**Datasets**

The datasets are downloaded from the link <https://archive.ics.uci.edu/ml/index.php> and are the same as what is used in HW1. The two datasets chosen are Car Evaluation and Adult. For both datasets, the features were limited to be able to run effective experiments.

1. Car Evaluation [1]

The dataset is derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.).

This dataset will be referred to as dataset 1. This is a multivariate classification dataset. It has 6 attributes with 1728 instances. The 6 attributes are buying, maintenance cost, number of doors, number or persons, luggage boot size and safety. The class values are inaccurate, accurate, good and very good.

1. Adult [1]

The dataset extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0)).

This dataset will be referred to as dataset 2. This is a multivariate classification dataset. It has 14 attributes with 48842 instances. The 14 attributes are age, workclass, education, education num, marital status, occupation, relationship, race, sex, capital gain, capital loss, hours per week and native country. The class values are >50K and <=50K.

**KMeans**

KMeans is a partition based clustering method. The data points get assigned to the centroids that have the smallest distance between them. For the experiment, I used Euclidean distance as the measurement. The KMeans algorithm can represent data in any dimension and thus this metric is used. To determine the optimal number of clusters for the two datasets, I used a combination of Elbow method and average silhouette score. The Elbow method was applied to the distortion.

Links used as a reference:

1. <https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html>
2. <https://towardsdatascience.com/silhouette-method-better-than-elbow-method-to-find-optimal-clusters-378d62ff6891>
3. <https://stackoverflow.com/questions/63471999/how-to-plot-clusters-and-centers-from-a-multi-feature-kmeans-model-with-matplot>

For dataset 1, using the Elbow method, the clusters of interest are 3, 8 and 10. Using the average silhouette score plot, the optimum cluster is found to be 3.

For dataset 2, using the Elbow method, the clusters of interest are 2, 3 and 5. Using the average silhouette score plot, the optimum cluster is found to be 2.

**Dataset 2 Dataset 1**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Chart, histogram

Description automatically generatedChart, bar chart

Description automatically generated**

**Expectation Maximization**

Expectation maximization is an algorithm that estimates the values of latent variables. Here we estimate the density using Gaussian Mixture Model. To determine the optimal number of clusters for the two datasets, I used a combination of Elbow method and average silhouette score. The Elbow method was applied to the AIC and BIC scores.

For dataset 1, using the Elbow method, the clusters of interest are 3, 5 and 10. Using the average silhouette score plot, the optimum cluster is found to be 3.

For dataset 2, using the Elbow method, the clusters of interest are 3 and 5. Using the average silhouette score plot, the optimum cluster is found to be 3.

**Dataset 2 Dataset 1**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart

Description automatically generatedChart, bar chart

Description automatically generated

**Principal Component Analysis (PCA)**

PCA is a linear dimensionality reduction method that transforms a set of variables into a smaller number of uncorrelated variables, known as principal components, while retaining as much of the variation of the original dataset. The number of components is determined using the explained variance and cumulative variance.

Links used as a reference:

1. <https://towardsdatascience.com/principal-component-analysis-pca-with-scikit-learn-1e84a0c731b0>
2. <https://www.oreilly.com/library/view/hands-on-unsupervised-learning/9781492035633/ch04.html>

For dataset 1, the number of components is determined to be 9, where the cumulative variance is around 95%. The scatter plot along the two principal components is shown below.

For dataset 2, the number of components is determined to be 30, where the cumulative variance is around 95%. The scatter plot along the two principal components is shown below.

**Dataset 2 Dataset 1**

Chart

Description automatically generatedChart, line chart

Description automatically generated

**PCA With KMeans:**

KMeans was run on top of the reduced dataset using PCA. The results are shown below.

For dataset 1, using the Elbow method, the clusters of interest are 3, 7 and 9. Using the average silhouette score plot, the optimum cluster is found to be 3.

For dataset 2, using the Elbow method, the clusters of interest are 3, 4, 5 and 8. Using the average silhouette score plot, the optimum cluster is found to be 2.

The average silhouette plot for both KMeans and Gaussian Mixture are same which is interesting. Only one of the plots is shown. The plots are generated in the plots folder when running the experiment.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**PCA With Expectation Maximization:**

Gaussian Mixture was run on top of the reduced dataset using PCA. The results are shown below.

For dataset 1, using the Elbow method, the clusters of interest are 3, 4, 6, 7 and 9. Using the average silhouette score plot, the optimum cluster is found to be 3.

For dataset 2, using the Elbow method, the clusters of interest are 2, 3, 5 and 7. Using the average silhouette score plot, the optimum cluster is found to be 2.

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

**Independent Component Analysis (ICA)**

ICA aims to separate information by transforming the input space into a maximally independent basis. Kurtosis is used to determine the number of components. It is a measure that defines how heavily the tails of a distribution differ from the tails of a normal distribution.

**Dataset 2 Dataset 1Chart, line chart, scatter chart

Description automatically generatedChart, line chart

Description automatically generated**

Links used as a reference:

1. <https://www.oreilly.com/library/view/hands-on-unsupervised-learning/9781492035633/ch04.html>

For dataset 1, the optimum number is found to be 7.

For dataset 2, the number of components of interest are 8, 10, 11, 21 and 37. The optimum number is found to be 10.

**ICA With KMeans:**

KMeans was run on top of the reduced dataset using ICA. The results are shown below.

For dataset 1, using the Elbow method, the clusters of interest are 3, 5 and 6. Using the average silhouette score plot, the optimum cluster is found to be 3.

