Unsupervised Learning and Dimensionality Reduction

*Assignment 3*

Er Li Suo  
903458480  
Nov. 3, 2018  
[esuo3@gatech.edu](mailto:erlisuo@gatech.edu)

**DESCRIPTION**

The assignment explores the unsupervised learning algorithms, clustering and dimensionality reduction algorithm by analyzing the results generated by applying the techniques to 2 datasets, the energy efficiency and seeds. The clustering methods applied are k-means clustering (kmeans) and Expectation Maximization clustering (em). The dimensionality methods used are Principle Component Analysis (PCA), Independent Component Analysis(ICA), Gaussian Random Projection (GRP) and Linear Discriminate Analysis (LDA). The 5 tasks listed in assignment description are analyzed in part 1 to part 5.

**DATA**

Energy Efficiency for Heat Load and Seeds Classification datasets are used in this assignment. Energy Efficiency has been used for assignment 1 and 2. Energy Efficiency dataset collects 8 features (relative compactness, roof area, overall height, orientation, glazing area and glazing area distribution) to measure the heating load of a house. For the classification and clustering validation purpose, the label for of dataset have been preprocessed to form 5 categories (low, med low, med, med high, high). For seeds classification dataset, the sample seed is classified with a soft X-ray technique with the measurements of 7 features (area, perimeter, compactness, length of kernel, width of kernal, asymmetry coefficient and length of kernel groove) into 3 labels, Kama, Rosa and Canadian. A total of 210 sample collected and 70 instances in each category.

**PART 1: Apply Clustering**

*K-means clustering*

For the K-means clustering, Euclidean distance is used to guarantee converging of results. The best estimated number of K is calculated and compared the within-cluster sums of squares (WCSS). The smaller the WCSS indicates the fewer outlier instances within a cluster. The ‘k-means ++’ initialization method is used to avoid the random initialization trap.

Energy Efficiency Dataset

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

In the graph on the left side, WCSS score for 1 to 30 clustering has been calculated. The best K is around 7. To visualize how the clustering has been applied, the above graph at the right shows the scatter points for the first two features, relative compactness and surface area in the Energy Efficiency dataset. We can see the instances of data are clustered based on their distance. Due to there are multiple points has the same value in the original dataset, many points are overlapping each other on the graph.

Seeds Dataset

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

Similar to the previous dataset, the WCSS scores are calculated based on 1 to 30 clusters. K = 3 provides largest fall on the WCSS score. Since the original dataset also has 3 labels, the dataset is divided into training set (80%) and testing set (20%). To visualize how the method performs, all the training instances are marked as little dots ‘.’ and all the test instances are marked as ‘X’. 6 features of the seeds are displayed as 2-D graph above. It is easily visualized that the majority of the testing instances could be clustering correctly. We can also observe many points that visually closer to one cluster have actually been placed in a different cluster. This could be caused by the distance of these points are closer to the center from the perspective of a different dimension.

*Expectation Maximization Clustering*

The BayesianGaussianMixture model from sklearn is used for this part of the assignment. The ‘tied’ covariance\_type (all components share the same general covariance matrix) is configured. All the models are converged within 10 iterations. The number of components are the same as k-means.

Energy Efficiency Dataset

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

Similar to K-means, the graphic representation of the clustering results is somewhat difficult to demonstrate as many instances in the dataset have the identical values for some of the features. The instance points are all overlapping each other perfectly. However, we can still roughly tell the closer the points, the bigger the tendency for the points to be clustered together.

Seeds Dataset

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

The clustering results for the 6 features are shown above. Since EM is a soft clustering algorithm, some outlier points, ie, the blue points appear in a group of red points, which are visually belong to certain cluster can actually have a bigger probability of being placed in a different cluster.

**PART 2. Apply Dimensionality Reduction**

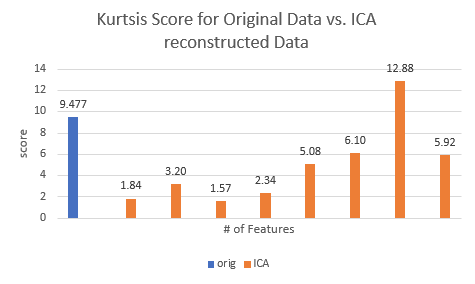
Dimensionality reduction algorithm generally transform a dataset to a new dataset with fewer dimensions/features to delight the computing burden with big datasets. PCA, ICA, GRP and LDA are analyzed below for 2 datasets.

Energy Efficiency Dataset

* PCA

|  |  |
| --- | --- |
|  | The table on the left listed the weight of each feature in the overall feature sets by comparing their orthogonal eigenvectors to maximize the amount of variance. For this particular dataset, we can observe the first 5 feature counts the majority of the weights. Such, n\_components = 5 has been used. The transformed datasets are used in part 3 to 5 for further analysis. |

* ICA



Kurtsis value is a measurement of ‘tailedness’ of a distribution. The reconstructed dataset is expected to have a Kurtsis values as large as possible, which represents the independence between each attribute. To find out what is the best number of dimension/feature, every kurtsis values from 1 to 8 features are calculated and displayed in the above graph. Since individual attribute could have negative Kurtsis values, all the values are taken as absolute values and are sum up. We can observe that the when n\_component = 7, we scored 12.88 which is higher than the score of the original dataset with the value of 9.477.

