Unsupervised Learning and Dimensionality Reduction

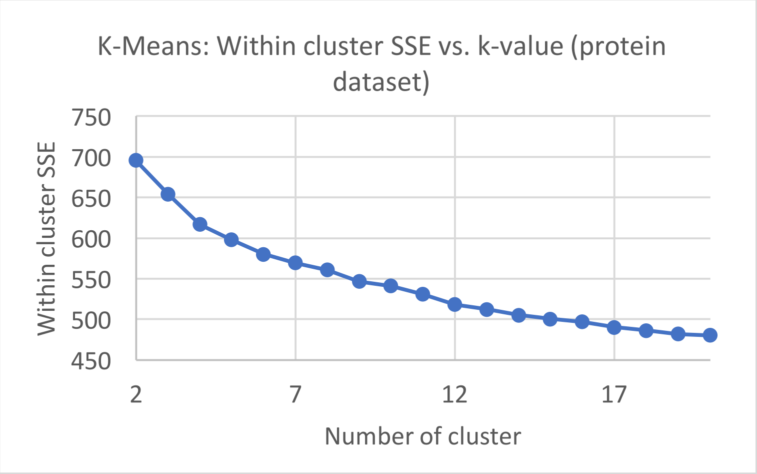
Georgia Institute of Technology CS 7641: Machine Learning Assignment Three

Xi Han

GT ID: xhan306

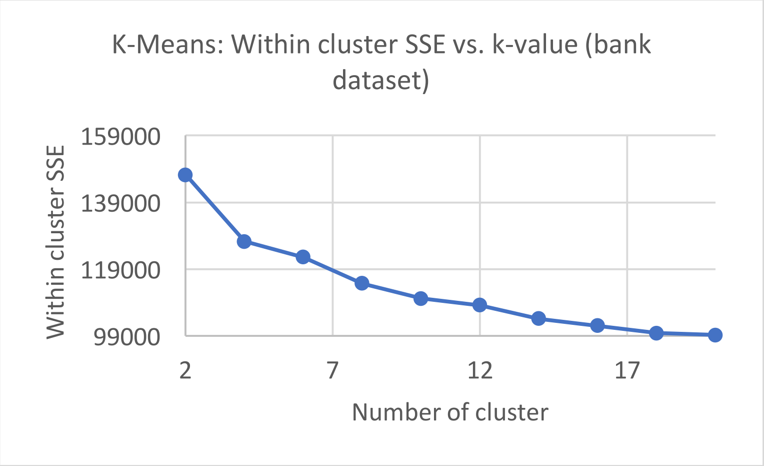
Introduction

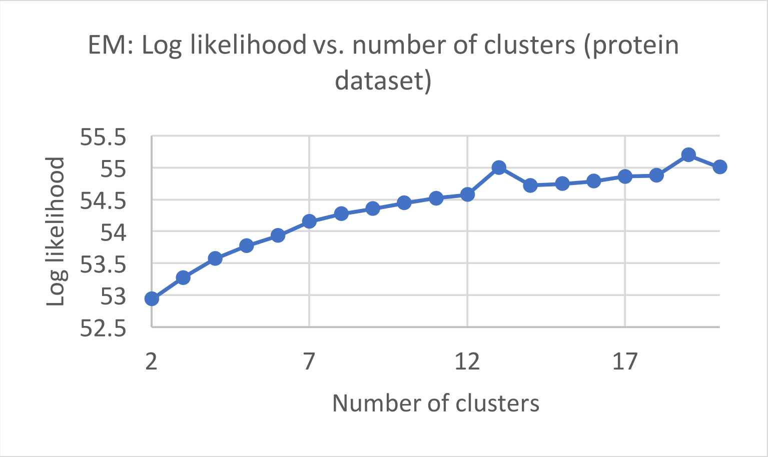
This report explores two clustering algorithms k-means clustering (K-Means) and expectation maximization (EM) and four dimensionality reduction algorithms principal component analysis (PCA), independent component analysis (ICA), random projection (RP) and information gain (IG) in machine learning for pre-processing data. In addition, both clustering and dimensionality reduction techniques were investigated further for training artificial neural networks in this report.



Dataset Description

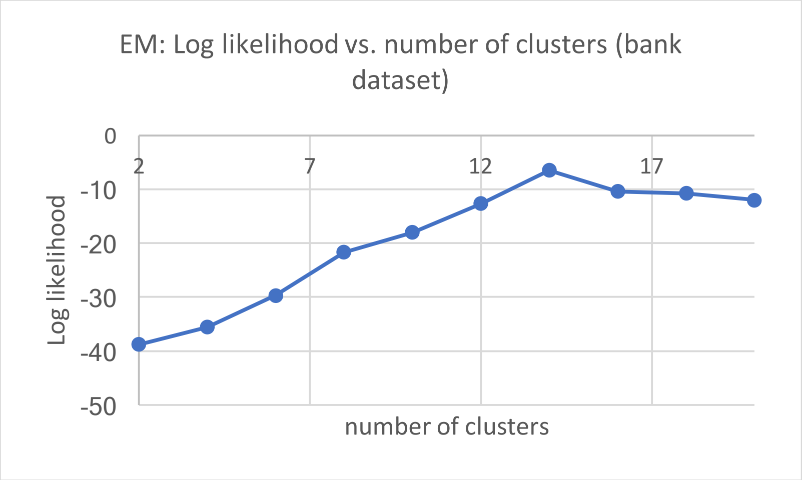
1. Protein dataset:
This dataset includes gene name, protein sequence and protein solubility for 3148 proteins. Producing some proteins *in vivo*, however, inefficient in industrial setting, resulting from low activity of the recombinant proteins. As biocatalysts and therapeutic agents, proteins with high activity have lower unit production cost. Developing biocatalysts with high activity has thus become an important goal for further development of many biocatalytic processes. Some empirical experimental strategies are time-consuming, expensive and often fail due to opaque reasons. A computational model *in silico* is





highly desired for predicting protein activity represented by protein solubility from its amino acid sequence. This dataset is used to predict protein solubility from protein sequence.
After data pre-processing mentioned in the previous reports, there are 20 attributes and one class represents whether the proteins are soluble.

