Assignment 3: Unsupervised Learning and Dimensionality Reduction

[Mingrui Sun]
[msun333@gatech.edu]

I. Introduction - Dataset Explanation

In this work, we reused the same two datasets from HW1 to explore the power of clusting method and dimensional reduction technique. The UCI adult income dataset([KO20]) and beans classification dataset([Koh96]) were used. What makes these dataset great for this work are the following:
1)Both of them have high dimension(a lot of features) so power of dimensional reduction method can all being shown. In addition to that, both of these datasets are imbalanced.
2)Adult dataset is a mixed features dataset(contain both numeric and categorical features) with 14 feature and a binary label of whether each entry/person earn more than 50k yearly income or not. This mixed features nature make the choice of euclidean distance(what the sklearn implementation of Kmeans and EM/Gaussian Mixture Model used) unsuitable. This make adult dataset interesting as it can be used to distance problem with distance metric select.
3)Beans dataset contained all numerical features(16 of them), so in this space euclidean distance would make sense.

II. Clustering for the Original Dataset(Step I)

In this section, Kmean clustering and Gaussian Mixture Model/GMM(a EM algo) were implemented on the original two datasets. For all numerical features in each dataset, the value was standardized to a mean of zero and a std of 1. For the beans dataset, sklearn verison of Kmeans and GMM was implemented.
For adult dataset, a kmeans model was implemented using python kmodes.kprototypes. The kprototypes verision of Kmean/kmode used a custom distance function that treat numeric features in euclidean and categorical features in dissimilarities space, it then combine euclidean distance with a weight gamma on dissimilarites as the distance score. In this way the kprototypes is a half and half between kmean(numeric) and kmodes(categorical). For EM, unfortunately i am unable to find a easy to used package for categorical data. so for the sake of this project, all categorical features was also standardized and treated as numeric to use sklearn GMM.
The number of cluster in this section was all choosing using the elbow method by ploting the number of cluster vs inertia/sum of square distance to the centeroid of each culster. The number of cluster K was then selected when the speed of inertia decrease decrease as K getting larger. Elbow method was used instead of the more robust silhouette score based method as it is a lot faster than that. Calculating silhouette score is pretty computationally intensive so it is better to stay from it.

A. *Beans Dataset*

1) Kmeans: The elbow chart of beans dataset on inertia was shown in figure 1. It can be clearly seen that the slope of the the inertia decrease slow down to linear at k=3. As k increase, the inertia naturally decrease as more cluster number would bring data closer to centeroid, but at some point(the elbow) the cost of adding cluster(overfitting) out weight decrease in inertia. The average Silhouette Score of different K value were also calcalated and shown in figure 2. K=3 got the highest(closest to 1) score which indicate a good separation of different cluster(dense cluster). Overall a K=3 was selected for this dataset. It was interesting to note that calculating the Silhouette Score took 24s compare to 0.7s to generated the elbow chart on inertia, so in the future only elbow method was used to select k if time is the main concern.
Comparison of the kmeans result to original label was shown in figure 3. It is kind of a mess, seems like the only unique beans class is BOMBAY as it was only shown in cluster 0. All other class have different level of abundance between each cluster: CALI/BARBUNYA present in all three cluster, DERMASON/SIRA/HOROZ/SEKER present in both 1/2 cluster. Overall the Kmeans(k=3) did a bad job in separating data label wise. This maybe coming from the fact that some features in this dataset may not be that relevant to the actual label discrimination. In Kmeans, each features was treated equally so a noise feature may add funny result. We would further see whether dimensional reduction method would help with this problem.

2) GMM Cluster: The Silhouette Score of different K on beans dataset using the Sklearn Gaussian Mixture Model(GMM) were shown in figure 4. At k = 2, the GMM model achieved the highest score so a k = 2 was selected for GMM. This result is smaller than the best k for Kmeans which is 3. In soft clustering(GMM), overlapping between cluster exist unlike in Kmeans a hard break between cluster was imposed. This cause our GMM ends up having smaller optimal K than Kmeans with the same data. Bayesian information criterion of different K was also generated.(Shown in Figure_5.png in code, wasnt shown due to page issue) Unfortunately the BIC chart seems not well behaved as it keep go down as K increase which suggested we to select a k that as large as possible which not make any sense. Metric like Silhouette score or BIC is only a rule

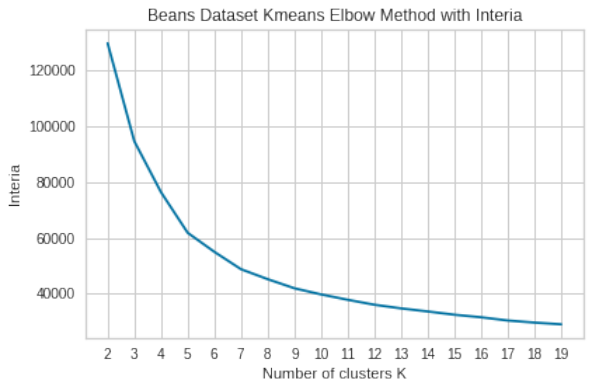

Fig. 1. Inertia vs Number of Cluster K(Kmean)

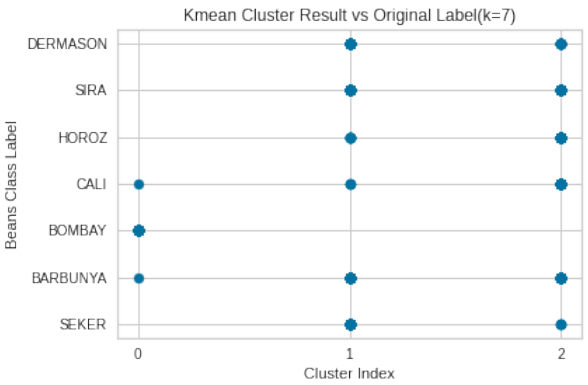

Fig. 3. Kmeans(3) Result vs Beans Label

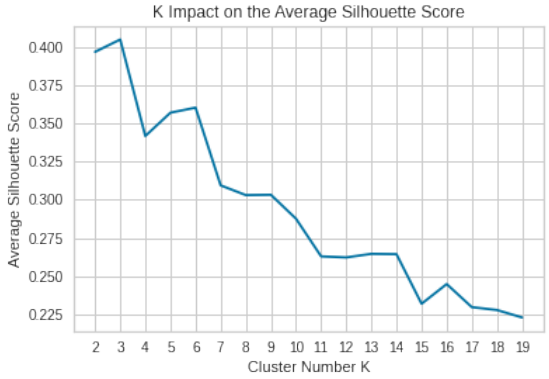

Fig. 2. Silhouette Score Different K on Beans Dataset(Kmeans)