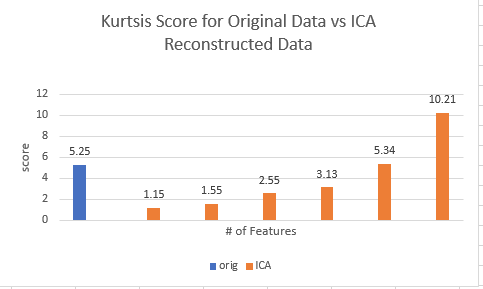
* GRP
* LDA

Seeds Dataset

* PCA

|  |  |
| --- | --- |
|  | The table on the left listed the weight of each feature in the overall feature sets by comparing their orthogonal eigenvectors to maximize the amount of variance. For the seeds dataset, we can observe the first 3 feature counts the majority of the weights. Such, n\_components = 3 has been used. The transformed datasets are used in part 3 for further analysis. |

* ICA



Similar to the previous dataset, to find out what is the best number of dimensions or features, every Kurtsis values from 1 to 6 features are calculated and displayed in the above graph. All the values are taken as absolute values and are sum up as well. We can observe that the when n\_component = 6, which scored 10.21 and it is higher than the score of the original dataset with the value of 5.25.

**PART 3: Cluster After Dimensionality Reduction**

Energy Efficiency Dataset

* PCA and EM

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

Above graphs demonstrate the clustering of EM after applying dimensionality reduction. We can still see that many instances are overlapping; however, it is clear the clustering pattern is more observable after the dimensionality reduction compared with the clustering graphs before applying dimensionality reduction.

* PCA and K-means

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

The graph generated with K-means clustering after PCA is more interesting. The clusters are NOT easily to be visualized. Taken the cluster with red dots as an example, even though they are not nearby each other on the displayed plane, they all follow a pattern, which could be argued as they are actually share similar properties in a different dimension, in this case, distance, which is the only measurement we consider. The graph should be able to improve by choosing different combination of the features.

Seeds Dataset

* PCA and EM

|  |  |
| --- | --- |
|  | Re-run the EM with the dataset after applying PCA. On the graph at left, the clustering results are much better. Less outliers and the borders between each cluster are more distinguished. |

* PCA and K-means

|  |  |
| --- | --- |
|  | Similar to the EM results, re-run K-means with the dataset after applying PCA also gives better clustering results. On the graph at left, it is easy to see there are less misclustered instances. Similar to the previous K-means graph, the test instances are marked as ‘x’ and training instances are marked as dot. |

**PART 4: Neural Network Classification after Dimensionality Reduction**

The whole purpose of dimensionality of reduction is to increase the prediction rate and at the same time, decrease the computing cost to deal with the curse of dimensionality. The wrapper, dimensionality reduction + classification algorithm, are applied here to examine the prediction ability.

Energy Efficiency Dataset

Without Applying dimensionality reduction, the performance of neural network (NN) predication is as below:

Training Accuracy = 90.21% Testing Accuracy = 86.36%

|  |  |
| --- | --- |
| * PCA | *Average Execution Time: 4.98 s* |
|  | The NN classification algorithm has been applied with different number of attributes, from 1 to 8. From the PCA analysis in the previous section, we know that the features offer the most weights are the top 5 ones. The same results can be concluded from the graph on the left. The predication accuracy is the best at n\_components = 5 and 7. Compared to NN without dimensionality reduction, the prediction rate is better. |
| * ICA | *Average Execution Time: 10.51 s* |
|  | Even though n\_component = 7 gives the best Kurtsis value when doing the ICA analysis, we can see the predication accuracy is not the best when n\_component = 7. 7 components give the second-best predication rate. However, regardless which n\_components we reconstructed, the predicating accuracy is worse than without applying ICA. One explanation could be that all the attributes in this particular dataset are already independent from each other (we can tell from its Kurtsis value). Reconstructing the dataset only increases the noise. |
| * GRP | *Average Execution Time: 5.11 s* |
|  | After applying random projection, the best predication accuracy is given at n\_component = 7 and it is improved compared to the NN without dimension reduction. In terms of speed, GRP is much faster than ICA and is similar to PCA and LDA. |
| * LDA | *Average Execution Time: 4.98 s* |
|  | LDA gives the same predication accuracy for n\_components greater than 3. Even though not significant, it is slightly better than the performance without applying dimensionality reduction. LDA with n\_components = 4 also reduces the dimension the most. |

**PART 5: Apply Neural Network after Combine the Cluster as New Feature**

Scale data between –1 to 1.   
apply pca with n\_feature = 5 (found from previous test)

Run cluster with n\_cluster = 5 (5 categories)  
get cluster labels, append to dr train data.   
rescale the data between –1 to 1  
apply NN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| k-means |  |  |  |  |  |
| alg | n\_c | cluster | time | train | test |
| PCA | 5 | 5 | 3.857 | 92.82% | 89.61% |
| ICA | 5 | 5 | 5.322 | 94.45% | 90.90% |
| GRP | 7 | 5 | 5.036 | 96.24% | 91.56% |
| LDA | 4 | 5 | 5.191 | 96.90% | 93.50% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| em |  |  |  |  |  |
| alg | n\_c | cluster | time | train | test |
| PCA | 5 | 5 | 3.814 | 92.82% | 89.61% |
| ICA | 5 | 5 | 5.462 | 94.45% | 90.90% |
| GRP | 7 | 5 | 5.48423 | 93.31% | 89.61% |
| LDA | 4 | 5 | 5.728 | 96.90% | 93.51% |