CS-7641
Unsupervised Learning and Dimensionality Reduction (Assignment #3)

Yin, Changyong

March 26, 2023

Introduction

In this section, we simply introduce the selected datasets (audit risk and personal loan datasets[1], unsupervised learning algorithms (k-means and expectation maximization algorithms), and dimensionality reduction algorithms (PCA, ICA, randomized projections and SVD) used in this report.

Datasets

Datasets tested in this assignment is the same as the ones in the assignment #1. As mentioned in assignment #1, both datasets come from the online community platform Kaggle.

Audit Risk Dataset (Kaggle Link)

The objective of this study is to assist auditors by constructing a classification model that can forecast fraudulent firms based on present and past risk factors.

Specifically, this dataset covered totally 772 firms, whose sector distribution are Irrigation(114), Public health(77), Building and Roads(82), Forest(70), Corporate(47), Animal husbandry(95), Communication(1), Electrical(4), Land(5), Science and Technology(3), tourism(1), Fisheries(41), Industrial(37) and Agriculture(200). And for each firm, there are 26 attributes (risk factors) are captured. Risk factors are evaluated and their probability of occurrence is calculated from both current and historical records, with each risk factor being represented by a numerical value. Moreover, 468 out of the 772 firms are not fraudulent firms.

Based on the characteristics of the data, it seems that this will be an interesting problem. One challenge is the correlations between the attributes presented in the data since correlation could introduce inaccuracies on the results of fro some algorithms.

Personal Loan Dataset (Kaggle Link)

The dataset originates from a Thera Bank initiative aimed at converting liability customers into personal loan customers while retaining them as depositors. The campaign resulted in a successful conversion rate of over 9%, which has motivated the retail marketing department to create more effective targeted marketing campaigns with a minimal budget to enhance the success rate.

The Personal Loan Dataset encompasses information on 5,000 customers, including demographic details such as age and income, the customer's relationship with the bank such as a mortgage or securities account, and their response to the previous personal loan campaign. Out of the 5,000 customers, only 480 (9.6%) agreed to take up the personal loan offer during the campaign. In other words, this dataset is not balanced.

Based on the characteristics of the data, the personal loan data has more instances than the audit risk data (5,000 vs 772) but less attributes (12 vs 26).

[1] They are the ones used in the assignment #1.

Unsupervised Learning Algorithms

In this report, the performance of k-means and expectation maximization (EM) algorithms are evaluated by using the selected datasets.

k-means Algorithm

The k-means algorithm is a popular unsupervised clustering technique used to group data points into a specified number of clusters based on their similarity. The algorithm works by iteratively assigning data points to the nearest centroid, which represents the center of each cluster. After each assignment, the centroids are updated based on the mean of the data points in each cluster. This process continues until convergence, where the changes of centroids are below some tolerances, or a maximum number of iterations is reached.

EM Algorithm

The Expectation-Maximization (EM) algorithm is a popular iterative optimization technique used to estimate the parameters of probabilistic models with latent variables. The algorithm works by alternately computing two steps: the E-step, where an estimate of the latent variable is computed given the current parameter estimates, and the M-step, where the parameters are updated to maximize the likelihood of the observed data given the estimated latent variables. The algorithm continues to iterate between these two steps until convergence, where the likelihood of the data no longer improves or a maximum number of iterations is reached.

Dimensionality Reduction Algorithms

In this report, the performance of principal component analysis (PCA), independent component analysis (ICA), randomized projections (RP), and singular value decomposition (SVD) are evaluated by using the selected datasets.

PCA

Principal Component Analysis (PCA) is a popular unsupervised machine learning technique used for dimensionality reduction and feature extraction. The algorithm works by identifying the principal components, which are the directions in the feature space that capture the most variance in the data. These principal components are ordered by the amount of variance they capture, with the first principal component being the direction that captures the most variance. PCA projects the original data onto these principal components, effectively reducing the dimensionality of the dataset while retaining as much information as possible.

