Unsupervised Learning and Dimensionality Reduction

Spring 2021 CS7641 Assignment 3

Hung-Hsi Lin hlin83@gatech.edu

ABSTRCT

This paper aims on implementing and analyzing two unsupervised clustering algorithms, K-means and expectation maximization, and four dimensionality reduction algorithms on two datasets. Then the processed datasets are implemented with Neural Networks supervised learning algorithm for accuracy study and analysis.

INTRODUCTION

The paper will be divided into 3 parts, first, the clustering algorithms: K-means and Expectation Maximization (EM) are applied on two datasets to see how these two algorithms perform in clustering instances. The first diabetes dataset is as same as assignment1, the instance size is 15000 and have only 9 features. The second one is phishing website with 11054 instances, 32 features and 2 classes. The output in both cases is binary, so they are classification problems. In second part, four algorithms for dimensionality reduction are studied, including Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projects (RP) and Information Gain (IG). And the clustering experiment will be reproduced again with the data being processed by these different dimensionality reduction methods. Finally, the clustered features with processed dimensionality reduction will be fed into Neural Networks and compared with the performance in terms of prediction accuracy and runtime.

SECTION 1: CLUSTER

The cluster algorithm is to gather the instances with same output together into the same group, and there are two different concepts get introduced in this section. First, in K means method, we target on separating n instances into K groups or clusters by minimizing the sum of squared Euclidean distance between each instance and the cluster it belongs to [1]. This is a hard clustering algorithm while each data can be only assigned to one cluster. On the other hand, EM enables soft clustering by calculating the probability in Gaussian distributions for each point on all clusters and the likelihood parameter will be iteratively maximized between the updating expectation step and maximization. [2]

In order to study how these two clustering algorithms work in real dataset, I implemented them with Scikit-learn module on two dataset used in assignment 1 and 2, and then compare the clustering performance, which is concluded in Figure 1. Before going details on the results, first thing should be clarified is how do we define if the clustering works well or no. For K means, we could use squared mean errors as indicator, which means the total distance between the point and its cluster centroid, lower means better clustering. For EM, likelihood was calculated, higher means the points is highly possible belong to the group it's assigned to. In Figure 1.a and 1.c, we could see the squared errors (blue lines) decreasing as the number of clusters is higher for both datasets. It means if we create more cluster, the instances could be more accurately assigned. Here we met a dilemma, we want the squared errors to be lower for better clustering accuracy, but we prefer the number of cluster to be as low as possible for

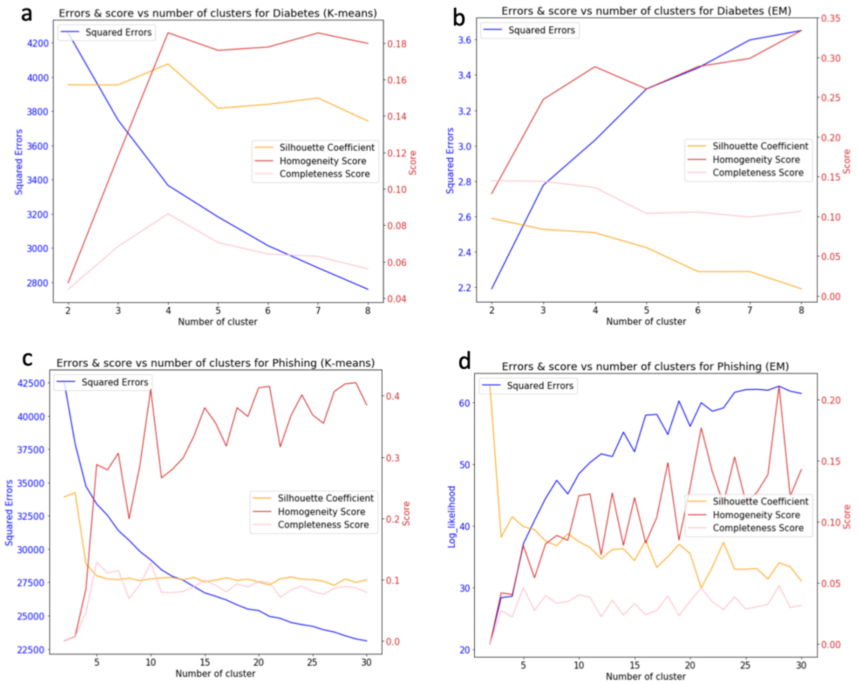

Figure 1 – Squared Errors and indicators analysis by using (a) K means cluster and (b) EM algorithm on diabetes and (c) K means and (d) EM algorithm on phishing datasets.

