Ashwin Rathie
CS 4641 – Machine Learning
Professor Byron Boots

Unsupervised Learning Analysis

Datasets:

Crime Dataset – This dataset describes crimes recorded in London by using features like zip code, borough, year, and month and describes the crimes in terms of different categories, or classifications. There are over 20,000 examples in this dataset.
Restaurant Dataset – This dataset includes information about restaurants. It includes 18 features, including location, price range, franchise or single, and handicap accessibility. In this analysis, handicap accessibility is the target feature.
Housing Dataset – This dataset is used in experiments 11, 12, and 13 and comes from assignment 1. It had over 20,000 data entries and includes features like square footage of the house, number of bedrooms and bathrooms, lot size, year built, size of the basement, and condition of the house. The target feature is sale price.

1.0– Clustering:

Clustering is, essentially, grouping data into group based on their graphical proximity. The goal is that each data point in one cluster is more similar to the other data points in its cluster than it is to the data points in other clusters. Its application to unsupervised learning is in its ability to create clusters than represent the data points that all share the target feature of that cluster. In our analysis, Euclidean distance is used to calculate the proximity of data points.

The clustering algorithms we will be exploring in this analysis are k-means and the Gaussian Mixture Model via Expected Maximization (henceforth referred to as “EM”).

K-means Clustering:

The k-means algorithm works by first initializing k number of arbitrary, random points to be the cluster centers. Then, each data point is assigned to one cluster based on its proximity to the cluster centers. The second step is to move the cluster centers to the “middle” of the data points assigned to it in the previous step. The middle is determined by the minimized sum of squares distance. This process is repeated until no data points change clusters through iterations (k-means clustering does converge).

EM Clustering:

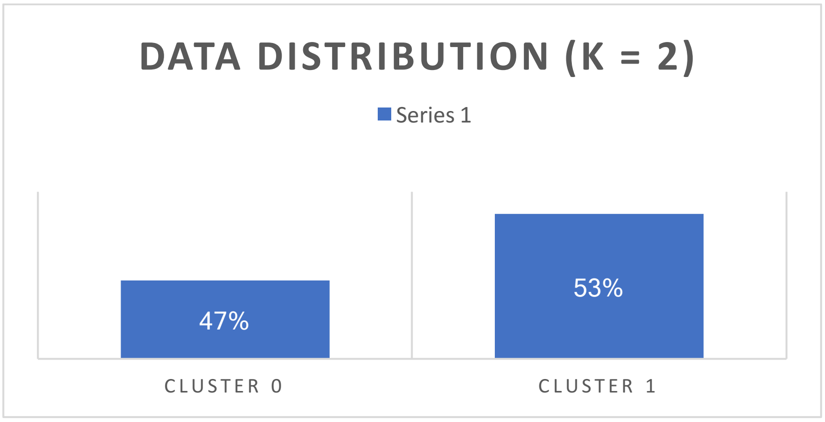
Expected maximization is similar to k-mean in the general steps it uses – initialize random centers, assign the data points, move the cluster centers, repeat – but assigns the data points using probabilities. The Gaussian mixture model is used to determine a probability that a point belongs to a certain cluster center and if that probability is within the predefined range, it is assigned to that cluster. EM is a soft clustering algorithms, which mean the clusters do not have absolute edges. It is possible for there to be data points that are not assigned to any cluster because the probability of belonging to any one cluster is not great enough the cross the threshold of actually being assigned to that cluster. The "distance" metric for this analysis is a logarithmic function of the probability.

In this analysis, sum of squared error (SSE) is used to find the silhouette score, which is a measure of the performance of the clustering algorithms and the optimal number of clusters is chosen manually by observing this score. A higher silhouette score indicates a lower SSE and is therefore better.