2. Bank dataset:
This dataset was used to predict whether the client would subscribe the term deposit product or according to personal information, social economic index recorded from Portuguese banking institution. There are 30891 observations in the bank. They 20 attributes including both continuous variables and categorical. For both dataset, 25% of original data was set as test data and 70% was training data.



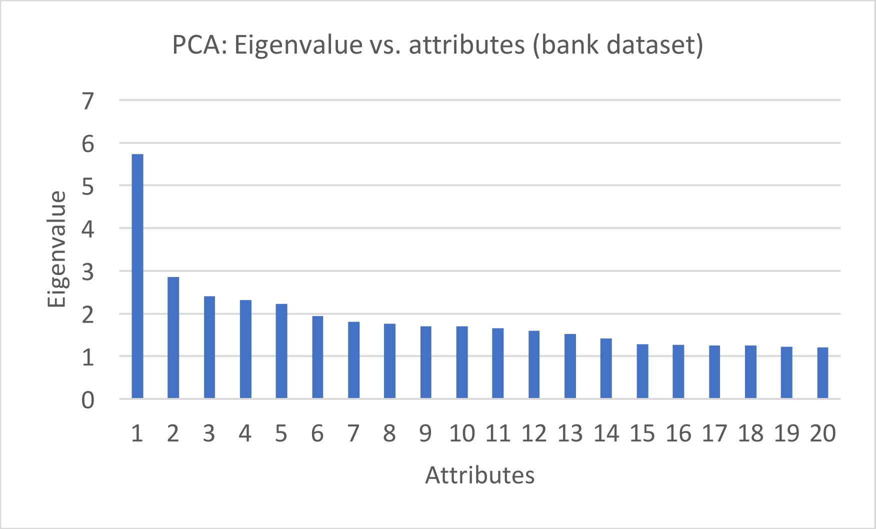
Part 1: Clustering

Clustering is a method of grouping the instances together such that instances which belong to same cluster are more similar to each other than those in other clusters. In this section, K-Means clustering and Expectation Maximization (EM) algorithms are explored in WEKA.

In K-Means, the similarity metric we have chosen to use for the analysis of clustering algorithms is the default one given by WEKA, which is the Euclidean distance, which uses a standard distance formula on attributes. This was found to be better than other available distance metrics such as the Manhattan distance or the Minkowski distance, both of which were deemed unsuitable for our dataset, as it was pre-normalized and made continuous, disqualifying the Manhattan distance, and we are not dealing with more generalized higher dimensions with our number of attributes/classes as would make the Minkowski distance suitable. Contrast to K-Means, EM is structured with probability distributions. It uses maximum likelihood parameters.

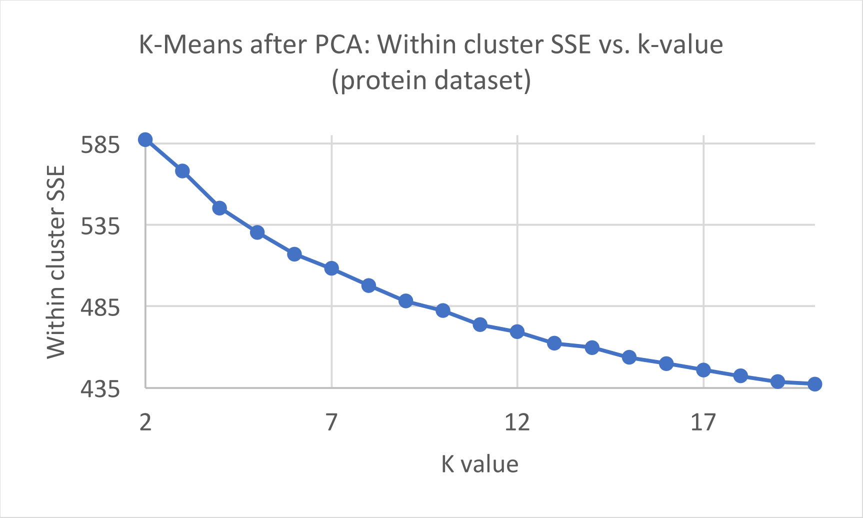
eigenvalue	proportion	cumulative
2.17484	0.10874	0.10874
1.96577	0.09829	0.20703
1.85785	0.09289	0.29992
1.58498	0.07925	0.37917
1.39356	0.06968	0.44885
1.13966	0.05698	0.50583
1.00789	0.05039	0.55623
0.91764	0.04588	0.60211
0.87815	0.04391	0.64602
0.83537	0.04177	0.68779
0.82553	0.04128	0.72906
0.76647	0.03832	0.76739
0.74815	0.03741	0.80479
0.70892	0.03545	0.84024
0.6928	0.03464	0.87488
0.67274	0.03364	0.90852
0.6386	0.03193	0.94045
0.60639	0.03032	0.97077
0.58467	0.02923	1

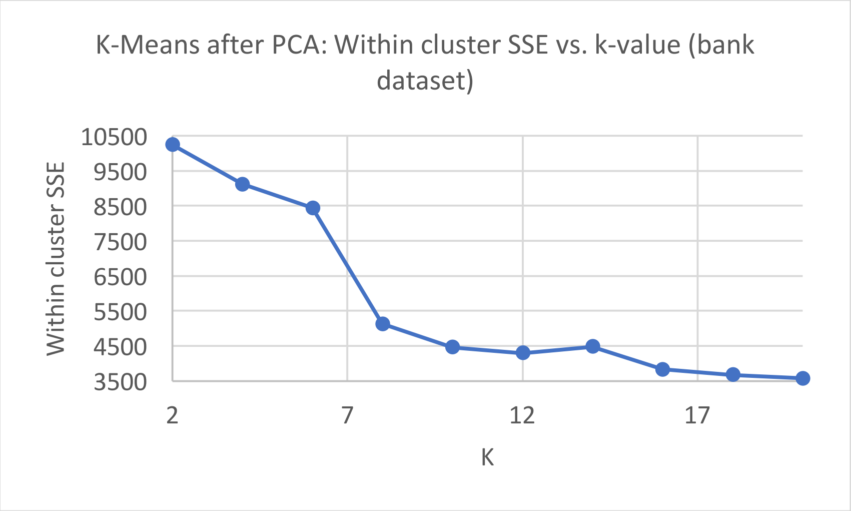
EM alternates between estimating the log-likelihood of current estimates (E step) and maximizing the likelihood based on the E step (M step).



