Examining Performance of Unsupervised Learning and Dimensionality Reduction Algorithms

*Zhi Li (zli3037@gatech.edu)

Abstract: **The report examines clustering and dimensionality reduction techniques applied to two datasets. Steps include clustering application, dimensionality reduction determination, utilization of reduced dimensions for clustering, neural network reconfiguration, and generation of new inputs by merging original features with cluster labels for subsequent neural network refitting. Then the performances of each step are compared. This report finds another effective to improve supervised learning which is to add cluster labels generated from unsupervised learning.**

I. INTRODUCTION

The report examines the two clustersing techniques and four dimension reduction techniques. The overall steps are as follows:

1) Apply clustering to the two datasets.
2) Conduct four dimensionality reduction techniques on each dataset, determining the optimal number of components to retain.
3) Employ the output from the second step as input for clustering on each dataset.
4) Reconfigure neural networks (NNs) using all the different numbers of transformed components generated from the four dimensionality reduction methods.
5) Generate a new input by combining original features with cluster labels. Subsequently, refit neural networks.

This report uses the same dataset from A1, the corresponding descriptions for the two datsets can be found in A1.

II. SELECTIED ALGORITHMS

A. Clustering Algorithms(step 1)

- Hierarchical Clustering, specifically Agglomerative Clustering, progressively merges the most similar data points or clusters until a hierarchy of clusters is formed, without requiring the number of clusters to be specified beforehand.
- Gaussian Mixture Models (GMM) is a probabilistic clustering algorithm that assumes that the data points are generated from a mixture of Gaussian distributions. Each Gaussian distribution represents a cluster in the data, and the GMM algorithm estimates the parameters of the Gaussians to fit the data.

B. Dimension Reduction Algorithms

- PCA, or Principal Component Analysis, is a dimensionality reduction technique that transforms data into a lower-dimensional space by identifying the most informative combinations of features, known as principal components while retaining the maximum variance in the data.
- ICA, or Independent Component Analysis, is a dimensionality reduction method that separates a multivariate signal into additive subcomponents, each representing statistically independent sources of variation, allowing for the identification of underlying patterns or signals.
- Randomized Projections is a dimensionality reduction technique that maps high-dimensional data to a lower-dimensional space by randomly selecting a set of projections, preserving the pairwise distances between data points with high probability.
- t-SNE, or t-distributed Stochastic Neighbor Embedding, is a manifold learning algorithm that aims to visualize high-dimensional data by representing each data point in a lower-dimensional space while preserving the local structure and pairwise similarities between data points.

III. RESULT COMPARISON

A. Clustering on Original Data

The author uses silhouette score and dendrogram to examine the performance of two clustering algorithms. Note only the plots with best sillhouette score is attached to reports. Other plots for different number of clusters can be found via author's github repository.

As shown in Fig 1, the dendrogram illustrates hierarchical similarity relationships, with evident cluster distinctions observed for the first DAC dataset when utilizing Euclidean distance thresholds of 15 and 20, resulting in three and two clusters, respectively. The silhouette score serves as a measure of the quality and compactness of the clusters, with higher scores indicating better-defined clusters and greater separation between them. In this context, the silhouette score reflects the degree of coherence within clusters and the degree of separation between clusters. Only the optimal outcomes for two clustering algorithms are highlighted in the silhouette plots, revealing that the optimal number of clusters for both algorithms is two. However, the scores are notably low, with values of 0.18 for Agglomerative clustering and 0.16 for GMM, indicatingthe two algorithms perform the same.