ICA

Independent Component Analysis (ICA) is another popular unsupervised machine learning technique used for feature extraction and signal separation. The algorithm works by decomposing a multivariate signal into independent, non-Gaussian components that are statistically as independent as possible. ICA is different from PCA in that it seeks to find components that are not only uncorrelated, but also non-linearly related to the original data. ICA can be used to separate mixed signals, such as separating speech from background noise, or to identify hidden factors that underlie a dataset.

RP

Randomized projections (RP) is a technique used in machine learning for dimensionality reduction, specifically for reducing the dimensionality of large datasets. The algorithm works by projecting high-dimensional data onto a lower-dimensional subspace using a randomly generated matrix. The matrix is chosen such that it preserves the pairwise distances between data points as much as possible, while also reducing the number of dimensions in the data. By using a randomized matrix, RP can be much faster than other dimensionality reduction techniques, such as PCA and SVD, which require computing the exact eigenvectors of the covariance matrix.

SVD

Singular Value Decomposition (SVD) is a popular matrix factorization technique used in machine learning. The algorithm works by decomposing a matrix into three parts: a left singular matrix, a diagonal matrix of singular values, and a right singular matrix. SVD can be used for a variety of tasks, including matrix approximation, dimensionality reduction, and feature extraction. SVD is particularly useful for dealing with large, sparse matrices, as it can identify the underlying structure in the data and extract the most important features.

Performance of Unsupervised Learning Algorithms

In this section, the silhouette score is used to evaluate the performance of clustering algorithms, which is a measure of how well a data point fits into its assigned cluster. The silhouette score ranges from -1 to 1, where a score of 1 indicates that the data point is well-matched to its cluster, while a score of -1 indicates that it would be better off in a different cluster. The silhouette score is calculated by comparing the average distance between a data point and all other data points in its own cluster to the average distance between the data point and all data points in the nearest neighboring cluster. A high silhouette score indicates that the clustering is appropriate, while a low score indicates that the clustering may not be optimal. The silhouette score is a useful tool for selecting the appropriate number of clusters in a dataset, as well as for comparing the performance of different clustering algorithms.

In this section, the k-means and EM algorithms are run over the k value from 2 to 20 to see how the performance varies between many clusters and very few clusters. In particular, the `init` parameter of the k-means algorithm was set to be *kmean++*, i.e., the initial cluster centroids are selected using sampling based on an empirical probability distribution of the points' contribution to the overall inertia, which speeds up convergence. And the GaussianMixture model from `SciKitLearn` with the default setting was selected to implement the EM algorithm.

The performance of k-means and EM algorithms applying to the audit risk and personal loan datasets are depicted in Figure 1 and Figure 2 respectively, which can be summarized as:

- For both datasets and algorithms, the silhouette scores showed the same trends, i.e., decreasing as the cluster number increase. But, for both datasets, the EM algorithm showed a significant smaller silhouette score for most of the cluster numbers. This is due to the EM algorithm assigns each data point a probability of belonging to each cluster, it may result in more overlapping clusters and less distinct boundaries among clusters, which can lead to lower silhouette scores. On the other hand, k-means assigns each data point to a single cluster, resulting in more distinct boundaries between clusters and potentially higher silhouette scores.

- For both algorithms and datasets, the calculation time increase as the cluster number increase due to more comparison. And, for both datasets, the calculation time for EM algorithm is much longer than that of the k-means algorithm when the cluster number is the same.

