Unsupervised Learning & Dimensionality Reduction

Unsupervised Learning is another important class of techniques in machine learning. It deals with unlabelled data, and is very useful in exploratory data analysis, which are often the first step in data analysis, and let the user knowing more about the nature of the dataset. In this project, two important class of unsupervised learning algorithms are of concern – clustering and dimensionality reduction.

Clustering is about a brief description of the data, and it group data points up into a few cluster. The details of the two clustering algorithms will be discussed in the later part.

Dimensionality reduction, on the other hand, is trying to represent the data with fewer variables, while still keeping as much information as possible. The details of the four dimensionality reduction algorithms will be discussed in the later part.

In this project, the same two datasets used previously are used to illustrate the power of these two class of algorithm.

Dataset 1 – Default of Credit Card Client

The first dataset is related to the default of credit card clients of customersâ€™ default payments in Taiwan. It will be interesting for banks to know which kind of clients are more likely to default their credit card payment and thus have a better business strategy.

This dataset includes the record of payment history of 30000 clients. The dependent variable is if the client obligated the payment or not, and there are 23 independent variables:

X1: Amount of the given credit including both the individual consumer credit and his/her family (supplementary) credit.
X2: Gender (1 = male; 2 = female).
X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
X4: Marital status (1 = married; 2 = single; 3 = others).
X5: Age (year).
X6 - X11: History of past payment.
X12 - X17: Amount of bill statement.
X18 - X23: Amount of previous payment.

Source: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

Dataset 2 – Equity Return Prediction under Fama-French Three Factor Model

The second dataset is related to prediction of equity return under Fama-French Three Factor Model. It is interesting to see if equity return can be predicted by some factors, instead of being some white noise. This is essential in testing if the market is efficient under efficient market hypothesis, and also it may lead to profitable trading strategies.

The linear form of the model considered is: $ER_t = \beta_0 + \beta_1 ER_{t-1} + \beta_2 SMB_t + \beta_3 HML_t$

where t denotes the time, and the sampling frequency is daily,
ER is the excess equity return (equity market return – risk-free rate),
SMB is return of Small [market capitalization] Minus Big,
HML is return of High [book-to-market ratio] Minus Low.

Since this project is about classification, the dependent variable is $1\{ER_t \geq 0\}$ (1 if the excess return is positive or zero, and 0 otherwise), i.e. it is to predict the market is going up or going down by the three variables. In addition, only data on of after 19700102 is used, and thus totally 11142 days of data is available.

One more point worth mentioning is that this is a time series dataset, which means that order matters. One potential issue here about such a time series dataset is that there are un-modelled autocorrelation (other than the AR(1) that is already include in the linear form), and there are potential structure change as time passes. The aim of this analysis is trying the find of the pattern in long run instead of any short-term structure.

Source: http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

Part 1 – Clustering

K-mean clustering and Expectation Maximization are used.

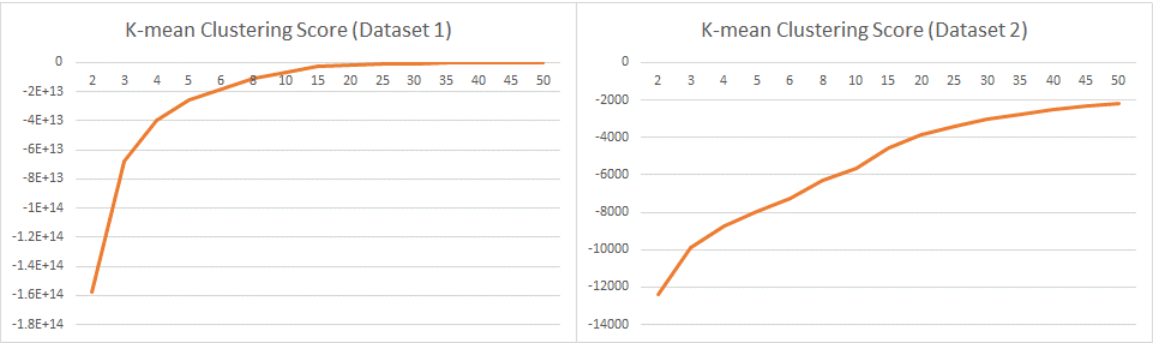
k-mean clustering

K-mean clustering is an algorithm that group the data points into k groups by iteration between classifying to the nearest mean and revaluating the mean.

