CS4647

Assignment 2

Kamal Sharma

July 12, 2017

DESCRIPTION

- Clustering of the provided *income dataset* and *Simeon digit dataset* by k-means
- Clustering of the mentioned 2 datasets by Expected Maximization framework
- Dimensional Reduction by PCA, ICA and RCA
- Clustering performed on the above two reduced dimensionality datasets
- Behavior of ANNs on the reduced dataset in contrast with full dataset of the *Simeon digit data*
- Applying clustering on the PCA reduced Simeon digit dataset

1 DATASETS

1.1 SEMEION HANDWRITTEN DIGIT DATA SET

This is one of the datasets used in assignment 1. It was created by Tactile Srl, Brescia, Italy and was later donated as a resource for machine learning research to Semeion Research Center of Sciences of Communication, Rome, Italy. A total of 80 individuals were asked to write down each digit (0,1,2...,9) twice on a sheet of paper. They were instructed to write down the digits carefully paying attention to details the first time while writing the same digit a second time in a fast and careless manner. These digits were then scanned and stretched to a 16x16 grid in grayscale of 256 values. Each pixel in the grid was then assigned a boolean value of either 1 or 0 according to a threshold function.

I find this dataset to be very interesting for unsupervised learning as I want to see if clustering can identify the classes and more than that if it can cluster similar kind of handwritings together giving more clusters than the possible values of the class (1,2...,9,0). Apart from clustering, I would like to see how much can the data be compressed as handwriting data is quite sparse.

ATTRIBUTES: There are 1593 rows in the dataset each belonging to a different instance. There are 266 columns out of which 256 correspond to each pixel in the 16x16 grid while the remaining 10 constitute boolean class attributes. For e.g., if the digit is identified as a 3 then the instance will have a 1 in column 259 (256 + 3rd digit) and a 0 in all other columns.

1.2 INCOME DATASET

2 PACKAGE USED

Waikato Environment for Knowledge Analysis
Version 3.9.1
(c)1996-2016
The University of Waikato
Hamilton, New Zealand.
Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

3 THE ISSUE OF MISSING VALUES

All missing values in the dataset were replaced by the mean if the values were numeric. The nominal missing values were replaced by their modes. This was done using the *ReplaceMissingValues* preprocessing filter in WEKA.

4 CLUSTERING OF THE PROVIDED *income dataset* AND *Simeon digit dataset* BY K-MEANS

4.1 INCOME DATASET

:The k-means clustering was done using the WEKA k-menas clustering algorithm. I ran the algorithm for different values of k, the number of clusters, and used the elbow method to determine that the $k = 10$, is the number of clusters after which we start to get diminished returns. Figure 4.1 the plot of k versus sum of square error:

$$J = \sum_{n=1}^{k} \sum_{i \neq j} |x_i^{(n)} - x_j^{(n)}|^2,$$

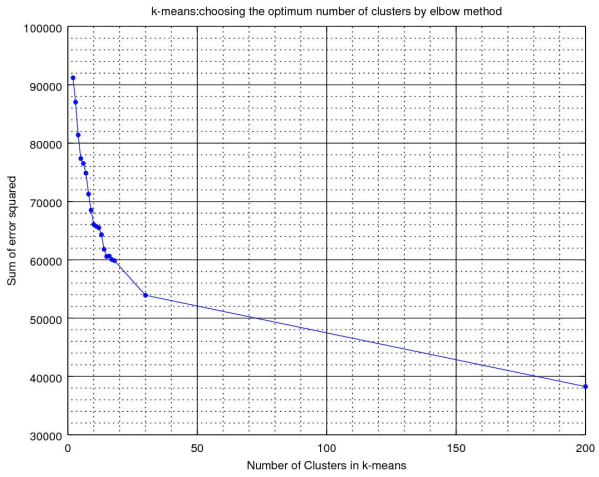

Figure 4.1: Determination of number of clusters in k-means clustering algorithm.

where $x_i^{(n)}$ is a data point belonging to cluster n. Here, y-axis represents sum of squares of each point's distance from every other point in the same cluster. There are quite a few *elbows* in this plot and so it is a matter of application accuracy and computing resources to choose number of clusters. Too many clusters are slow to build and give diminishing returns as k increases.