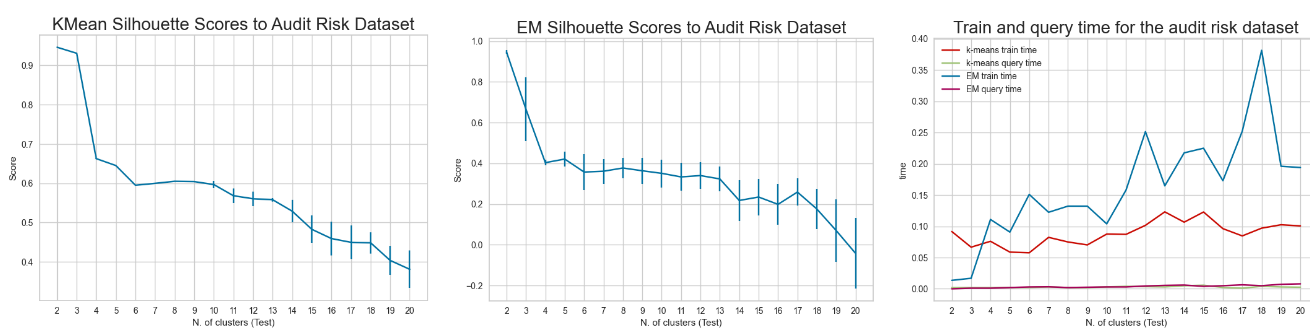

Figure 1: The first two figure showed the silhouette score for the k-means and EM algorithms for audit risk dataset, respectively. And the last figure showed the train and query time per cluster time for both algorithms.

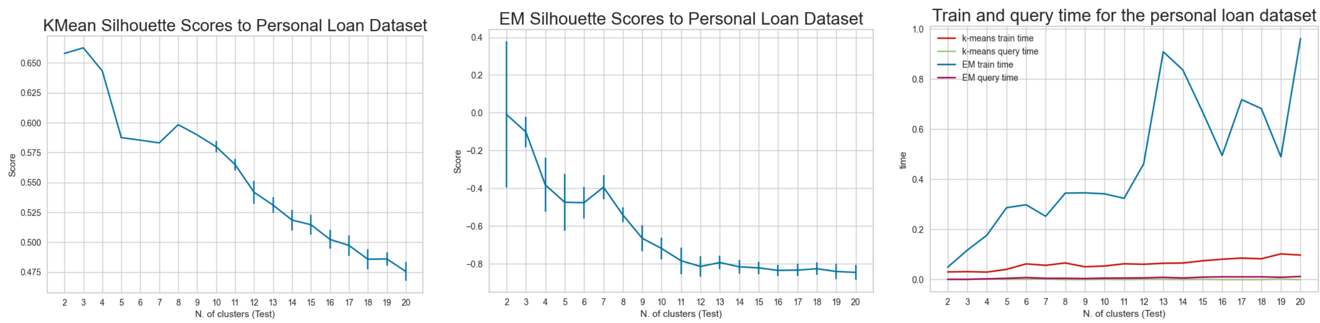

Figure 2: The first two figure showed the silhouette score for the k-means and EM algorithms for personal loan dataset, respectively. And the last figure showed the train and query time per cluster time for both algorithms.

Combine these observations on the silhouette scores and the calculation time, also the distance between GMMs and BIC scores for the EM algorithm in the submitted package, the optimal cluster number per dataset and per algorithm can be summarized as:

	k-means algorithm	EM algorithm
Audit Risk Dataset	2	2
Personal Loan Dataset	3	2

Table 1: The optimal cluster numbers per clustering algorithm and per dataset.

Furthermore, Table 2 depicted the cluster sizes per dataset and algorithm, and the relationship between the data original label with the cluster labels. Please note, the rows are for the original labels, while the columnare are for the cluster labels per dataset and algorithm. According to this table and the figures in the package generated through the function `SilhouetteVisualizer`, we have

- For audit risk dataset, the optimal cluster number is 2 for both algorithms, and they generate the same clusters with sizes of 5 and 767. In general, it suggests that there may be a meaningful difference between the two groups of data points, and that further investigation may be warranted to understand the factors that differentiate them. Considering one of the clusters only contain very few data points (5), they may be outliers.

- For personal loan dataset, the k-means and EM algorithms give different clusters (k-means algorithm divided the data into three clusters with size 2305, 2694 and 1, the EM algorithm divided the data into two clusters with size 4358 and 642). k-means and EM algorithms give different clustering results may due to differences in their initialization, assumptions about data distribution, handling of outliers. In particular, k-means is sensitive to outliers since it minimizes the sum of squared distances between data points and centroids. In contrast, EM can handle outliers better by assigning low probabilities to outliers.

