Unsupervised Learning and Dimensionality Reduction

Feng Zhang
fzhang326@gatech.edu

1. Introduction to data

1.1 Iris problem:

This problem is using several attributes to predict the class of iris plant, this dataset is a very typical dataset and best known database in the pattern recognition literature. There are four predictors, which are sepal length in cm, sepal width in cm, petal length in cm and petal width in cm. The outcome includes three classes which are Setosa, Versicolour and Virginia.

1.2 USA arrest problems:

This dataset is consisted with aggregated data for assault, murder and rape rate per 100,000 residents among 50 US states in 1973. It also provides information about the percentage of population living in the urban areas.

I like this two dataset for several reasons. First, out of interest, I am a botanist before and have a great interests in the plant and collected data for my own research. Also, after living in the USA, I found the differences among district in USA varies, some of the place is pretty dangerous while some are not which is different from my hometown and country. Second, this two datasets are pretty mature and don't have much tricky in cleaning and exploring and the best approach has achieved, what I can do is to check whether I could get the best model using my models.

2. Models

2.1 K-mean problem

For the iris problem, my question is whether I could cluster all samples into three clusters. The number of k in the K-mean problem is **3** since I have three species in the dataset and I want to check whether the K-mean algorithm can categories all my samples into these three classes. In the code, I also specify the nstart =25 which is the number of starting assignments, this will decrease the sensitivity of the random starting assignments.

2.2 Expectation Maximization

The EM algorithm is an iterative method to find the maximum likelihood for the estimates, it has two steps. First step is to find the conditional expectation of the model based on the data and then run the maximum algorithm to find the maximum value of the parameter. Of course, it cannot

find the maximum before running iterations. It can solve many problems, especially can be used in estimating the missing value in the dataset. Also, it can be used to do the cluster.

In the model, I did not specify the prior in the function, it will specify Gaussian finish mixture model for the EM model. The metric for the best fit is the Bayesian information criterion.

3. PCA

Principle component analysis is a statistical method which is used to perform dimension deduction. It compares the eigenvectors and eigenvalues in the data. In the model, it will return the percentage of variation which can be explained by the variables. This means that the higher percentage of variation explained by the variable, the more important this variable is in the data. All settings are default values.

4. ICA

Independent component analysis is another dimension deduction algorithm which is used to separate independent sources from their mixture. In the ICA, the number of components are set as 3 which is consistent with the previous approaches. The alpha which is the negentropy is 0.1 when the function of the G function is "Logcosh", also, the rows of the data matrix X will not be standardized beforehand. The scalar giving the tolerance is set as 1e-4 which is pretty small to converge.

5. Randomized Projections

The randomized projection is another feature extraction algorithm which based on the projection matrix is fulled with independent and identically distributed random values. The classification algorithm used in the randomized projection is CNN.

6. Random Forest

The last feature deduction algorithm I used here is the random forest. This algorithm can return the importance of each covariates and can be used to decide the most important features in the model.

3. Clustering

According to the algorithm described above, I run the two algorithms in my dataset.
For the first USarrest dataset, I failed to visualize the cluster

	Murder	Assault	UrbanPop	Rape	cluster
Alabama	13.2	236	58	21.2	2
Alaska	10.0	263	48	44.5	2
Arizona	8.1	294	80	31.0	2
Arkansas	8.8	190	50	19.5	3
California	9.0	276	91	40.6	2
Colorado	7.9	204	78	38.7	3

graph so I just show the dataset I created which label the number of cluster for each observations.

For the EM algorithm, I also can create the similar results: Even through they are assigned different labels between two algorithms, they are pretty similar according to their assignments.

```
> data.frame(fit$classification)
                fit.classification
Alabama                          1
Alaska                           1
Arizona                          1
Arkansas                         3
California                       1
Colorado                         1
Connecticut                      3
Delaware                         3
Florida                          1
Georgia                          1
Hawaii                           3
Idaho                            2
Illinois                         1
Indiana                          3
Iowa                             2
Kansas                           3
Kentucky                         3
```