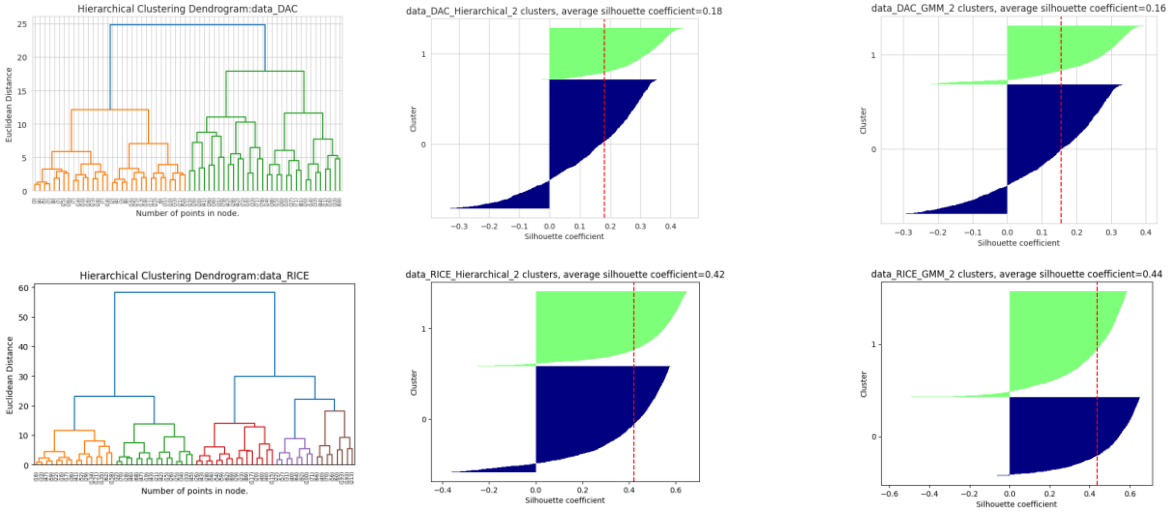

Fig. 1 Clustering Results Using Orginal Features with Best Silhouette Scores for two datasets

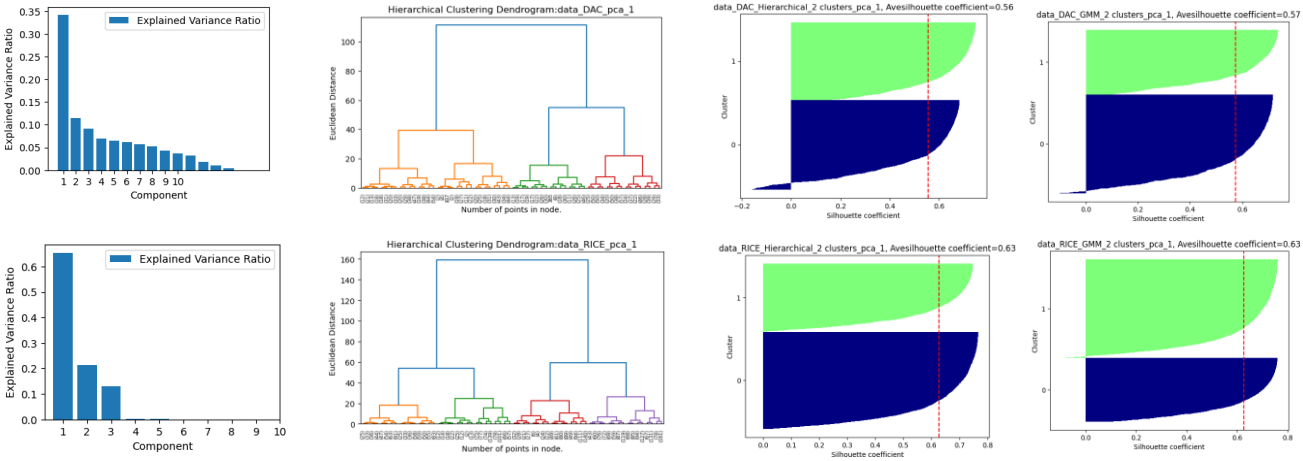

Fig. 2 PCA + Clustering Results Using Orginal Features with Best Silhouette Scores for two datasets