1.1.K-Means

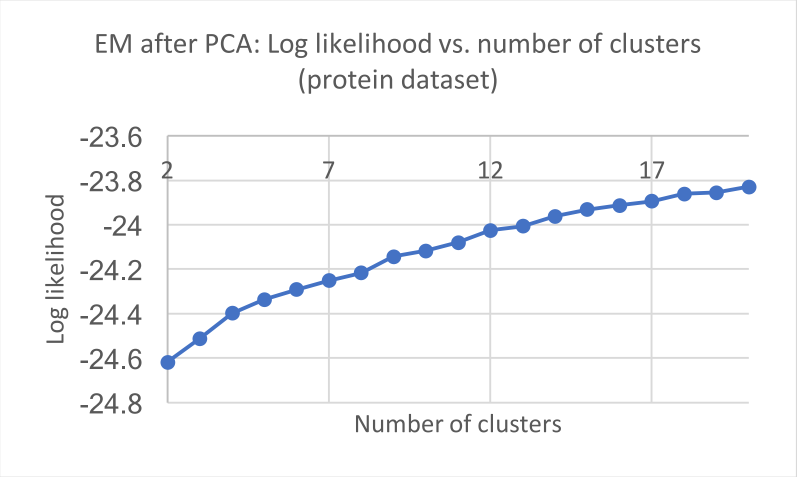
K-means, being one of the simplest clustering algorithms, begins by picking k centres at random. Each centre then claims all the points of data that are closer to it than to other centres. For each of these clusters, the centre point is then recomputed based on the averaged or mean location of all cluster points. The algorithm then begins again by claiming points with these newly recomputed centre points. This process continues until convergence, or when the centre points no longer move. With k-means clustering, there is only one hyperparameter that we can tune, which is the value of k, corresponding to the target number of clusters. We then attempt to find this optimal number of clusters for our data. It can be seen from our figures, with the increase of clusters, the sum of squared errors within each cluster gradually decreased. The elbow method was used to find the appropriate number of clusters for our datasets. We can distinguish a clear elbow point at which the error no longer decreases significantly. For the two datasets, this elbow point was determined to be k = 17 for the protein dataset, and k = 16 for the bank dataset.

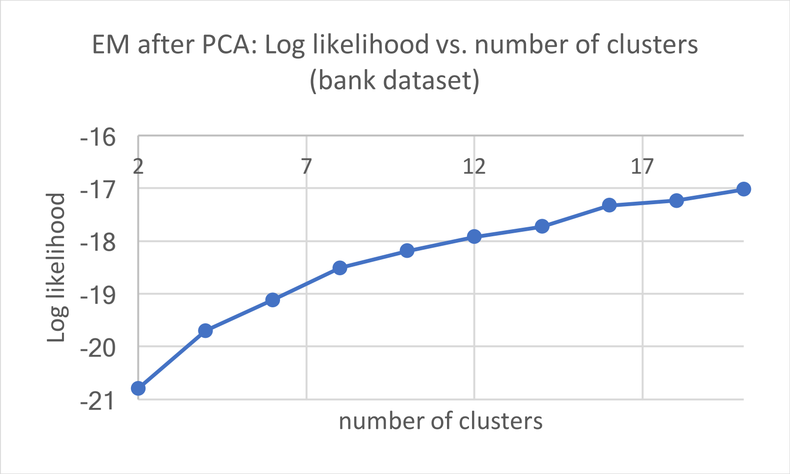




1.2.EM

Unlike K-Means, EM is a soft clustering algorithm, which effectively allows a point to be in potentially as many clusters as possible (as opposed to hard clustering, where a point is in a cluster or not). Then, for expectation maximization, instead of a data point being considered to be entirely in one cluster or not, for some k number of clusters, we assign to each point k probabilities, which signify the probability of that point being in a certain cluster, which in effect actually corresponds to a Gaussian distribution for each cluster, and the probability represents the probability that that distribution could've generated such a point. In the training process, EM is obviously slower than K-Means. When the cluster number is 14, we can see the angle in the log probability curves for both datasets.



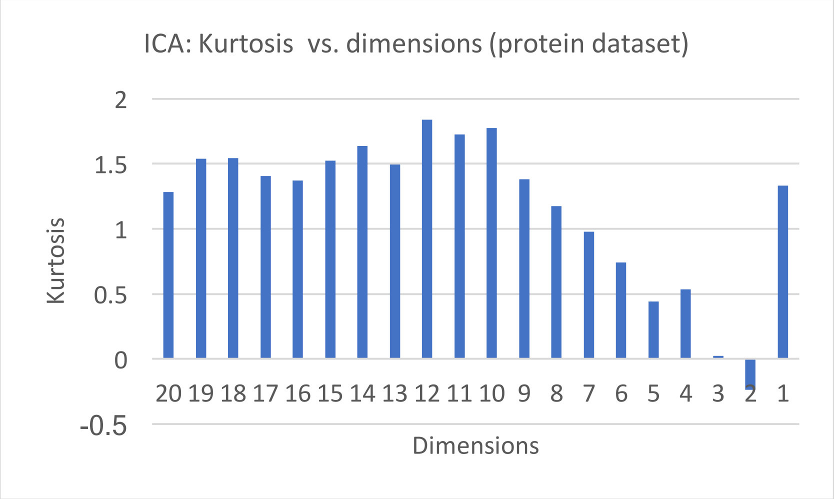


Part 2: Dimensionality Reduction and Clustering

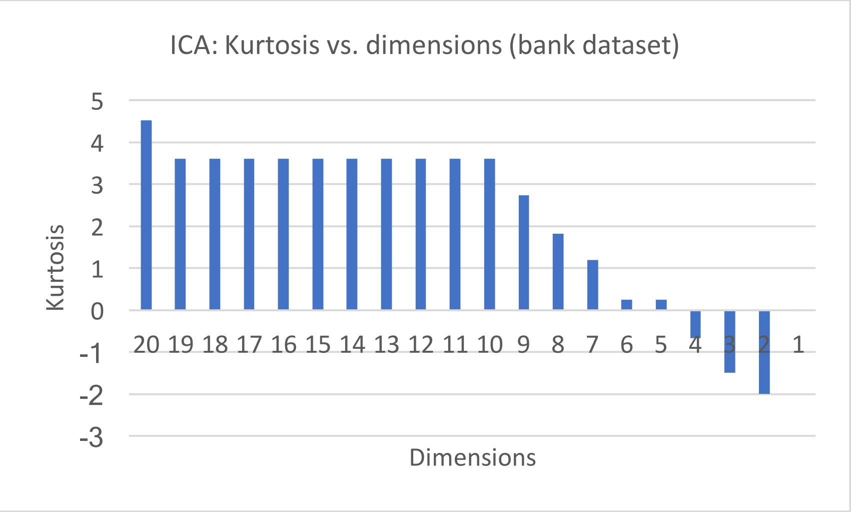
Dimension reduction algorithms transform the input data to fewer dimensions. There are multiple ways of doing this, which are typically split into two general ideas: feature selection and feature transformation. Four algorithms PCA, ICA, RP and IG were implemented in "Preprocess" of WEKA.