For dataset 2, using the Elbow method, the clusters of interest are 3, 4 and 7. Using the average silhouette score plot, the optimum cluster is found to be 7.

The average silhouette plot for both KMeans and Gaussian Mixture are same which is interesting. Only one of the plots is shown. The plots are generated in the plots folder when running the experiment.

Chart, line chart, scatter chart

Description automatically generatedChart, line chart

Description automatically generated

**ICA With Expectation Maximization:**

Gaussian Mixture was run on top of the reduced dataset using ICA. The results are shown below.

For dataset 1, using the Elbow method, the clusters of interest are 3, 4, 5 and 7. Using the average silhouette score plot, the optimum cluster is found to be 3.

For dataset 2, using the Elbow method, the clusters of interest are 3, 4, 6, 7 and 8. Using the average silhouette score plot, the optimum cluster is found to be 7.

The kurtosis plot can be found in the plots folder in the code directory.

**Chart

Description automatically generatedChart, histogram

Description automatically generated**

**Gaussian Random Projection (GRP)**

GRP reduces dimensionality by projecting the original input space on a randomly generated matrix where components are draw from a distribution

The results are compares using the reconstruction loss.

Links used as a reference:

1. <https://scikit-learn.org/stable/modules/random_projection.html>
2. <https://www.oreilly.com/library/view/hands-on-unsupervised-learning/9781492035633/ch04.html>
3. <https://asifrehan.com/how-to-compute-reconstruction-error-for-random-projection/>

**Dataset 2 Dataset 1**Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**GRP With KMeans:**

KMeans was run on top of the reduced dataset using ICA. The results are shown below.

For dataset 1, using the Elbow method, the clusters of interest are 3, 5, 6 and 7. Using the average silhouette score plot, the optimum cluster is found to be 3.

For dataset 2, using the Elbow method, the clusters of interest are 3, 5, 8 and 9. Using the average silhouette score plot, the optimum cluster is found to be 2.

The average silhouette plot for both KMeans and Gaussian Mixture are same which is interesting. Only one of the plots is shown. The plots are generated in the plots folder when running the experiment.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**GRP With Expectation Maximization:**

Gaussian Mixture was run on top of the reduced dataset using ICA. The results are shown below.

For dataset 1, using the Elbow method, the clusters of interest are 3, 5 and 6. Using the average silhouette score plot, the optimum cluster is found to be 3.

For dataset 2, using the Elbow method, the clusters of interest are 3, 4 and 7. Using the average silhouette score plot, the optimum cluster is found to be 2.

The distortion plot can be found in the plots folder in the code directory.

Chart

Description automatically generatedChart

Description automatically generated

**Mini Batch Dictionary Learning**

Finds a dictionary that performs well at sparsely encoding the fitted data. The algorithm learns the sparse representation of the data.

The results are compares using the reconstruction loss.

Links used as a reference:

1. <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.MiniBatchDictionaryLearning.html>
2. <https://www.oreilly.com/library/view/hands-on-unsupervised-learning/9781492035633/ch04.html>
3. <https://asifrehan.com/how-to-compute-reconstruction-error-for-random-projection/>

**Dataset 2 Dataset 1**

**Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated**

**Mini Batch Dictionary Learning With KMeans:**

KMeans was run on top of the reduced dataset using ICA. The results are shown below.

For dataset 1, using the Elbow method, the cluster of interest is 7. Comparing with the average silhouette score plot, the optimum cluster is found to be 7.

For dataset 2, using the Elbow method, the clusters of interest are 3, 4 and 5. Using the average silhouette score plot, the optimum cluster is found to be 2.

The average silhouette plot for both KMeans and Gaussian Mixture are same which is interesting. Only one of the plots is shown. The plots are generated in the plots folder when running the experiment.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Mini Batch Dictionary Learning With Expectation Maximization:**

Gaussian Mixture was run on top of the reduced dataset using ICA. The results are shown below.

For dataset 1, using the Elbow method, the clusters of interest are 4, 5, 6 and 7. Using the average silhouette score plot, the optimum cluster is found to be 7.

For dataset 2, using the Elbow method, the clusters of interest are 3, 5, 6 and 10. Using the average silhouette score plot, the optimum cluster is found to be 2.

The reconstruction error plot can be found in the plots folder in the code directory.

**Chart

Description automatically generatedChart, bar chart

Description automatically generated**

**Neural Network – MLP Evaluation**

Below are the accuracy scores and model times for the experiments using the MLP classifier from scikit learn - neural network.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Accuracy - In Sample** | **Accuracy - Out Sample** | **Fit Time** | **Query Time** |
| Principal Component Analysis | 0.789081886 | 0.653179191 | 3.029490709 | 0.002018929 |
| Independent Component Analysis | 0.710504549 | 0.570327553 | 3.686540842 | 0.001997471 |
| Gaussian Random Projection | 0.779156328 | 0.755298651 | 4.129885674 | 0.00303483 |
| Mini Batch Dictionary Learning | 0.669975186 | 0.524084778 | 0.798716307 | 0.001969337 |
| KMeans | 0.70306038 | 0.693641618 | 0.626730442 | 0.001999617 |
| Expectation Maximization | 0.70306038 | 0.693641618 | 2.853698969 | 0.002007484 |

Overall GRP has the best accuracy scores though it takes longer than the other methods. Mini batch dictionary learning has the worst performance metrics. However, this could be because of the batch size provided along with the small dataset chosen.

Both clustering methods have similar accuracy scores with the difference being in the model execution times. The scores are likely to improve with dimensionality reduction as it doesn’t look like the clustering is optimal for the given features.

**References**

1. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.