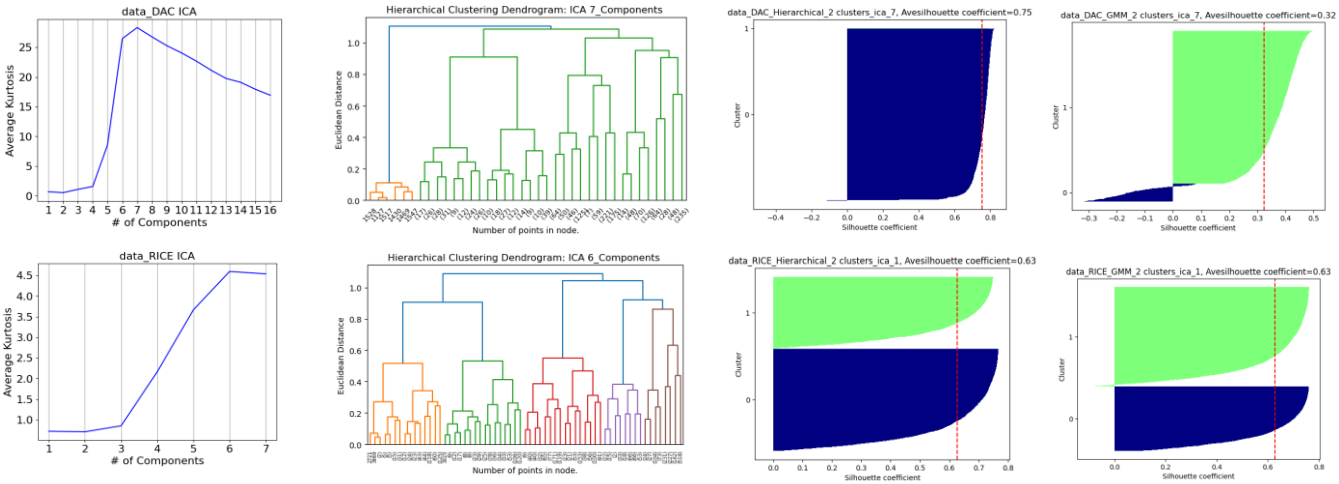

Fig. 3 ICA + Clustering Results Using Orginal Features with Best Silhouette Scores for two datasets

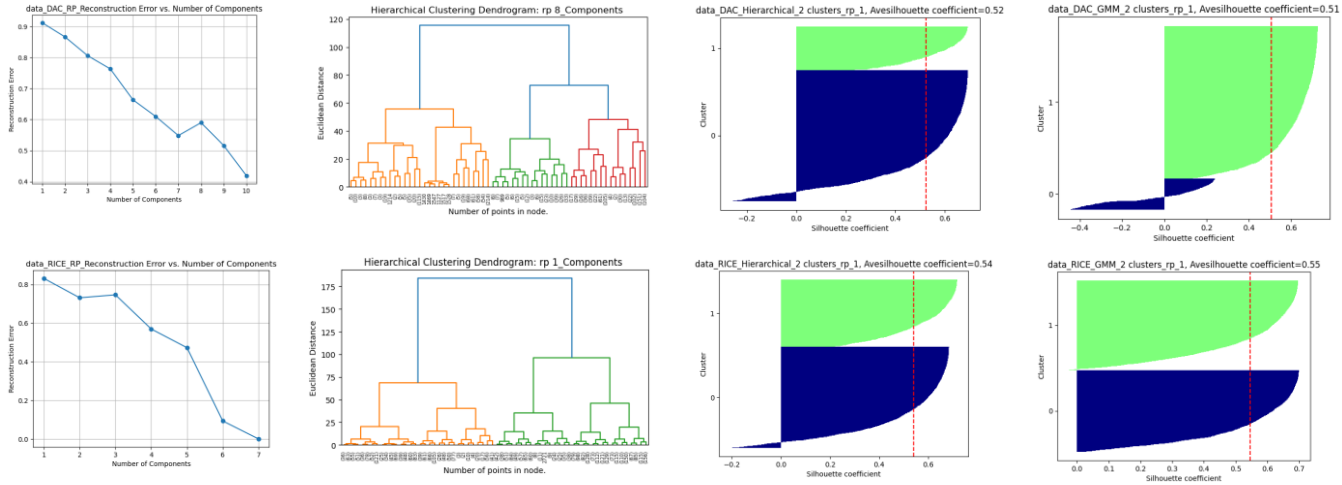

Fig. 4 RP + Clustering Results Using Orginal Features with Best Silhouette Scores for two datasets