dimension reduction. Therefore, in general, I will look for the point where the squared error drops dramatically, and then become flattened afterwards. From 1.a and 1.c, cluster of 4 for diabetes and 3 for phishing looks like the points we are looking for. Same story for 1.b and 1.d while using EM algorithm, we are looking for a point where likelihood jumps dramatically and becomes more flattened. Just like squared errors, it is also hard to define the best point though.

Therefore, other indicators are required for the ease of analysis. Completeness score (pink) is the indicator shows the degree on if instances with same labels are all assigned into the same cluster. Homogeneity score (red) shows if each cluster contain only one type of label. And Silhouette coefficient (orange) represents how similar the instance belongs to its cluster centroid compared to others. For all these indicators, higher value means better clustering performance, so I plot all three in the same figure at the right side of y-axis. In diabetes dataset, EM performs better than K-means algorithm in most of indicators, demonstrate the fact that for this dataset, there might be a lot of point at the border of multiple clusters, make the hard-clustering performs worse. I also run the K-means multiple times randomly, and the results could vary a lot, showing the fact that K-means might get stuck in local optimum point and the performance is highly dependent on the initial cluster generated. For phishing dataset, the conclusion is reversed, looks like the K-means method has better performance as the number of clusters is higher. In my opinion, the reason is that EM has to calculate the probability of distribution on each point with all clusters, as the number of cluster is higher, then it's hard for the algorithm to accurately calculate which cluster has the highest likelihood for each instance. As the number of cluster is small, the performance of EM is similar to K-means for phishing dataset.

SECTION 2: DIMENSTIONALITY REDUCTION AND CLUSTERING

In this section, PCA, ICA, RP and IG will be studied independently first to study the theory behind them and how to properly utilize these algorithms. Then all four will be combined with clustering mentioned before and the performance on clustering will be analyzed.

2.1 Principal Component Analysis

PCA first calculate principal components and utilize these components to re-project the original data. By doing it iteratively, the features could be reduced into lower dimension by using fewer orders of principal component as new axis. [3] Therefore, the first step is to determine how many dimensions we could reduce to. In figure 2.a and 2.d, the calculated eigenvalues on different numbers of principal components (PC) can be seen. The optimun case is first few PCs exhibit most part of eigen values, so we could rule out the others in new feature sets. The blue bars shows that in diabetes, most of features play important roles, and for diabetes, we could rule out some contributions from lower value of PCs. The red line shows the calculated mean squared errors under different selection of PC numbers, and we are also looking for the point showing dramatic drop then flattened. To visually inspect how the clustering works, I plot the figure using 2 PCs in 2.b ad 2.d, and it seems like the it does decent jobs but not that good in both datasets. As it plotted in 3-D using 3 PCs, shown in figure 2.c and 2.f, the clustering works well in diabetes dataset. And for phishing dataset, it didn't look that good in separating different classes, which means more PCs required for better clustering. Based on the preliminary analysis, number of PC of 6 and 20 are selected for diabetes and phishing dataset, respectively. The implementation of PCA with EM and K-means will be discussed at the end of this section.