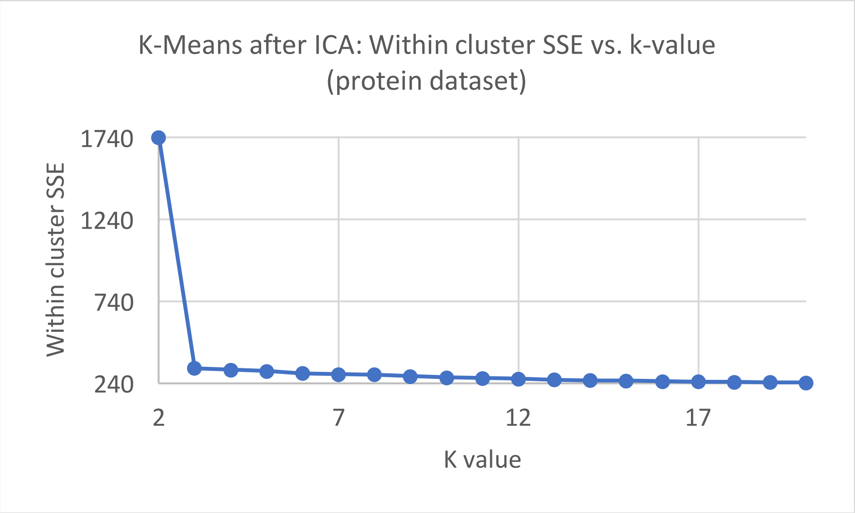
2.1.Principal component analysis (PCA)

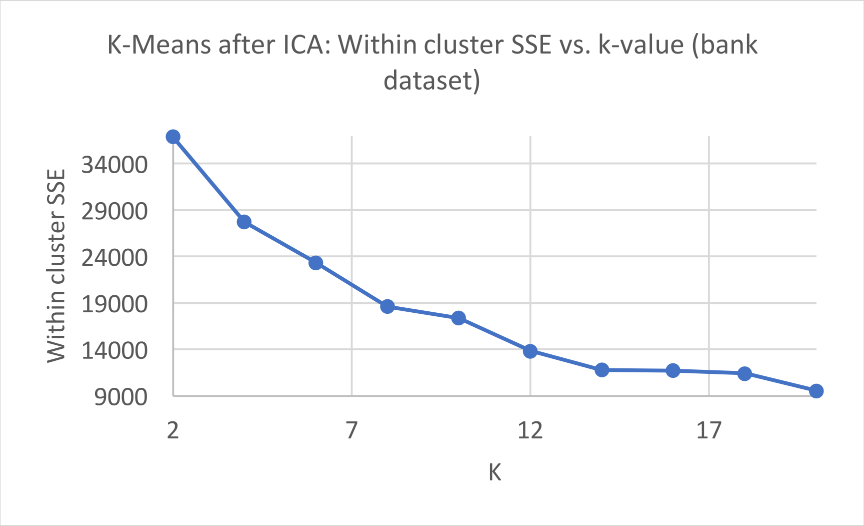
Principal component analysis (PCA), a feature transformation algorithm that interprets the problem as an eigenproblem, and attempts to find a linear transformation of the features, interpreted as axes of a plot of the data, such that the projections of the data onto the transformed



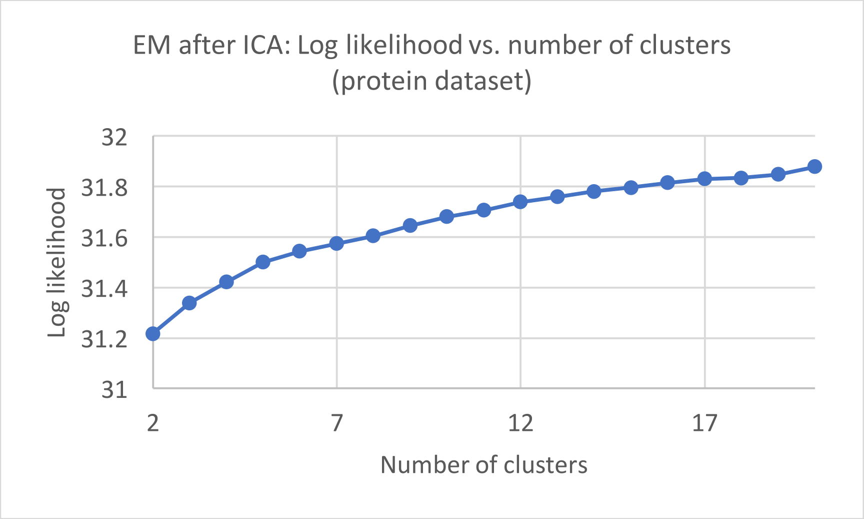
axes have maximum variance. These transformed axes are known as the principal component vectors. After finding a first principal component vector, the result of a linear transformation onto the original vector of features, we then look for orthogonal principal component vectors to add more dimensions with which the data is modelled. Each of these vectors have an eigenvalue which decreases the more principal component vectors that are found. The eigenvalue of each attribute is listed in the table for protein dataset and plotted for bank dataset. According to the default setting in WEKA, 95% information were remained, and therefore 18 attributes were remained. Because the biological meaning for our dataset, the sum of 20 attributes for each protein should be 1. 19 attributes are actually independent in the 20 attributes, which is also shown in the table above. Finally, 18 attributes were selected for further clustering. For the bank dataset, when the attributes were 12, the curve of eigenvalue gradually became flat and 12 attributes were chosen. Clustering algorithms are applied on the transformed data after PCA with 18 attributes for protein dataset and 12 attributes for bank dataset. For protein dataset, as we can see from SSE and log probability, the curve has an ambiguous angle. PCA transformed data has similar performance curves as the original dataset. However, with PCA, SSE is lowered and log probability is increased for the protein dataset. This indicates that PCA makes it easier to cluster the data. For the bank dataset, PCA seems not beneficial for the clustering.