4.2 SIMEON DIGIT DATASET

For the Simeon digit dataset, the number of clusters can be found in the same way. It comes out to be near 15 by visual inspection of the elbow plot. Let's take $k = 15$. The sum of squared errors for 15 clusters for this dataset is

$$J_{digitfull} = 63194.99.$$

5 EXPECTATION MAXIMIZATION

The EM was also used the same way as k-means. The algorithm was run several times to find out the optimum number of clusters that give the maximum returns for increasing one cluster (the elbow) method. The difference is that here instead of sum of errors squred, we use the parameter *log-likelihood* for a given number of clusters. The process is summarized in figure 5. The leftmost point in the figure corresponds to $k = 2$. Running the algorithm again for three gaussian clusters improves the *log-likelihood* from -47 to -39. Further increase in number of clusters results in a zig-zag plot of log-likelihood until it jumps again for $k = 18$ clusters. To find a smoother curve I ran the algorithm many times with different random seed initializatoins. However, to me it seemed like the after $k =$, the returns per increase in

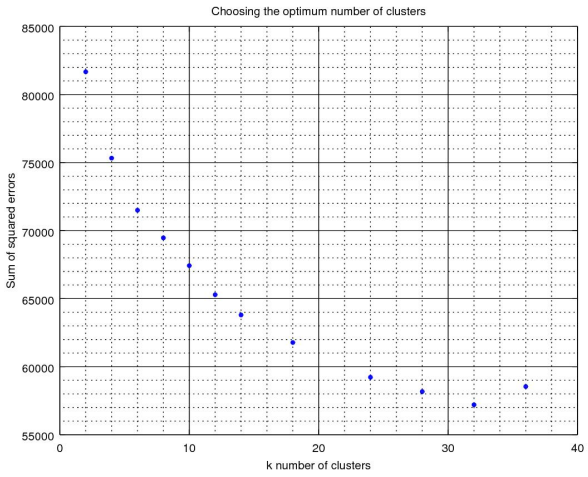

Figure 4.2: Elbow method applied on digit dataset to get the optimum value of k. $k = 15$ is the optimal number as far as I can tell.

the number of clusters are diminishing. This analysis led me to conclude that the optimum number of clusters should be $k = 3$. This, however, is very different from what I concluded from k-means in the previous section. There $k = 10$ was my choice although now its a little obvious that the k value could also have been near three based on the plot in figure **??** which has multiple elbows. By changing the parameter *clusterNum* in WEKA to -1, it runs a search automatically spewing out the optimum number of clusters for expectation maximization. I found that that algorithm calculates optimum number of clusters to be $k = 2$. Therefore, it rests on the user to determine the number of clusters based on the application or task at hand.

6 Dimensionality Reduction

According to the assignment, we have to use principal component analysis (PCA), independent component analysis (ICA) and the randomized projections (RP) on the two datasets.

6.1 PCA

I used the principle component analysis algorithm in WEKA library of packages to find the eigenvectors and the eigenvalues of the covariance matrix of the data. The PCA algorithm gives us the directions in the data space alomng which the variance of data is maximized. The lines are parallel to the eigenvectors of the covariance matrix and are called principal components. These components can be ranked in the order of their decreasing eigenvalues such that the first principal component represents the maximum variance, the second one a little less and so on. Looking at the cumulative variance we can choose the number of principal components and discard the rest.

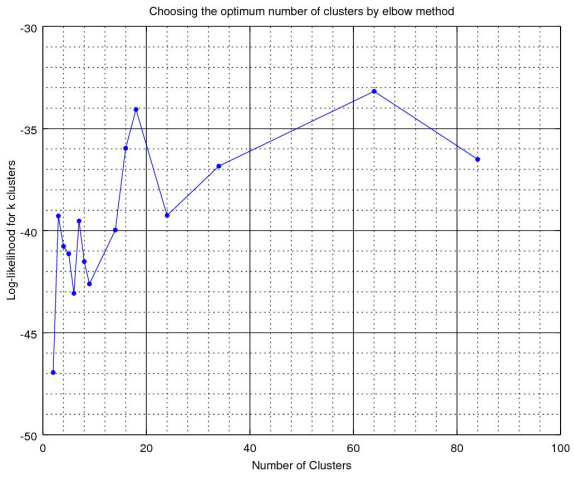

Figure 5.1: Log-likelihood plotted against number of clusters for Expectation Maximization.