Expectation Maximization

Expectation maximization is an algorithm that group the data points into k groups in a probabilistic sense, i.e. soft clustering. It assumes that each of the data points of probably in a certain cluster by a certain probability.

The scores of the two algorithms on the two datasets at different k are shown as follows:



As expected, as the number of cluster increase, the intra-cluster variation decreases, and thus the score increase, but the rate of increase become slower and slower and stabilize at the end.

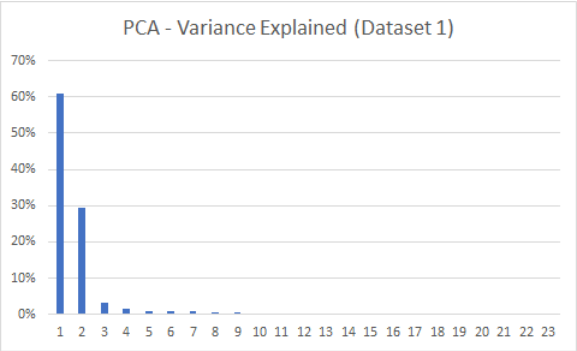
The number of cluster for both k-mean and expectation maximization can be determined by elbow method – choosing the point that adding another cluster doesn't give much better modelling of the data.

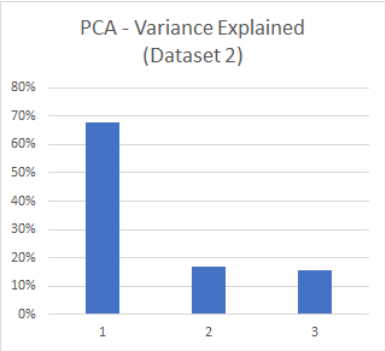
The suggested k for Dataset 1 is 15, and that for Dataset 2 is beyond 50. This is as expected as credit risk (Dataset 1) are generally regarded as easier to be modelled in statistical sense, while market movement (Dataset 2) are normally more difficult to model, which is consistent with the efficient market hypothesis.

Part 2 – Dimensionality Reduction

Four dimensionality reduction algorithms are tested in this project. They are Principal Component Analysis, Independent Component Analysis, Randomized Projections and Factor Analysis.

Principal Component Analysis (PCA)

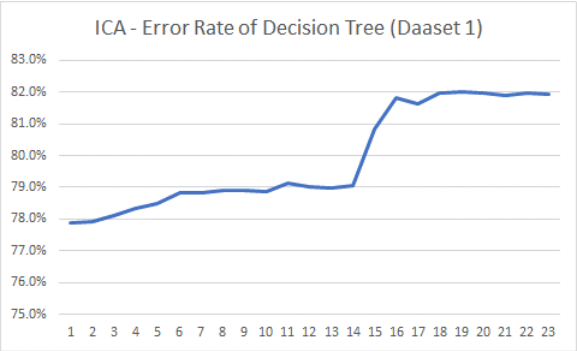


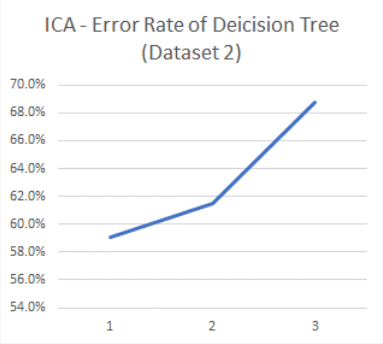


For the first dataset, the first two principal components already explained most of the variance, i.e. carry most of the information. This is somehow reasonable as quite some variables of the dataset are clearly higher related among each other (e.g. payment delay on 1st month and payment delay on 2nd month). Therefore, we can reduce the number of dimension greatly.

