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.

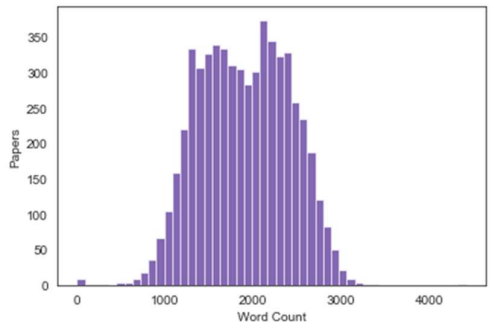

Figure 1: Distribution of word counts

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.

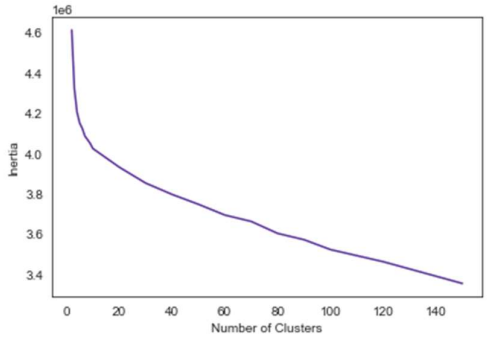

Figure 2: The elbow point

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

	Word	Class
3681	feature	12
3682	features	12
4671	image	12
4675	images	12
5587	learn	12
5590	learned	12
6890	object	12
8316	recognition	12
10666	trained	12
10667	training	12

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

	Word	Class
381	amplifier	6
382	amplifiers	6
388	analog	6
1323	capacitance	6
1326	capacitors	6
1486	charge	6
1521	chip	6
1522	chips	6
1539	chris	6
1566	circuit	6
1567	circuitry	6
1568	circuits	6
2726	digital	6
2950	drain	6
3110	electron	6
3111	electronic	6
3600	fabricated	6
3601	fabrication	6
3733	film	6
3816	floating	6
4377	hardware	6
4573	hot	6
4927	injection	6
5119	inverter	6
5568	lazzaro	6
5919	mahowald	6
6130	mead	6
6456	mos	6
6458	mosis	6
7937	programmable	6
8415	refresh	6
8633	resistor	6
8634	resistors	6
9304	shibata	6
10053	substrate	6
10681	transconductance	6
10694	transfers	6
10707	transistor	6
10708	transistors	6
10827	tunneling	6
11209	voltage	6
11210	voltages	6

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

	Word	Class
611	artifact	24
612	artifacts	24
2244	course	24
2245	courses	24
3287	eog	24
3344	erp	24
3345	erps	24
4976	instructed	24
5314	jung	24
5812	locked	24
5935	makeig	24
9000	scalp	24
11161	vigilance	24

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] ([Web Link]).