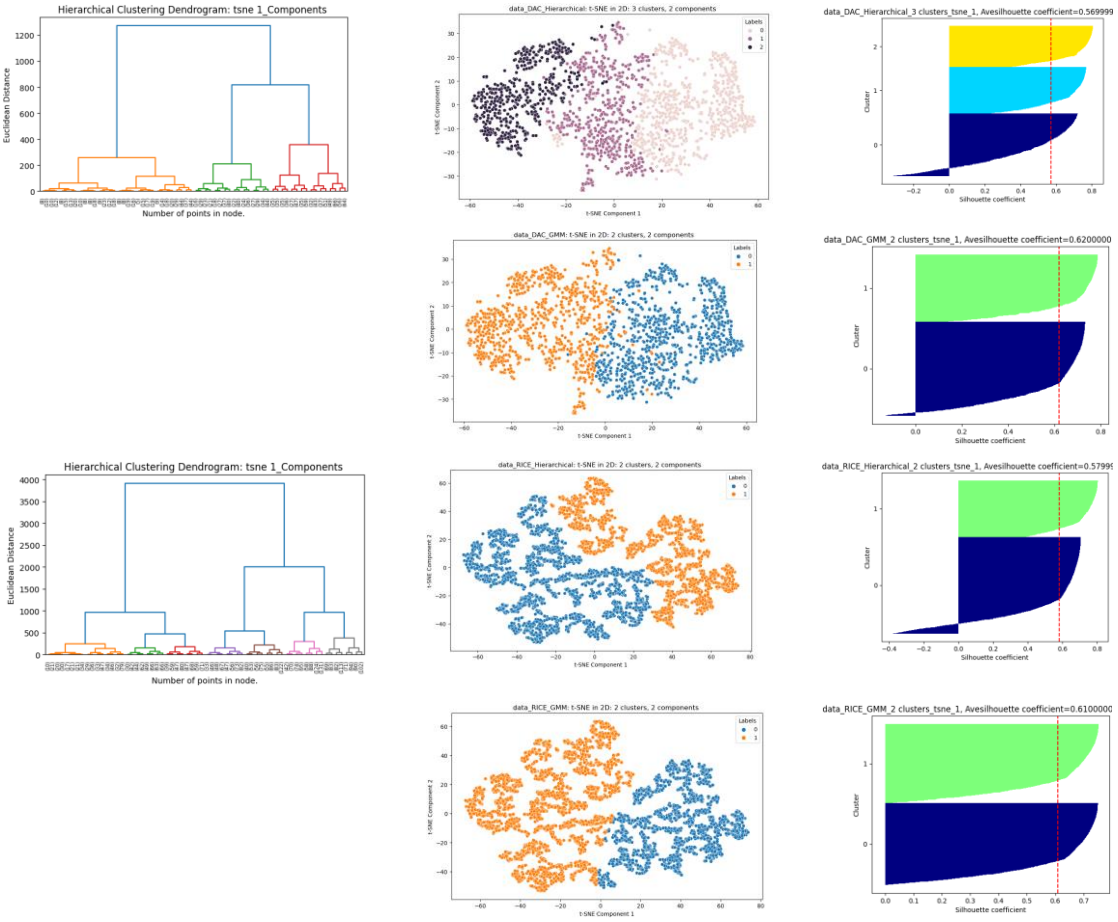

Fig. 5 TSNE + Clustering Results Using Orginal Features with Best Silhouette Scores for two datasets