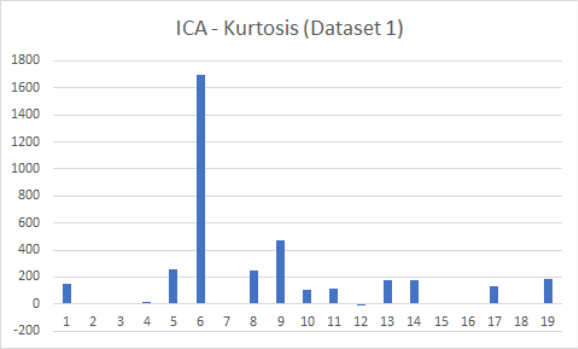
However, this is not the case for the second dataset. The first principal component is the most important one, explained nearly 70% of the total variance, but the other two remaining ones are not weak also. This somehow also shows that the force affecting the financial market is complex.

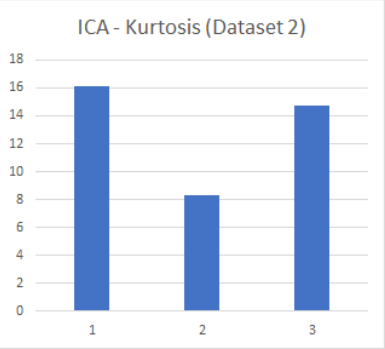
Independent Component Analysis (ICA)





The first task of ICA is to determine the number of independent components. This is done by fitting a learning algorithm (where decision tree is used in this project) for each total number of independent components and the one that with a higher accuracy is selected. The graphs on the previous page shows that a total of19 and 3 independent components are suitable for dataset 1 and 2 correspondingly.



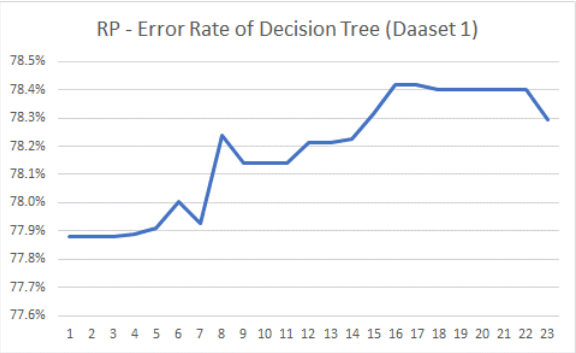


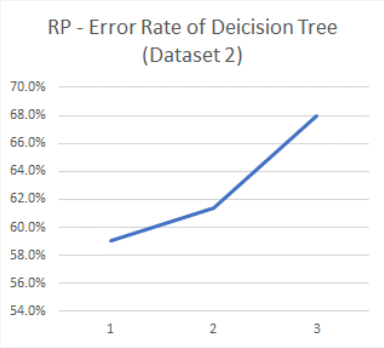
The next step is to extract the components from the total pool. This can be done by selecting the one that are more likely independent, i.e. the non-Gaussian one (as the Gaussian ones are more likely to be actually combination of other factors due to Central Limit Theorem).

For the first dataset, a group of the components have significantly much high kurtosis (which are larger than 100), than others, indicating their non-Gaussian properties, and thus they are chosen (totally 11 of them).

For the second dataset, all three components have a moderate kurtosis, and since there are only three variables, it is advised to just take all three of them.

