Ryan Fischbach
Dr. Pauca
CSC375
9/11/2020

Deep Learning Project 1 Writeup

1. Background

Clustering is an unsupervised machine learning technique where data points are compared using a distance metric and grouped into clusters. These groups feature similar data points and can be extremely useful when no labels have been provided. There are many different clustering algorithms, all with their strengths and weaknesses. 5 of the most common are K-Means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Mean-Shift Clustering, Expectation-Maximization (EM) Clustering via Gaussian Mixture Models (GMM), and Hierarchical Clustering. Hierarchical Clustering and K-Means Clustering will be evaluated and tested for this project.

1.1. Hierarchical Clustering

Hierarchical Clustering is a connectivity method using the idea that a distance metric can indicate the relation between samples. In other words, closer samples are more similar than farther samples. Hierarchical clustering houses two different methods in it: agglomerative (bottom-up) or divisive (top-down). In agglomerative clustering, each sample exists in its cluster, and clusters (samples) are merged as the algorithm proceeds. The earlier the samples are merged, the more similar they are. In divisive clustering, all samples start in the same cluster, and samples are split into clusters as the algorithm proceeds. Both types can be represented as a dendrogram or a tree-like graph that represents the clusters that exist within the data as well as their respective similarities (with the closer pairs having links closer to the leaves in Agglomerative). There are two key parameters for hierarchical clustering models: a distance metric and linkage type. The distance metric tells the algorithm how to compute the distance between two samples, with different metrics yielding different results based on different underlying equations. Some examples of distance metrics are Euclidean distance, Manhattan distance, and Cosine. Linkage criteria give the algorithm the metric used to merge/unmerge the samples. Some examples are ward, complete, average, and single. Each dataset is different, so the parameters that will maximize the accuracy of the model (yield the best predictions) will vary and should be investigated.

1.2. K-Means Clustering

K-Means clustering is a centroid-based method, using a fixed number of clusters input initially. The clusters are initially placed randomly by using random samples, and each subsequent sample is placed into the cluster closest to it via a distance metric. Once all samples are assigned to a cluster, the algorithm finds a center location of each of these clusters such that the squared distances from each cluster center to samples are minimized. If a sample is now closer to another cluster center, it will move to that new cluster. This process is repeated until a new cluster center yields no reassignment of samples to new clusters. The number of clusters input into the algorithm is fixed, serving as a drawback for this method. Additionally, initial cluster assignments are random, so each run of the algorithm can produce different results.

2. Methodology

To compare these two techniques and determine the best parameters for each, the same dataset was used. MNIST is a dataset containing images of hand-drawn numbers (0 – 9) with corresponding labels. These images (28 x 28) have been transformed into size (784, 1) where each entry is the intensity of that corresponding pixel. In this case, the number of clusters is known, allowing us to use K-Means clustering. The size of this MNIST dataset is 60,000, but training a model on a dataset this large is impractical and does not produce a significantly better result than a smaller piece. A smaller subset of 10,000 images from the MNIST dataset was selected. 6 sets of 10,000 images were tested to see if any significant difference in accuracy could be achieved by using a specific subset. The first 10,000 images produced the highest accuracy score and were chosen. The relative frequency of each number was calculated via the corresponding training labels, and the first 10,000 images used are representative of the 60,000-image dataset.

When using agglomerative clustering, the number of clusters remained constant at 10 across all tests. Different affinity (distance functions) and linkage were used to determine the maximum accuracy score. The affinity parameters tested were Euclidean, Manhattan, and Cosine and the linkage parameters tested were Ward, Average, Complete, and Single. All combinations were tried, yielding 10 total tests. The algorithm is not aware of which number (0-9) it is grouping, so it places each into a random cluster. A method was used to correctly map the clusters returned by the model back to its “true” label. In other words, the algorithm could map the image 9 into group 0, so the method mapped that 9 from group 0 to 9. An accuracy score (number of correct groupings / total number of samples) was then calculated by comparing the returned clusters and the correct labels given by the MNIST data. A confusion matrix was also calculated to indicate which numbers the model had trouble grouping.

Similarly, for K-Means clustering, the number of clusters remained fixed at 10 throughout all tests. Different random state, max_iter, and algorithm parameters were investigated to determine the optimum values. Random_state gives a random number generation for the initialization of the centroids used as the center of the clusters. An integer passed into this parameter makes the randomness deterministic. 5 and 1234 were used for this value. Max_iter tells the algorithm the maximum number of iterations the k-means algorithm should use on a single run. This number is 300 by default but was moved up to 10,000 (the size of the batch of images used). The algorithm parameter can move between “full” and “Elkan”. Elkan is used by default but full was investigated as well. Similarly, the same method was used to map the clusters returned by the model to their true labels. From there, an accuracy score and confusion matrix were calculated to determine the best parameters and best accuracy of the technique.

3. Results

The accuracy scores and confusion matrices of both techniques (using optimal parameters) along with their different parameters combination results are below.