6.1.1 PCA ON THE INCOME DATASET

The income dataset has 14 real attributes some of them numeric while others binary (see dataset description). The other two attributes are index numbers (or serial number of data points) and the prediction class. We can include or exclude the index values as they won't affect the PCA for a randomly arranged and aquired dataset. However, the *prediction* class needs to be excluded from the PCA because we do not want it to be used as an attribute and get mixed up into new principal components. There are two reasons for this: (1) we will use the data later for clustering and including class attributes in clustering will severly affect the output. We don't want to do that because we are performing unsupervised learning, (2)we do not want to introduce a bias which might prevent the clustering into smaller physically meaningful clusters.
In order to apply the PCA on our data we need to convert all data to numbers. The numeric attributes will work as they are but the nominal attributes can be changed to binary such so that mathematical operations can be performed on them in the algorithm. The conversion of all nominal attributes to binary attributes leads to an increase in the total attributes to 103. Now we can perform the PCA on this modified data which gives us a 103 principal components arranged in the order of their eigenvalues (or ranks). Dimensionality reduction is now reduced to pick the highest few components.

BUT HOW MANY SHOULD WE CHOOSE? This question can be answered by plotting the variance retained by chosen principal components against the number of principal components. Figure 6.3 shows the plot where we can easily see that after including the topmost 23 principal components, anymore increase gives us only a small benefit in variance covered. Thus we can simply ignore the principal components from 24 till the end because they contribute negligible variance to our transformed data.

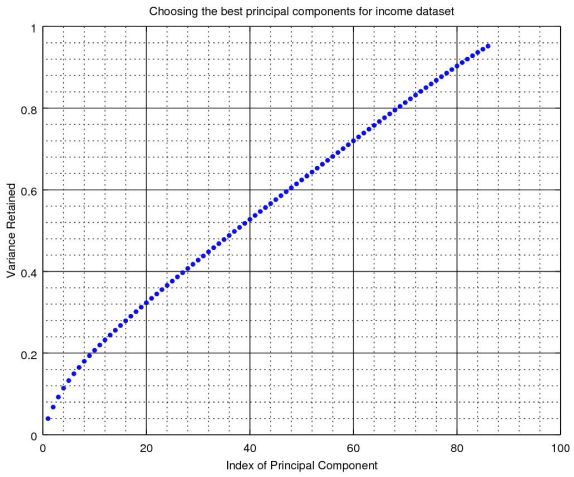

Figure 6.1: Retained variance plotted against number of retained principal components for .

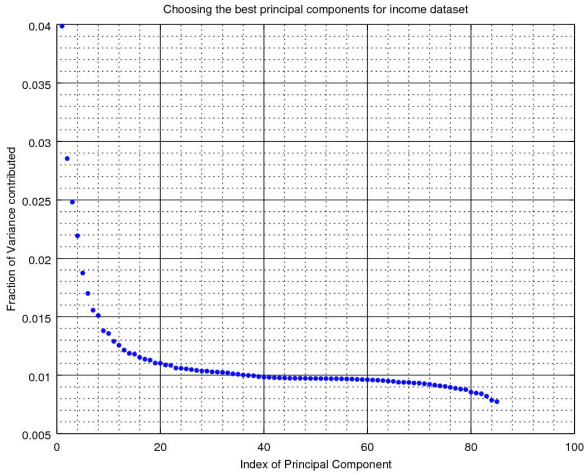

Figure 6.2: Retained variance plotted against number of retained principal components for .

6.1.2 PCA ON THE SIMEON DIGIT DATASET

The digit dataset is a specially large dataset with binary data from pixel's on or off state. It has 256 binary attributes for each pixel in a 16×16 grid and one nominal class attribute that identifies the digits in the set:

$$\{0,1,2,3,4,5,6,7,8,9\}$$

. It is obvious that there is a lot of empty space when a digit is written down in a square cell. Therefore, I expect there to be a substantial amount of compression possible in this case. The principal component analysis gives us 256 components out of which I selected first 30 principal components and ran them through a neural network.