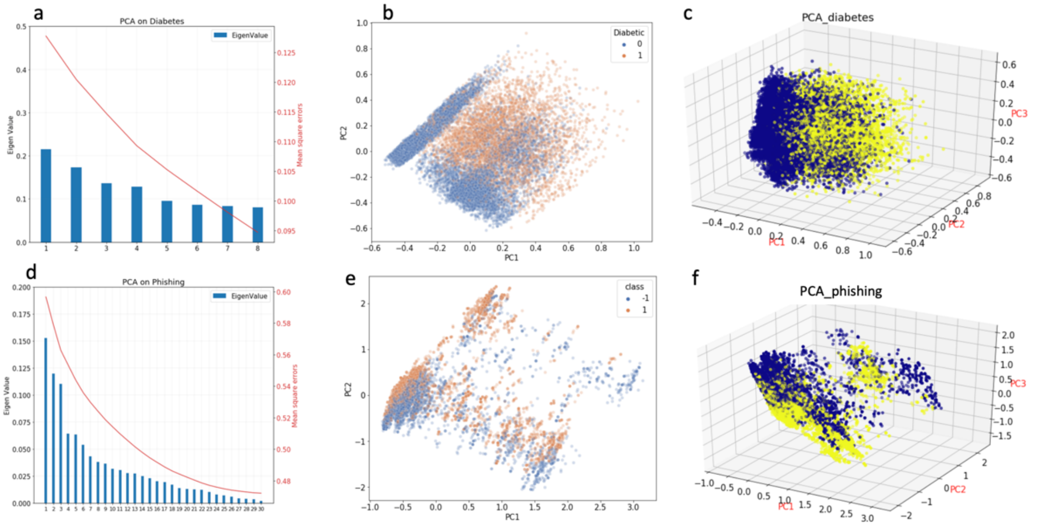

Figure 2 – Eigenvalue calculation for different number of principal components on (a) Diabetes and (b) Phishing datasets. Plot of clustering result using 2 principal components on (c) Diabetes and (d) Phishing datasets, and using 3 principal components for (d) Diabetes and (e) Phishing datasets.

2.2 Independent Component Analysis

The algorithm of ICA tries to maximize the difference between components, and back calculate all the independent component (IC), or Kurtosis to reconstruct the original data. [4] FastICA in sklearn module was used and the results are shown in Figure 3. In 3.a and 3.d, I first tried to reconstruct the original dataset using different number of Kurtosis. Just like PCA, as the number of Kurtosis increases, the mean squared error decreases, but dimensionality reduced is less, there's an optimum point which needs to be defined. For the clustering plot in b,c,e and f, seems like the clustering didn't do a good job in both cases, means the original dataset is hard to be reconstructed using just 2 or 3 independent components. More components are necessary for rebuilding the original dataset. Here, in order to reduce the dimension, the number of IC selected for diabetes is 6 and 22 for phishing dataset. After combining with K-means and EM, the results will be shown later.

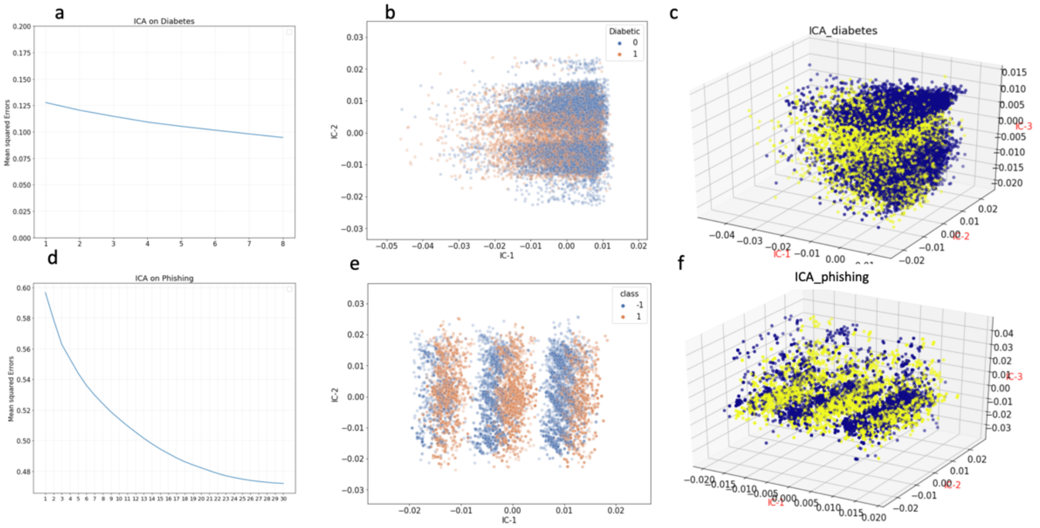

Figure 3 – Mean square error for different number of Kurtosis on (a) Diabetes and (b) Phishing datasets. Plot of clustering result using 2 ICs on (c) Diabetes and (d) Phishing datasets, and using 3 ICs for (d) Diabetes and (e) Phishing datasets.