Table 2 Comparison results for steps 4 and 5							
Step 4	Step4 with all converted features		Step 5	Step5 with Hierarchical labels		Step5 with GMM labels	
	f1-score			f1-score		f1-score	
	Train	Test		Train	Test	Train	Test
DAC_PCA	0.91	0.90	DAC+new cluster label	0.91	0.90	0.91	0.90
DAC_ICA	0.85	0.86					
DAC_RP	0.91	0.90					
DAC_TSNE	0.90	0.88					

Similarly, for the RICE dataset, employing Euclidean distance thresholds of 15 and 20 results in silhouette scores higher than those obtained for the DAC dataset. This improvement can be attributed to the RICE dataset's relative simplicity, characterized by its smaller feature set of only 6 features compared to DAC's more than 20 features.

The similarity in results between the two algorithms can be attributed to the original features of the data, which may present a "messy" or complex underlying structure that is challenging for both algorithms to effectively capture and differentiate.

B. 4 Types of Dimension Reduction Algorithms(step2+3)

In this section, the four DR algorithms are conducted followed by two clustering algorithms with the best numbers of components

Fig. 2 shows how PCA performs in two datasets. Firstly, we can see from the explained variance ratio plots, that for both datasets, the 1st principal component is the largest one, and it is significantly larger than other components. It indicates that the 1st principal component contains the most important information or variability within the data, making it a crucial factor in capturing the underlying structure or patterns present in the datasets. Moreover, based on the Elbow method, only the first principal components are retained for the following clustering analysis.

Surprisingly, the results improved a lot. For the DAC dataset, it increased by 250%. For RICE data, it increased by 40%. The difference is attributed to the data points' structures. DAC is larger and has more features/dimensions. High-dimensional data often contains irrelevant or noisy features that can obscure the underlying structure. Dimensionality reduction techniques such as PCA help filter out noise by focusing on the most informative features, namely the first principal component. This helps find the underlying structures

Figure 3 illustrates the performance of Independent Component Analysis (ICA) across two datasets. Initially, the average kurtosis plots indicate higher kurtosis values for both datasets, suggesting greater non-Gaussianity, which often signifies meaningful signal sources. Therefore, selecting the number of components with sufficiently high non-Gaussianity is preferable. Accordingly, for the DAC dataset, 7 components are chosen, while for the RICE dataset, 6 components are selected.

As seen in Fig 3, even though it achieves the highest score of 0.75. it is not a good result since disadvantaged communities shouldn't be that less. It looks like ICA clusters some outlier as a single cluster. ICA is sensitive to noise.

We can see from the Fig 3 that the ICA's results are worse than PCA. It is because ICA is sensitive to noise in the data, as it assumes that the non-Gaussian sources of variation represent meaningful signal sources. As indicated in Fig 1, DAC data contains significant noise or irrelevant features, ICA was struggled to separate signal from noise effectively.

Since ICA explicitly aims to separate signal from noise by assuming that the sources of variation in the data are statistically independent. This makes ICA more effective in separating meaningful signal sources from noise components. However, ICA's performance in those two datesets are affected becasue the noise in the datasets violates its assumptions, that is, being non-Gaussian or correlated with the signal sources.

Fig. 4 shows how Randomized Projection performs in two datasets. Firstly, we can see from the reconstruction error plots for both datasets, it doesn't have an elbow. So I use just one component. It shows the results are as good as PCA due to its ability to preserve pairwise distances between data points, making it suitable for capturing the local structure of the data. These factors enable Randomized Projection to achieve results similar to PCA while offering advantages such as faster computation and greater adaptability to different types of datasets.

Fig. 5, it shows the t-SNE's performance on each dataset with two clustering algorithms. To assess the quality of the t-SNE embedding, typically we should do it visually or by evaluating clustering performance. It is different from other methods like PCA or ICA. To visualize the performance, we see the clusters are well separated, with very less datapoints blended together. It is also shown that it has the best silhouette plots with the high average score, the clusters are more uniform with fewer negative scores.