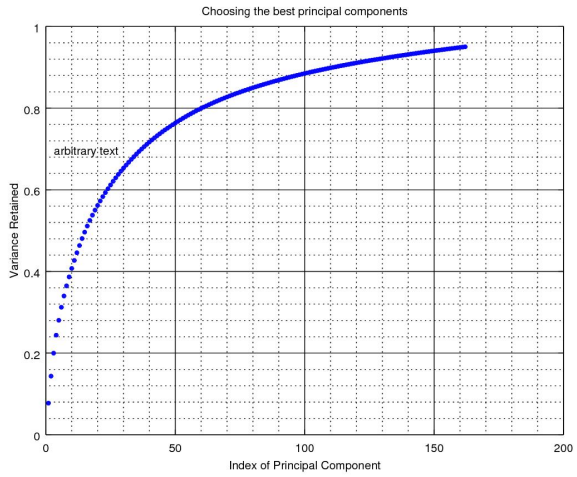

Figure 6.3: Retained variance plotted against number of retained principal components for Simeons digit data. Notice the curve as compared to the straight line for income dataset.

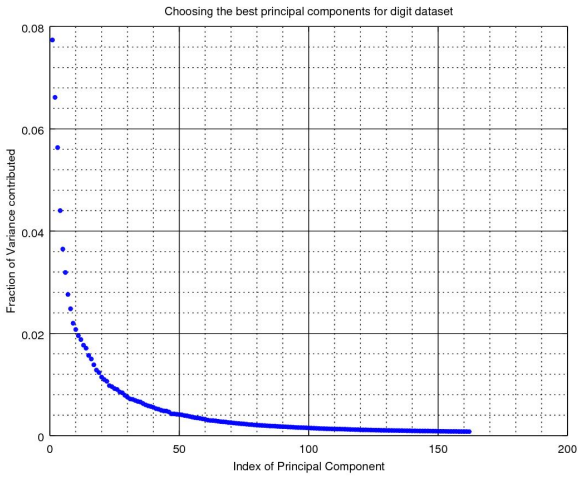

Figure 6.4: Contribution to variance plotted against number of principal components for Simeons digit data. Contribution from higher principal components is clearly more.

6.2 ICA

WEKA has ICA as a preprocess filter which transforms the data into independent components. Of course the assumption that ICA will be useful for a dataset depends on whether it is composed of different independent sources and that the data is non-gaussian.
following by visually analyzing the 2D plots of various sources (ICA generated attributes). The

most obvious difference between PCA and ICA is that the classes are not separated at all in ICA plots. If same kind of rules govern the two data, i.e., they don't have different sources or the coherence in the data is lost, then I would expect this kind of behavior from the data. Another thing that is very conspicuous is that PCA plots are stretched along various directions while the ICA plots seem like globular or elliptical clusters with equal amounts of points from both classes.

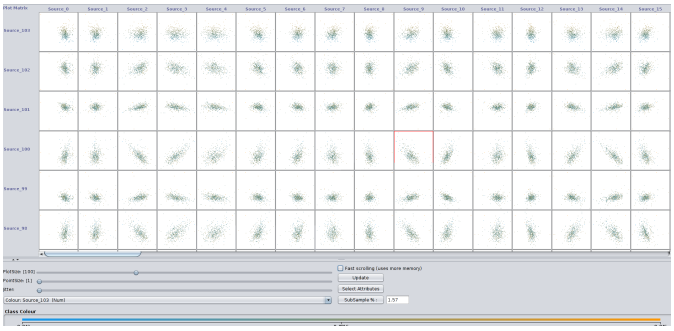

Figure 6.5: Almost well mixed clusters visible in ICA plots for income dataset.

Comments related to specific questions in assignment 2 The clusters I generated in the income dataset had a class attribute *prediction* which I ignored while generating the clusters. In some cases I can clearly see that the clusters have majority of one class while in others it is a pretty mixed collection of datapoints. This was further more clear after doing PCA. I plotted few pairs of principal components in 2D plots and indeed it is possible to see some clusters. I could see some long and thin collections of data separated by low density spaces. Although the lable for income data were just binary, there are a lot of sub-groups in the data with specifc nationalities, age, gender, etc. that can be taken as clusters. The lables don't generally fit them but there is a clear separation of classes in some of the plots. See figure 6.2.