2.3 Random Projection

RP reduces the original dataset dimensions into lower dimensions generated from random Gaussian matrix. [5] Here, I implemented the random_project method from sklearn on two dataset and the results are summarized in figure 4. First, the mean square errors are calculated under different settings on dimension for RP algorithm. For diabetes, the MSE dropped a lot at dimension of 3 then slightly getting flattened afterwards, and for phishing dataset, the similar condition was found at the dimension of 19. For 2-d and 3-d clustering results for both datasets, we can clearly see that the RP did a decent job, but not as good as PCA shown in figure 2. Means that few dimensions of random Gaussian misture is not enough to represent the original data for these two datasets. The comparison and also the implementation with K-means and EM again, will be introduced in later section.

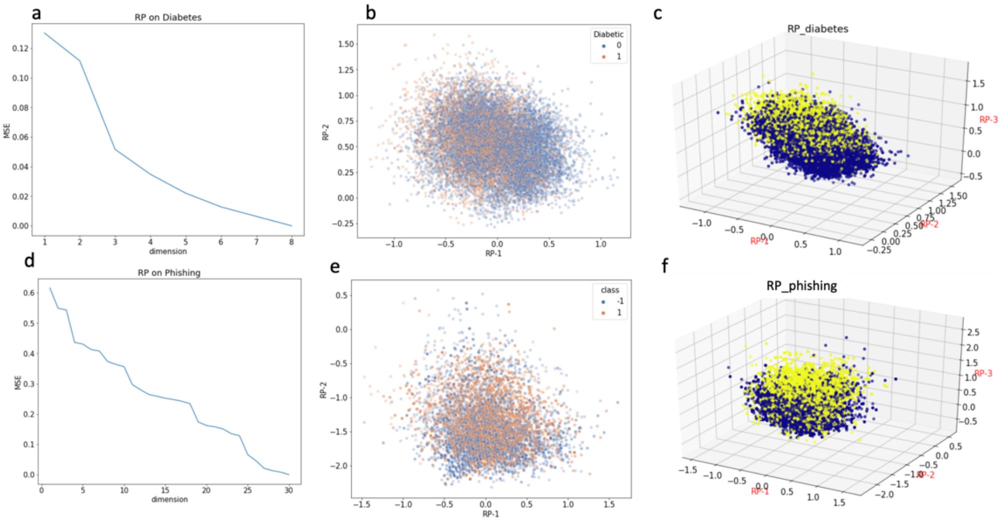

Figure 4 – Mean square error for different dimensions using RP on (a) Diabetes and (b) Phishing datasets. Plot of clustering result using 2-dimension on (c) Diabetes and (d) Phishing datasets, and using 3-dimension for (d) Diabetes and (e) Phishing datasets.

2.4 Information Gain

Just like the information gain learnt from supervised learning part, I select a tree-based feature selection for dimensional reduction. The idea is to import tree-based structure, and see which features play more important role in determining the final classification. [6] In Figure 5.a and 5.b, I first plotted all the information gain, which is equal to the importance of each feature in both datasets. From diabetes, we could see that the 1st and 8th feature weights slightly more than other features, but in general, all feature are equally important in diabetes dataset, while the conclusion made here is similar to the discussion in previous PCA section. On the other hands, for phishing, it's clearly showing that two of the features weights a lot more in determining the classification on the instance, means there are a lot of features in phshing dataset could be pruned without affecting too much on prediction accuracy. To apply this IG algorithm with K-means and EM, I select the top 6 features in diabetes, and top 10 features in phishing dataset for the comparison discussed in later section.