This is because t-SNE preserves the local structure of the data when reducing its dimensionality. For instance, in the case of the DAC dataset, t-SNE focuses on identifying similar communities and keep them close in the low dimension. In this way, t-SNE effectively reveals local structures and clusters, when visualizing high-dimensional datasets in a lower-dimensional space.

C. Step 4:refit NNs on new input space from step 2

Table 1 shows the results from A1 as the baseline. Training acuracy is 0.90, Test acuracy is 0.85. one thing should be noted here is that, in A1, I already conducted PCA and only retain 1st principal component.

Table 1：Result from A1		
	f1-score	
	Train	Test
A1	0.90	0.85

So it is not surprised the PCA results are similar to A1, the improvements are slightly increased train accuray, and allievate overfitting further. Because PCA in A3 step 4, it keeps all the components, so have more features and reduced the overfitting.

For ICA, it has worse acuracy compared to A1 and other algorithms in A3. This is because ICA is sensitive to noise in the data, as mentioned in Section A, DAC data contains significant noise or irrelevant features, ICA was struggled to separate signal from noise effectively since the noise in the dataset are not aligned well with its Gaussian assumption.

For Randomized projection, it achieves the same performance as PCA. Similar to PCA, it preserves the pairwise distances between data points. Despite utilizing random projections, Randomized Projection efficiently reduce the dimensionality of data while maintaining its essential structure, resulting in accurate representations. Moreover, the robustness of Randomized Projection to noise

and outliers allows it to filter out irrelevant features effectively, that is the reason why it outperforms the ICA.

For t-SNE, it achieve the same training accuracy as PCA and RP, but lower testing accuracy. This is attributed to the following reasons: 1) t-SNE focuses on preserving local relationships between nearby data points, which can lead to more intricate and detailed embeddings. 2) the datasize is relatively small, with smaller datasets, it may be more prone to overfitting due to the limited amount of data available for learning the manifold structure.

D. Step 5 refit NNs on new input space from step 1

In the step 5, we respectively, add cluster labels as new features to the original DAC dataset.

It is supremely good and robust. Both reach the best training and testing accuracy with almost no overfitting. It may be attributed to the following reasons: 1) Adding cluster labels as new features provides additional discriminative information to the dataset, so the model can better differentiate between classes during training and testing. Secondly, the addition of cluster labels can act as a form of regularization, it helps mitigate overfitting by providing additional constraints to the model. Thirdly, the model can learn more meaningful representations of the data that capture both the original features and the clustering structure. Lastly, cluster labels can help the model focus on relevant patterns in the data while ignoring noisy or irrelevant features.

IV. CONCLUSION

After examining the performance of unsupervised learning and dimensionality reduction algorithms. The conclusions can be drawn as follows:

It presents another effective to improve supervised learning is to add cluster labels generated from unsupervised learning.

For dimension reduction, PCA excels in capturing global structures, while t-SNE is proficient in visualizing local structures and clusters. RP may be more robust to noise, while ICA is more sensitive to data distributions.

Independent Component Analysis (ICA) may excel in detecting noise or outliers within the data, while t-distributed Stochastic Neighbor Embedding (t-SNE) is often preferred for visualization tasks. Randomized Projection (RP) may be the method of choice for large-scale dimensionality reduction, whereas Principal Component Analysis (PCA) serves well in capturing overall variance and simplifying complex datasets.

For clustering algorithms, Gaussian Mixture Models (GMM) are flexible, assuming data generation from Gaussian mixtures, capturing diverse clusters. Hierarchical clustering offers interpretability through dendrograms, aiding understanding. GMM scales better for large datasets, while hierarchical clustering may create clusters of similar sizes, potentially misrepresenting data.

REFERENCES

[1] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[2] Mitchell, T.M. and Tom, M. (1997) Machine Learning. McGraw-Hill, New York.