- As for the relationship between the cluster labels with the data original labels, not much can be obtained from the audit risk dataset as it has very small size cluster. But for the personal loan dataset, in each cluster group, the original distribution is similar for the k-means clustering while it is not for the EM clustering. Therefore, we can not expected the cluster labels always line up with the data original labels. This is mainly due to that the cluster labels heavily depend on the metric used in the clustering algorithms. Usually, the data have many attributes, which may not be suitable to measure by the same methodology. For example, one of the personal loan dataset is zip code, which is not appropriate to be measured by the Euclidean metric.

	k-means + Audit		EM + Audit		k-means + Loan			EM + Loan	
	0	1	0	1	0	1	2	0	1
0	0	467	467	0	1649	1881	0	3178	352
1	5	300	300	5	656	813	1	1180	290
Total	5	767	767	5	2305	2694	1	4358	642

Table 2: The optimal cluster numbers per clustering algorithm and per dataset.

Based on the results, some datasets may generate small clusters that may not be meaningful or informative for the analysis. In some case, this is due the property of the tested datasets, in more cases, it is due to the clustering algorithms or data quality issues. Therefore, some improvements can be made to the algorithms to avoid or address too small clusters:

- Adjust the similarity/distance metric: the choice of similarity/distance metric can influence the size and shape of the resulting clusters.
- Use hierarchical clustering: hierarchical clustering is a clustering algorithm that creates a tree-like structure of nested clusters, and it allows for more flexibility in defining the size and shape of clusters. By setting appropriate thresholds on the tree, one can avoid creating too small or too large clusters.
- Merge small clusters: After the initial clustering, it is possible to merge small clusters into larger ones if they are not meaningful or informative. This can be achieved by either manually combining clusters or by using post-processing techniques such as agglomerative clustering.
- Remove outliers: Outliers are data points that do not fit well into any cluster, and they can create small and noisy clusters. Removing outliers before clustering can help prevent the formation of too small clusters.

In summary, avoiding too small clusters in clustering analysis involves careful selection of clustering parameters, metric, and algorithm, as well as post-processing of the results. It is important to consider the nature of the data and the research question when choosing the appropriate strategy to avoid too small clusters.

Performance of Dimensionality Reduction Algorithms

In this section, dimensionality reduction algorithms (PCA, ICA, RP and SVD) are used to construct lower dimensional data spaces. Firstly, after applying dimensionality reduction algorithms (PCA, ICA, RP and SVD) to the audit risk dataset, the transformed spaces can be described as

- For PCA algorithm, the new space is defined by the principal components, which are linear combinations of the original features. The principal components are orthogonal to each other, which means that they are linearly independent and do not contain any redundant information. The individual and cumulative variance ratio is depicted in Figure 3, which showed that the first 8 principal components contributes more than 90% of the total variance.
- For ICA algorithm, its output space is a new set of signals that are statistically independent of each other, with no redundancy. ICA reduces the dimensionality by finding independent components that underlie the observed signals. The results of the different 26 kurtosis are:

  -1.279, 48.521, 97.904, 63.370, 275.392, 49.054, 4.492, 104.304, 60.901, 655.161, 122.871, 27.550, 59.622, 11.827, 0.411, 303.924, 68.069, 17.645, 4.160, 267.438, 245.665, -1.232, 2.811, 143.519, -1.181, 11.104
- For RP algorithm, the newly obtained space is a lower-dimensional space that approximately preserves pairwise distances between points in the original space. If we run randomized projections (RP) several times, we would get different projections each time due to the randomness of the technique.
- For SVD algorithm, the space obtained by applying SVD algorithm is a lower-dimensional space with orthogonal features that capture most of the energy in the original data. In particular, the The individual and cumulative variance ratio is depicted in Figure 3, which showed that the first 2 principal components contributes more than 97% of the total variance.

