Andrew Cudworth
CS 7641 Machine Learning (Spring 2020)
Assignment 3 Unsupervised Learning

CS 7641 Assignment 3 Unsupervised Learning

Intro/Data:

This paper explores unsupervised learning via clustering and dimensionality reduction on Airbnb data and poker hand data. K-means and expectation maximization methods are used to look for clusters on the original and reduced dimension data. Principal Component Analysis (PCA), Independent component analysis (ICA), Random Projections (RPA), and Factor Analysis (FCA) are explored for dimensionality reduction. The reduced data with and without clusters are used to retrain a previously generated [10] Neural Network to see if the unsupervised learning exploration lead to better performance.

The first dataset is a summary of Airbnb listings in 2019 in New York city [3]. The target to be predicted is "room type" which can be "Entire Home", "private room", or "Shared Room" encodes as 0,1, and 2 respectively. The features shown in Table 1 are all continuous values whose difference is a meaningful distance metric. There are 49k unique records broken into an 80/20 train test split. Clustering and dim reduction are applied to the training data in all analysis.

Table 1 Airbnb Data Record Example

price	Min Nights	Reviews	Reviews/month	Host Listings	Room Type
149	1	9	0.21	6	Private room

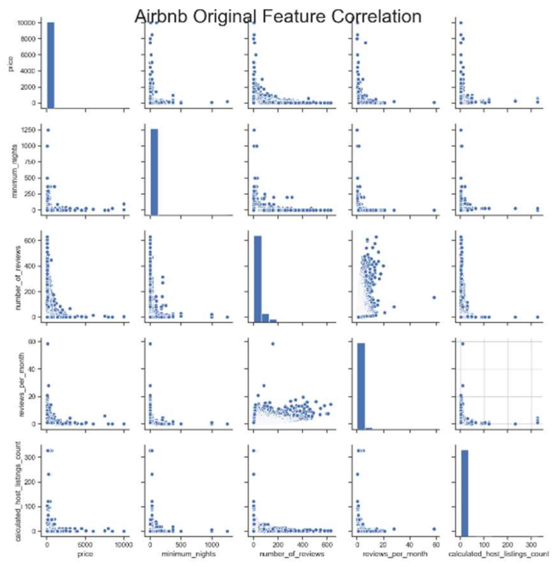
The second set of data are real hands of poker drawn from a well shuffled deck with a label of the suit and value for each card dealt matching expected probability distributions [10]. For any hand better than 2 of a kind, the hand is labeled 1 for a "good hand" else it is labeled 0. The type of hand is predicted when 4 of 5 cards are known. There are 25k records with 1547 unique hands with an example in Table 2. The data is highly imbalanced with only 2.85% of samples being good hands. Again, all experiments use an 80/20 train test split.

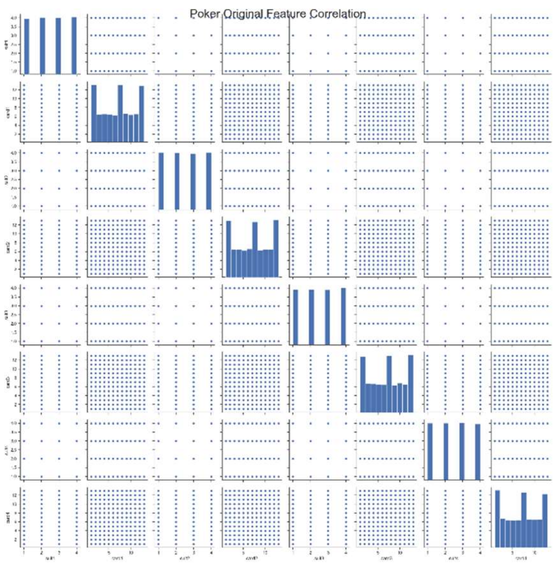
Table 2 Poker Data Record Example (Potential Royal Flush)

Suit1	card1	suit2	card2	suit3	card3	suit4	card4	hand
1	10	1	11	1	12	1	13	1

Correlation among original features is summarized in Table 3 below for general shape reference.

Table 3 Airbnb (Left) and Poker (Right) Original Feature Correlations





Andrew Cudworth
CS 7641 Machine Learning (Spring 2020)
Assignment 3 Unsupervised Learning

K-Mean:

Equation 1 K-Means Inertia [6]

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Parameters:
1 μ_i represesnts cluster C average x_i is each sample

Equation 2 Silhouette Score [11]

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

Parameters: 2 a is intra cluster average distance; b is the nearest cluster distance; i is a sample record