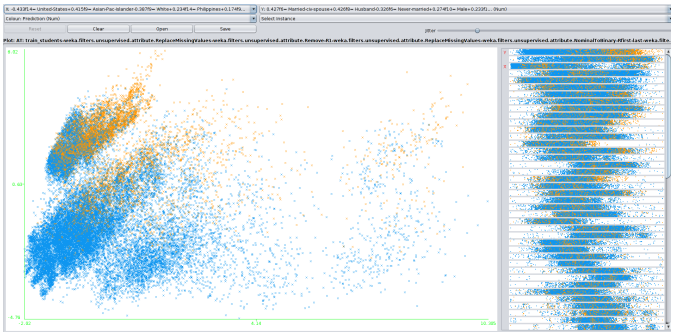

Figure 6.6: Possible clustering visible through principal component plots. Here the class is very much segregated among the clusters.

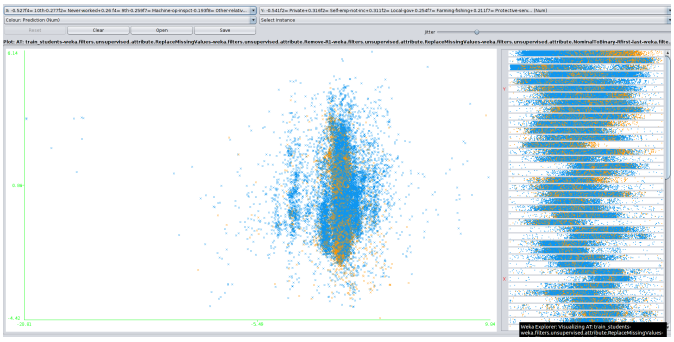

Figure 6.7: An example where sparse long clusters are visible but classes not separated.

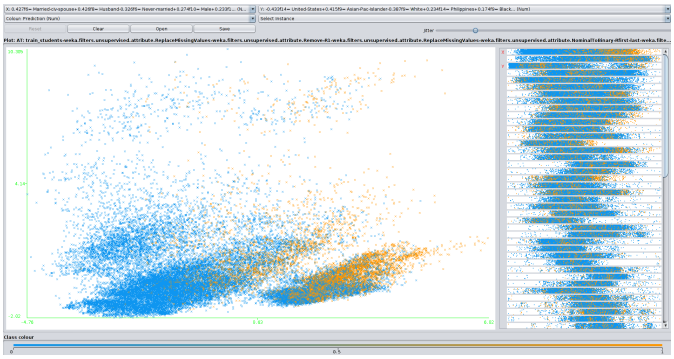

Figure 6.8: A two dimensional plot of PC1 and PC3 showing colors as classes.

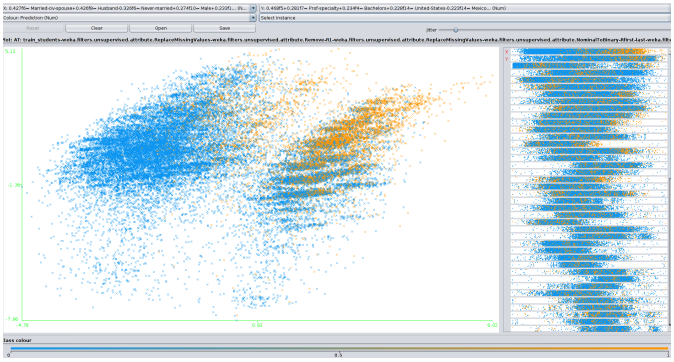

Figure 6.9: A two dimensional plot of PC1 and PC2 showing colors as classes.

6.3 RCA

The plots from RCA are either stretched along the x-axis or the y axis. One can see parallel clusters which in most cases clearly have a majority of a particular class. It seems like in this case, random projection is working better than Independent component analysis to separate out the data belonging to different classes.

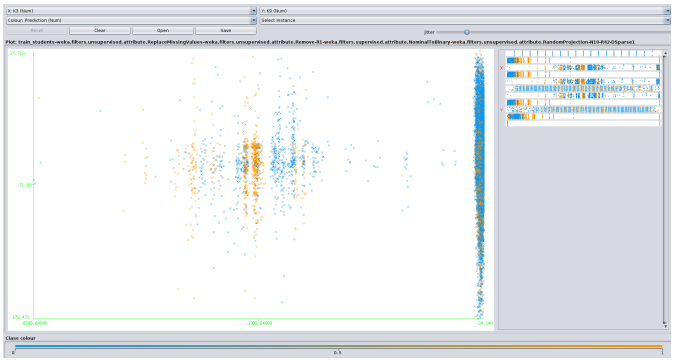

Figure 6.10: Vertical and horizontal clusters visible in RCA plots.