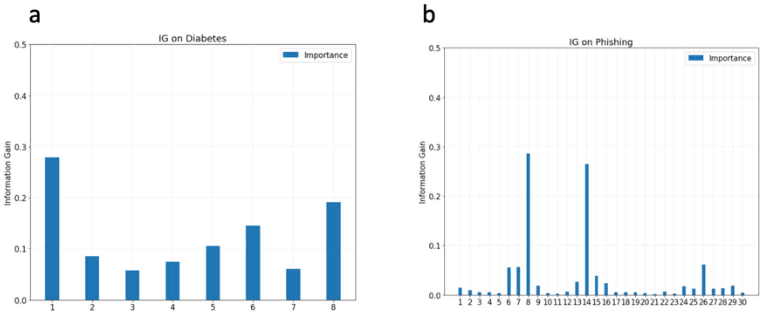

Figure 5 – Information gain for the features using IG on (a) Diabetes and (b) Phishing datasets.

2.5 Comparison of dimensionality reduction and clustering algorithms

I plotted the figures just like Figure 1, including squared errors, log max likelihood, Silhoutte coefficient, homogeneity score, and completeness score for each combination of dimensionality reduction and clustering algorithms. The detail on each graph could be found from the links provided in README file, and the results are summarized in the Table I for diabetes and Table II for phishing dataset. First, for PCA with K-means and EM in diabetes, there's not too much difference in all indicators, but in phishing dataset, the K-means outperforms the EM algorithm. The reason may result from the fact that there are too many clusters and the instance might have close distance with each cluster, making the likelihood calculation ambiguity. For ICA, there is also not too much difference between two clustering algorithms, one thing should be noted is that the performance of ICA is worse than PCA, implies that we can't easily rebuild the original dataset with just few new created features in both datasets. For RP, the performance is similar to PCA, in both two datasets and clustering algorithms, implying that the random Gaussian matrix also does a decent job compared with PCA, which is more expensive in calculation. Regarding to IG, it also works surprisingly good as PCA and RP. I first ran the information gain calculation on all features and select the important ones for clustering. It ends up working decent and comparable with other dimension reduction algorithms. It could be explained that the features in the dataset are already pretty important and exhibiting low noise, so there's no big difference observed between using good features (IG) or using the reconstructed and optimized features (RP, ICA, PCA).

Table I. Dimensionality Reduction + Clustering algorithm on Diabetes dataset

Algo	PCA		ICA		RP		IG	
Algo	Kmeans	EM	Kmeans	EM	Kmeans	EM	Kmeans	EM
Silhouette coefficeint	0.18	0.15	0.14	0.1	0.25	0.26	0.24	0.12
Homogeneity Score	0.178	0.3	0.17	0.28	0.07	0.1	0.18	0.14
Completeness Score	0.09	0.1	0.07	0.12	0.03	0.07	0.07	0.25

Table II. Dimensionality Reduction + Clustering algorithm on Phishing dataset

Algo	PCA		ICA		RP		IG	
Algo	Kmeans	EM	Kmeans	EM	Kmeans	EM	Kmeans	EM
Silhouette coefficeint	0.12	0.07	0.07	0.03	0.15	0.12	0.13	0.12
Homogeneity Score	0.26	0.05	0.12	0.2	0.25	0.20	0.25	0.25
Completeness Score	0.12	0.02	0.15	0.07	0.1	0.08	0.12	0.13

3 NEURAL NETWORK PERFORMANCE

Finally, all dimensionality reduction and clustering algorithms are introduced to Neural Network (NN) and performed on both datasets. The NN classifier setting for all cases are the same, with [250,500] hidden layer, relu activation function, and learning rate of 0.05, and other parameters are default. The runtime is the model training time, and the testing accuracy is calculated from cross-validation f1 scores for 10 times. First, the original NN was ran as a reference using original dataset. And the datasets were applied for two clustering algorithms, then fed into NN for calculating the prediction accuracy and runtime. In Figure 6.a and 6.b, we could see the results from both datasets. First, compare with original NN, K means and EM, in both clustering cases, the NN using original dataset with accuracy of 92.5% and 96% (red bars) outperforms a lot than either using K-means (72%, 63%) or EM (74%, 58%) as the clustering algorithms. It means that most of the features are important there might be a lot of instances locating in ambiguous region, makes the classification harder for both clustering algorithms. In this case, we can't just directly utilize the clustered features in Neural Network analysis. The runtime for K-means is more than EM, and if we selected fewer cluster centroids for clustering algorithms, the runtime could be shorter, but the accuracy might be worse, depending on the distribution of data points and the noise of dataset.