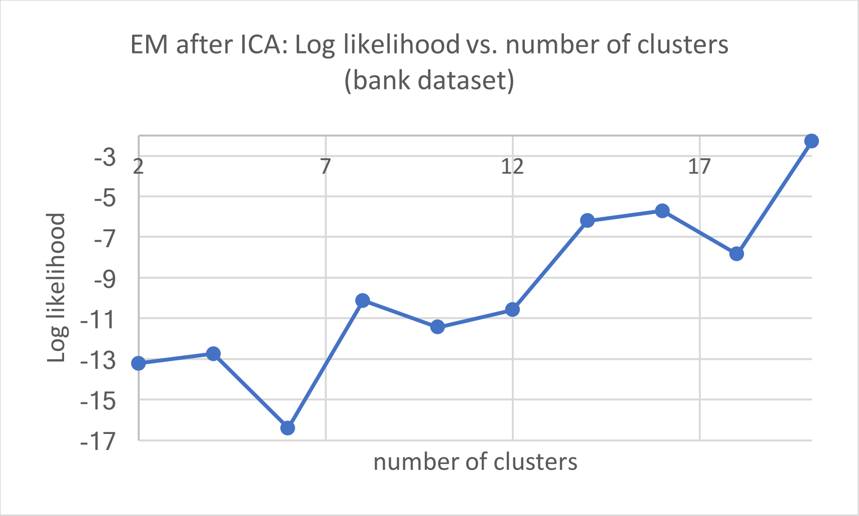




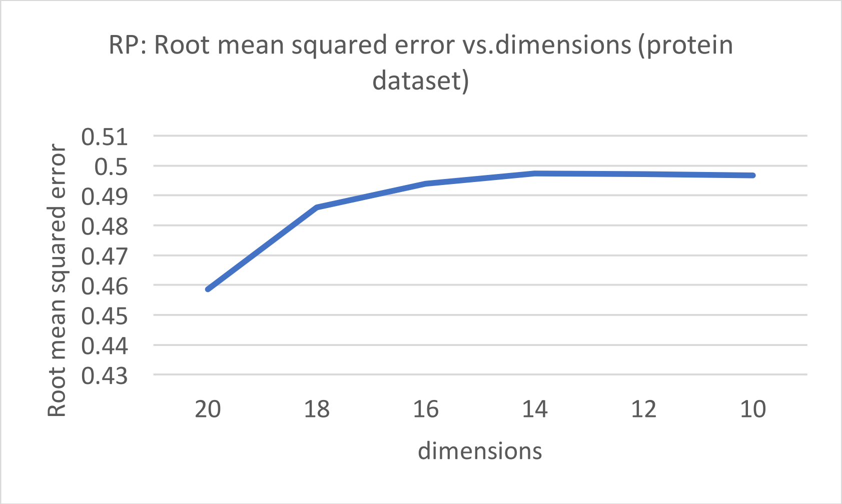


2.2. Independent component analysis (ICA)

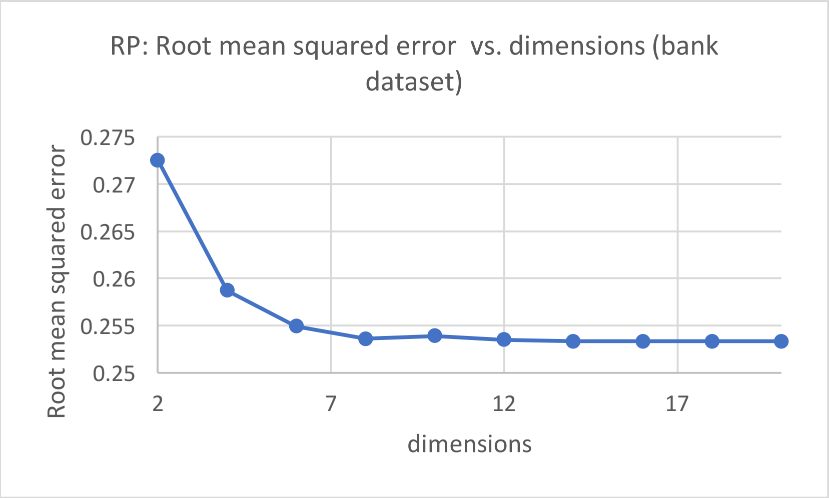


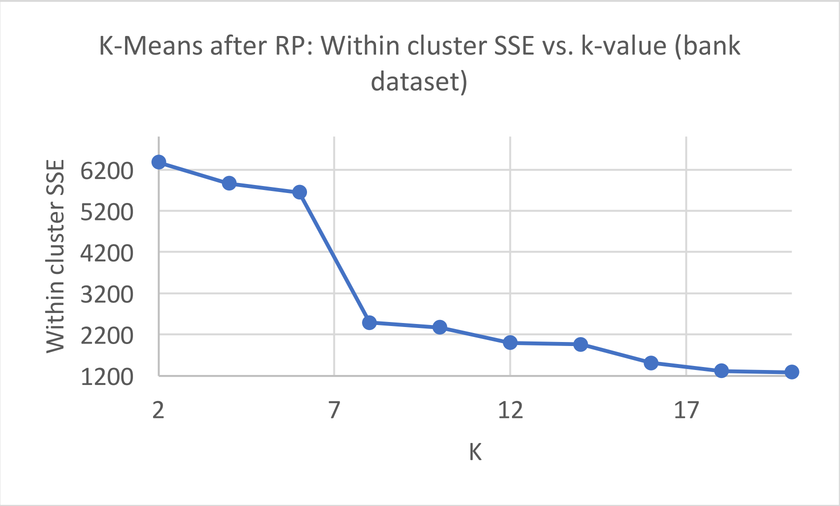


Independent component analysis (ICA) is another algorithm for feature transformation that has largely the same goals as PCA, but attempts to achieve it in a different way. ICA finds the independent components (also called factors, latent variables or sources) by maximizing the statistical independence of the estimated components. We may choose one of many ways to define a proxy for independence, and this choice governs the form of the ICA algorithm. The two broadest definitions of independence for ICA are minimization of mutual information and maximization of non-Gaussianity. We then ran the ICA algorithm on each dataset, computing the independent components for each dataset; we then looked at the kurtosis of each independent component, attempting to determine its similarity to a normal distribution. Gaussian signals have zero kurtosis, Super-Gaussian signals have positive kurtosis, and Sub-Gaussian signals have negative kurtosis. The goal of projection pursuit is to maximize the kurtosis, and make the extracted signal as non-normal as possible. According to the figures of kurtosis with dimensions for protein dataset, 12 attributes were selected since there was a peak of kurtosis. For bank dataset, 10 attributes are chosen as



