status) is linear. If the relationship is highly nonlinear, logistic regression may not perform well.

LDA: Assumes that the data within each class follows a multivariate normal distribution and that the covariance matrices are equal across the classes. If these assumptions are violated, LDA may not provide accurate results.

QDA: Similar assumptions to LDA but relaxes the assumption of equal covariance matrices across the classes.

These methods might fail as with 1000 genes as predictors, we might have an issue with dimensionality, where the number of predictors is large compared to the number of samples. This can lead to overfitting and difficulties in estimating parameters. If the underlying assumptions of the methods are violated, the models may not perform well. And if there is high collinearity among the genes, it can affect the interpretability of the models.

1. The approach taken combines hierarchical clustering, PCA and logistic regression to classify samples based on the gene expression data. Hierarchical clustering is applied to identify potential clusters in the dataset. The resulting clusters are used as a categorical response variable for logistic regression. To handle the high dimensionality of the data, PCA is used for dimensionality reduction. The principal components are used as predictors in the logistic regression model. The misclassification rate is calculated based on the confusion matrix, indicating the accuracy of the model in predicting the clusters.

Confusion matrix:

A screen shot of a computer

Description automatically generated

Misclassification rate: 0.99421

The high misclassification rate suggests that the current model may not perform well in accurately classifying samples into the specific clusters.

Top of Form