Unsupervised Learning and Dimensionality Reduction

Yi Zhao

[Yzhao644@gatech.edu](mailto:Yzhao644@gatech.edu)

# 1. Introduction

This paper discusses how to use unsupervised learning algorithms to identify extra information from dataset. Also discussed is using feature reduction techniques to select or project existing features to different dimensions to provide better representation.

In both problems, a key question to answer is how to choose the right number for clustering and feature reduction. I used elbow method with metrics like Silhouette score, Akaike information criterion, Inertia (without touching the ground truth label) and Adjusted\_mutual\_info\_score (used to measure how much cluster is related to ground truth labels) for clustering. For feature reduction, I used Variance distribution, Kurtosis, Transformation error to find the right number to reduce the features to.

The clustering method I used are KMeans and Expectation Maximization. The feature reduction techniques are Principle Component Analysis, Independent Component Analysis, Randomized Projection, Decision Tree Feature Importance.

For the second part, I integrated the clustering label and reduced dimension data and use the newly constructed data feed a neural network to compare the performance with original dataset in terms of training speed, model accuracy and precision and recall.

The dataset I used are Adult dataset that predicts individual’s income level with census data and Wine dataset that measures wine’s quality by wine’s chemical features.

Keyword: Unsupervised Learning, Feature Reduction

# 2. Dataset

The first data is obtained from UCI Adult Income Dataset. The data is extracted from 1994 Census database. The task is to predict whether a person will make more than 50K a year.

The features include many geographic information, like age, race, education, origin country etc.

What I like about the data set is there are decent amount of data points, about 30k data points. For feature wise, it has a mix of categorical data and numerical data. After combining the countries and removing the not available values and standarized, the final data is has a dimension of 59.

Second data are the results of a chemical analysis of wines quality.

The goal is to predict the quality of the wine by providing a score ranging from 3 to 8.

The features include chemical levels of ingredients from the wine, like alkalinity of ash and alcohol level.

Since the features are all numerical, it is a great data for clustering. The total feature size is 12.

# 3. Clustering

3.1 KMeans and Expectation–maximization algorithm

Kmeans is using distance to decide the cluster for each data points. Starting with initial center points, it collects the close points and update to new center points. It converges when no update happens.

The nice thing about using distance is the speed, as it’s fast to calculate Euclidean distance between two data points. The downside of is it’s hard to decide which cluster it belongs to when distance is same to both group. Initialization would have a big effect on how the algorithm converges. Also, it suffers when the features are not scaled to unit value, as it might put more emphasis on one over another feature. Also when the cluster is not in convex shape, it’s hard for the algorithm to work. So proper standardization and later feature reduction is recommended for Kmeans algorithm.

Expectation–maximization algorithm is a two-step algorithm. First to have an expectation about clusters, for example mean and variance for a Gaussian distribution. Next is to compute the parameters that maximize the expected log-likelihood. Since it’s using MLE to get the best data, the training time is longer than Kmeans but it will have a better understanding underlying data by of the fitting model with the best parameter.

3.2 Metrics to consider

As mentioned earlier, the key point of the problems is how to decide number of clusters. In 3.1, I discussed the difference between EM and KMeans. There are certain metrics that measures distance well, some measure likelihood better.

Silhouette analysis is based on the distance of the data point, it is very friendly to linear based clustering such as K-Mean. As a measure strategy, its performance on density natural data might not be very ideal.

Inertia is another good metric for KMeans which is essentially a Squared Error metric checking the point’s distance with the cernter.

For EM, I used Akaike information criterion as metric. AIC is calculated with number of parameters – likelihood. So the smaller, the better.

With the above metrics, I used Elbow method to determine the K, which is where the curve goes to flat.