1.1 – Experiment 1: K-Means used on Restaurants Dataset

K NUMBER OF CLUSTERS	SILHOUETTE SCORE	TIME TAKEN TO RUN (SEC)
2	.09177	.1287
3	.0873	.1447
4	.0815	.1489
6	.0678	.1980
8	.0612	.2217
12	.0412	.2538

I tested K cluster values from 2 to 12 to choose the optimal number of clusters, then use this number to proceed with the silhouette analysis. As a result, 2 clusters were found to be the best grouping as it had the greatest silhouette score. The clustering into two groups on the classification problem stands to reason, as the problem is one of binary classification. The fact that this data has been previously used to create a binary classification with high accuracy makes sense given that it is easily and best separated into two clusters. The highest performing model, 2 clusters, also took the least amount of time to compute at .1287 seconds.



The above figure is the distribution of the data after k means was run with k = 2, which was found to be optimal. The cluster that is meant to group the data of locations that are not handicap accessible had 47% of the data in it, and the other had 53%. We know that the original dataset was 50% to 50%, so this clustering is still rather accurate. The discrepancy may be in the fact that the data was initially largely qualitative, but was transformed into quantitative binary data

1.2 – Experiment 2: K-Means used on Crime Dataset

K NUMBER OF CLUSTERS	SILHOUETTE SCORE	TIME TAKEN TO RUN (SEC)
2	.0481	7.9121
4	.1291	16.1311
6	.1092	35.2324
8	.0482	44.8283

The results of k-means on the crime data shows low silhouette scores, just as the results of experiment 1 showed. At 20,000 points, running this was very computationally intense, taking over ten minutes for 8 clusters (maybe PCA will help…), so I randomly selected 1,000 of the data points and re-ran the experiment. The resulting scores were generally similar to the full dataset scores. Using the silhouette method, k = 4 was selected as the optimal clusters.

1.3 – Experiment 3: EM used on Restaurants Dataset

# OF CLUSTERS	PROBABILITY (LOG)	TIME TAKEN (SEC)
2	4572.1344	1.29
4	4711.1939	2.44
6	4803.9642	4.12

8	4889.0321	6.22

As EM can sometimes have difficulty converging, this data did not seem to converge. The results show that as the number of clusters grows, the performance continues to increase. However, this could be because of the relatively low number of example in this dataset. At under 150 examples, 8 clusters are nearly within the same order of magnitude as the total number of data, and an increased number of clusters would naturally make sense as clusters can be more specific, so the algorithm may be overfitting in a sense.

1.4 – Experiment 4: EM used on Crime Dataset

# OF CLUSTERS	PROBABILITY (LOG)	TIME (SEC)
2	26819.87	87
4	27998.12	176
6	28574.82	302
8	28732.28	629

The results of experiment 4 do no align with those form experiment 2, when k-means was used on this dataset. K-means resulted in an ideal number of clusters at 4, whereas EM's score continues to increase even at 8 clusters. However, if the "elbow method", the method that selects the point at which diminishing returns mean a slightly inferior performance at a dramatically lower amount of running time is preferable, is used to determine number of clusters, 4 would be optimal in EM just as it is in k-means.

1.5– EM vs. K-Means Conclusion

Overall, K-means proved to be a better clustering algorithms based off of a lower computation time and lower error rate, as it chose the apparent ideal number of clusters on both datasets rather than just one. While both resulted in an optimal cluster number of 4 for the Crime dataset, they disagreed on the restaurants dataset as EM appeared to continually increase the number of clusters in search of ideal performance. We know that k-means result of 2 clusters is correct because the target feature of handicap accessibility is binary classified. The result of 4 clusters being optimal for the Crime dataset also makes the most intuitive sense because there are 4 different types of crime classifications. EM's failure could be the result of "overfitting" the data in a sense, as the number of clusters was within the same order of magnitude as the number of data examples. Running the tests again with a larger second dataset than the restaurants dataset could yield more information about the two algorithms.

2.0 – Dimensionality Reduction: Applying Principal Component Analysis

Dimensionality reduction is a technique that aims to reduce the number of features in a dataset that will be given to a learning algorithm in the hopes that this will increase the algorithms accuracy and decrease computational complexity and runtime. This proves to be one of the solutions for bypassing the curse of dimensionality.