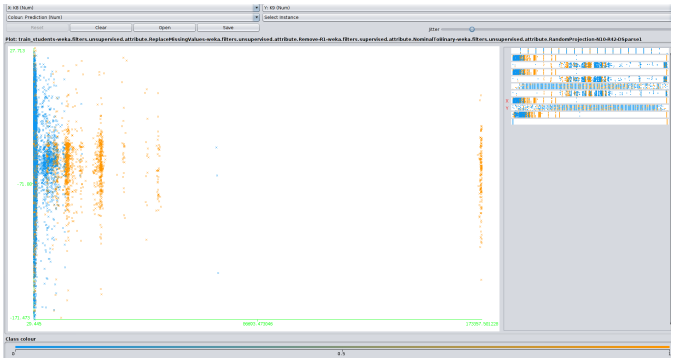

Figure 6.11: Another example of RCA data plots. This plot also shows very well separated classes in elongated clusters.

6.4 Effect PCA, ICA and RCA filtering on data distribution

1. **PCA:** All of the nominal attributes in the income dataset were converted to binary and the PCA was performed. Before PCA because of the binary nature of most data, the distribution of data among various attributes was not well behaved. The were separated histograms. After PCA, almost of the data transformed into beautiful almost bell shaped distribution. There were very few multimodal distributions mostly in the components carrying highest variance. As you go down the rank the distributions start to develop *kurtosis* specially *leptokurtic*. There is more or less no skew in the distribution. Refer figure 2:

2. **ICA:** The independent component analysis performed on the income dataset results in a close to non-gaussian distribution of projected components. The distribution of compponents after ICA is applied has high kurtosis, i.e.,*leptokurtic*. This is to be expected because ICA tends to find the axes of maximum non-gaussianity. The distribution of data is tall and sharp with no multi-modal behavior. See figure **??**

3. **RCA:** Randomized projections (RCA) is just a tool to mix the data well such that ap-

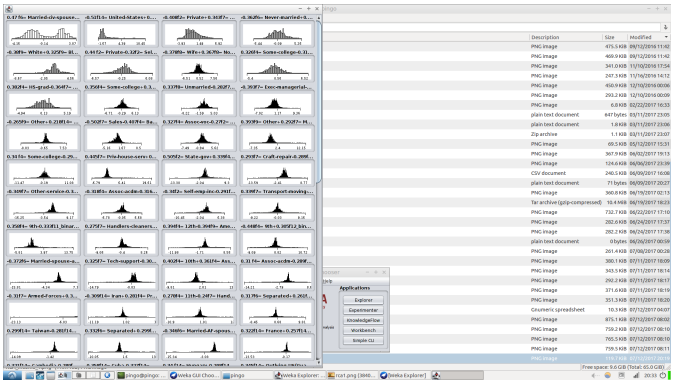

Figure 6.12: Multimodal distributions and kurtosis in low variance components in PCA data.

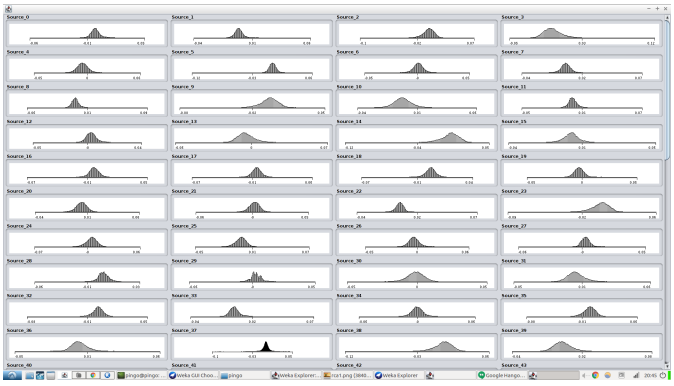

Figure 6.13: Multimodal distributions and kurtosis in RCA data.

preciable variance is not lost if any of the attributes is dropped. RCA is very useful and simple because most of the datasets have some attributes with high variance while others with a small variance. This algorithm just cuts the losses in case of dimensionaity reduction is required. The distribution of components is unlike what we usually see.