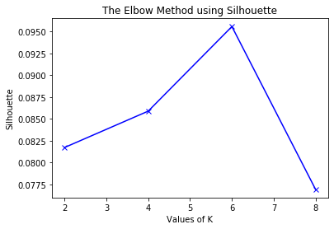
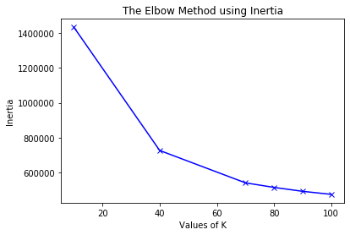
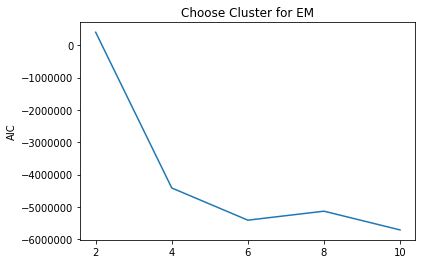
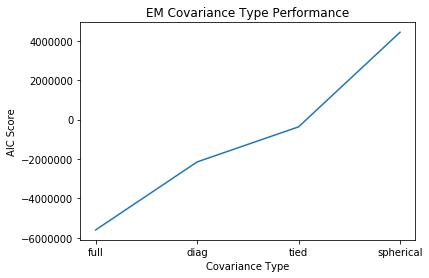
I used SKlearn’s KMeans and GaussianMixture for clustering method. For EM there’s another hyper-parameters to discuss, covariance-type. Different choice determines how much we care about the covariance between different clusters and feature correlation within one cluster.

3.4 How the K is selected

My intuition is that in the dataset, there is a feature native\_country, which could be a great candidate for clustering. So I started to find clusters smaller than 10. Also having a lower K is nice to visualize the data.

Target label is excluded when training for clustering.

As discussed in 3.3, four elbow methods are implemented to decide the best parameters, 2 for reach. Result as follow.

For Kmeans algorithm, Inertia always decreases as cluster number increases. With Sihouette metric, the best cluster is determined as 6.

For EM algorithm, cluster is also best at 6. I also compare variance type vs metrics. With ‘full’ as covirance\_type, the EM performs the best. This makes sense, with many one hot encoding features and feature engineer to remove duplicate or highly correlated features, there’s not much correlation between features for our dataset. So with full covariance, the model can better measure the correlation between internal feature and outer clusters.

3.5 Result analysis (cluster match, some cluster has most of the positive, show distribution or number of 1 and 0s)

I use three methods for result analysis. First to measure mutual information between cluster and target label. I used adjusted\_mutual\_info\_score to check that. My intuition is that the cluster doesn’t have much relation to the target variable, as features describes census information, which not necessarily related to the income exactly. It makes more sense for the data to represent cluster by geographic than income itself. Plus as the best clusters are determined as 6 while target label only has two value, it’s unlikely they match each other.

My scores are about 0.1318 for both algorithms which is inline with my idea.

Secondly, I check overall distribution of point to each clusters, and also target label with clusters.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| EM0 | 4561 | 5815 | 5976 | 1752 | 6609 | 7 |
| EM1 | 544 | 415 | 935 | 250 | 5697 |  |
| Kmeans0 | 16037 | 6960 | 7 | 13 | 1643 | 60 |
| Kmeans1 | 1817 | 5783 |  | 10 | 191 | 40 |

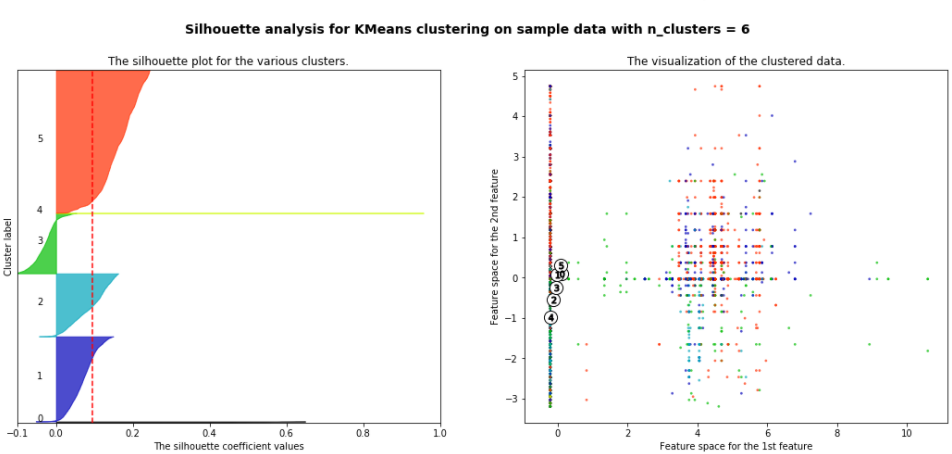
The first insight is Kmeans Cluster 0 and EM cluster 0-3 has about the same distribution of the target variables. Changes are higher for the point to be labeled as 0 if they are assigned to those groups.