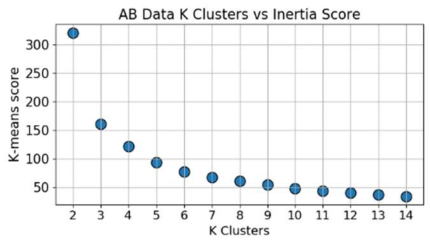


Figure 1 Air Bnb Data Inertia Score vs K

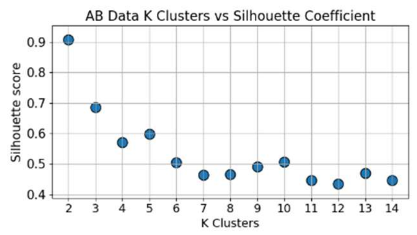


Figure 2 Air Bnb Data Silhouette Score vs K

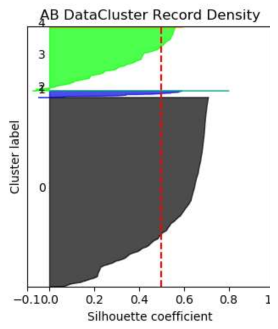


Figure 3 Airbnb Data 5 Cluster Sample Density

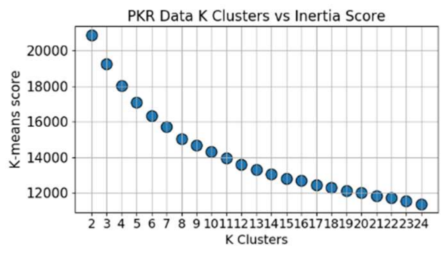


Figure 4 Poker Data Inertia Score vs K

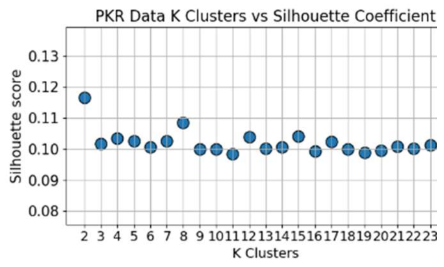


Figure 5 Poker Data Silhouette Score vs K

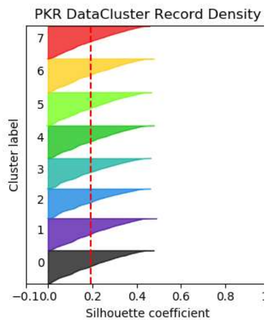


Figure 6 Poker Data 8 Cluster Sample Density

The K-Means clustering method was tested on both data sets with no assumed k value using Euclidian distance of the features. As implemented, the k-means method attempts to create clusters of equal variances via minimizing inertia defined in equation 1. The inertia is treated as the K-means Score in the elbow plot in figures 1 and 4. Conceptually, this score decreases with tighter clusters. A silhouette score was also generated over the same range were silhouette score depends on both cluster variance and distance to other clusters with range (-1,1) worst to best. Conceptually this favors tight clusters that are well separated. This is used to identify the “best” k value near the “bend” in the elbow of Figures 1 and 4 where k-means score is low as well as the total clusters. The distribution of records in the clusters was then visualized in figure 3 and 6. Generally a K is sought with almost no negative silhouette score among clusters.

Figure 7 presents Airbnb normalized feature value density for each cluster. The most variation can be seen among price and minimum nights. (other features not pictured) This makes sense as price has the largest range of values and contributes most to "distance" in the k means metric. The Poker data exhibits similar results via the card value which also has a greater range than suit. Notably, clusters are formed around whether each card is generally high or low. The pattern in Figure 8 repeats for all permutations of cards 1-4 being generally high or low. Neither data set formed clusters around labels.

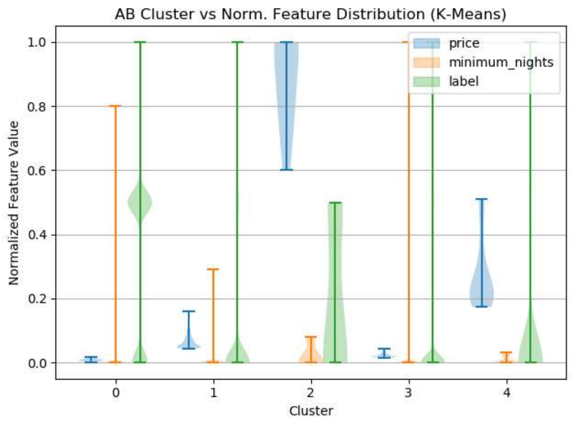


Figure 7 Airbnb Data K-Means Normalized Feature Distribution vs Cluster

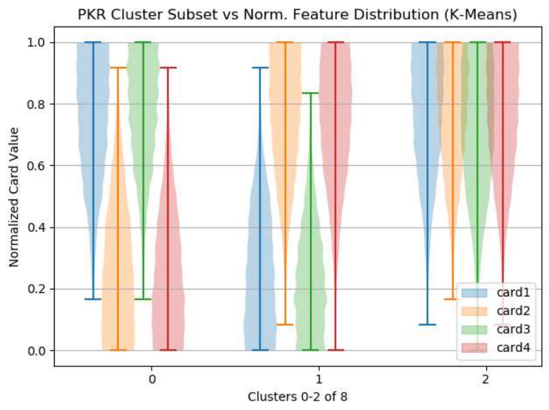


Figure 8 Poker Data K-Means Normalized Feature Distribution vs Cluster Subset (3 of 8 clusters)

