Assignment 3

Sai Chen

**Section 1. Introduction**

In this analysis, I applied four different dimension reduction / feature selection methods and two unsupervised clustering methods to two datasets. Below I will describe and compare the results from those analysis. I will also compare the relative performance of those methods when used as feature selection algorithms for neural network using one of the datasets.

***Section 1.1. Datasets***

The two datasets used in this assignment are from assignment 1.

*Section 1.1.1. Digits dataset*

In this dataset, 1797 images of hand-written digits in the range of 0-9 are provided with their matching labels. The size of each image is 8 by 8, with each pixel ranging from 0 to 16. There are ~180 samples for each digit. This dataset can be loaded by sklearn load\_digitsfunction.

*Section 1.1.2. Breast cancer dataset*

There are 569 samples in this dataset. Each sample is a collection of 30 medical attributes computed from a digitized image of a breast mass FNA sample. The diagnostic conclusion is also given for each sample, namely whether the patient has breast cancer or not. This dataset can be loaded by sklearn load\_breast\_cancerfunction.

***Section 1.2. Methods***

*Section 1.2.1. Clustering methods*

I used two clustering methods in this assignment: K-means and Gaussian Mixture model (computation is based on Expectation Maximization). The Euclidean distance is used for all analysis below, because this is the only option in sklearn library. The Euclidean distance is a reasonable choice for K-means method, since K-means algorithm looks for new centers by averaging over the points in the matching clusters. This approach is consistent with a loss function using L2 norm. Whether other distance metrics are better for Gaussian Mixture algorithm cannot be explored due to the limitation of sklearn library.

The best number of clusters (namely, K) for each method is determined by several evaluating metrics: elbow method, Silhouette method and BIC method.

To use the elbow method, I plotted the sum of square distances of each point to its assigned cluster center against the number of clusters. It is obvious that the total distance decreases when more clusters are added to the algorithm. However, there might exist some point at which the rate of the decrease slows down, which is the elbow point on the plot. Then I pick the corresponding number as the number of clusters to be used. This method is not very reliable, so I used it as supporting information.

In the Silhouette method, a silhouette score is calculated for each sample. We define the distance of a point towards a cluster as its average distance to all points in that cluster. For each sample, Let A be the distance to its own cluster, and B be the minimal distance towards all other clusters. The silhouette score is calculated as (B - A) / max (A, B). From this formula, it is clear that the silhouette score is close to 1 when the point is much further away from other clusters than its own cluster. In practice, I calculated the average silhouette score over all samples for each number of clusters and pick the one that has the largest average score.

The Bayesian Information Criterion method is only used in Gaussian Mixture models. BIC controls model complexity by adding a penalty term based on the number of parameters used in the model, which counterbalances the increase in data likelihood caused by more complicated models. The BIC scores are calculated using the bic function in GaussianMixture estimator in the sklearn library.

For K-means analysis, I used elbow method and Silhouette method to find the best number of clusters. For Gaussian Mixture model, I used Silhouette method and BIC method. When different methods give different results, I gave Silhouette method higher priority, and also took into account the number of class labels in the original dataset.

*Section 1.2.2. Dimension reduction methods*

Three feature transformation methods are used in this assignment: Principle Component Analysis (PCA), Independent Component Analysis (ICA), and Random Projection (RP).

For PCA, the number of components was chosen so that altogether they can explain over 90% of the variances in the dataset. For ICA, I first decomposed the raw data into (#features - 1) source components, and then used a feature selection algorithm (mentioned below) to pick out the set of components for further analysis. For RP, I did not find a consistent way to choose the best number of components. With the rationale that RP can have as good performance as PCA with the same number of clusters, I used the same number from PCA for all analysis in RP. I used a two-level random forest classifier to pick the best random seed used for downstream analysis.

*Section 1.2.3. Feature selection method*

I used Recursive Feature Elimination with Cross-Validated data (RFECV) as my feature selection method. The idea of RFECV is to look for subset of features that has the best performance when given to an external estimator for prediction. The external estimator can produce the importance of each feature during training, and RFECV can remove the least important feature step by step until the performance on CV-based classification reaches the peak. The nice property of RFECV is that there is no need to specify the number of features to be kept: the cross-validation procedure can find the optimal number for you. In this analysis, I used a 100-tree 2-level random forest estimator as the external estimator and 5-fold cross validation.

*Section 1.2.4. Data processing*

In all analysis, the raw features are standardized so that each feature has sample mean of zero and variance of one.

*Section 1.2.5. Matching generated cluster labels with raw class labels*

Since the clustering algorithm generates arbitrary cluster labels, it is difficult to match them with the original class labels. This is especially true when I used different numbers of clusters than that of the original dataset. To match the two sets of labels, I counted the number of samples falling into each class (from the original dataset) within each generated cluster and relabel the cluster by the class label with the highest number of samples. In this way, I minimize the classification error over the whole dataset. In the figures below, the centers predicted from clustering methods are color-coded based on this label matching strategy. In another word, algorithm-generated cluster centers are colored based on their matching class labels from the original dataset.