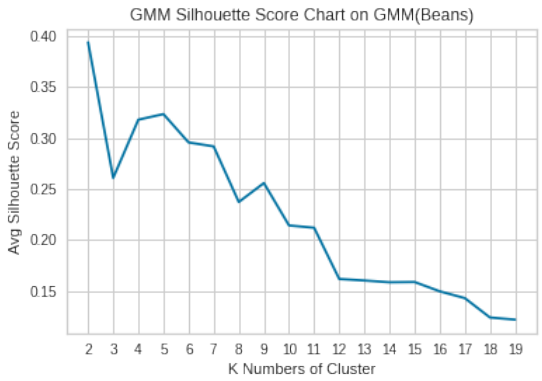

Fig. 4. Silhouette Score Different K on Beans Dataset(GMM)

of thumbs, so users may choose or believe whichever rule based on their preference. Due to the unsupervised nature of clustering method, it is impossible to have a define say about any of this selection. Overall a k = 2 was selected for our final GMM model on Beans dataset.

So how do this GMM model result lines up with or original labels. Like in the KMeans section, the only class of beans that can be clearly separated is BOMBAY(BOMBAY is the only class that are not in both clusters, the label distribution wasn't shown here due to page length issue, it is in the figure_6.png in code). GMM model(K=2) also did a bad job in separating data based on their class.

B. *Adult Dataset*

1) kprototypes(Kmeans+Kmodes): To select K value, kprototypes cost(the kprototypes version of distance to centeroid) relationship with K was shown in figure 5. Using the elbow method, K=3 was determined to be the number of clusters to use. The resulting kprototype cluster was summarized in figure 6 in terms of its label distribution corresponding to different cluster. As the Fig 6. shown, those three cluster don't manage to separate any data based on their original label. in all cluster, a around 75% of <=50k label was achieved which is basically the same as percent of this label in the dataset population. The failure of kprototypes method in separating label sense maybe due to the nature of this dataset, whether one person earn more than 50k or not may be due to more on 'luck' or features not shown in this dataset. Assume the lucky people distribute equally in each group then we would ends up always a 33% of 'lucky' people in every group. Another reason maybe though we have good features in this dataset, some of the features may not be relevant to separation of labels, further investigation of dimension reduction or features selection method may be helpful in later section.

2) GMM: As stated before, the adult dataset contained mixed features, so any method that used euclidean distance may cause problem. A better way to use GMM would be to do the exact same idea as Kprototype model, in which we treat categorical features and numeric features differently. For example we can treat numeric features as from a Gaussian distribution but categorical features as binomial distribution and do thing in a EM way. But a easy to use mixed Gaussian/binomial model package wasn't found so we unfortunately have to treat categorical features as numeric to apply GMM model for this section.

To obtain the proper K value for GMM on adult dataset, Bayesian Information Criterion of different K value was

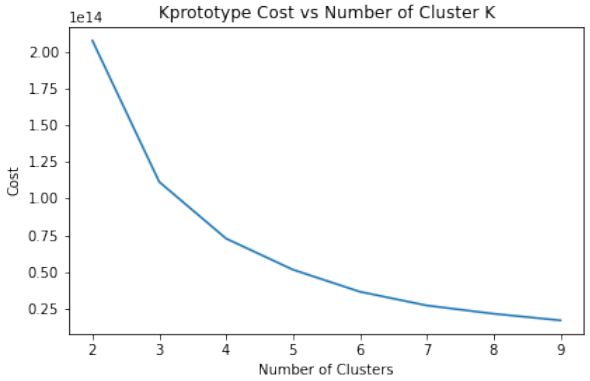

Fig. 5. KPrototype Cost Vs K

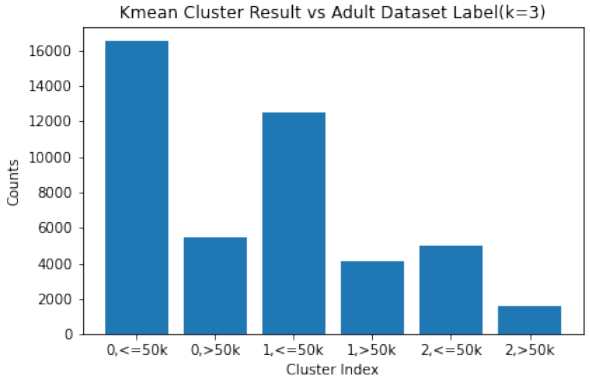

Fig. 6. Label Distribution of Kprototype(k=3) Cluster

summarized in figure 7. As the figure shown, full covariance and diagonal covariance achieve lower BIC value compare to all covariance type. In the same time BIC curve shape wise diagonal covariance curve is more well behavior as it reach a minimal at k = 6 then start increase due to information cost of the extra cluster added. So a diagonal covariance type and k = 6 was the proper choice given by BIC value. This result is further validate by calculating the silhouette score of the diagonal covariance GMM model in figure 8. The elbow for figure 8 show at k = 4 but k = 6 is another elbow after the first one, so in this same these two method agree with each other. Combine with what we learn in Beans dataset, BIC assignment on k seems to always prefer a larger K than what elbow method on silhouette score suggested. In the end a k = 6 was selected.

The resulting clustering label compare to original data label were summarized in figure 9. Label separation wise, we do achieve great separation in some cluster like cluster 0(45% >50k) and cluster 2(65.7%) which have noticeable difference in the >50k label compare to the sample population(25% >50k). This is significant different from the kprototype(k=3) result as at least some separation was achieved label wise. Further investigation in dimensional reduction method may show whether this problem with separating label was due to poor features or problem with high dimension dataset. Also the problem of mixed features may also be played into this problem.