EXPECTATION MAXIMIZATION:

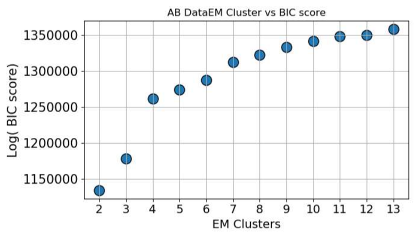


Figure 9 Airbnb Data BIC Score vs C

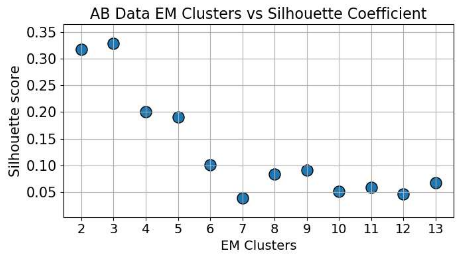


Figure 10 Airbnb Data Silhouette Score vs C

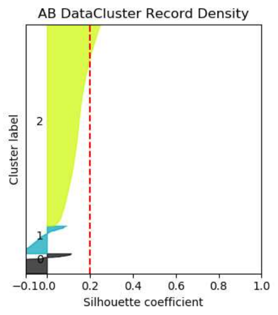


Figure 11 Airbnb EM Data 3 Cluster Density and Scores

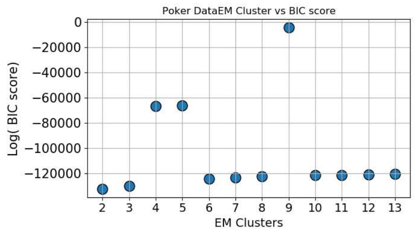


Figure 12 Poker BIC Score vs C

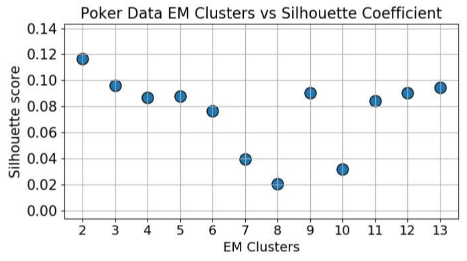


Figure 13 Poker Data Silhouette Score vs C

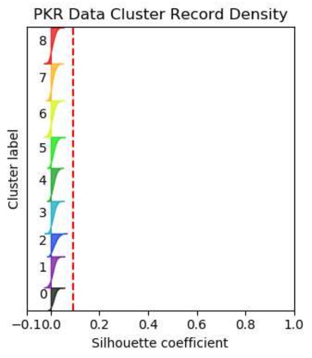


Figure 14 Poker Data EM 9 Cluster Density and Scores

The Expectation-Maximization clustering method was tested on both data sets with no assumed cluster count using Euclidian distance of the features. The EM method attempts to assign a cluster probability to each sample by alternating gaussian estimates and reassigning samples to clusters. Bayesian Information Criteria [4] is treated as a score for clusters in the elbow plot in figures 9 and 12. Conceptually, this score punishes more clusters [4]. A silhouette score was also generated. This is used to identify the "best" cluster near the "bend" in the elbow of Figures 9 and 12 where BIC score is high and total clusters is low. The distribution of records in the clusters was then visualized and inspected as in the process for K-means.

Figure 15 presents Airbnb normalized feature value density for each cluster. Variation can be seen among all features besides reviews/month. This makes sense as EM relies less on distance the K-means and utilized all features more. The Poker data did not obviously cluster well. No obvious elbow formed in figure 12. However, Clusters pictured in Figure 16 each exclude an entire suite in some form. This may be meaningful. Neither data set formed clusters around labels.

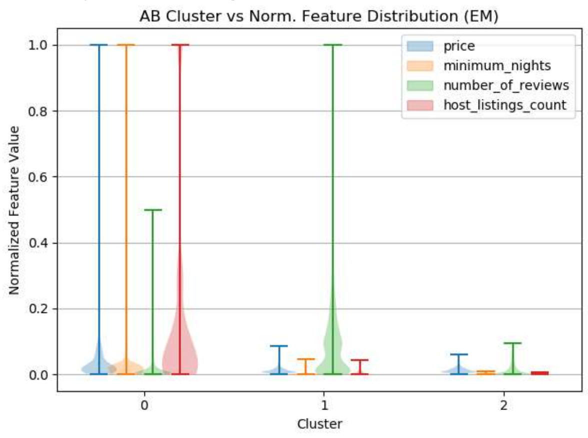


Figure 15 Airbnb Data EM Normalized Feature Distribution vs Cluster

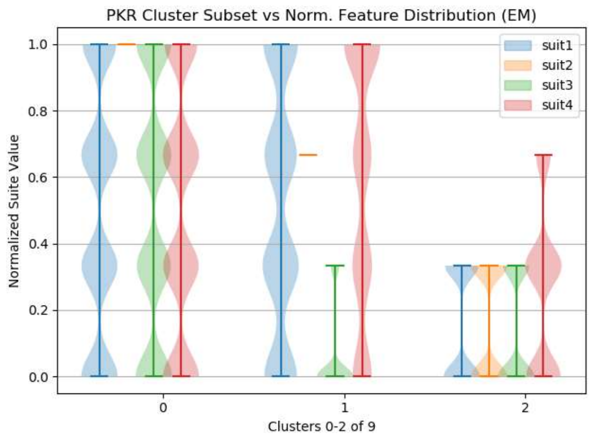


Figure 16 Poker Data EM Normalized Feature Distribution vs Cluster Subset

REDUCED DIMENSION RECLUSTERING:

The methods described above were repeated for data reduced via PCA,ICA,RPA, and FCA detailed in later sections. With Results summarized Table 4 and plots available in the src/re_clster folder.

Table 4 Reduced Dimension Re-Clustering Best Cluster Count and Average Silhouette Score

Airbnb Data					
		K-Means		Exp-Max	
	Featr. Count	Clstr. Count	avg. Sil	Clstr. Count	avg. Sil
Orig	8	8	0.45	9	0.10
PCA	4	5	0.71	5	0.30
ICA	4	7	0.62	3	0.71
RPA	4	7	0.62	6	0.25
FCA	4	11	0.44	5	0.22

Poker Data					
		K-Means		Exp-Max	
	Featr. Count	Clstr. Count	avg. Sil	Clstr. Count	avg. Sil
Orig	8	8	0.12	9	0.09
PCA	4	16	0.31	11	0.24
ICA	4	16	0.31	13	0.24
RPA	4	18	0.19	4	0.18
FCA	4	*18*	*0.19*	2	0.25

FCA with K means poker data lead to better results than any other method for increasing NN ROC_AUC performance. Clusters are similar to the results explored in figure 8 with FC component 1-4 rather than card value 1-4.

REDUCED DIMENSION RECONSTRUCTION ERROR LOSS:

PCA,ICA,RPA, and FCA, were all evaluated for up to n-1 number of components where n is the original number of features with all features of the dataset normalized between 0 and 1. At each step, the mean squared error loss of the reconstructing the data was evaluated and pictured in Table 5. In the case of RPA the average loss was taken over 50 runs of the analysis to calculate the loss. ICA,PCA, and RPA had nearly identical loss values despite entirely different projections. This is because each follow the same method for projecting features while seeking to maximize or randomize different properties of the projections. In each case (PCA,ICA,RPA) the closer to original feature count the projection is, the easier it is to reconstruct because each reduction in projections "compresses" a feature into a new space adding error. [12] FCA has the opposite result because each "factor" has some error in it. So more factors compound the error when reconstructing.

Table 5 Mean Squared Error from Reconstruction (Loss)

	Recn. Loss							
	PCA		ICA		RPA		FCA	
cmpnts	Airbnb	Poker	Airbnb	Poker	Airbnb	Poker	Airbnb	Poker
2	2.256E-04	8.280E-02	2.256E-04	8.280E-02	2.032E-03	8.276E-02	9.15E+01	2.01E+00
3	1.150E-04	6.520E-02	1.150E-04	6.520E-02	1.267E-03	6.519E-02	6.53E+02	2.66E+00
4	3.776E-05	4.880E-02	3.776E-05	4.880E-02	6.315E-04	4.884E-02	1.22E+03	3.32E+00
5	NA	3.630E-02	NA	3.630E-02	NA	3.626E-02	NA	4.32E+00
6	NA	2.390E-02	NA	2.390E-02	NA	2.387E-02	NA	5.28E+00
7	NA	1.150E-02	NA	1.150E-02	NA	1.155E-02	NA	5.28E+00

PRINCIPLE COMPONENT ANALYSIS:

Table 6 PCA Explained Variance Value and Ratio for individual PCs

	Ab		Pkr	
PC	ev	evr	ev	evr
0	3.88E+04	92.10%	1.45E+01	23.70%
1	2.32E+03	5.50%	1.43E+01	23.30%
2	6.89E+02	1.60%	1.42E+01	23.20%
3	2.98E+02	0.70%	1.33E+01	21.70%
4	NA	NA	1.30E+00	2.10%
5	NA	NA	1.27E+00	2.10%
6	NA	NA	1.26E+00	2.10%

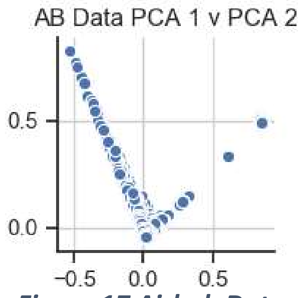


Figure 17 Airbnb Data Projected to PCAs

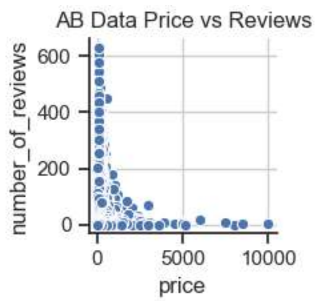


Figure 18 Airbnb Price vs Reviews (No Transform)

PCA seeks to maximize the variance within a component so each component was evaluated based on its proportion of the variance via explained variance[7] pictured in Table 6. Conceptually the ratio of explained variance is the % of variance among the data captured in the respective component. From this, 2 and 4 principal components were selected for Airbnb and Poker data respectively as they captured > 93% of the total variance.