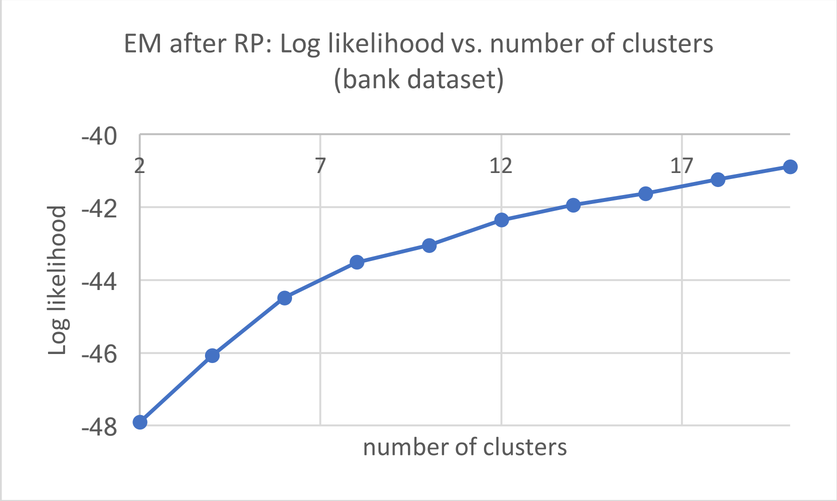
the kurtosis reduced significantly after point 10.



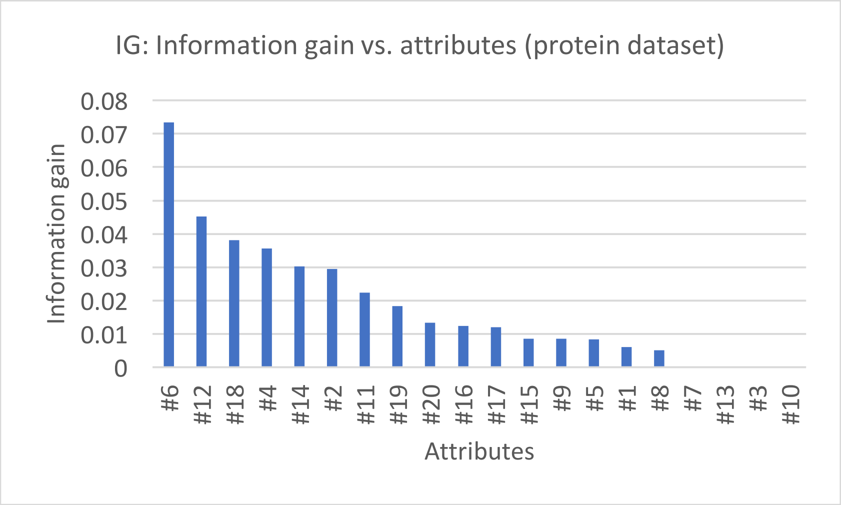


Clustering algorithms are applied on the transformed data after PCA with 12 attributes for protein dataset and 10 attributes for bank dataset. For protein dataset, from the SSE and EM log probability plots, we use elbow method and there is an obvious angle at cluster 2. It is reasonable since there are 2 classes in the original dataset, which indicates whether the protein is soluble. For the bank dataset, it is not very obvious to tell the good cluster number using elbow method for SSE and EM log probability plots. Some other methods can be explored to find the suitable number of clusters or attributes for dimension reduction such as plotting the accuracy of an algorithm after clustering and dimension reduction when the elbow method is not helpful.

2.3. Randomized Projections (RP)

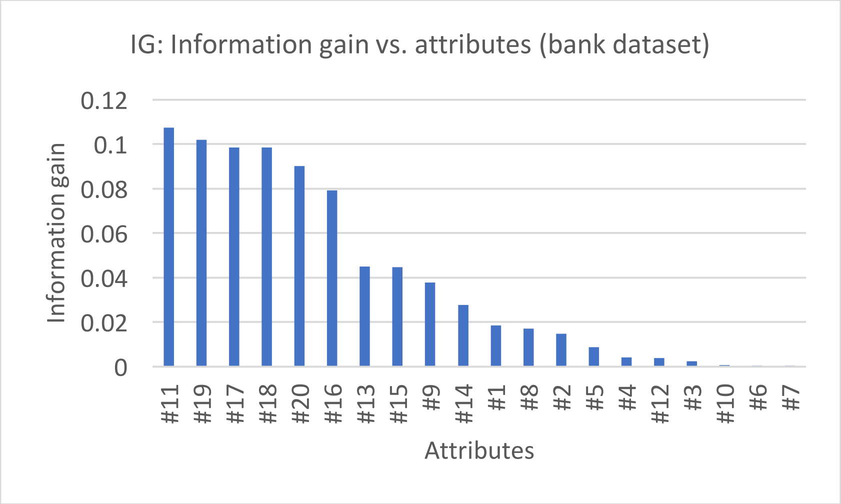


While PCA and ICA are quite powerful algorithms in their own right, RP runs significantly faster and works well enough in practice. True to its name, the algorithm takes a specified target number of dimensions, and uses that information to generate a randomized matrix that is then used to transform and project the features into another feature vector in a smaller feature space of the specified number of dimensions. Random projection is implemented in WEKA. We first set the dimension number to the number of attributes, then remove the attribute one by one to get the classification error. According to the figures of protein dataset, the error increased significantly after dimensions were reduced. Therefore, 20 attributes were remained for the clustering, which is the same with the original one because no attributes were removed for this dataset. For the bank dataset, 10 attributes were chosen since the error is still low until remaining dimensions were 10. After RP, there is an obvious elbow at cluster 6 for K-Means and EM.