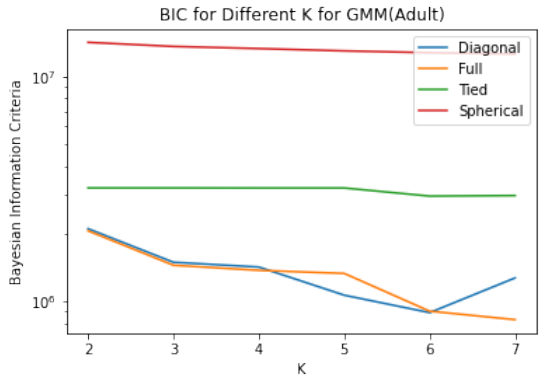

Fig. 7. Bayesian Information Criterion for GMM on Adult Dataset

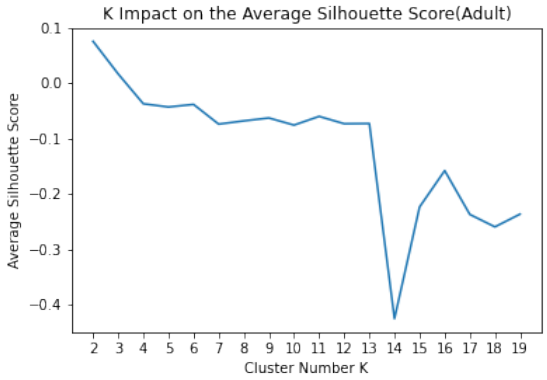

Fig. 8. Silhoute Score of Different K Value for GMM(Adult)

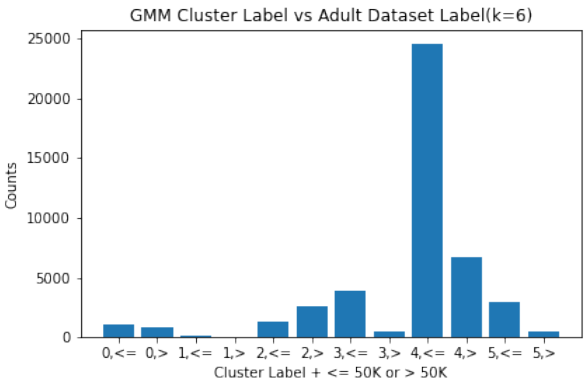

Fig. 9. Label Distribution in Each Cluster(Adult GMM K=6)

III. Dimensionality Reduction(DR) and Feature Selection(Step II and III)

All cluster methods in last section on both dataset dont seems to have great result in term of separating original data

label. This problem may be coming from irrelevant features or bad features, so DR come to the recuse as it reduce number of features(filtering) and also create new features(how depending on algo used) that may be more relevant so we can achieve better result. Also a another benefit of DR algo is that it made visualize cluster a lot easier as our human seems to can only understand 2d chart. In this section, Principle component analysis (PCA), Independent component analysis (ICA), random component analysis(RCA or random projection) was all implemented and apply to our two dataset with hard (Kmeans) and soft (GMM) clustering. Also decision tree was implemented and used as a features selection algo. For the adult dataset which contained mixed features type, methods like PCA/ICA/RCA may not produce great result for reasons that we discussed before. A better implementation would be only use PCA/ICA/RCA on the numeric features and leave categorical alone. Or use Factor Analysis of Mixed Data(FAMD) to do DR instead of these method. So for this section, PCA/ICA/RCA of the adult dataset would only be done with numeric features.

A. *PCA*

The idea behind PCA is to project data into a new axis that the variance on that axis is being maximum. The projected value on this axis was then called the first PC. Then the second PC would be a axis that orthogonal to the first PC but also maximum the projection variance, the third ..... We ends up have a endless series of PC(direction in space) that our data can project with maximum variance in that direction. By selected a series of PCs that have smaller number than original data features dimension number, DR was then achieved. How exactly we should select PC is more art than science as there are no agree upon way to do this and most of the time it depend on the data.

1) Beans: To discuss performance of PCA, we have to first figure out how many PCs that we selected. The percentage variance explained of the first 10 PC was shown in figure 10. The first 3 PCs explained 85% of the variance the data. A rule of thumb for choosing PC number would be to take the first PC that explained 85% of the variance so 3 PCs would be great according to that. Silhouette chart was generate on the PCA(n=3) transform beans dataset in figure 11, it turns out the chart have exactly same shape and the same turning points as figure 2. This result suggest that PCA retained all the same information in the original dataset so of course it make sense to choose the same K = 3 value as in figure 3. The distribution of different label in the first two PC axis was shown in figure 12. We can clear see a separation of the green class compare to all others class in the PC1 vs PC2 plane, but all the rest of the class seems to smash together into a big cluster at the left side but still some level of separation of was achieved in the first three PC.

Kmeans cluster (k = 3) was done on the PCA transform dataset. Clustering result in the PC1/PC2 plane was shown in figure 13. The Kmeans ends put all upper right blob into one cluster and further separated the bottom left blob into two cluster. Comparing Fig 12/Fig 13 to fig 3, we can see that kmeans cluster separation of each real label are better in the PCA space than the orginal feature space as the BOMBAY class was clearly separated into its own cluster. This show the power of PCA in terms of its ability on features selection. Also 3 PCs dimension provide more visualization possibility compare to our original 14 dimension.

Ok so PCA do improve cluster performance, one interesting thing to test is that by expanding k to 5 (same number of label) to see if all resulting cluster predict exactly the beans label. This the K = 5 Kmeans reasult were shown in figure 14. Comparing Fig 14 and Fig 12, we can see that a lot of overlapping are happening in the bottom left bolb in each cluster in terms of beans label. This maybe causing by the fact that beans label in the bottom blob may be too similar statistically, so it is really hard to distinguished between them in a hard clustering method like Kmeans. Or maybe the features in the beans dataset were not the best features to separate them. Also in PCA, PC are being ranked by the explained variance, so a features in features that have high variance would have more to say in Top PC, information in some low variance features also be important in terms of classify beans label but are represented less in the final selected PCs.

A GMM model was also built in the first three PCs space. K = 3 was selected by calculating silhouette and apply elbow method to it. This k selection is different from what in the original space which the peaks of silhouette score stand at 3 but not 2. The resulting cluster was shown in figure 15. Comparing to the kmeans result, the GMM cluster looks bigger when the center of each cluster seems to be in similar position, also edge between closer cluster red/blue overlap more compare to kmeans well defined cluster. These edge show more semblance to grouping edge in figure 12(beans label) which a lot of overlapping are happening between class. This shown some benefit of using soft clustering technique(EM) in worlds that label have high similarity compare to each other. Also another benefit of Soft clustering is that it output result in terms of a probability of a instance is in each class, which can be used as a measured of how confidence we are of each instance in each cluster. We can use this information to further circle out points that are problematic (i.e. out of two std(<Ĩ-95%)). These two features of EM algo like GMM bring more degree of freedom when using them in the real world that things are not always black and white.