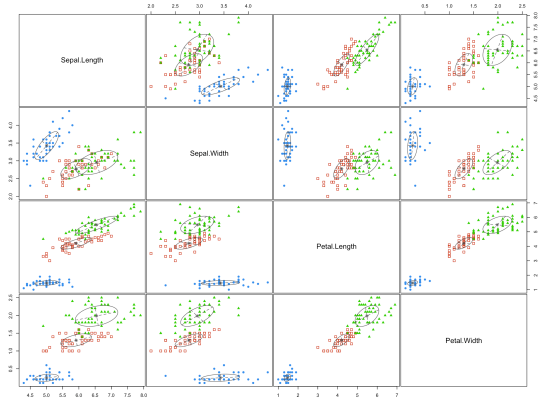
For the second dataset, I performed similar steps and created similar datasets using two algorithms. The rest of the settings in the model are same as the first dataset.

In the programming, I compared the results from two approaches, the classification similarity was calculated manually since no label was given in the data. For example, since Alabama was in the 2nd cluster in the first algorithm, and 1st cluster in the second algorithm, as well as Alaska and Arizona, so I make sure that they are clustered in one group, just different number of clusters in the results. In this case, I will label it as the same classification. According to this senses, I did not write the code but checked and calculated based on my intuitive sense about the categorization of the dataset. From my calculation, around 84% of the districts are in the same categories while 16% are ambitious between two algorithms.

Since we have no labels for the USarrest dataset, I can adjust the number of clusters in the model to avoid undercutting and overfitting problems. I checked the figures see whether it is good to set the number as 3 or more. However, as I increase this parameters, the more variation it will have between two algorithms in categorizing the samples. The minimal variation was at number 3.

In the iris dataset, it is interesting to find some things because I have the label for each spice. I applied two algorithms and found different results. For PCA, there are 25/150 samples which are not consistent to their labels while in the EM algorithm, 150/150 samples are consistent with the given labels so it is reasonable to say that the EM algorithm works better in this dataset using default model settings.

The right figure is a classification plot to reveal different clustering results among all predictors. From the results figure, we can see that in some predictors, the categorization is not fully efficient since it is still has some overlaps. In the EM algorithm, I tried different priors in the model but did not improve the accuracy in categorization. In the meantime, if I don't specify the number of clusters in the model, it will then assign 4 categories for the results. I think this is consistent with the



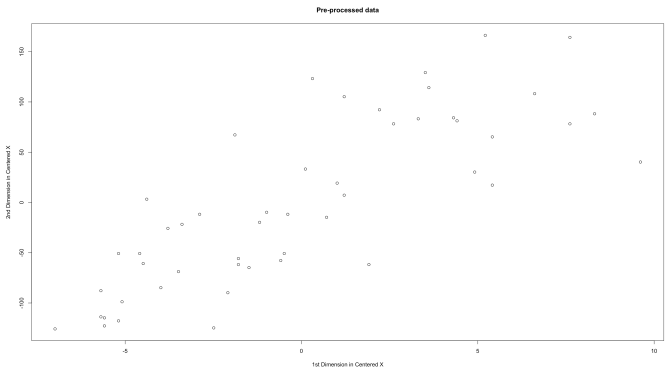
results from the KMeans since all inconsistent categories are in the 3rd clusters which EM assigned as the fourth categories. From this results, I have evidence to suspect whether we could give the Virginia category a sub-category to differentiate each other.

4. Dimension Deduction

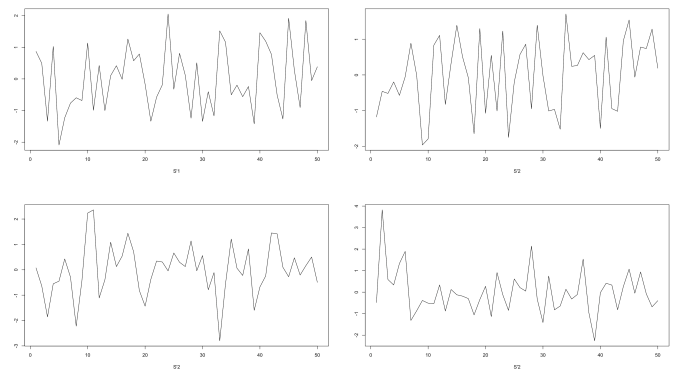
For the USArrests data, for the PCA, from the results of the variation distribution, the first component (Murder) explains 62% if the total variation, the second component (Assault) explains the 24% of the total variation, the third component (UrbanPop) explains 8% of the total variation and the four component accounts (Rape) for 4% of the total variation. In other words, the Murder and the Assault explains 86% of the total variation in the dataset and first three predictors explain the 95% variation.