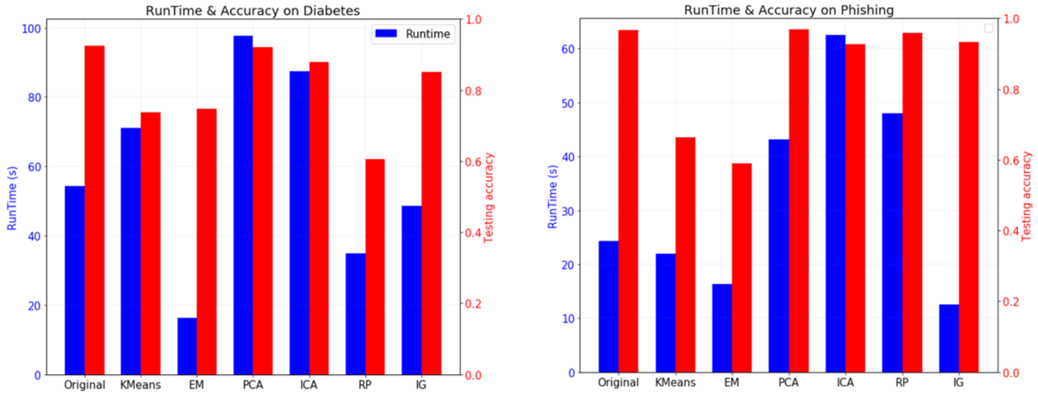

Figure 6 – Information gain for the features using IG on (a) Diabetes and (b) Phishing datasets.

Based on the previous conclusion, we need to implement dimensionality reduction algorithm to optimize the features used in NN. The results for the diabetes dataset are concluded in Figure 6.a by applying 4 dimensionality reduction ways with K-means. PCA performs pretty well compared with original settings. It could be expected because in section 2.1, we already saw that first few principal components could do a good job in classifying clusters. And the size of features is not bad, therefore PCA seems like a good option for these two datasets. The performance of ICA is a little bit worse, because the number of independent components selected in this model is not enough to rebuild the original dataset. If more IC could be introduced, I believe the ICA should perform way better because it has the power to deduct the original theory on the datasets (more expensive in time though!). IG works surprisingly good to me, demonstrates the fact that the original features provided in the dataset are already really representative, there's not too much need for further optimization for both cases. The worst accuracy is shown is the RP, and it works relatively worse than PCA as the dataset features are not many. The training runtime for RP is way shorter than other algorithms, so RP may be the proper candidate while the feature size is large, if the size is small, like in diabetes cases (only 8 features), it performed pretty bad.

For phishing dataset in 6.b., we can clearly see that the accuracy for all algorithms besides pure K-means or EM are pretty close and good. In IG section, I mentioned that in this phishing dataset, there's 10 features weight a lot more than other 20 features, and that's why in all four algorithms, we don't have to pick a lot of components for training and eventually still get good performance. More clusters intend to increase the time exponentially because of curse of dimensionality, which needs to be properly selected though. In the case of PCA + K-means, the accuracy is even higher than the case using original dataset for training NN model. It demonstrates the fact that the prediction performance could possibly be better with proper dimensionality reduction and clustering applied, by removing the noise and provide useful information with less feature sizes.

4 REFERENCES

1. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
2. https://machinelearningmastery.com/expectation-maximization-em-algorithm/
3. https://en.wikipedia.org/wiki/Principal_component_analysis#:~:text=Principal%20component%20analysis%20(PCA)%20is,components%20and%20ignoring%20the%20rest.
4. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html
5. http://users.ics.aalto.fi/ella/publications/randproj_kdd.pdf
6. https://machinelearningmastery.com/information-gain-and-mutual-information/