2) Adult: The eigenvector distribution of adult dataset in terms of % variance explained was shown in figure 16. Each PCs in figure 18 have almost the same percent variance explained. This can be due to the fact that we only have 5 numeric features so we can only generate 5 PC unlike in beans dataset that we only use 10 PCs of the possible 14. Also variance on this 5 features maybe similar so no one features overpower. Based on the 85% rule of thumb, 4 PCs was selected to use. Then we only manage to reduce one dimension by doing PCA.

To further do Kmeans on the numeric features, silhouette

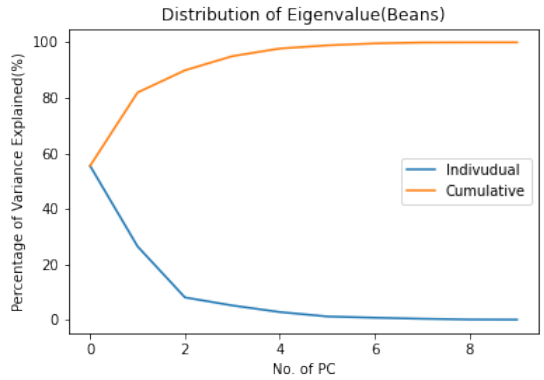

Fig. 10. Percentage of Variance Explained for the First Ten PC(Beans))

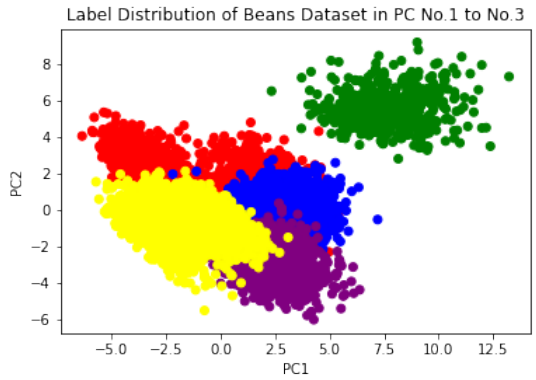

Fig. 12. Label Distribution in PCA space(Beans)

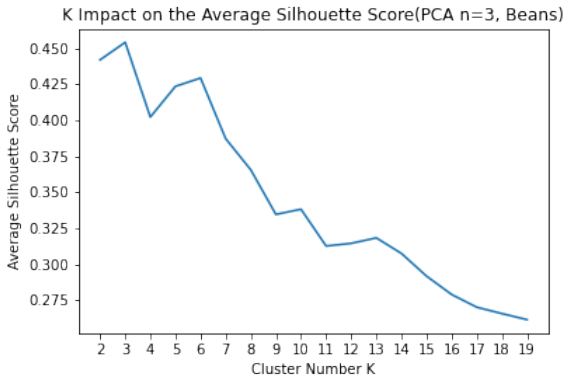

Fig. 11. Average Silhouette Score of Kmeans(PCA n=3 Beans)

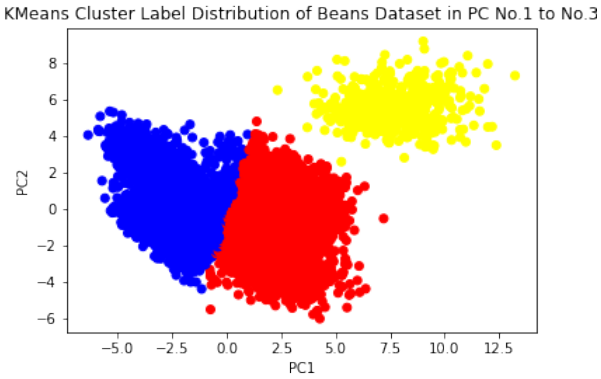

Fig. 13. Cluster Distribution in PCA space(Beans K=3 Kmeans), Color here reference to Different Cluster Label

score was calculated and we found that it peaks at K=2. So K=2 was selected as the number of cluster to use. Figure 17 showcase why good perdiction is hard in the adult dataset: only some >50K instance (the upper yellow blob) can be easily separated from other data when majority of red and yellow just mixed all together in the bottom. When you are in the lower value region(PC2/PC3) then these two class are overlapping so prediction would be mostly base on 'luck'. The Kmeans cluster label distribution in PC2/PC3 space was also generated but not shown here as it have high similar to the label distribution.

GMM cluster was also done on the PCA transformred adult data. K = 2 was selected based on the calculate silhouette score. Soft clustering may works better when your target(>50k or <=50k in this case) is coming from a probability distribution(binomial in this case). Is it really the case? We compare them by calculating the accuracy of using cluster label to predict data label. The kmeans(k=2) model returned 75.42% accuracy and GMM(k=2) return 77.5% accuracy. The GMM do did better in this task compare to Kmeans. But on the other hand, the data contain 75.2% of <=50K label which is almost the same as GMM and Kmeans accuracy. Both of them are can only be seems as weak learner in terms of separating label.

B. ICA

A simple way to explain what ICA is: ICA project data into components(axis) that the projection on those axis are independent. Unlike PCA that any dataset can be apply to, ICA can't be done if those independent component don't exist in the first place(if all features is just noise). Unlike in PCA that the maximum amount of PC can be generated is only limited by features dimension, number of IC for each dataset is a smaller number and most of the time we used all the IC generated.