Agglomerative Clustering:

The optimal parameters to produce the highest accuracy score received during tests were affinity = Euclidean and linkage = Ward. This combination yielded:

Accuracy Score: 0.6958
Confusion Matrix:

```
array([[ 979,    1,    1,    6,    2,    0,    9,    0,    3,    0],
       [   0, 1088,   12,    5,    1,    0,    0,   19,    2,    0],
       [  13,    2,  885,    9,    2,    0,    1,   66,   13,    0],
       [   1,    0,    9,  961,    5,    0,    1,   32,   23,    0],
       [   0,    1,    2,    0,  470,    0,   16,  491,    0,    0],
       [   2,    0,    3,  277,    3,    0,    9,  323,  246,    0],
       [  10,    1,    1,   13,    2,    0,  978,    1,    8,    0],
       [   0,    3,    2,    7,   40,    0,    0, 1018,    0,    0],
       [   2,    4,    3,  279,   11,    0,    6,   60,  579,    0],
       [   1,    1,    2,   20,  395,    0,    2,  555,    2,    0]])
```

The following are the resulting accuracy scores from all tested parameters. For more details and their respective confusion matrices, see the Jupityer Notebook.

Parameters (Affinity, Linkage)	**Accuracy Score**
Euclidean, Ward	0.6958
Euclidean, Average	0.2297
Euclidean, Complete	0.4109
Euclidean, Single	0.1136
Manhattan, Average	0.2287
Manhattan, Complete	0.3649
Manhattan, Single	0.1136
Cosine, Average	0.2222
Cosine, Complete	0.3853
Cosine, Single	0.1136

K-Means Clustering:
The optimal parameters to produce the highest accuracy score received during tests were Random_state = 1234. This parameter yielded:

Accuracy Score: 0.5677

Confusion Matrix:

```
array([[1536,    5,   10,   88,   47,    0,   67,    3,  232,    6],
       [   0, 2265,    3,    3,    1,    0,    3,    1,    4,    1],
       [  19,  269, 1329,  111,   73,    0,   47,   17,   57,    7],
       [  10,  190,   60, 1327,   27,    0,   17,   12,  387,   46],
       [   1,  118,    6,    0,  731,    0,   28,  542,    5,  514],
       [  25,  344,    4,  581,   52,    0,   43,   43,  564,  119],
       [  28,  169,   17,   11,  216,    0, 1488,    0,   42,    0],
       [   1,  152,   15,    1,  205,    0,    0,  955,    2,  762],
       [  7, 228,  11, 469,  53,   0,  18,  67, 989,  80],
       [  8,  73,   1,  39, 489,   0,   4, 587,  12, 801]])
```

The following are the resulting accuracy scores from tested parameters.

Parameters (Random State, Max_Iter, Algorithm)	
1234, Default (300), Default (Elkan)	0.5677
5, Default, Default	0.5675
5, 10000, Default	0.5675
5, Default, Full	0.5675

4. Analysis and Discussion

For Agglomerative Clustering, the highest accuracy score obtained out of the tested parameters (with Euclidean affinity, Ward linkage) was 0.6958. In other words, out of the 10,000 samples, the algorithm correctly grouped 6,958. This is a high percentage relative to the other parameters tested, but it does leave room for improvement. The confusion matrix highlights this, suggesting that the algorithm had trouble grouping some numbers. The algorithm did not correctly group 5 or 9. 5s were predominantly placed into either 3, 7, or 8 with 9s being predominantly placed into 4 and 7. Additionally, more 4s were placed into 6 than 4. This confusion matrix highlights patterns of error with agglomerative clustering and offers room for improvement with more fine-tuning of parameters or the use of another clustering method. Additionally, parameters matter a lot when attempting to optimize Agglomerative Clustering. The difference between the accuracy scores of the best and worst parameters was 0.5822. This makes sense because of the importance of distance to compare samples. The linkage tells the algorithm which distance to use between observations, merging samples that have the lowest distance. The affinity tells the algorithm how to calculate this distance, serving an important metric for comparison. By changing the distance used and how the distance is calculated, the samples clustered together will change, thus impacting the accuracy of the algorithm.

For K-Means clustering, the highest accuracy score obtained out of the tested parameters (with Random_state = 1234) was 0.5677. In other words, out of the 10,000 samples, the algorithm correctly grouped 5,677. This accuracy was lower than Agglomerative Clustering, suggesting Agglomerative Clustering is better for this application. However, K-Means beats out Agglomerative Clustering in runtime complexity (O(n^2)) vs (O(n^3)). The confusion matrix associated with these optimal K-Means parameters also points out some patterns of errors. Once again, the algorithm did not correctly group 5 or 9. 9s were grouped into 4 and 7 while 5s were grouped into 2, 4, and 8. Additionally, the different parameters used by K-Means do not impact the accuracy of the model as shown in the table above. With the Random_state constant, a change in Max_Iter and Algorithm did not produce any increase in accuracy.

Both methods point out a key drawback of unsupervised learning. Despite unsupervised learning being a step up over no technique at all, unsupervised learning can struggle when there are patterns or resemblances between samples in different clusters. For example, digits 3, 5, and 8 can often be confused due to a similar shape. Unsupervised methods receive no feedback from labels, so the model does not learn to distinguish between these digits while training.