**IBM Unsupervised Learning – NIPS Conference Papers Clustering Projects**

This study looks into using clustering techniques in order to cluster words that are used in similar papers. Specifically, Kmeans(++) clustering, hierarchical agglomerative clustering and DBSCAN are used in an effort to group similar words within the dataset using the papers they occur in NIPS.

**1 Introduction**

This dataset can be accessed at <https://archive.ics.uci.edu/ml/datasets/NIPS+Conference+Papers+1987-2015> [1] and contains the distribution of words in the full text of the NIPS conference papers published from 1987 to 2015. The data contains 11463 unique words (words that appeared less than 50 times were not included) and spans over 5811 NIPS conference papers. The number of appearances of a word in a paper is counted.

**2 Data Exploration and Cleaning**

The dataset was found to have no missing or N/A values and was fully numerical (excluding the labelling of the words).

The distribution of word counts was found in order to determine whether it was necessary to scale the data – using the distance metric of Euclidean distance, it would only make sense to use the original scaling given all of the papers were around the same length.

|  |
| --- |
|  |
| Figure 1: Distribution of word counts |

Given the wide range of (key) word counts, it became clear that a scaling method was required whereby papers with high quantities of as word would be closer together whereas those with low quantities would keep more of their original geometric distance. As a result, the data was scaled with a log1p transformation for Kmeans clustering (as opposed to a standard logarithmic transformation due to the large quantity of zero values in the data). For the ither clustering methods, instead the distance metric used was cosine distance.

**4 Clustering Models**

**4.1 KMeans Clustering**

In order to determine the best number of clusters to use for the KMeans model, various models were fit with different numbers of clusters and 13 clusters was found to be the optimum number.

|  |
| --- |
|  |
| Figure 2: The elbow point |

In this case, the Kmeans++ algorithm was used in order to try and find the optimal solution.

|  |
| --- |
|  |
| Figure 3: Kmeans clustering example class |

It was found that Kmeans did provide some useful classes of related words though some were less so.

**4.2 Hierarchical agglomerative clustering**

|  |
| --- |
|  |
| Figure 4: HAC cluster example |

**4.3 DBSCAN Clustering**

Density-Based Spatial Clustering of Applications with Noise was the fastest of the clustering techniques – likely because it is absolute and does not need to iterate as such.

However, likely due to the clusters of different densities, 6459 outliers were detected. An example class is shown in Figure

|  |
| --- |
|  |
| Figure 5: DBSCAN cluster example |

**5 Key Findings**

DBSCAN was by far the fastest algorithm. However, from a subjective point of view, Hierarchical agglomerative clustering appeared to have the best groupings of related words. Both of these methods had wide imbalances in the size of the clusters though and perhaps the larger clusters were not of particular usefulness.

**6 Possible Flaws**

Kmeans clustering assumes that the clusters are “spherical” in shape and this is a particular drawback. Additionally, when not using mini batches, it takes a much longer time. It is also unclear on the best number of clusters to choose, even after evaluating the inertia of the different choices.

Hierarchical agglomerative clustering appeared to produce the best clusters in our case. However, like Kmeans, it takes quite a long time relatively speaking., especially for larger datasets.

DBSCAN Clustering, whilst fastest required a number of input parameters and so took a long time to optimise. It also produced a lot of outliers (or in the case of other parameters one large cluster) meaning it was not as useful in this case.

**7 Next Steps**

For next steps, one could do the reverse of this analysis and instead use clustering on the papers – finding similar groups of papers.

Additionally, this clustering of words could be used in the case of word encoding for uses such as neural machine translation.

The dataset is also highly specific as all papers are from the same journal. Further steps to take might be to consider papers in alternative fields.

**References**

[1] 'Poisson Random Fields for Dynamic Feature Models'. Perrone V., Jenkins P. A., Spano D., Teh Y. W. (2016). [[Web Link]](arXiv:1611.07460) ([[Web Link]](https://arxiv.org/abs/1611.07460)).