1) Adult: ICA was performed on the the numeric parts of the adult dataset like in the PCA section. The sklearn fastICA returns 5 ICs which their kurtosis statistic was plotted in figure 18. IC No.1 and No.3 achieved the highest kurtosis(20 and 140) which indicate these two ICs are heavily tailed and not like gaussian at all (kurtosis of 3, and can be seems as noise). So IC1/IC3 plane was used reduce dimension of this dataset. The IC1/IC3/IC4 projection was shown in figure 19. (IC4 was only here for plotting as it make density of point easier to see). We can clearly see that projection on these two axis dont have any correlation at all as they are the most independent component of the bunch. When IC1 = 0, we can clearly see two class separating with some degree of overlaping. Label

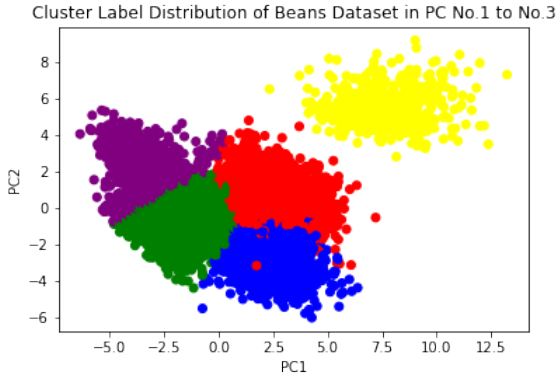

Fig. 14. Do K=5 in PCA Space give the Perfect Cluster Base on Beans Label?

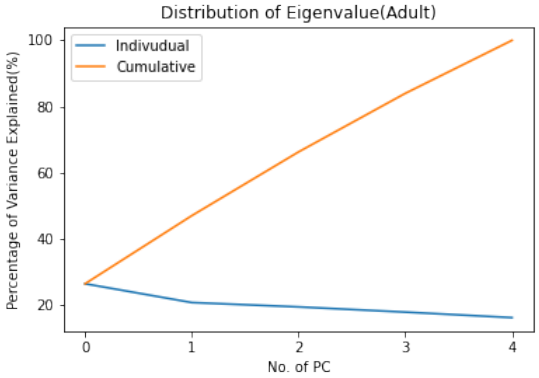

Fig. 16. Percentage of Variance Explained for the First Five PC(Adult)

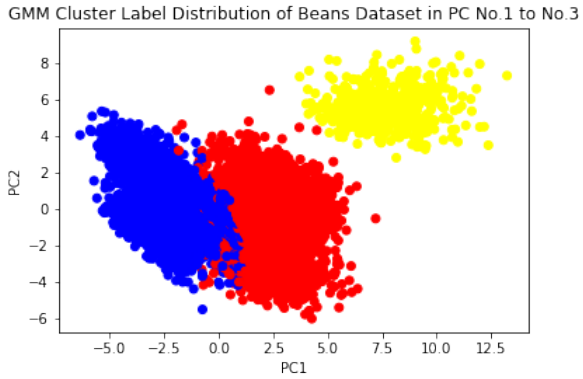

Fig. 15. Cluster Distribution in PCA space(Adult K=3 GMM)

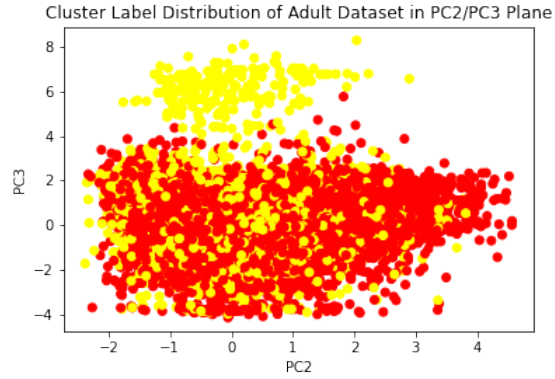

Fig. 17. Label Distribution in PC2/PC3 Plane, Red/Yellow Corresponds to <=50K and >50K Income Respectively

separation wise ICA seems to do a better job than PCA in last section.

Kmeans Clustering was also done on the ICA transform dataset. In figure 19 we can clearly points out 3 cluster so this would be the K value to use. Resulting cluster label distribution in this plane was plotted in figure 20. A GMM on IC1/IC3 projection was also done. The cluster distribution wasnt shown as it get almost the exact same cluster but with the blue cluster get a tiny bit larger at IC1 ≈0.01.

2) Beans: The resulting kurtosis distribution in each IC was shown in figure 21 The highest top three kurtosis value was IC5/IC13/IC15 so these two axis was selected. Label distribution in this plane was shown in figure 22. Only the red and green class was managed to clearly separted in this space, the other four class all cluster in the bottom right region. In this space, it can be clearly seems that 3 different cluster exist (the red long tail, the green long tail, and the bottom left blob), so K = 3 was selected. Kmeans Clustering and GMM was further done on the ICA transformed data in this space. GMM clustering ends up give better resemblance to the two long tailed + a blob clusters like we anticipate as figure 23 shown.

C. *RCA*

PCA and ICA both have its own bias in terms of what component(axis to project to) to choose, for PCA it favor components that maximize projected variance and ICA favor a high kurtosis. All these things may not be relevant to for our dataset, so just project the data into random k dimension(k smaller than orginal features dimension) would be a great way to get away for these bias in the same time achieve dimension reduction without losing information. The greatest features of this method is that it is really fast, but performance of it is highly depend on our luck so a lot of rerun with different random seed is needed for a reasonable performance. In this section gaussian random projection was implemented using Sklearn to reduce dimension to 3.

1) Beans: In order to show the amount of variability RCA method can have, the reconstruction error of Gaussian Random Projection was shown in figure 24 in which error from a 100 rerun was shown. If we require a reconstruction RMSE larger 0.95 so we don't lose too much information after dimension reduction, in this 0 to 100 range, only 5 random seed achieve this. On top of that if we want a good separation between label using these 5 seed, we need to examine performance of these five seed separately which would require a lot of manuel labor

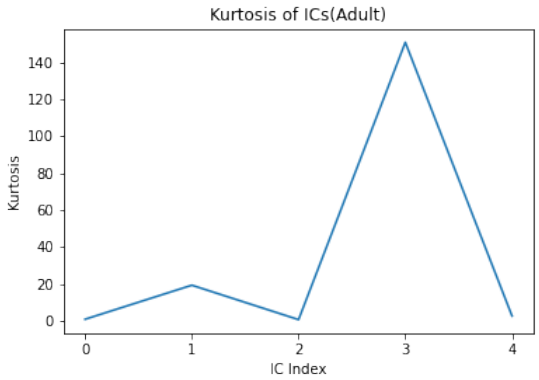

Fig. 18. Kurtosis of the Result IC(Adult)