**Section 2. Results**

***Section 2.1. Digits dataset***

*Section 2.1.1. Raw data, no transformation or selection*

First, I used K-means on the raw data for clustering. As can be seen from the figure below, both elbow method and silhouette method indicate that a large K (K = 22) is appropriate for the data (much larger than 10). In the bottom figure, the silhouette score of each sample within each cluster is sorted and plotted, and different clusters are colored differently. The existence of negative silhouette scores for several clusters in all subplots in the silhouette figure suggests that the data is not quite separable using the raw data. In the top right panel, the raw data is projected onto a 2-dimensional space by PCA and colored by class label from the original dataset. The large diamonds are the predicted centers by K-means algorithm. Each cluster is given a class label based on the matching strategy discussed in the result section. Thus, some samples might have the wrong label if they are not in the class that has the largest number of samples in a particular cluster, similar to the classification errors we encountered in the classification problems. I labeled these points using a ‘x’ in the plot. It is clear that digits 4, 7, 2 can be well separated in most cases, while it is almost impossible to separate 3, 8, 9 using an 8x8 image. It is also interesting to see that sometimes 0 and 6 are similar to the algorithm, which makes sense if the hand-written patterns are not regular.

|  |  |
| --- | --- |
|  |  |
|  | |

Since there is no clear elbow point in the elbow plot for most the analysis, I decided to use silhouette method to pick the number of clusters for K-means in all the following analysis.

Figure below shows the result from GM method. It gives similar conclusion as the K-means algorithm. In the top left panel, I used the BIC method to pick the number of the clusters. Besides choosing K, it is also important to determine the type of covariance matrix used in the GM model. For each K I also used all four possible covariance structures to compute the BIC score. In this case, the BIC method picks 28 clusters with diagonal covariance matrix for the GM model. In the right panel, I plotted the 95% confidence ellipse for the GM model projected onto 2-dimentional space by PCA. We can see that except for having more clusters, the result is similar to what we got in the K-means algorithm. To save space below, the result for selecting number of clusters will not be shown and all the results can be found in the uploaded folder.

|  |  |
| --- | --- |
|  |  |
|  |

*Section 2.1.2. PCA*

Figure below shows the result for PCA. It requires >30 components to cover >90% of variances in the dataset. So, I used 30 components for the downstream analysis. The eigenvalues are also plotted below. As we can see from the left panel, eigenvalues for the later components are close to zero, meaning that variances along those dimensions are small. It is thus reasonable to throw away those components for downstream analysis, because no information could be extracted if no variation exists for a particular feature.

|  |  |  |
| --- | --- | --- |
|  |  |  |

The Silhouette method selects K = 24 for K-means and K = 9 for GM. In theory, both K-means and GM methods should generate the same results for raw data and PCA-transformed data with all components retained. This is because PCA is only a rotation operation, so that the geometric structure of the dataset does not change when all components are kept. In fact, when I kept all components from PCA and used the same number of clusters in both clustering algorithms, the results are exactly the same for raw data and PCA data.

In practice, PCA is used as a dimension reduction algorithm. However, since later components contribute very little to the total variance, they also contribute little to relative distances among sample points. Due to the inseparable property of the digit dataset, the Silhouette method gives different numbers of clusters. Yet we can still see that when the number of clusters are close (in K-means), the centers are similar between raw data and PCA data.

*Section 2.1.3. ICA*

Figure below shows the results for ICA. ICA looks for components that are independent and non-Gaussianity. The implementation used in sklearn is to maximize neg-entropy of the resulting components.

|  |  |
| --- | --- |
|  |  |
|  |  |

Kurtosis is commonly used as a measure of non-Gaussianity. Here I calculated the kurtosis so that a standard normal distribution has value zero. We can see that there are ~10 components that have very large kurtosis. In the top right panel, I plotted the data projected onto the two components with the largest two kurtoses. We can see that the first component has a large kurtosis because there is one ‘outlier’ in the dataset. This suggests that kurtosis is highly sensitive to outliers, and thus could not be used as standard for choosing optimal number of components. This is why I used the RFECV method discussed above.

In the two bottom panels, I plotted the data projected onto two randomly chosen components after feature selection. Silhouette method chooses K = 10 for K-means and K = 20 for GM. It seems like the first components picks out number 6 and second picks out number 4. Due to the independence nature among components, other digits that are not distinguishable by these two components are clustered together. Overall, ICA does not perform well on this dataset, which might be due to the inseparable nature for some digits.

*Section 2.1.4. Random Projection*

Figures below shows the results for random projection method. As discussed above, I used a two-level random forest classifier with 100 trees to choose the best random seed used for downstream analysis. As can be seen from left panel, the distribution of prediction accuracy using different random seeds is quite wide, suggesting that there is a descent amount of variation across each run of random projection.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Silhouette method chooses K = 13 for K-means and K = 7 for GM. I plotted data projected onto the first two components generated from the selected random projection run. From the figures, we can see that even though the components are generated randomly, there are still good separation of different classes. Similar to results from using raw or PCA transformed data, digits 4, 7, 2, and 6 can be easily picked out.

*Section 2.1.5. RFECV*

Figures below show the results for feature selection method. Silhouette method chooses K = 12 for both clustering methods. I plotted the data projected onto the first two components from a PCA transformation after feature selection. Not surprisingly, the results are similar to what we got by using raw data or PCA transformed data, with maybe a slightly better separation in the middle of figure where it is difficult to separate the sample points.

|  |  |
| --- | --- |
|  |  |

The above results suggest that it is beneficial to select features for clustering methods based on the predictive power of each feature relative to the class labels. However, in practice it is almost impossible to know the cluster labels beforehand. Thus, it might be more important to find other metrics irrelevant to classifications as standard for feature selection.