In this analysis, we will be exploring Principal Component Analysis. PCA a feature transformation algorithm, so we aren't removing any features, but we are altering them so that they project onto a lower dimensional space. The PCA score is based off of the resulting variance, and a lower PCA score is better than a higher one.

2.1– Experiment 5: PCA used on Restaurants Dataset

# OF COMPONENTS	VARIANCE	PCA SCORE	RUNTIME (SEC)
2	1.17, .58	248.2	.33
3	1.17, .58, .51	255.8	.21
4	1.17, .58, .51, .04	312.6	.39
5	1.17, .58, .51, .04, .23	322.2	.32

The data in Figure 9 demonstrate that past 3 components, the variance of the features there is a sharp increase of PCA score– components 3, 4, and 5 are not adding value. 3 components result in a similar score, and because we may prefer to not excessively reduce dimensions, could also be considered optimal over 2 components.

2.2 – Experiment 6: PCA used on Crime Dataset

# OF COMPONENTS	VARIANCE	PCA SCORE	RUNTIME (SEC)
2	.35, .20	8099.6	14.39
3	.35, .20, .17	7956.2	12.44
4	.35, .20, .17, .14	7724.9	10.28
5	.35, .20, .17, .14, .15	7893.4	11.32

The crime dataset appears be optimized by PCA at 4 dimensions due to its low PCA score. However, all of the PCA scores are very, very high, which may devalue the results and point towards there being a lesser benefit from using PCA on this dataset.

2.3 – Applying PCA conclusion

Applying PCA to the restaurants dataset shows that 2 or 3 principal components are optimal for reducing dimensionality. Although it may not have been necessary on this relatively small dataset (150 entries), it will still likely result in a quicker runtime for clustering algorithms. The optimal number of components for the crime dataset appears to be 4. At 20,000 data entries, this dataset previously took prohibitively long when I attempted to run clustering algorithms on it. With dimensionality reduction, we can expect to see this reduced.

3.0 – Clustering after PCA:

3.1– Experiment 7: K-Means used on Restaurants Dataset after PCA

K NUMBER OF CLUSTERS	SILHOUETTE SCORE	TIME TAKEN TO RUN (SEC)
2	.1043	.0985
3	.0871	.1287
4	.0832	.1275
6	.0611	.1489
8	.0605	.1738
12	.0404	.2293

*We will be evaluating the performance of the clustering algorithms using the same metrics as the experiments in section 1. Note that silhouette score is based off of the sum of squared error between the data points and the cluster centers; a higher silhouette score indicates a lower SSE and is therefore better.

The first thing to be noted about these results is that the overall silhouette scores appear to be of a similar distribution and range compared to the pre-dimensionality reduced experiment. This is a good sign, as the goal of dimensionality reduction is to reduce the dimensions of the data while maintaining the data itself.

We expected to see a dramatic decrease in the time taken to run due to decreased complexity and computational load. Although there was a decrease in overall time taken to run, it was trivial and would be inconsequential in practice. This is because the original dataset was about 150 data entries with 18 features, so dimensionality reduction wasn’t really necessary due to the low amount of data.

3.2 – Experiment 8: K-Means used on Crime Dataset after PCA

K NUMBER OF CLUSTERS	SILHOUETTE SCORE	TIME TAKEN TO RUN (SEC)
2	.0429	2.92
4	.1151	5.39
6	.1202	14.72
8	.0388	19.15

Although the overall range of silhouette scores is very similar to experiment 2 (before PCA), the conclusion is slightly different. These results would suggest that 6 clusters would be the optimal k value, whereas 4 was indicated to be in experiment 2. The difference in their respective silhouette scores is relatively small, exposing the main weakness of dimensionality reduction algorithms. These algorithms are exceptional at reducing size of data with minimal cost to the values of the data itself; however, there is still a cost. The change in scores before and after PCA was very slight, but because the scores were already so close, the conclusion was different as a result of running PCA.