With these PC number counts Airbnb data was reduce to 2 PCs. While it is not intuitively obvious which features PCs are constructed from, price and number of reviews had the highest standard deviation in the initial data. The principal components of Airbnb data are easily visualized in Figure 17 compared to the initial correlation between price and number of reviews. The shape could be interpreted as a rotation of the original features.

The poker data was reduced from 7 to 4 features. The pair plot of the reduce features is presented in Figure 19. It is harder to interpret what the PCs are in this transformation. The PCs have their values nearly normally distributed and natural clusters exists between the PCs. The standard deviation among cards would be slightly higher than among suites, but of the same order. Given that the hands are random, these components likely represent projected gaussians of some suite and card value combinations. As can been see in the diagonal of the pair plot, the components themselves have normally distributed values.

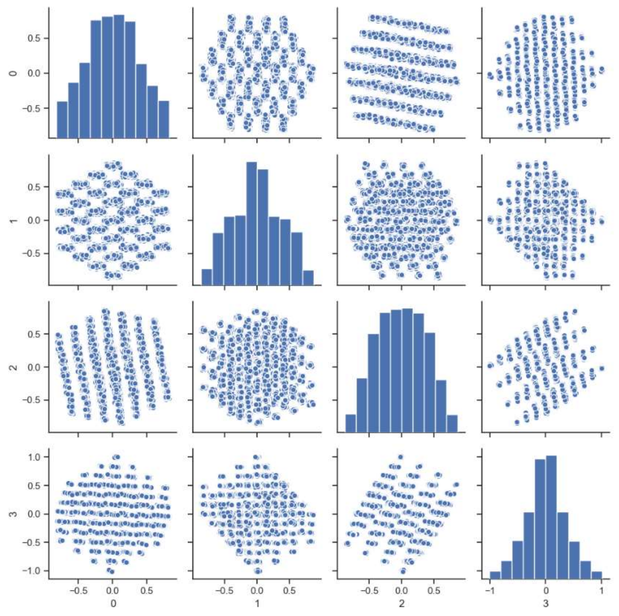


Figure 19 Poker Data PCA Projection Pair Plot

INDEPENDENT COMPONET ANALYSIS:

ICA seeks to find statistically independent components via linear combinations of existing features. [11 p. 7] Statistical independence can be assessed based on sharpness of the peak of a component's distribution. Separate sharp peaks would be intuitively well separated. Kurtosis was measured for each IC out of n and the average was taken as presented in Table 7. The distance from 0 of the kurtoses or "distance from gaussian" was used to determine the best number of ICs for each dataset. Both datasets were best separated at 4 ICs.

Table 7 Average Kurtosis of ICs for N Components

ICA	cmpts	2	3	4	5	6	7
Ab	avg_kurt	6.86E+01	6.94E+01	2.92E+02	NA	NA	NA
Pkr	avg_kurt	-7.60E-01	-8.20E-01	-1.40E+00	-1.20E+00	-1.20E+00	-1.10E+00

Figures 20 and 21 show the 100-bin histogram for each IC of their respective data sets. For Airbnb data, Each IC has a sharp peak with little variation indicating good independence. They are linear projections of the original data. So their meaning is not easily interpretable from the original data. However, their properties can be seen by the histograms showing tightly clustered projections of the normalized data.

In contrast, each IC of the poker data 4 distinct peaks. This does not indicate good independence between components but does reveal the data represent 4 cards dealt randomly. Each IC is a card that can be one of four suits with a value distributed normally with respect to the average or 4 random cards in simpler terms!!