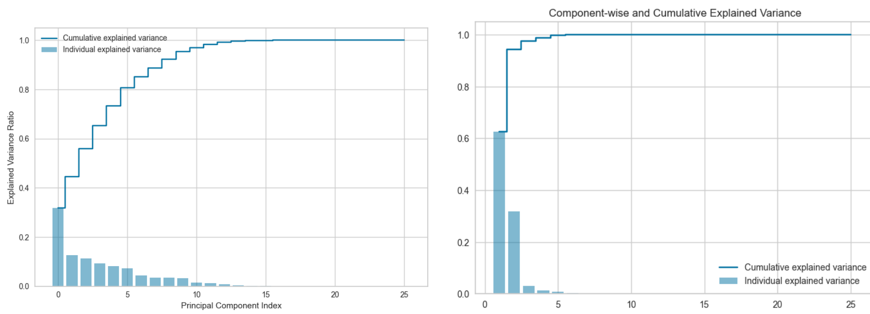

Figure 3: The explained variance ratio with respect to number of components for the PCA and SVD algorithms.

These dimensionality reduction algorithms are also assessed based on the test accuracy of a decision tree classifier (80% and 20% of the total data are used for training and testing respectively) which is a comprehensive approach. The range of components being tested (from 2 to 20) is a good starting point to evaluate the trade-off between reducing the dimensionality of the data and maintaining classifier accuracy. Additionally, the explained variance ratio of the PCA and SVD are also taken into account.

For audit risk dataset, the test accuracy after diemensionality reduction and the explained variance ratio are depicted in the Figure 4 and Figure 3, which can be summarized as:

- For PCA, the test accuracy in general showed an increase trend with respect to the number of the components. When the number of principal components is 9, the corresponding test accuracy is close to 1.0 and the explained variance ratio is larger than 0.97. Therefore, 9 is thought to be the optimal number of principal components for the audit risk dataset.

- For ICA, the test accuracy is more volatile with respect to number of components comparing to other algorithms. This is due to ICA assumes that the data can be represented as a linear combination of independent components. However, it may not always hold true in practice and result in ICA identifying different independent components and having a more variable. This also make the selection of the optimal number of components more difficult. For the audit risk dataset, 10 is thought to be the optimal number of components.

- For RP, the test accuracy showed an increase trend at beginning and then became stable as the number of components increase. Therefore, 9 is considered to be the optimal number of components with a test accuracy close to 1.0. Meanwhile, running RP several times can be useful for parameter tuning and ensemble learning purposes. It can help us find the best parameter settings and produce a more robust and stable dimensionality reduction.

- For SVD, the test accuracy in general showed an increase trend with respect to the number of the components. And 10 is considered to be the optimal number of components with test accuracy larger than 0.99 and explained variance ratio close to zero.

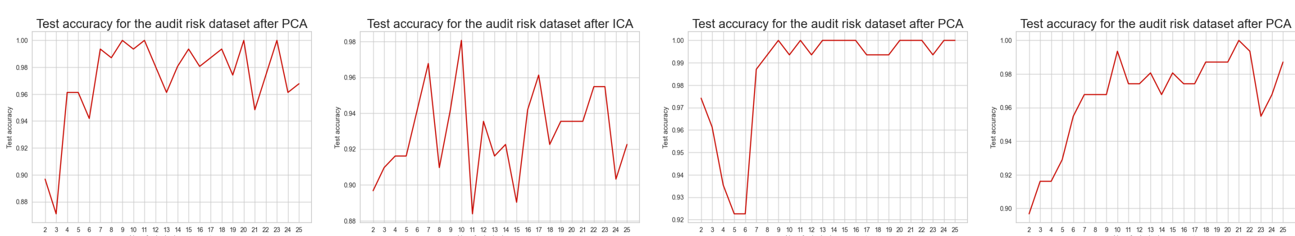

Figure 4: Test accuracy by applying the decision tress classifier to the audit risk data after dimensionality reduction (PCA, ICA, RP and SVD).

Similar analysis can be done to the personal loan dataset (corresponding figures saved in the submitted package), and the optimal number components per dataset and algorithm can be summarized in Table 3. **The optimal component per dimensionality reduction algorithm and dataset obtained in this section will be used in the following sections.**

	PCA	ICA	RP	SVD
Audit Dataset	9	10	9	10
Personal Loan Dataset	10	11	6	4

Table 3: The optimal component numbers per dimensionality reduction algorithm and dataset.