For the Iris data, from the results of the variation distribution, the first component (Sepal.length) explains 72.96% if the total variation, the second component (Sepal.Width) explains the 22.85% of the total variation, the third component (UrbanPop) explains 8% of the total variation and the four component accounts (Rape) for 3.69% of the total variation. In other words, the length and the width explains 95.81% of the total variation in the dataset and first three predictors explain the 99.4% variation. In other words, I believe the first two predictors are enough in the model for further analysis since these two are already pretty enough in explaining the variation.

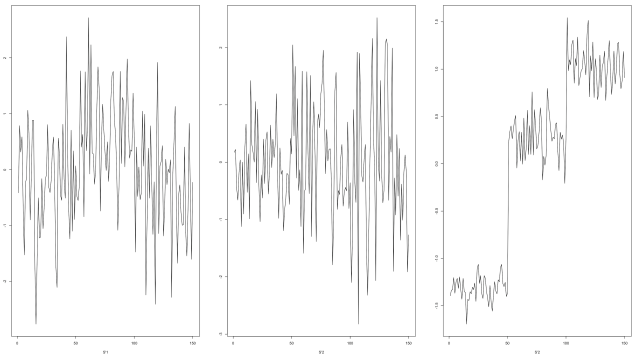
In the ICA for data USArrests, I applied package FastICA, non-gaussianity is measured using approximations to neg-entropy (J) which are more robust than kurtosis-based measures and fast to compute. So from he package, I draw the first and second dimension distribution in the preprocessing dataset. For the Iris dataset, we could perform the similar analysis and figures.



Besides this figure, we could also analyze the source signals. From the four figures in the right-hand sided, we could see may only the second predictor show some patterns, the rest three variables did not show a clear patterns so we may not extract independent tones from their linear mixture distribution. I highly doubt it is because of the small sample size in the dateset. Based on this assumption, I



run the algorithm on the second dataset which is iris. The results are promising for this time even through it did not show clear patterns to reveal the source signals.



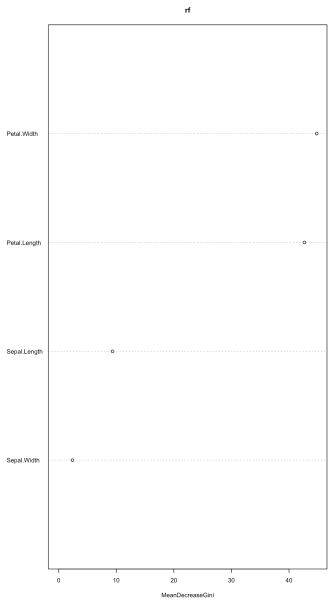
After running the dimension deduction, I rerun the PCA. For the Iris dataset, I select the first two important variables which corresponding to 95% variation. For the USArrests dataset, I also select first two components which account for 86% variation. After making subset selection, I rerun the analysis.

For the USArrests dataset, I got the first component accounts for 90% of the variation while the second is 10% which gives the first component more weight than previous full dataset. For the Iris dataset, after performing the subset selection, I rerun the PCA and got the results that the first component accounts for 55% while the second accounts for 44% of the total variation. The results seems to decrease the weight of the first component compared to the full dataset while increase the weight for the second component.

From above results, we found that after performing the dimension deduction, the PCA results will be different. Then I run the Randomized projection based on the given data and tested whether the results will change after running the dimension deduction.

Since only the Iris has a label so I will only include this dataset in the report. Based on the full dataset, we could have an overall 92.11% accuracy with 95% CI as (0.7862, 0.9834). The overall performance is good. After performing the dimension deduction, there is no change in the accuracy.