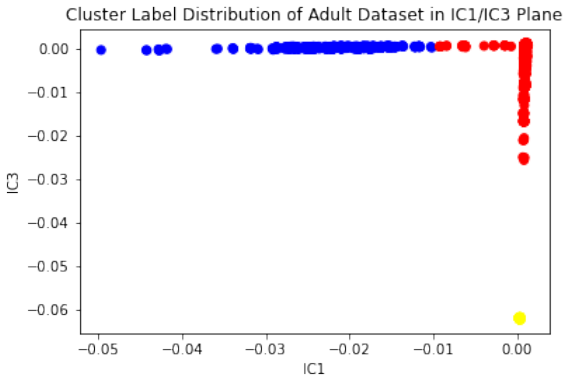

Fig. 20. Cluster Label Distribution in IC1/IC3 Plane

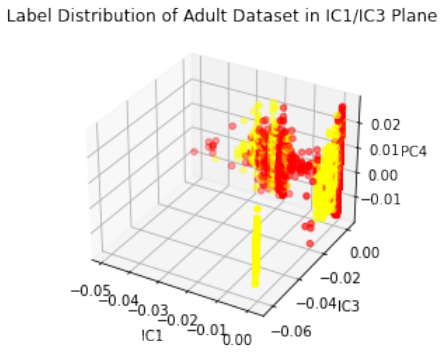

Fig. 19. Label Distribution in IC1/IC3/IC4 Space, Yellow Point Here is > 50K Class.

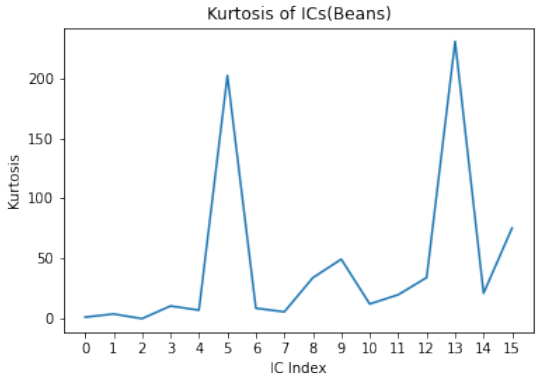

Fig. 21. Kurtosis of the Result IC(Beans)

compare to in ICA or PCA that the algo itself already have bias on what is good. But in terms of wall time require to do this transformation, RCA (took 0.3s for 100 rerun) greatly outperform ICA or PCA (1 transformation ĩs).

For this section a random seed of 37 was selected to generate RCA projection as it produce the lowest error in figure 25. Label distribution in the RCA space was shown in figure 26. In this RCA space, no notable label separation can be seems at all. Compare to PCA or ICA label distribution, RCA did terribly as its performance is mostly random. In real world appilication, more extensive fine tuning on the random seed is needed for RCA.

Kmeans and GMM was also being done on Beans dataset in RCA space. Unlike in PCA or ICA, we can clearly see good number of cluster to use, it is not clear so we have to go back to calculating silhouette score of different K. A k = 2 was selected based on those calculation for the RCA transformed beans dataset for both kmeans and GMM. The GMM result was shown in figure 26. The difference between GMM cluster and Kmeans cluster(not shown in figure) was that GMM give larger red cluster when centroid of each cluster stay almost the same. Overall in terms of label separation, in RCA space that we tried, everything was trouble.

D. *Features Selection-Decision Tree*

Another way to do dimension reduction is to do features selection which we filter bad features out. In this work, we used decision tree as a way to do that. One great features of this kind of filtering method is that we dont have to deal with the problem of mixed features like in PCA/ICA. In this section, decision tree classifer was built with sklearn in a supervised manner using information from HW1, the return feature importance rank was used to filter bad features. For this section, all near zero importance features was removed to achieve dimensionalilty reduction.

1) Beans: Using hyper parameters from HW1, a DT model was trained with the beans dataset and features importance was shown in figure 27. Since we would be losing information after the filtering process, only features with a near zero importance was removed. This filter process ends up reduce data dimension from 16 to 7. One interesting things to know is that the last four features in here are synthetic features from nonlinear combination of other feature(all are ratio) based on some domain knowledge. Among these syn features that we do have some that contribute nothing even though domain knowledge argue the opposite.

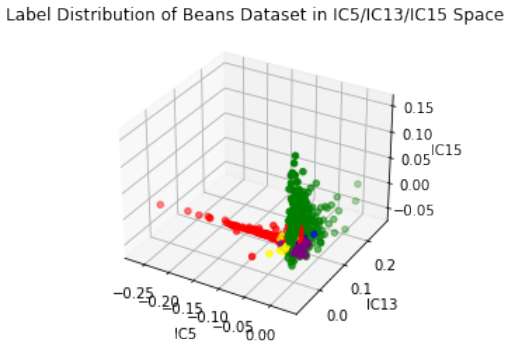

Fig. 22. Label Distribution in IC5/13/15 Space(Beans)

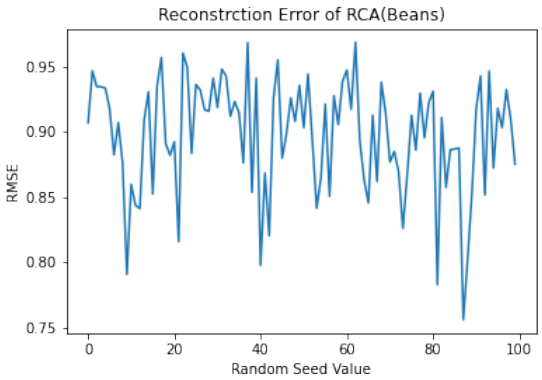

Fig. 24. How Much Reconstruction Error Can RCA Produce?