3.3 – Experiment 9: EM used on Restaurants Dataset after PCA

# OF CLUSTERS	PROBABILITY (LOG)	TIME TAKEN (SEC)
2	4572.1344	1.29
5	4711.1939	2.44
7	4803.9642	4.12
10	4889.0321	6.22

Like the equivalent experiment in section 1, EM does not appear to converge on this dataset. In experiment 3, we noted that the very low amount of data examples may be enticing the algorithm into “overfitting”, matching individual clusters with less data points. This appears to be happening again as PCA will have done nothing to mitigate the issue. It is further reducing the dimensionality, which wouldn’t combat the overfitting described at all. Although the algorithm did not converge to give us a concrete answer, it is a good outcome that the results between the algorithm before and after PCA are consistent.

3.4 – Experiment 10: EM used on Crime Dataset after PCA

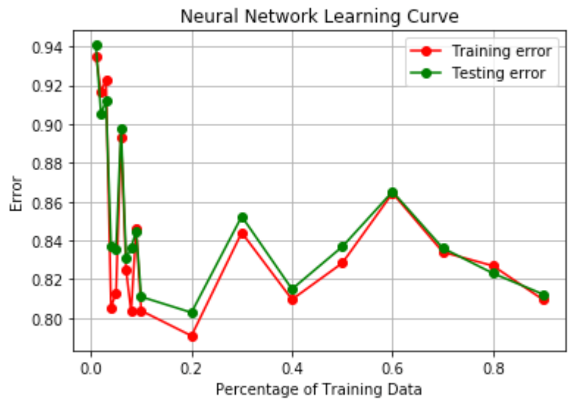
# OF CLUSTERS	PROBABILITY (LOG)	TIME (SEC)
2	26139.48	25
4	27611.02	42

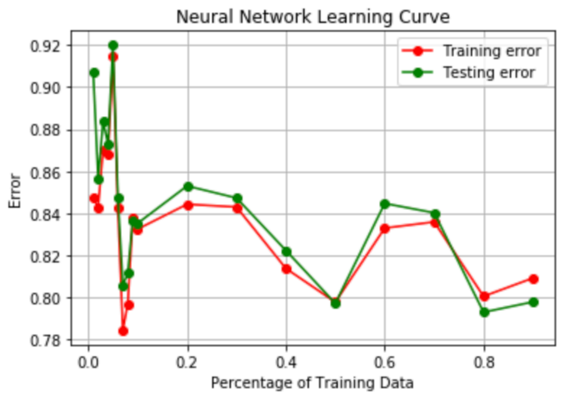
6	28392.78	98
10	28583.49	174

In section 1, when running clustering algorithms before PCA, we realized that running k-means and especially GMM via Expected Maximization was taking a prohibitively long time to run. This forced us to cut the dataset to 1,000 data points, bringing the runtime to a manageable level. The runtime of the EM algorithm on this data after PCA was applied was dramatically reduced as seen above. For consistency, this experiment was run with the same reduced dataset of 1,000 examples, but it now seems feasible to run the clustering algorithm on the full dataset if needed. This highlights the crowning advantage of using PCA. The reduction of time and features allows us to use data to its fullest – i.e. we could use 20,000 data points instead of 1,000 – which is preferable as it gives us the truest conclusion to whichever problem we are modeling.

4.0 – Experiment 11: Applying PCA then training Neural Network

In our first analysis, we looked at the performance of supervised learning algorithms on a "housing prices" dataset. In this analysis, we ran the same neural network learning algorithm after applying PCA to the housing dataset. The pre-PCA performance is shown below (left) and compared to the post PCA results (right).





The pre-PCA NN results can be characterized by volatile then stabilized performance that gave a .83 error rate. Admittedly, this is a very high error rate as we concluded that there was perhaps no strong correlation between the features in the dataset and the price the house was sold for. However, we can use it here as this analysis is about comparing the performance before and after dimensionality reduction.