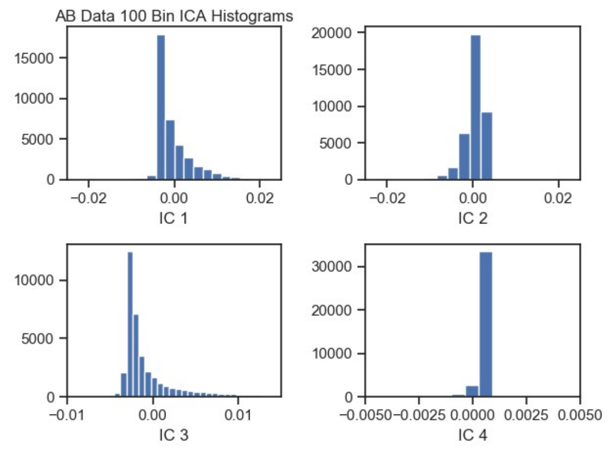


Figure 20 Airbnb IC Projection 100 Bin Histograms

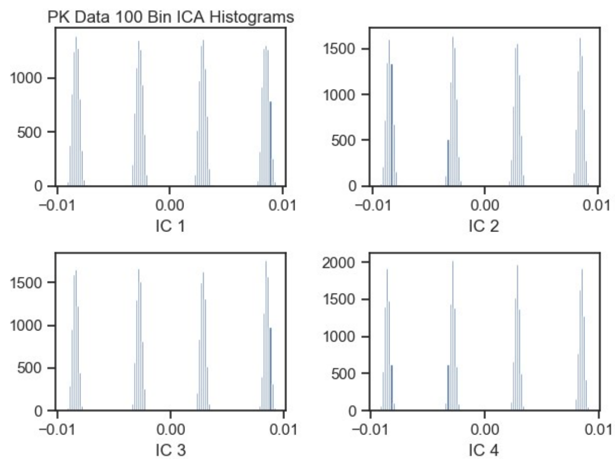


Figure 21 Poker IC Projection 100 Bin Histograms

RANDOM PROJECTION ANALYSIS:

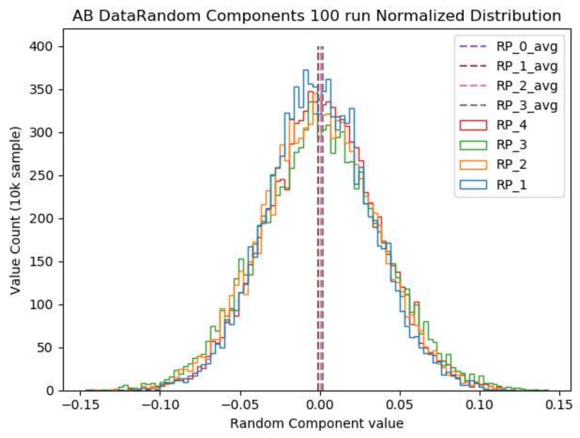


Figure 22 Airbnb Random Component Axis 100 run Histogram

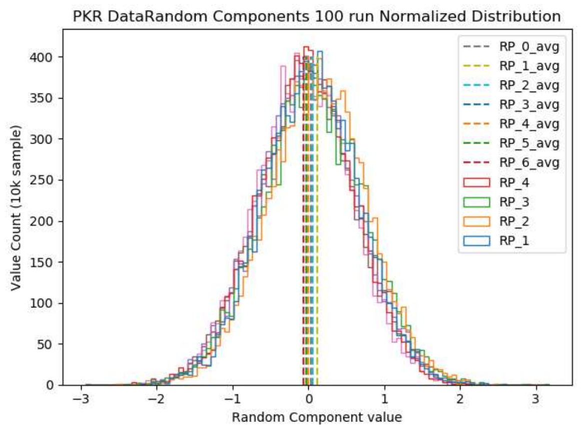


Figure 23 Poker Random Component Axis 100 run Histogram

RPA seeks to reduce dimensions via projections like PCA and ICA, but the projections are random. Projections are not easily evaluated, but the process is far faster than other methods. Variation in projections between runs was measured by taking the average and standard deviation of each point in a projection across 100 runs (32k points for Airbnb and 20k for Poker). The average of the average and average of the standard deviation was then taken for each entire component across all points. A normal distribution was fit to these values and plotted in Figures 22 and 23. As expected by the law of large numbers and high sample size, the components approach a normal distribution with a consistent average near 0 and spread across all components.

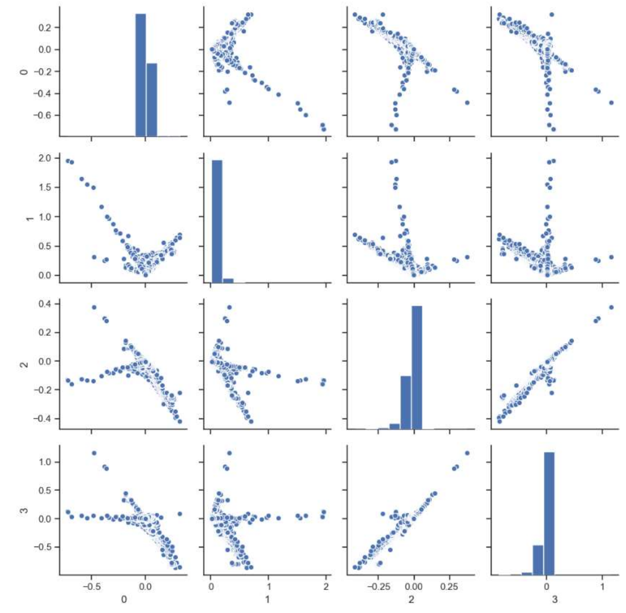


Figure 24 Airbnb Random Components Pair Plot (4 Components)