2.4. Information Gain (IG)

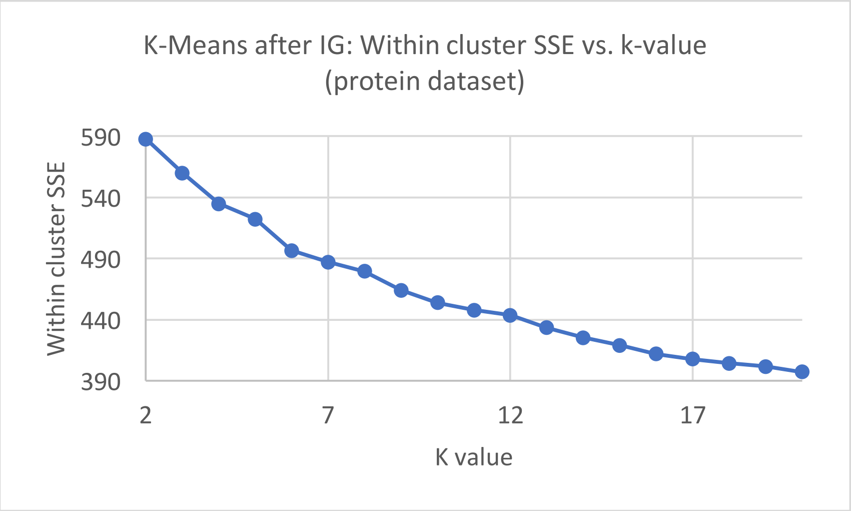
While PCA, ICA, and RP are all feature transformation algorithms, information gain is a feature selection algorithm. Information gain comes from information theory, and may be better known as the Shannon entropy. It is a statistical measure making use of conditional probability that ranks our features, by how impactful they are in determining the class. That is to say, the greater this value for information gain, the more important that attribute is in predicting the outcome. The information gain is ranked in the figures for both dataset. For protein dataset, there is 4 attributes with information gain 0, which was removed and 16 attributes were selected for clustering. For bank dataset, the first 14 attributes have larger information gain were remained according to the figures. After IG, SSE is lowered and log probability is increased for the protein dataset. This indicates that IG makes it easier to cluster the data. For the bank dataset, IG does not improve the clustering.



Part 3: Neural Network Learner Results

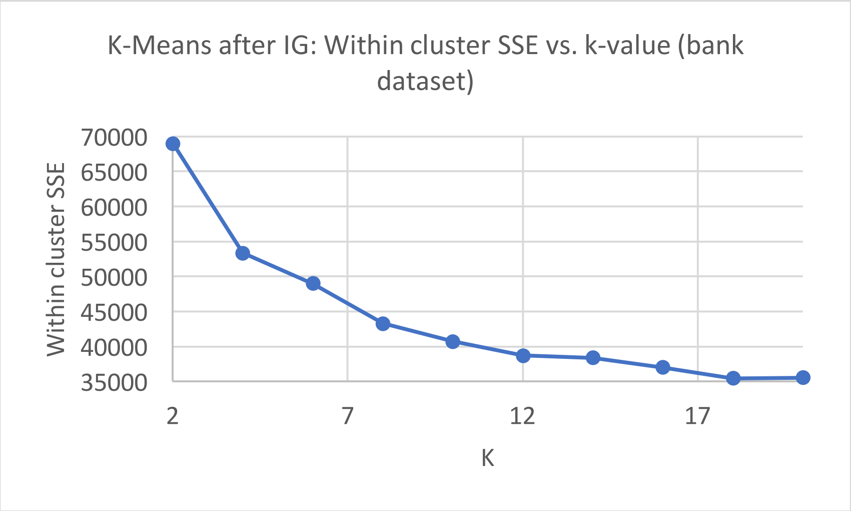
3.1.Results after dimension reduction

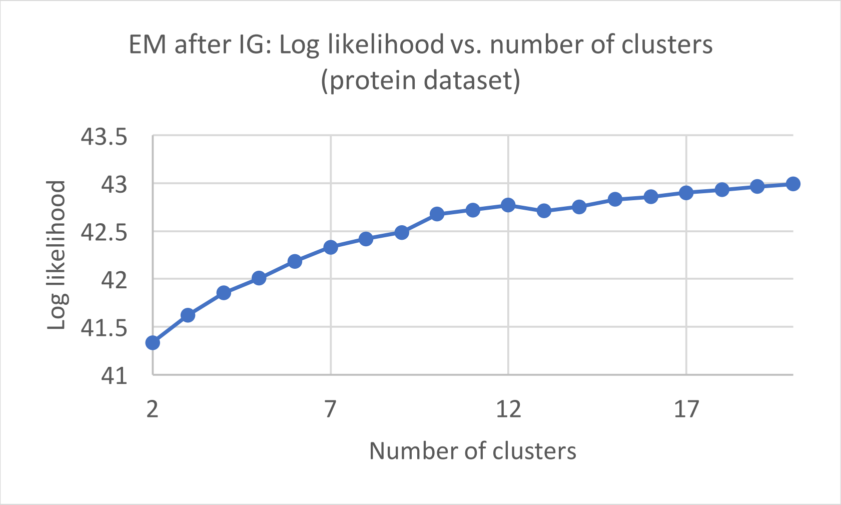
In this part, we pick protein dataset to train a neural network and with four dimensionality reduction algorithms. We apply four different algorithms on the dataset, then do a forward search from only one component or attribute to all 20 components or attributes with neural network classifier. The classifier is using default nodes in the hidden layer, with learning rate = 0.3 and momentum = 0.2 in Weka. We use 75% dataset as training and do not use cross validation because it takes much longer time and the neural network model is not the main focus here. Different dimensions were plotted in figure for four dimension reduction algorithms. It can be seen that when the attributes were removed, the accuracy of neural networks (NN) gradually decreased. For the protein dataset, IG seems have best performance among the four algorithms, which can achieve a relative high accuracy with low dimension. It demonstrates that IG remained the most important information in the process of dimension reduction. For all algorithms, 12 seems a general number of attributes which can remain most information.