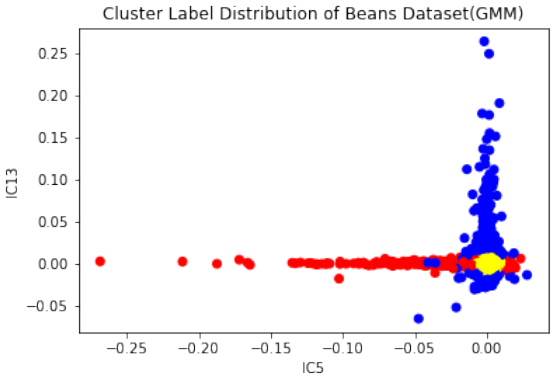

Fig. 23. Cluster Distribution in IC5/13/15 Space(GMM in Beans)

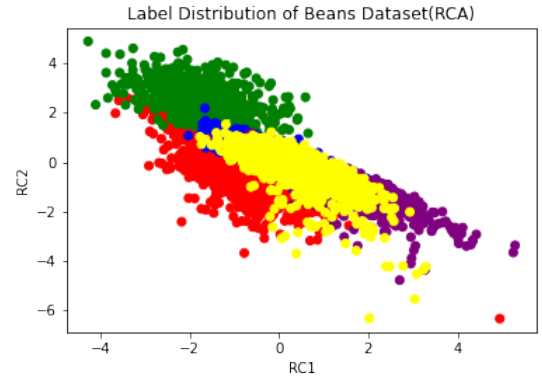

Fig. 25. Label Distribution in RCA Space(Beans)

2) Adult: The features importance of adult dataset was shown in figure 28. We removed near zero importance feature and result in 5 features left compare to original 13 features. These 5 important features was identify as education number, family relationship, capital-loss, capital-gain, working hours per week. GMM and Kmeans clustering was further performance on the features filtered space.

IV. Neural Network with DR on Adult Dataset

In this section, NN was trained with the transformed/filter adult dataset after dimensionalilty reduction technique to see whether these method help the classifier or not. For all NN model shown in this section, grid search hyper parameter tuning was done on each dataset(transformed) to ensure the best performance. In the first part of this section, the transformred/filter dataset was directly used to replace the original data. For the second part, we used the cluster label generated from all clustering method(2 cluster method * 4 transformred/filter method) as a new features and added it to adult dataset.

A. Directly Work With Transformed Data(Step 4 and 5)

In this section, the transformed Adult dataset(PCA, ICA, RCA, DT Filter Features Selection) was used to replace the original adult dataset to be used in NN training to classify 50K earning label. For each transform dataset, the best hyperparameter setting obtained from grid search cross validation on the training set was used. Also a baseline NN was training using the original data to compare with these dimensionalilty reduction method. PCA/ICA/RCA transformed all reduce the features dimension to 8 when feature selection get it down to 5. In theory PCA/ICA/RCA can reduce feature dimension even more, but in the case of adult dataset that these method was only introduce to the numeric which hinder its ability to further reduce dimensionaility. The learning curve of these NN model on training dataset was shown in figure 29. All NN model here was determined to be sufficiently trained as all have well behaved learning curve(not shown in this report). On the training curve side, the bsealine seems to achieve higher training/testing f1 score compare to all transformed data. But on the other hand all transformed NN achieved better generalization compare to to baseline which have higher difference between training score and testing score. Overall in the training part the filtering method achieve the best overall performance and are outperforming PCA/RCA/ICA. T In the final testing shown in figure 30, PCA/ICA all achieved similar performance compare to baseline. DT feature importance filtering do outperform all method here in both Training and Testing. his maybe coming from two reason :1) The features importance ranking used in

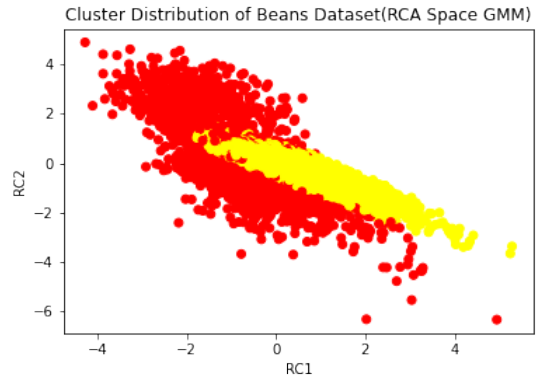

Fig. 26. Cluster Label Distribution in RCA Space(Beans GMM)

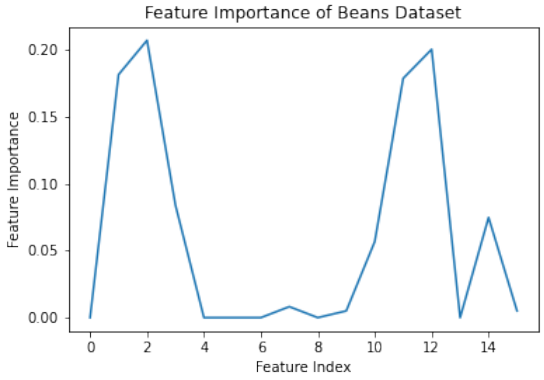

Fig. 27. Features Importance of Beans Dataset

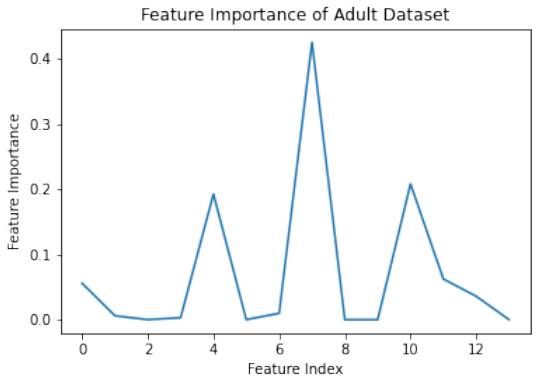

Fig. 28. Features Importance of Beans Dataset

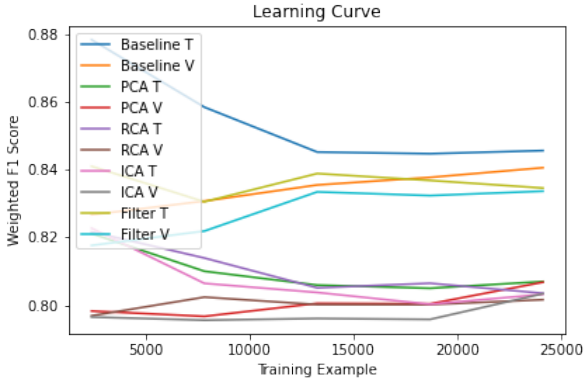

Fig. 29. Learning Curve of NN on Various Transformed Dataset, T/V here stand for training/validating Respectively

this method make use of the training set label which leaks information to the selection process. Cheating make it better in the classification problem. 2) PCA/RCA/ICA was not done on the all features so this comparison is not fair. In the same time, RCA have the worst performance among these method here as it performance depend on random factor.