Performance of Clustering Algorithms after Dimensionality Reduction

In this section, we rerun the clustering experiments, but on the data after dimensionality reduction. Firstly, the dimensionality reduction algorithms are applied to the two selected datasets, and the new data spaces are used for the clustering experiments. Please note the optimal component numbers per dimensionality reduction algorithm and per data obtained in previous section are used for the dimensionality reduction step.

For audit risk dataset, the silhouette scores for each dimensionality reduction algorithm and clustering algorithm combination is depicted in Figure 5. The first row for k-means clustering algorithm after applying PCA, ICA, RP and SVD dimensionality reduction algorithm respectively. And the only change for the second row is the k-means clustering algorithm was replaced by the EM clustering algorithm. The same algorithms and processes can be applied to the personal loan dataset, with results depicted in Figure 6.

The results can be summarized as:

- In general, the silhouette scores from the EM clustering algorithm is lower than those of the k-means clustering algorithm. As mentioned in previous sections, this is due to the EM algorithm assigns each data point a probability of belonging to each cluster, it may result in more overlapping clusters and less distinct boundaries among clusters, which can lead to lower silhouette scores. On the other hand, k-means assigns each data point to a single cluster, resulting in more distinct boundaries between clusters and potentially higher silhouette scores.
- Per each dataset and clustering algorithm combination, the PCA and SVD depicted similar performance. This is due to they are related to each other, and in fact they are mathematically equivalent. Both PCA and SVD are used for dimensionality reduction by selecting a subset of the principal components or singular vectors that capture most of the variance in the data.
- For both datasets and both clustering algorithms, the ICA dimensionality reduction algorithm showed a much lower silhouette scores comparing to other dimensionality reduction algorithms. Maybe more fine-tuning and parameter selection are needed to obtain better performance. Therefore, more investigation is needed in this point.
- In general, the performance of RP is not very sensitive to the number of clusters. This may be due to the ability of RP to preserve pairwise distances and the linear nature of the method may make it less sensitive to the number of clusters, as long as the clustering algorithm is able to use the preserved pairwise distances effectively.

Moreover, the clustering experiments on the datasets projected onto the new spaces created by PCA, ICA and SVD do not generate the same clusters as before. This is due to that the generated clusters heavily dependent on the distance between points, but these three algorithms do not preserve the distance between points. In this sense, the RP algorithm are most likely to generate the same clusters as before, as the RP algorithm is designed to preserve the pairwise distance between data points. But, for our tested datasets, the clusters are not kept as the pairwise distance are not 100% preserved.