Randomized Projections (RP)





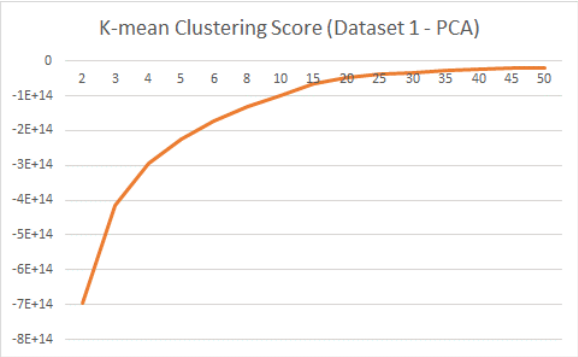
Similar to ICA, a learning algorithm (where decision tree is used in this project) is fitted for each total number of independent components and the one that with a highest accuracy is selected. The graph looks similar to the ones for ICA, except that the accuracy rates are really similar across different number of components for dataset 1, unlikely the ICA case.

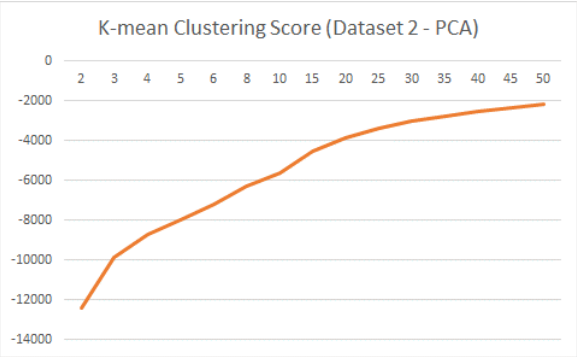
Factor Analysis (FA)

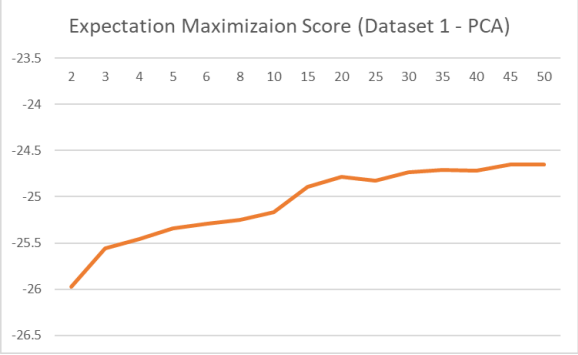
Part 3 – Clustering after Dimensionality Reduction

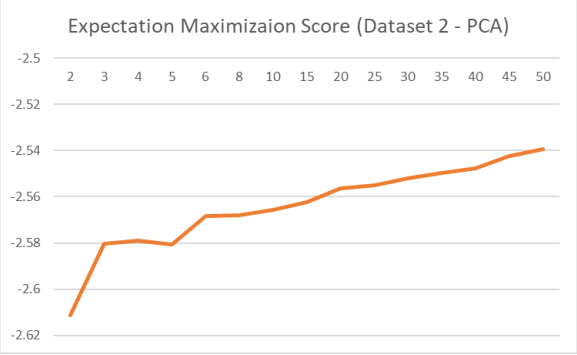
It will be interesting to see how the clustering algorithm perform on the dataset after dimensionality reduction instead of the original one