7 Clustering on reduced dimensionality datasets

7.1 Income dataset

PCA was performed on the reduced income dataset. We chose a much smaller number of attributes after PCA and went down from 103 attributes to 24 attributes (principal components). I would have expected that this would change the k-menas clustering. However, after performing the elbow method to find the optimum number of cluster, I still found that $k = 10$, see figure 7.1. The sum of squared error for ten clusters after dimensionality reduction and before PCA (full data) are:

$$J_{PCA} = 3274.289, J_{Fulldata} = 66057.375.$$

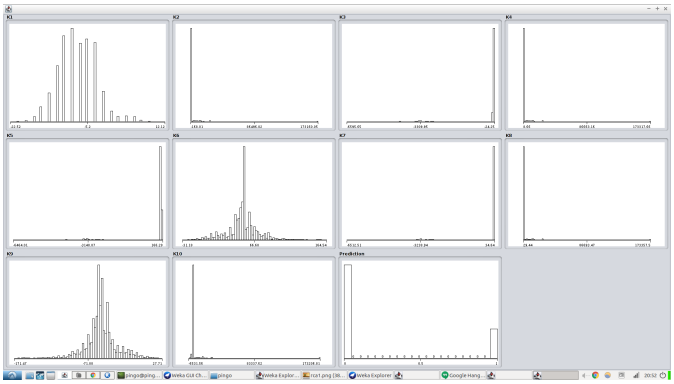

Figure 6.14: Attribute distribution after randomized projections (RCA)

This is a tremendous gain in the compactness of a cluster. I conclude, therefore, that reducing dimensions using PCA perhaps removes away some of the extra distances between points inside each cluster. This can be attributed to the reduced number of dimensions and also projection of distances along axes that minimize intra cluster distances. Dimensionality reduction using PCA was, therefore, very effective against clustering of the income dataset.

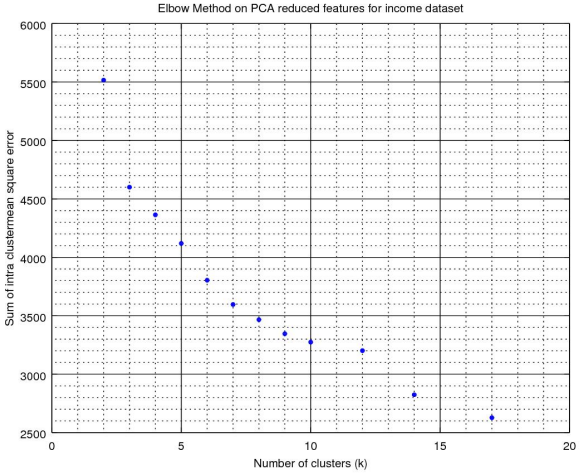

Figure 7.1: Elbow method for k-means clustering on the PCA reduced income dataset. Still gives $k = 10$.

Clustering on the income dataset reduced by RCA was performed by k-means for a number of k-values. See the elbow plot in figure 7.1. It shows the elbow method used on post RCA filtered income dataset. It is very clear that the optimal number of clusters is $k = 6$ this time. The sum of square errors for the new $k = 6$ clusters is 550.289. For $k = 10$ the sum of error squared turns out to be 394.84 which is further smaller than the one we got after applying PCA on the income dataset.

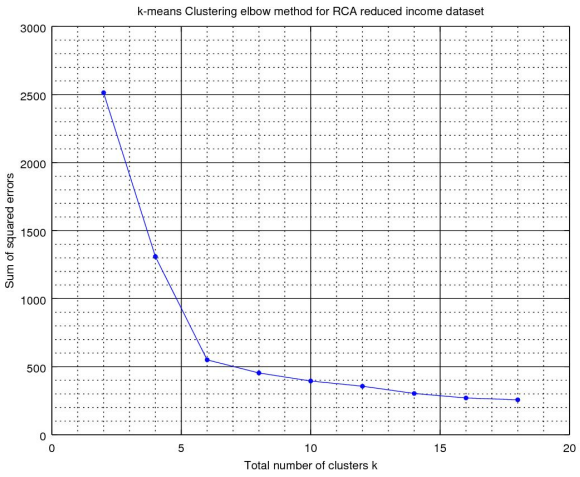

Figure 7.2: Elbow method for k-means clustering on the RCA reduced income dataset. This time the elbow position is at $k = 6$.