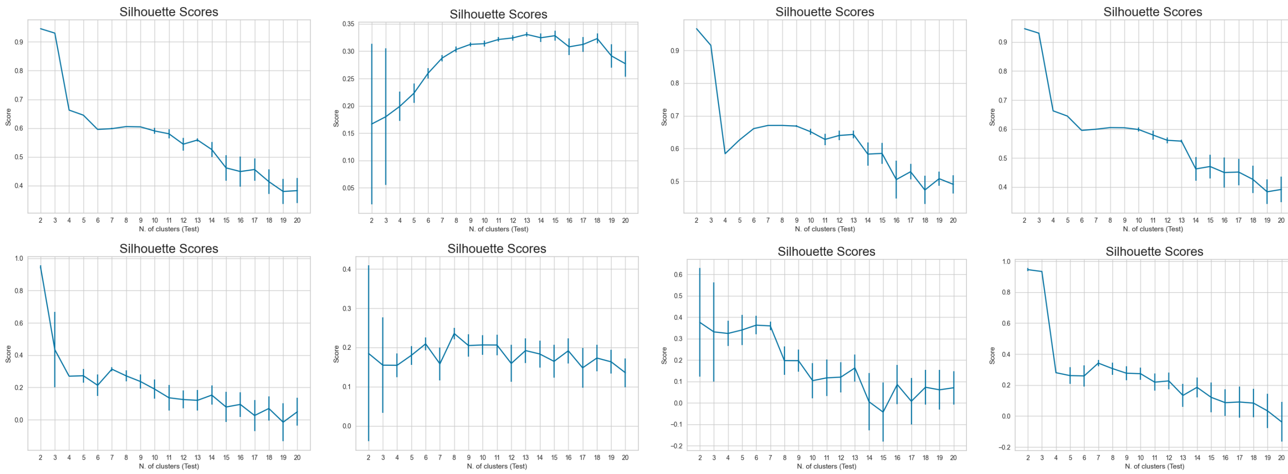

Figure 5: The silhouette score of applying the k-means and EM algorithms to the audit risk dataset after applying the dimensionality reduction algorithms (PCA, ICA, RP and SVD) respectively.

Figure 6: The silhouette score of applying the k-means and EM algorithms to the personal loan dataset after applying the dimensionality reduction algorithms (PCA, ICA, RP and SVD) respectively.

Performance of Neural Networks on Projected Space

In this section, the dimensionality reduction algorithms (PCA, ICA, RP and SVD) are applying to the personal loan dataset to obtain lower dimensional projected datasets. Please note the optimal components for each dimensionality reduction algorithm obtained in previous section will be used for the experiments in this section. In other words, the optimal dimensions for PCA, ICA, RP and SVD are 10, 11, 6 and 4 respectively.

The projected datasets are then used to train the neural networks with two hidden layer (sizes 5 and 2), and *logistic* `activation`. The computation time and the associated test accuracy (by using 80% data for training and 20% data for testing) can be summarized in Table 4.

	PCA	ICA	RP	SVD	No Dimensionality Reduction
Test Accuracy	0.708	0.695	0.716	0.704	0.677
Time	15.1	25.4	17.4	17.4	16.6

Table 4: The test accuracy and computation time of neural networks with (PCA, ICA, RP and SVD) or without dimensionality reduction.

We can concluded that:

- Comparing to the test accuracy without dimensionality reduction, a slightly better test accuracy is obtained after the dimensionality reduction. Overall, reducing the dimensionality of the data can lead to improved test accuracy by reducing overfitting, reducing the curse of dimensionality, improving feature representation (identifying the most important features), and improving model interpretability. However, it is important to carefully choose the dimensionality reduction technique and the number of dimensions to retain, as removing too much information from the data can lead to a loss of performance.
- Overall, reducing the dimensionality of the dataset can potentially make the neural network faster to train, but it depends on the specifics of the dataset and the neural network architecture. However, dimensionality reduction algorithms may also introduce additional computational costs in the pre-processing stage, which could offset the training time gains.

Performance of Neural Networks on Projected Space + Cluster Labels

In this section, the projected datasets through applying the dimensionality reduction algorithm are extended by add the cluster labels as a new feature. And then the extended datasets are used to train the neural networks. As in last section, the personal loan dataset is used for the training. The neural networks are set the same as in last section, the number of clusters are set up as the optimal ones obtained in previous section, i.e., 3 and 2 are used for k-means and EM algorithms respectively.

When the clustering algorithm is k-means or EM algorithms, the results can be summarized as in Table 5 and Table 6 respectively.

	PCA	ICA	RP	SVD
Test Accuracy	0.688	0.717	0.702	0.704
Time	14.8	19.3	15.2	15.6

Table 5: The test accuracy and computation time of neural networks training with the extended projected datasets (obtained through applying PCA, ICA, RP and SVD algorithms). And the cluster labels generated by the k-means algorithm are used to extend the projected dataset.

	PCA	ICA	RP	SVD
Test Accuracy	0.723	0.706	0.691	0.699
Time	17.9	19.9	16.0	16.6

Table 6: The test accuracy and computation time of neural networks training with the extended projected datasets (obtained through applying PCA, ICA, RP and SVD algorithms). And the cluster labels generated by the EM algorithm are used to extend the projected datasets.

Combine Table 4, Table 5 and Table 6, it seems that including the cluster labels to the projected datasets for the neural networks does not really improve the performance of the neural network leaner. The test accuracy for both k-means and EM algorithms are similar to the result of running neural network learner to the projected database without the cluster labels. In some cases, the test accuracy slightly increased, while in some other cases the test accuracy slightly decreased.