Lastly, I performed the random forest in order to check whether I could retrieve similar results. Specifically, we see the mean of decrease Gini as the metric to measure the importance. From this metric, we find that the petal.length is the most important and followed by the petal.width which is different from the previous approach.
For the USArrests dates, the same results could be retrieved from the random forest which is the murder is the most important feature, followed by the assault and the last is UrbanPop.



4. Reclustering after Dimension Deduction

After running the dimension deduction, we can have a basic idea about the importance of each predictor, then we could apply the clustering afterwords to check whether we will have a better categorization.

Here I will only compare the change on the Iris dataset since it is labeled. From the results of K-means, worse results show after performing the dimension deduction. Similarly, we have a worse clustering results from the EM algorithm. This is clear since when we apply the whole dataset, the EM algorithm could perfectly (100%) cluster the samples into three categories which are corresponding to the categories of the species. I guess the reason for this decrease is because of the loss of information. Even though the dimension deduction could help remove the noise and decrease the curse of high-dimension but for the small dataset, we may not that need the dimension deduction. I guess if we have a high-dimension in the dateset such as 100+ predictors, then the dimension deduction may help reduce the redundancy in the model and improve the efficiency and accuracy.

5. Redo the assignment 1 dataset: Titanic

From the analysis, before running the dimension deduction, when we cluster the dataset, we can have a 62.605% accuracy for the titanic dataset. Similar to the k-means, we only have a 37.25% accuracy for the clustering.

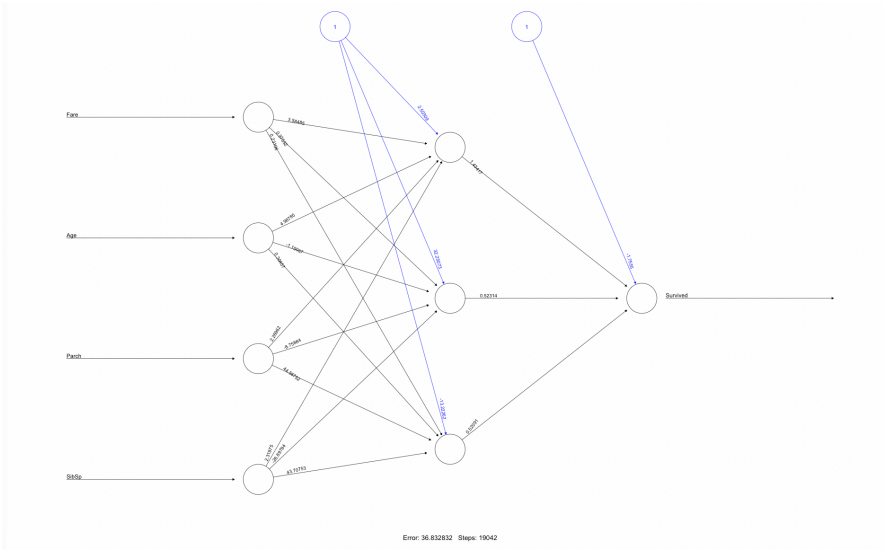
```
Importance of components:
                          PC1    PC2    PC3    PC4
Standard deviation     1.2794 1.0522 0.8182 0.7659
Proportion of Variance 0.4092 0.2768 0.1673 0.1467
Cumulative Proportion  0.4092 0.6860 0.8533 1.0000
```

From the dimension deduction results, we could see that the first component accounts for 40.92% of the total variation while the second accounts for 27%. Even though there is no clear signal to show that one or more predictors are dominant, if we have to choose, then the first three components accounts for about 85% of variation.

I also compare the results from the random forest. The importance figure shows that the Far and age are the most two important features and followed by the Parch, this results are identical to the previous results.

	IncNodePurity
Fare	27.215627
Age	17.586660
Parch	6.927001
SibSp	8.509703

From above analysis, we will select Far and age the predictors in the model and add the clustering we created in the previous step to check whether the accuracy and other metrics will improve significantly.



The fight figure is the neural network visualization.
Before the dimension deduction, the overall accuracy is 68.18% which is higher than the 60.83% after the dimension deduction in this example. The overall speed did increase due to smaller data after performing the dimension deduction but the results did not increase. My thought is that the data is not big enough to support the dimension deduction so based on this limited dataset, it will only loss the information to run the dimension deduction.