Another benefit of using dimensionaility reduction technique on task like classification with NN is its can hugely reduce training run time. For the adult dataset, grid search optimal hidden layer strcture layer of the transform dataset is all smaller than the baseline(16,16):the optimal NN hidden layer structure was determined as (16,16),(6,6),(8,8),(4,8),(8,8) for the baseline(14 features), PCA(8 features), ICA(8 features), RCA(8 features), filtering(5 features). This lead to shorter training time of all DR treated dataset(average 1.53s) than the baseline(2.17s trained) and reduced memory usage. (This is not apple to apple as also training rate have impact on this) In task like NN training, fine tuning of hyperparameters are alway required (a huge number of rerun of model was required) so a 25% time reduction would free up tons of computation resource and could pay a lot of dividend in the future.

B. Using Clustering Label as additional feature for NN

Label features that generated in section I for adult dataset(Kprototype(k=3) and GMM(k=6) clustering) was used in this section as a new feature to train NN classification learner. For each NN trained, its optimal hyperparameter setting was obtain through grid search which is a hidden layer size of (16,16) for both two case and the baseline. The learning curve of these three NN model was shown in fig 31. In fig 31, we can see that all the baseline and cluster label added training/validating line all heavily overlap with each other which indicate similar training classification performance. But overall the added label do improve the classfication performance on training set a tiny bit with the more cluster GMM version seems to have more improvement than the Kprototype version. We further tested their performance on the testing set and result were summarized in figure 32. In the testing set, actually the baseline perform the best despite those tiny lead of added label shown in the training process, so why that is the case? We have to come back to one discussion we have in HW1 which is about synthetic features. In the beans dataset there are four synthetic feature that are coming from non linear combination of original features(ratio), in HW1 we showed that these synthetic features do improve the performance of

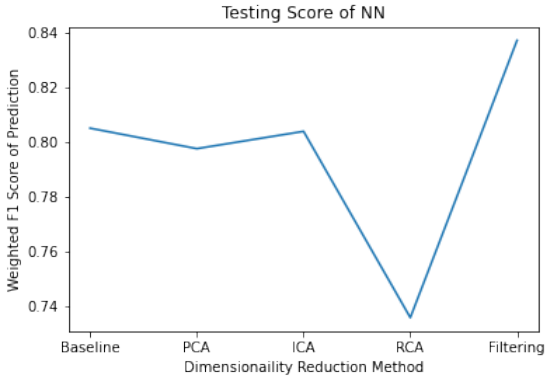

Fig. 30. Testing Score of NN Trained on Transformed Data

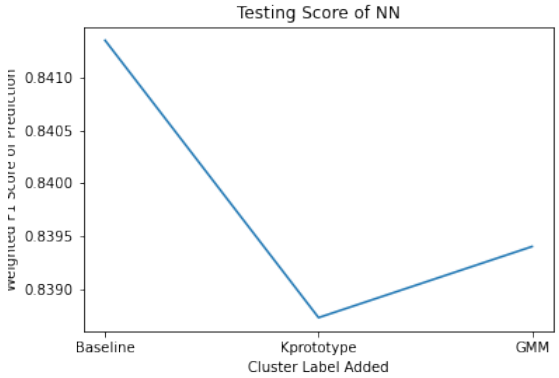

Fig. 32. Testing Score of NN Trained with Added Cluster Label

classifier. (also in figure 29 we shown three of them have high feature importance value) If those domain knowledge do work then doing non linear combination on existing features would added information to the data so would increase classifier performance. (As it increase weight on existing useful feature and decrease weight on unimportant features) Well for our case here, the cluster label is also a synthetic features coming from non linear combination of existing features(clustering). But the problem is that we dont really have domain knowledge so we don't know whether distance in the cluster space are important for classification or not. What the cluster label features do to our final result could be similar to copy and paste a features that have no feature importance, which would increase the chance of overfitting in training set but decrease testing performance. So one way these cluster label would help would be in dataset that distance used in the cluster do make sense for classification, but unfortunately those are not true for our dataset.

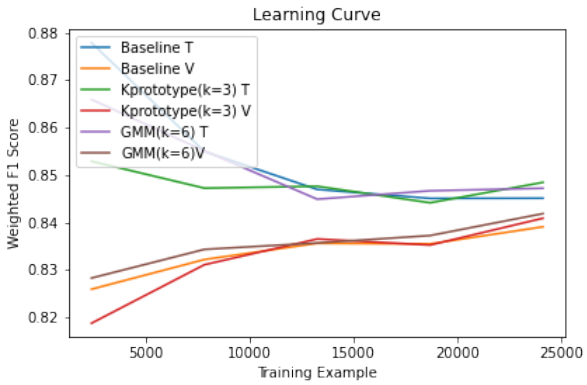

Fig. 31. Learning Curve of Cluster Label Added NN

V. Conclusion

In this work, we implemented two cluster method kmean/kprototype and GMM on two different dataset. Also four deminsionalitly reduction technique on each dataset to reduce the complexity. One interesting problem in this work is coming from mixed features type(adult dataset), euclidean distance would be problematic so the kmeans/GMM and PCA/ICA/RCA become problemetic for those dataset. In this work we got away from this problem by using kprototype clustering and only apply PCA/ICA/RCA to numeric features. But a better way to deal with this would be alway stick to algo that support mixed data type, for example Factorial Analysis of Mixed Data(FAMD) would be a replacement for PCA in mixed data. Also one interesting to know is that it seems like R language have more support about these sorts of mixed data type problem than Sklearn. Seems like there are needs to support those mixed type algo in sklearn from my usage in this work.

One great features of DR algo was shown in this work which it is the ease to perform visualization after that. I am clearly to stuggle to ploting clustering result in Step I for 14 dimension, but right PCA/RCA/ICA reduce the space to 3D, things can then be easily plotted like in figure 14. This make user have a easier job in terms of data exploration. But unfortunately for the NN classifcation job, features selection seems to do a better job than PCA/ICA which is my before guess due to their wide usage. (Probable due to mixed feature type) Also DR algo reduce the space and time cost of whatever future process you need to done on the dataset which would always be favorable.

Overall, unsupervised learning and dimensionalilty reduction are both useful technique in data exploration.

VI. References

References

[Koh96] R Kohavi. "Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid". In: (Dec. 1996).

[KO20] Murat Koklu and Ilker Ali Ozkan. "Multiclass classification of dry beans using computer vision and machine learning techniques". In: *Computers and Electronics in Agriculture* 174 (2020), p. 105507. ISSN: 0168-1699.