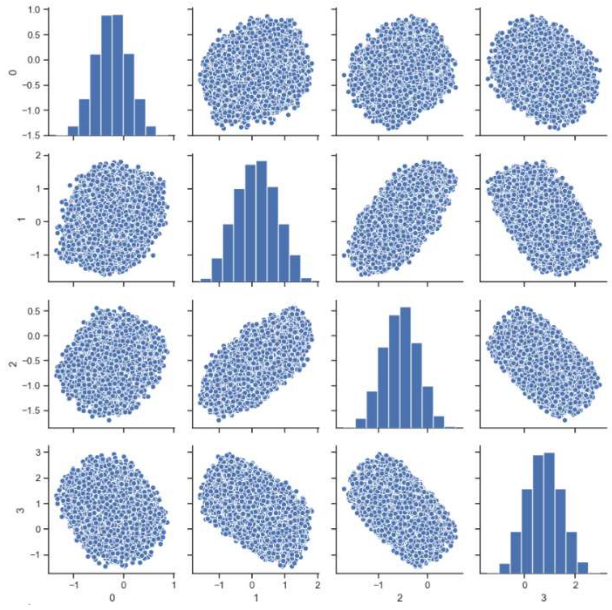
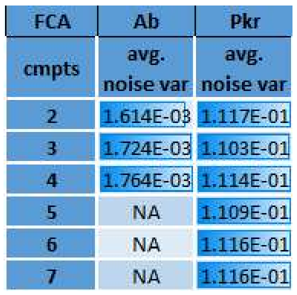


Figure 25 Poker Random Components Pair Plot (4 Components) – Gaussian Projections

Because the projections are random, no pattern is exactly expected. However, from the Airbnb data, a rotation of original features appears to occur in figure 24. This is rather like PCA rotations in Figure 17. Projections of the poker data are normally distributed and appear to be 2-D multivariate gaussian distributions from Figure 25. While not obviously interpretable, some of the random nature of suite and card value is being captured in the projections.

FACTOR COMPONENT ANALYSIS:

Table 8 Factor Analysis Average Component Noise Variance



FCA	Ab	Pkr
cmpts	avg. noise var	avg. noise var
2	1.614E-03	1.117E-01
3	1.724E-03	1.103E-01
4	1.764E-03	1.114E-01
5	NA	1.109E-01
6	NA	1.116E-01
7	NA	1.116E-01

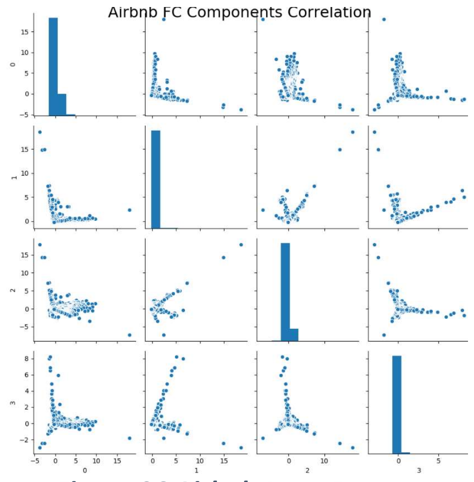


Figure 26 Airbnb Data Factor Analysis Pair Plot 4 Components

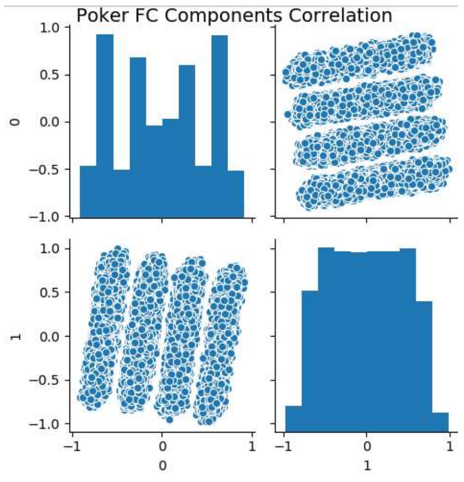


Figure 27 Poker Data Factor Analysis Pair Plot 2 Components

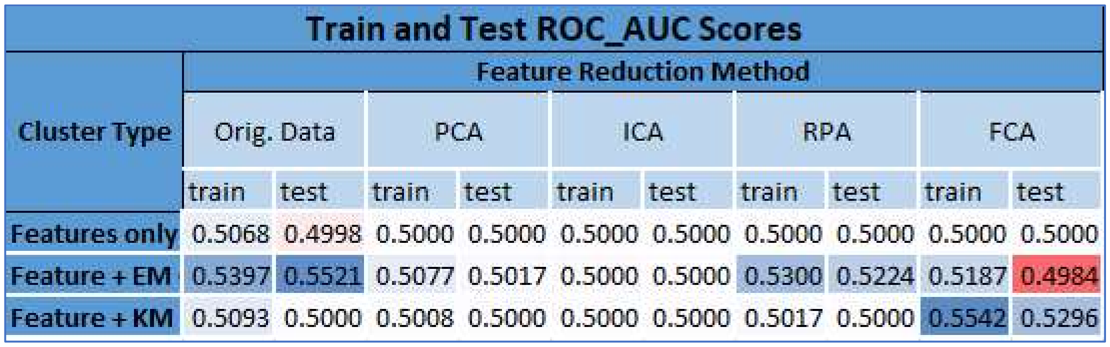
FCA Models the data as a continuous variable whose value is a normal distribution and estimates a multivariate normal distribution from the samples allowing for noise. Similar to PCA it attempts to minimize variance in component projections. However, where PCA assumes a diagonal covariance matrix among components FCA does not. Projections in FCA thus need not be orthogonal as in PCA allowing more expressive component relationship [SOURCE]. Notably, FCA still produces orthogonal projections in the case of Airbnb. Maximum average noise variance among components was used to choose 4 and 2 components on the Airbnb and poker data. Conceptually, this noise is the entire variance of the model.