On the other hand, the computation time slightly decreased when k-means algorithm is used for the cluster labels. This may be due to the cluster labels are highly correlated with the existing features, thus it may provide redundant information to the model, leading to sparsity in the training data. In such cases, adding the new feature can actually decrease the complexity of the data, making it easier for the model to learn and reducing training time. As for EM clustering algorithm, a longer time is needed to run the EM algorithm itself, which may be longer than the time saved in the neural network training. Therefore, time decreased in some case while time increase in some other cases.

Conclusion

In this report, the datasets in assignment #1 (audit risk and personal datasets) are reused for the tests of clustering and dimensionality reduction algorithms.

For each clustering algorithm and each dataset, the optimal cluster numbers are determined by comparing the silhouette scores with various cluster numbers (from 2 to 20). The silhouette score is a measure of how well a data point fits into its assigned cluster. Taking the silhouette scores and computation time into account, for audit risk dataset, the optimal cluster numbers are both 2 for k-means and EM algorithms; for personal loan dataset, the optimal cluster numbers are 3 and 2 for k-means and EM algorithms respectively.

For audit risk dataset, two cluster groups with size 767 and 5 are assessed to be the optimal clustering under both k-means and EM algorithms. For personal loan data, three and two clustering groups are assessed to be the optimal ones under K-means and EM algorithms respectively. One observed issue in the optimal clustering is that some clustering groups have a very small size, such as 5 or 1. Some strategies, such as adjusting the similarity/ distance metric, using hierarchical clustering, merging small clusters and removing outliers, can be used to address this issue.

For the dimensionality reduction techniques (PCA, ICA, RP and SVD), the test accuracy by applying decision tress classifier is the mainly criteria used to determine the optimal component numbers per dimensionality reduction algorithm and per dataset. The explained variance ratios for PCA and SVD, and the kurtosis for independent components are also taken into account. See Table 3 for the optimal components numbers per algorithm and per dataset.

The clustering algorithms (k-means and EM) are also applied to the projected datasets generated through the dimensionality reduction algorithms. The generated clusters may be different from the ones before the dimensionality reduction algorithm is applied. This is due to that the generated clusters heavily dependent on the distance between points, but PCA, ICA and SVD do not preserve the pairwise distance among points. In this sense, the RP algorithm are most likely to generate the same clusters as before, as the RP algorithm is designed to preserve the pairwise distance between data points. But, for our tested datasets, the clusters are not kept as the pairwise distance are not 100% preserved.

Finally, the projected datasets obtained through the dimensionality reduction algorithms and its extension to include the cluster labels are used to train the neural networks. In general, the results are in general similar to the ones obtained by training the neural networks by the original dataset. In this test, the personal loan dataset is used for the training. Overall, reducing the dimensionality of the data can lead to improved test accuracy by reducing overfitting, reducint the curse of dimensionality, identifying the most important features, and improving model interpretability. But in the same time, dimensionality reduction algorithms may also introduce additional computational costs in the pre-processing stage. In this section, the optimal cluster numbers and optimal components obtained in previous sections are used for the tests.

In summary, K-means is a simple, efficient, and robust algorithm for clustering data with a fixed number of clusters and data with convex shapes. EM is more flexible and can handle complex data distributions, missing data, and non-convex clusters but can be computationally expensive and require more expertise to implement. On the other hand, PCA, ICA, RP, and SVD are all unsupervised machine learning algorithms used for dimensionality reduction. PCA and ICA are linear methods while RP and SVD are non-linear methods. PCA and ICA can separate mixed signals while RP and SVD cannot. RP is faster and less computationally expensive than SVD, but SVD is more accurate and can be used for both linear and non-linear data. Ultimately, the choice of algorithm depends on the nature of the data and the specific requirements of the problem at hand.

References

[1] Vincenzo Lavorini. Gaussian Mixture Model clustering: how to select the number of components (clusters).

[2] Wikipedia. Jensen–Shannon divergence.