Kmeans cluster 1 and EM cluster 4 has about the same distribution. And they will like to be labeled as positive then other cluster, but still only 50% chance.

Also the mapping above is have the same label, meaning Kmeans make what EM 4 cluster as 1, while EM makes kmeans cluster 0 to 0-3 separately.

Cluster 2 and 5 are also the same 7 individuals.

Lastly I visualize my Kmeans result. The center are all located to 0 (capital gain). This will be used to compare with next section when ICA and PCA has a better representation of data.



# 4. Dimensionality Reduction

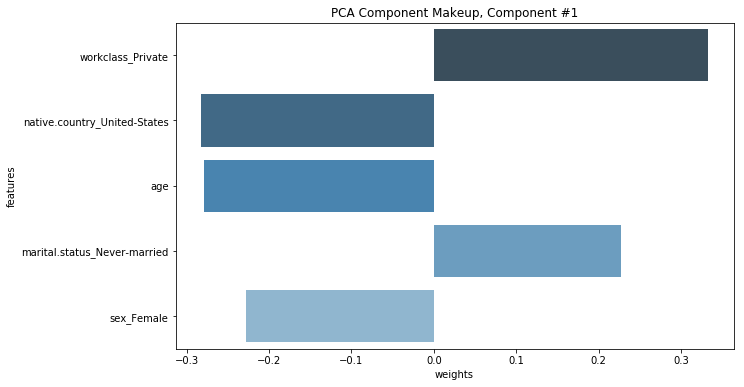
# Increase dimensions has many benefits. Firstly, one hot encoding would transfer categorical data to numerical data, which is basically required for every machine learning model. Secondly, data could be only separatable in higher dimensions.

# But in higher dimension, calculation could be slower; noisy feature could cause high variance; sparse matrix could cause other calculation and performance problems. This curse of dimensionality results necessity for feature reduction.

4.1 Principle Component Analysis

PCA use SVD techniques to change original data to eigen vector dimensions, so that original information could be represented with less features but as much (measured by total variance percentage) as possible.

I used sklearn’s PCA function and PCA.explained\_variance\_ratio\_ to find the best number of features. In my experience, 80% is a good representation of the original information. In our case, I need 34 features to achieve that.

One thing to notice is that largest eigen value only takes about 18% information and rest is less than 10%. This could be a sign there aren’t much noise in the features and even with transform, there aren’t many decisive features.

I also visualize the first and most important features.

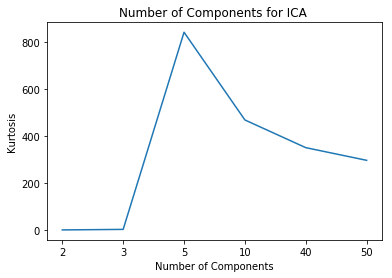
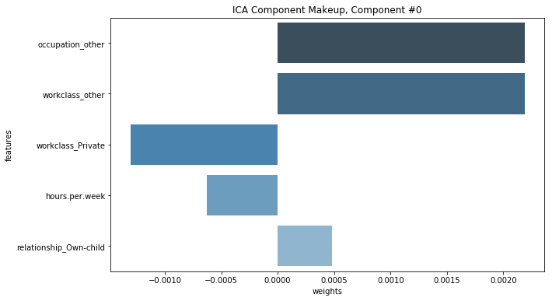
So the most important eigen vector consists of work\_class, native\_country and age, which makes sense to me, as this targeting US, elder single employee that works for private institution.