FCA finds similar orthogonal projections to ICA and PCA on the Airbnb data suggesting reducing dimensions in Airbnb data is best achieved reducing the variance in along a single projection between any two features. FCA find 4 distinct gaussian distributions between 2 components visualized in Figure 27. These are similar results to ICA however, they are easily found with only 2 dimensions allowing greater dimension reduction of the original data. Interestingly, when comparing Figure 27 to the PCA projections in figure 19, PCA nearly models the same distribution between component 2 and 3, but values remains tightly grouped within the projection of the normal rather than random due to the orthogonality constraint.

NEURAL NET ANALYSIS:

A neural Net with 8x8x10 node layers previously tested on poker data achieved an ROC AUC test score of 49.9%; slightly worse than guessing the most frequent label [10]. For each dim reduction algorithm, a new Neural was trained and tested. The NN structure was adjusted to include 2 hidden layers with the number of reduced dimension and/or cluster instead of 8x8. All other properties of the NN were held constant. Number of clusters follows the results in table 4. Train/Test ROC was determined for the reduced data as well as reduced data + cluster. The results are summarized in Table 9.

Table 9 Neural Net ROC_AUC Dimensionality Reduction Results



Train and Test ROC_AUC Scores										
Cluster Type	Feature Reduction Method									
	Orig. Data		PCA		ICA		RPA		FCA	
	train	test	train	test	train	test	train	test	train	test
Features only	0.5068	0.4998	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
Feature + EM	0.5397	0.5521	0.5077	0.5017	0.5000	0.5000	0.5300	0.5224	0.5187	0.4984
Feature + KM	0.5093	0.5000	0.5008	0.5000	0.5000	0.5000	0.5017	0.5000	0.5542	0.5296

The Poker data is highly imbalanced with only 2.85% of the records being labeled "good hand." So the increase in ROC_AUC via clustering methods is significant improvement in overall performance. As a side not training and testing accuracy actually decreased by 2.16% and 1.14% respectively.

CONCLUSION:

Poker and Airbnb data were clustered via K-means then Expectation Maximization with cluster count chosen by inertia and BIC score respectively. Each dataset had their dimensions reduced by PCA,ICA,RPA and FCA. Reduction methods had their reconstruction loss compared with ICA,PCA, and RPA loss increasing with dimension reduction. FCA had the opposite behavior. New dimensions were chosen based on explained variance, kurtosis, and average noise for PC,ICA and FCA respectively. New Clusters were generated for all reduced poker data. A previously trained neural net was retrained and test on the new data with general improvement in the ROC_AUC score. Most notably, test roc went from 49.98% for the original data to 52.96% for FCA reduce data with 18 EM clusters. Future investigation would include run and computation time of the reduction and clustering methods.

Andrew Cudworth
CS 7641 Machine Learning (Spring 2020)
Assignment 3 Unsupervised Learning

SOURCES

[1] Hyvärinen, A. (2000). Constraint Independent Component Analysis Algorithms and Applications. doi: 10.1002/9781118679852.ch12
[2] 6.6. Random Projection¶. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/random_projection.html
[3] Derksen, L. (2019, April 29). Visualising high-dimensional datasets using PCA and t-SNE in Python. Retrieved from https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b
[4] Sarkar, T. (2019, September 6). Clustering metrics better than the elbow-method. Retrieved from https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6
[5] Selecting the number of clusters with silhouette analysis on KMeans clustering¶. (n.d.). Retrieved from https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py
[6] 2.3. Clustering¶. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/clustering.html#k-means
[7] Hincks, I. (2019, October 1). matplotlib - making labels for violin plots. Retrieved from https://stackoverflow.com/questions/33864578/matplotlib-making-labels-for-violin-plots
[8] PCA projection and reconstruction in scikit-learn. (2019, March 19). Retrieved from https://stackoverflow.com/questions/36566844/pca-projection-and-reconstruction-in-scikit-learn
[9] Fodor, I. K. (2002). A Survey of Dimension Reduction Techniques. doi: 10.2172/15002155
[10] Cudworth, A. T. (n.d.). *Cs 7641 Project 1*.
[11] 2.3. Clustering¶. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient
[12] 2.5. Decomposing signals in components (matrix factorization problems)¶. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/decomposition.html#fa