Finally, I used the k-means clustering on the data reduced by the Independent component analysis. This time the clustering was still better than that on the original data but the winner of this competition was RCA followed by PCA. Both these algorithms just made the clusters tighter and well defined. **But why is it so?** This can be attributed to my hypothesis that perhaps the dataset was gaussian in the first place and thus there was not much non-gaussian behavior thta can be extracted from the data. Hence, clustering did not perform well as compared to PCA and RCA.

7.2 DIGITS DATASET

As expected from the previous example for income dataset, the PCA made the clustering tremendously faster and with a much lower sum of squared error:

$$J_{PCA} = 877.828, J_{digitfull} = 63194.99.$$

. Again PCA has proven to be a good tool to perform better clustering of data. The same reason still applies here. Dimensionality reduction by PCA reduces distances inside clusters and removes a lot of extra dimensions that results in more compact clusters.

8 TRAINING A NEURAL NETWORK ON DIGIT DATASET AFTER DIMENSIONALITY REDUCTION

The results for the neural network on the reduced digit dataset are summarized below:

Kernel	# attributes	10-fold Cross Validation Accuracy	runtime
Original Semeion dataset	256	92.66%	230.1 seconds
PCA reduced Semeion dataset	30	91.34%	6.85 seconds

The prediction accuracy of the neural net build on the original data as compared to the PCA reduced dimension data have almost the same accuracy (~ 92%) given other conditions are the same. Furthermore, the time taken by the reduced data to build a neural network is roughly 33 times more on the original dataset. Since ANN are very resource intensive to build, PCA is a very good method to filter out data and save orders of magnitudes of time and resources. I did not see any performance improvement in ANNs after dimensionality reduction using PCA, although it took much faster.

9 USING CLUSTERS AS ATTRIBUTES FOR THE DIGIT DATASET

9.1 K-MEANS CLUSTERS AS ATTRIBUTES TO PCA REDUCED DIGIT DATASET

As a reminder, we used PCA on the digits dataset and then chose top 30 components as a dimensionally reduced dataset. We also found out that optimum number of clusters for original dataset is $k = 15$. Now, due to shortage of time, I am not going to deduce the optimum k using the elbow method. Instead I am directly using $k = 15$ clusters here on the PCA reduced digit dataset. The ANNs applied on the dataset with 15 clusters added as attributes improves the efficiency of the ANN. It took much less time (68 secons) to train the ANN with an accuracy of 93% which was not seen anywhere before in this assignment. Thus clustering makes Neural networks worth using because it sigficantly reduces the time required to train the neural net with slight improvement in performance.

9.2 K-MEANS CLUSTERS AS ATTRIBUTES TO RANDOMIZED PROJECTION REDUCED DIGIT DATASET

The randomized projection filter reduces the entire set of attributes to to 10 attributes from K1 to K10 and an additional class attribute. Performing k-means with, again $k = 15$ clusters and then use these clusters as attributes takes a small amount of time to run. However, the performance of the ANN on this dataset was terrible with an accuracy of only 43%. In hindsight, it seems like I should have expected this as the randomized projection although reduces the variance but it also corrupts by its influence on the ANN model. It imposes a relationship between datapoints which are geometrically close to each other. Geometry is good to cluster data into types but in this case takes us away from real predictions.

9.3 K-MEANS CLUSTERS AS ATTRIBUTES TO INDEPENDENT COMPONENT ANALYSIS REDUCED DIGIT DATASET

I wanted to try the the same procedure with Independent Component Analysis but ICA filter does not work with my dataset quite well. It always tends to delete the class attribute which is required in the end for classification. I added the cluster attribute using k-means and trained

the ANN but as there is no class attribute, there is nothing to compare to and get the cross-validation scores.

10 CONCLUSION

In case of very large (huuuge) datasets, it is almost impossible to use any kind of classification algorithm directly because of noise, complexity, time and resources. In all such situations clustering and dimensional reduction of the data is a very strong tool to be used in different combinations depending on our requirement to get some sense and prediction out of the data. Thank you Karl for your gesture.