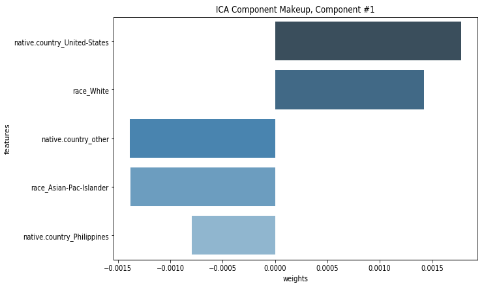
In the later tree importance, this will be revisited with more proof.

4.2 Independent Component Analysis

ICA is also using linear combination of original features to map them into a new dimension. The assumption is that the original data is sum of independently non-Gaussian distribution. So the metric to decide the number of features to keep is Kurtosis of the transformed data.

I used elbow method again to get the number as 5.





Also a visualization of first two ICA components are shown. Occupation and Work class is evident that component is talking about working. Second component is talking about geo-location of the individual.

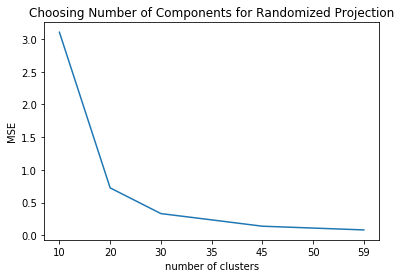
Geo-location and working are quite independent.

4.3 Randomized Projections

RP randomly selected features and use Johnson-Lindenstrauss lemma, to preserve the distance from higher dimensions in order to reduce dimensions.

The metric I used is the Mean Squared Error between transformed matrix and original matrix. The transformed matrix is calculated by GaussianRandomProjection.components times output of the output.

I run the RP multiple times and take the average of the errors.

Due to the simplicity, RP is faster to train than other algorithms. I use the same components and dataset to gather the time spent. RP is at least ten times faster than other two.

|  |  |  |  |
| --- | --- | --- | --- |
|  | RP | FastICA | PCA |
| Seconds | 0.014 | 0.248 | 0.156 |

4.4 Decision Tree Feature Importance

# 5. Clustering after Feature Reduction

Cluster is easier after feature reduction. As it’s projected to new dimensions that are independent or orthogonal to easch other. Easier to map to a cluster.

It can be shown with visualization and silhouette\_score (kmeans)increase with same number of clusters.

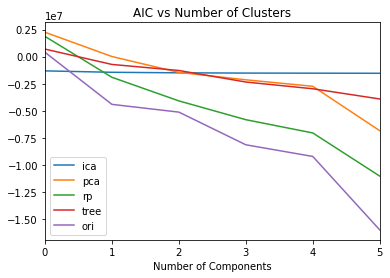
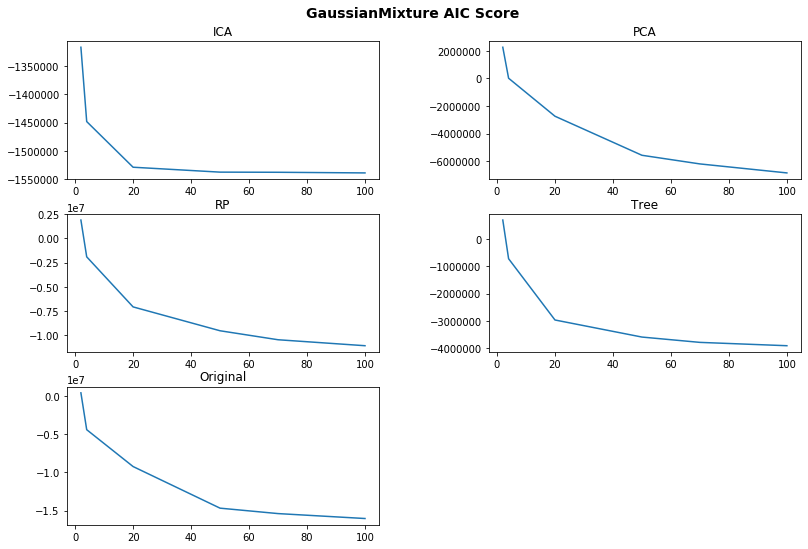
Except trees, as it’s not changing anything. (don’t show trees)