Principal Component Analysis





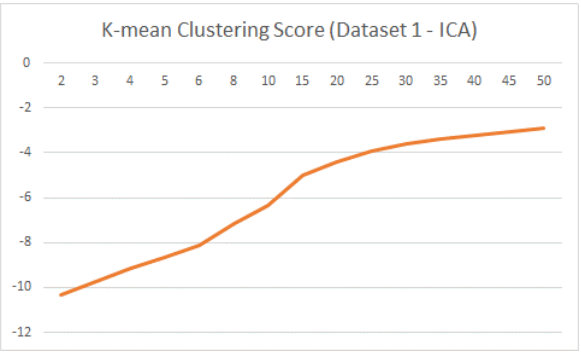


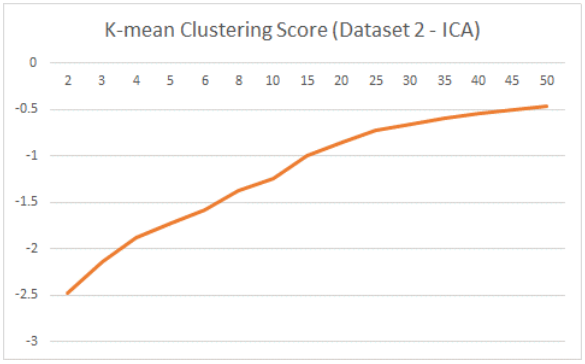


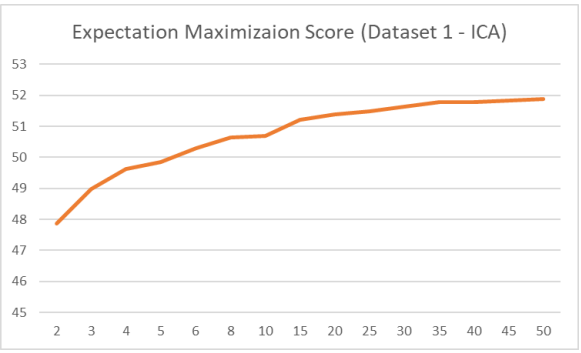
As the graphs shows, the results are similar to the one before transformation by PCA, except that the graph of expectation maximization for dataset 1 is much smoother.

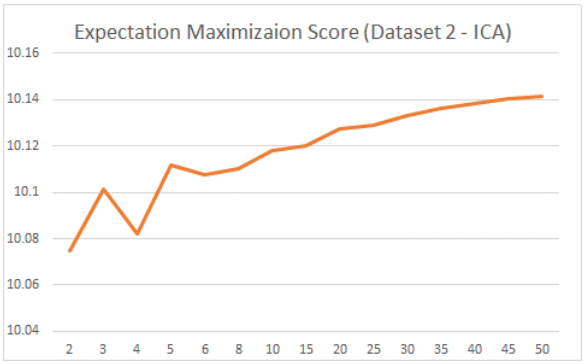
This somehow also shows that the process of PCA is not losing much information in reducing the dimensions.

Independent Component Analysis









As the graphs shows, the results are mostly similar to the one before transformation by ICA, except that the slope on the graph of k-mean for dataset 1 is steeper, which indicated that there maybe more cluster in dataset 1.

Part 4 – Neural Network after Dimensionality Reduction

Dimensionality reduction is a very useful technique in supervised learning. It is because many supervised learning algorithms suffer the curse of dimensionality issue, where the speed and/or the accuracy of the algorithm is largely affected; and with a suitable dimensionality reduction technique, the curse of dimensionality can be avoided.

An artificial neural network model with two hidden layers (both with eight perceptron) is fit to the Dataset 1, which is subjected to various dimensionality reduction techniques, and are compared to the original case.

The whole dataset is treated by dimensionality reduction techniques and then the transformed variables are fit to the neural network instead of the original one, except for the base case. The accuracy of the model is one minus the testing sample error estimated by 10-fold cross validation.

	Base Case	PCA	ICA	RP
No. of Variables	23	2	11	16
Testing Sample Accuracy	67.6%	69.8%	81.2%	77.7%
Elapsed Time (s)	9.93	4.70	26.26	30.07

Both PCA, ICA and RP report a higher testing sample accuracy and longer running time when compared with the base case. The improvements of the accuracy are very likely because of the reduction in dimension, which make the sample space denser, which compensate the loss of information in the process. This shows that dimensionality reduction is very helpful in neural network model,

Except the original case, the one with lower dimension are faster.

One more point worth mentioning here is that the result for RP is different when it is run again and again, and the figures shown here are just one sample of that. Nevertheless, the test sample accuracy is larger than that of the base case for most of the time.

CHAN, Siu Hang (Sammy), Frankfurt am Main, 2018-04-02 05:54