Assignment 4

James Lowrey

Representative sampling was used to reduce memory usage and decrease clustering time. Sampling allows the number of data points processed to be much smaller than the original corpus and to maintain results similar to what would be expected if clustering was ran on the entire database. In this lab, the data size was reduced to a little less than ¼, a final size of 5000 samples. First the data was partitioned into 20 sections, and the same number of samples were selected from each partition without replacement. The partitioning is to ensure that the data is relatively representative, and aims to prevent randomly choosing many nearby data points. Data was not replaced in the sampling to prevent selecting the same piece of data multiple times, further improving the representativeness of the sampling (3). Each partition has about 1078 samples, and 250 samples are selected from each partition, almost 24% of the partition. If data replacement was used there would be a non-negligible chance of reselecting the same data.

**Time of Raw (Non-Sampled) Clustering Algorithm Prediction (seconds, 21578 samples)**

|  |  |  |
| --- | --- | --- |
|  | Euclidean | Cosine |
| K-Means | 30.2773 | 27.8734 |
| DBSCAN | 137.5356 | MemError |

**Time of Sampled Clustering Algorithm Prediction (s****econds, 5000 samples)**

|  |  |  |
| --- | --- | --- |
|  | Euclidean | Cosine |
| K-Means | 7.7462 | 8.5513 |
| DBSCAN | 11.7726 | 4.2628 |

The sampled data was then vectorized to allow clustering to occur. The word (feature) and class vectors of each document were vectorized before being fed to the SKLearn clustering toolkit. Word/feature vectorization was done by first finding all unique words in the processed/reduced article data, defining a consistent ordering of the values, and filling in the feature vector with the word's TF-IDF value if it was found in the document. Class labels were simply transformed into integers. In the original corpus, each document can have 0 or more class labels, but after vectorization, documents without labels were assigned the “0” label to allow for easier manipulation of data.

SKLearn's packages were used for the clustering implementations. The two clustering algorithms used were K-Means (Euclidean) and DBSCAN. K-Means is a fast, relatively simple clustering algorithm that separates a given number of clusters based upon distance from a calculated centroid. Initial centroid choices are important as poor centroids can lead to poorly clustered data (as the centroid can lie between data). SKLearn uses a default of 10 different centroid seed restarts which are chosen “in a smart way” using its k-means++ algorithm (4). K-means++ likely runs k-means on a smaller, representative data sample. DBSCAN is another clustering methodology that finds core data points of high density and expands clusters outwards from them. It is a strong algorithm for finding clusters of similar density, is resistant to noise, and can create irregularly shaped clusters (5). DBSCAN finds core points based upon it neighbors: there must be a minimum number of neighbor points within an epsilon distance. SKLearn uses NearestNeighbor algorithm modules to find neighboring points.

One drawback of SKLearn is the lack of Spherical K-Means, a K-Means algorithm with Cosine distance metric. In this lab, Spherical K-Means was emulated through usage of L2 normalization of data and SKLearn's Euclidean K-Means. Usually, spherical K-Means implies a cosine distance function and the constraint that centroid vectors are forced to lie on the unit circle (L2 normalized). SKLearn does not have a default Spherical K-Means implementation, but using L2 normalization on the original data, the default K-Means becomes proportional to the Spherical K-Means, and the results are similar (2). L2 normalizing the data before clustering makes Euclidean K-Means proportional to the Cosine Distance function. When the magnitude of direction/feature vectors are constrained to equal 1, the square of the Euclidean distance is equal to twice the magnitude of the cosine distance.

Proof of the Relationship Between Squared Euclidean Distance and Cosine Distance (2)

This initial normalization of the data is important as it allows the subsequent Euclidean K-Means to achieve results proportional to Spherical K-Means. As the results are proportional, clusterings will be similar (2). Spherical K-Means typically performs better on text classification as representative feature vectors are classically very sparse (3).

A variety of metrics were used for clustering quality. K-Means is commonly evaluated based upon the varying clusters' Sum of Squared Error values. SSE sums the squared euclidean distance of every point from its centroid's midpoint. Average centroid radius is a related, similar measure and was also calculated. K-Means tends to create globular clusters, so calculations measuring point distance to its centroid do a fair job of informing on the relative quality of the clustering. For DBSCAN, where clusters can be irregularly shaped radius and SSE are not useful measures. Entropy was thus used as a cluster quality measure for both algorithms. Entropy measures the homogeneity/spread of different classes within a cluster. The Entropy of the original, unclustered data was found and used in conjunction with the Entropy of the clustered data to determine Information Gain of each clustering schema.

Different clustering schema were compared to find the optimal parametrization of the clustering algorithm. The original, un-sampled data has 121 class labels (including the No-Class label) while the random, representatively sampled version tended to fluctuate around 85-95 classes. If the classes are assumed to be internally similar, it would make sense if the number of clusters were close to the number classes. However, as more clusters are added to K-Means, or the epsilon increased or minNeighbors increased in DBSCAN, the run-time of the algorithms dramatically increases. Using a saved representative sample, different clustering parameters were compared across several runs of the algorithm.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | #classes |  |  |  |  |  |
| K-Means |  |  |  |  |  |  |
| K-Means (Cosine) |  |  |  |  |  |  |
| DBSCAN |  |  |  |  |  |  |
| DBSCAN (Cosine) |  |  |  |  |  |  |

Sources:

1. “Margin-based local regression for adaptive filtering”. Proceedings of the 12th International Conference on Information and Knowledge Management, pages 191-198, 2003. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.307.1537>

2. Ashok N. Srivastava, Mehran Sahami. “Text Mining: Classification, Clustering and Applications”, pages 162-163. [https://books.google.com/books?id=BnvYaYhMl-MC&pg=PA162&lpg=PA162&dq=clustering+l2+normalization+spherical+k+means&source=bl&ots=oj1IxpWWgg&sig=7r7u\_3gCwBQu9NgJg-nrZrCgPyo&hl=en&sa=X&ved=0CEEQ6AEwBGoVChMIzerw8aH6yAIVyz8-Ch3UDg2c#v=onepage&q=clustering%20l2%20normalization%20spherical%20k%20means&f=false](https://books.google.com/books?id=BnvYaYhMl-MC&pg=PA162&lpg=PA162&dq=clustering+l2+normalization+spherical+k+means&source=bl&ots=oj1IxpWWgg&sig=7r7u_3gCwBQu9NgJg-nrZrCgPyo&hl=en&sa=X&ved=0CEEQ6AEwBGoVChMIzerw8aH6yAIVyz8-Ch3UDg2c" \l "v=onepage&q=clustering l2 normalization spherical k means&f=false)

3. Mary Parker, “Sampling With Replacement and Sampling without Replacement”. <https://www.ma.utexas.edu/users/parker/sampling/repl.htm>

4. “sklearn.cluster.Kmeans”. Scikit Learn. <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

5. “sklearn.cluster.DBSCAN”. Scikit Learn. <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html>