5.1 elbow again

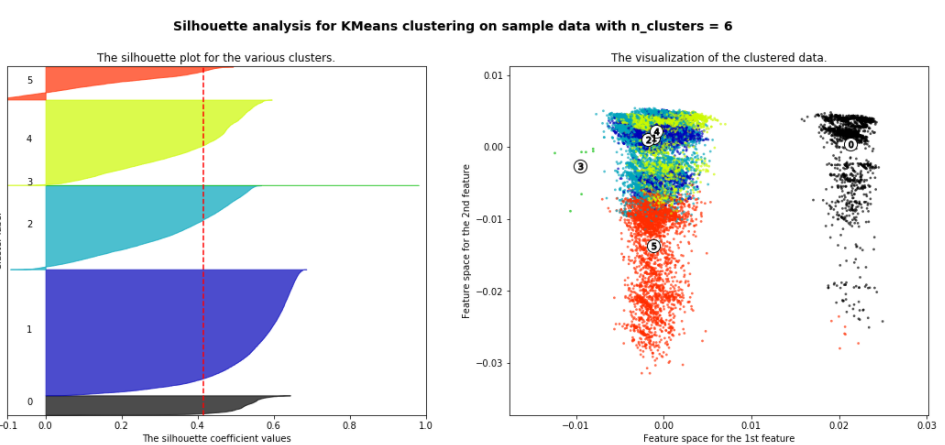
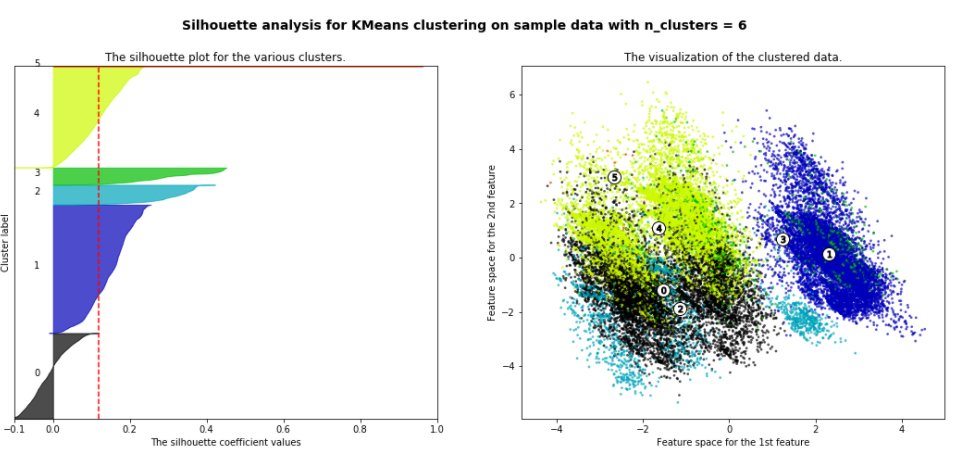
For kmeans tree and ori choose about 20 clusters, others about 50. Inline with em result.

For EM, it’s similar effect, as one can see ica has a sharp drop in the 2-20, as it doesn’t need many clusters to reach the elbow point, whereas pca and rp needs about 50 clusters to reach the turning point.

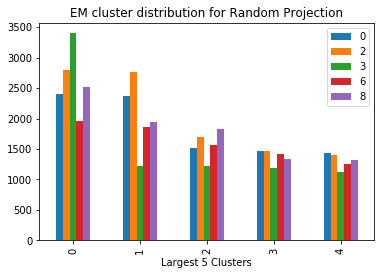
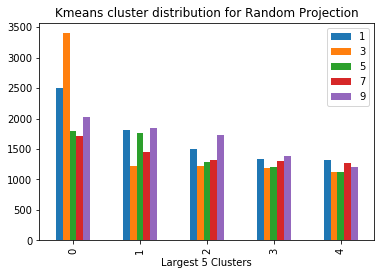
Best number of clusters has a relationship with number of clusters. If bit wise feature, number of clusters = 2\*\*n