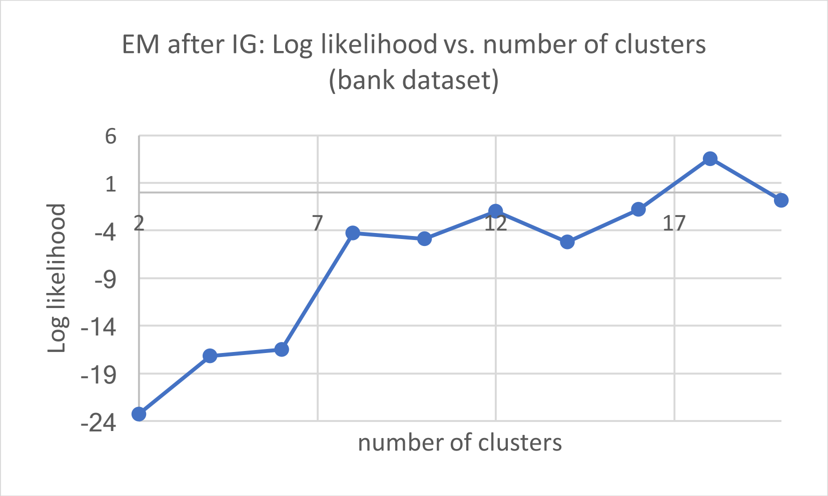
3.2.Results after clustering

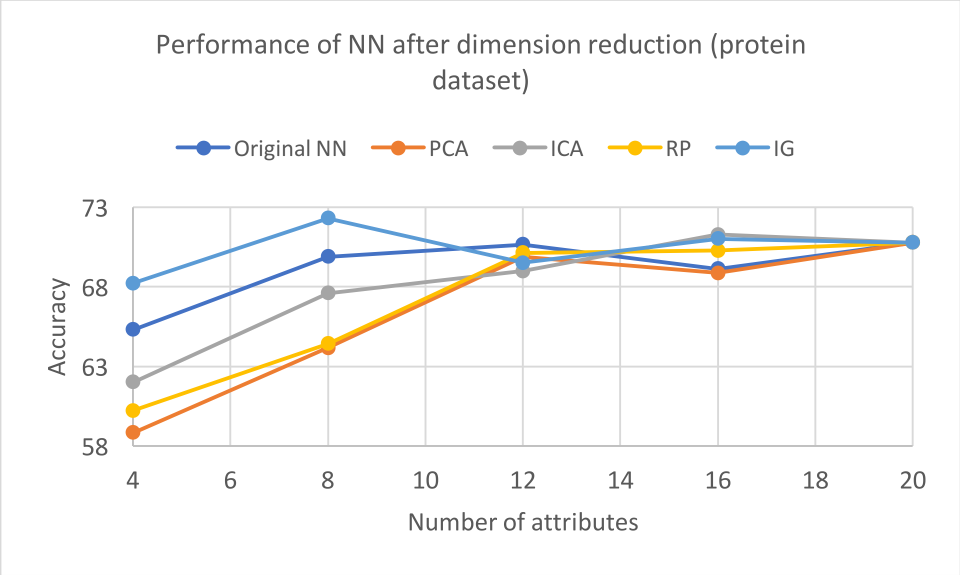
In this part, we explore how performance changes as clustering is introduced as an attribute. We use clusters as an additional attribute in addition to the original 20 attributes by “AddCluster” in Weka. Different number of clusters were explored here and plotted in the figure. It can be seen that when the number of clusters increased, the accuracy of NN always decreased. It was demonstrated that cluster 2 is more suitable for this dataset, and is exact the number of classes in the original dataset, soluble class and insoluble class. It is reasonable since the class in the original dataset divided the dataset into two obvious different groups. Therefore, when the elbow method is not helpful for find a suitable number of clusters, selecting a machine learning model and measure the accuracy after clustering a good approach.



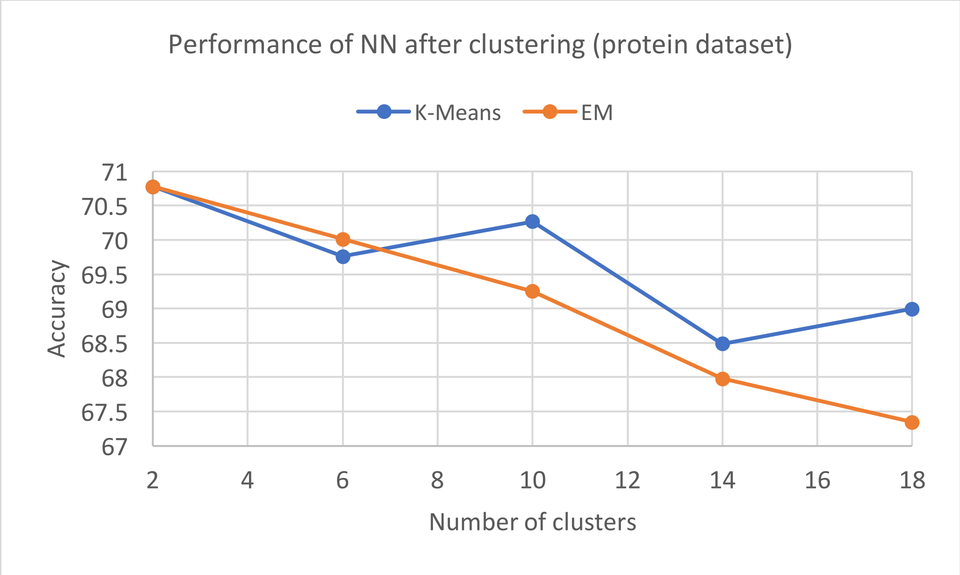


Part 4: Conclusion





Information gain is shown to have the best accuracy performance among all four dimensionality reduction algorithms. PCA also shows relatively good performance and it has shorter training time. For PCA and ICA, we also find out the low-ranking components (by eigenvalue or kurtosis) do not have worthy information and can be discarded for further dimension reduction. The Random projection performance varies but it actually shows some good accuracy results. We also find out PCA and IG transform the data such that it has lower K-Means SSE and higher EM log probability. The clustering added as an additional attribute did not help achieve better performance as the information of two clusters in the original dataset may already be included. And more clusters decreased the performance of NN.


Performance of NN after clustering (protein dataset)
K-Means
EM
Accuracy
Number of clusters