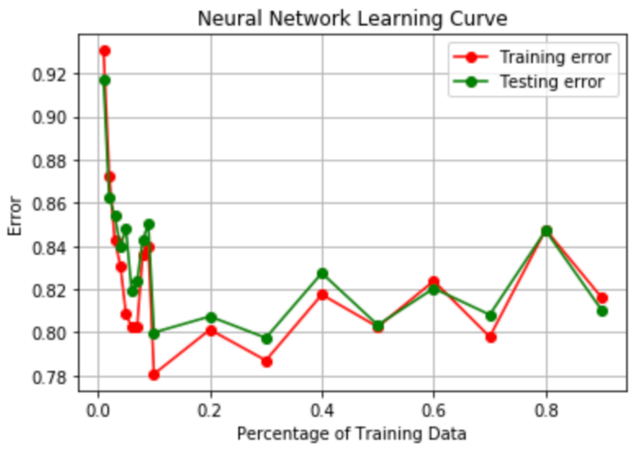
After PCA was applied to the dataset, the results are still volatile at first, then stabilize. The error rate was about .83-.84, which is trivially worse than the results from before PCA. It should be noted that while the error rate is about the same, the runtime was not. In assignment 1, I noted that the NN took the second longest to run, after SVMs. This time, it took noticeably less time on this fairly large dataset (over 20,000 entries).

5.0 – Using Clustering as Dimensionality Reduction Algorithms

5.1 – Experiment 12: Using K-Means as Dimensionality Reduction Algorithm

In 5.1 and 5.2, we will attempt to use clustering algorithms as dimensionality reduction techniques. We will apply these algorithms to the same dataset as experiment 4.0, the housing prices dataset, for consistency.

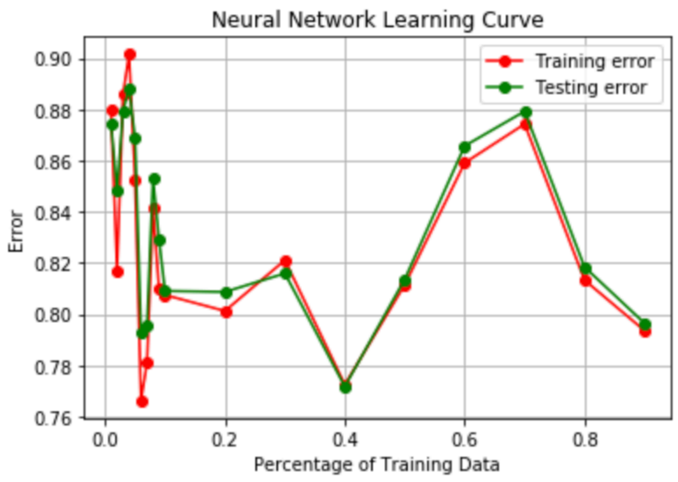
To use clustering as dimensionality reduction, we take our high dimensional feature vectors and apply the clustering algorithm, which in this experiment is k-means. Then we represent the old data points in terms of its distance for each cluster center. The k distances for the k cluster centers represent a new vector in dimension k, thereby reducing the overall features using clustering. We will use k = 10 in order to minimize the risk of oversimplifying data.



The NN performance results on the reduced data is shown on the left. The learner reacted rather well and was stable around .81 error, which is actually even lower than the non-reduced data performed. I believe this is a coincidence for this occurrence only, not that k-mean clustering will always result in great NN performance. Once again, the runtime was less, but not as quick as the NN running on post-PCA data.

5.2 – Experiment 13: Using EM as Dimensionality Reduction Algorithm

In this experiment, we will do the same as we did in experiment 13, instead using GMM via EM on your features rather than k-means. Again, we will use 10 clusters to minimize the risk of oversimplifying data.



The results shown to the left are inconsistent and never really stabilize, with its peak (after the initial jaggedness) at .88 error. It appears that EM was not quite as effective of an DR algorithm as k-means was. Additionally, the NN on the EM reduced data took just marginally less time that the NN takes on the original data. However, a positive is that the results, while never stabilizing, are still within the range of the original test, so using EM on features didn't critically damage the data.

Works Cited:
Datasets:
https://www.kaggle.com/bharathnr/restaurant-and-consumer-data#geoplaces2.csv
https://www.kaggle.com/jboysen/london-crime/version/1

Code Source:
https://github.com/parasj/machine-learning