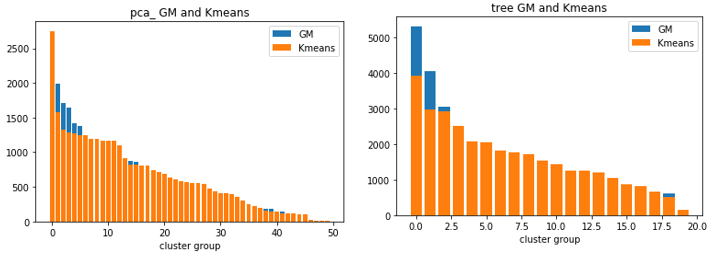
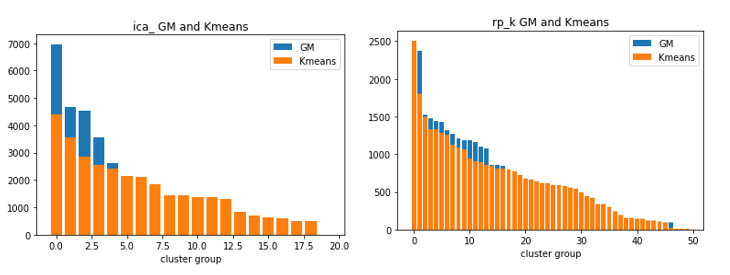
 

5.2 show visualization

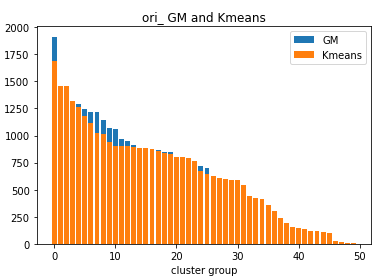
 

Rp for clustering when running 5 times.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ICA EM | ICA KMEANS | PCA EM | PCA KMEANS | TREE EM | TREE KMEANS | ORI EM | ORI KMEANS |
| 0.244 | **0.337** | 0.204 | 0.256 | **0.275** | 0.276 | 0.209 | 0.232 |



a. Distribution similar b. reduced features has more data points in the first few clusters. They are clustered better then original. (2000 vs 2500)

Sihoulle score also show that.

# 6. Neural networks with feature reduction

Variance reduction in pca is not significant, meaning that there are not much noise in the features to reduce.

Clustering info could be learnt through neural net, (why accuracy not improved)

Best feature only origin need two layers

Cluster doesn’t help

# 7. neural networks with cluster

Summary

Reference

<https://en.wikipedia.org/wiki/Akaike_information_criterion>

<https://www.researchgate.net/post/How_can_I_test_the_GMM_clustering_result_using_a_measure_different_to_BIC_Is_appropriate_to_use_Silhouette_beyond_to_BIC>

Cluster is easier after feature reduction. As it’s projected to new dimensions that are independent or orthogonal to easch other. Easier to map to a cluster.

It can be shown with visualization and silhouette\_score (kmeans)increase with same number of clusters.

Except trees, as it’s not changing anything.

For kmeans tree and ori choose about 20 clusters, others about 50. Inline with em result.

For EM, it’s similar effect, as one can see ica has a sharp drop in the 2-20, as it doesn’t need many clusters to reach the elbow point, whereas pca and rp needs about 50 clusters to reach the turning point.

More feature does end up with a lower aic score. But sihousette score is low since it’s not very sepratable.

Best number of clusters has a relationship with number of clusters. If bit wise feature, number of clusters = 2\*\*n

Rp has the lowest aic, meaning it’s matching the distribution well, but the si score is lowest, meaning it’s not cluster well?

Check speed of running algorithms. Rca should be faster than others.

Clustering info could be learnt through neural net, so use original data clusters

Visualization is different between original and feature reduction.

Try different variance for EM (more is not necessary better, but we will stick with it.)

Variance reduction in pca is not significant, meaning that there are not much noise in the features to reduce.