Clustering vs Classification of Newspaper Articles By Topic

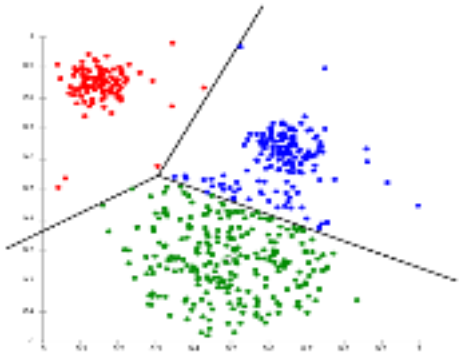
Jared Welch

News Data presents unique challenges compared to numerical data in the realm of machine learning and classification of that data. Due to the nature of the text, reducing the noise of the dataset is also a primary concern. Many words are not valuable. In order to classify the data, this project will aim to first classify by topic, based upon word frequencies in the article. There are many different approaches for estimating the weights of the keywords within articles. The primary considerations when choosing between these techniques include the difficulty in implementation, the overhead of its calculations, and the accuracy of the results. Accuracy in this case means achieving valid topic classification of the articles, which accurately reflect the true contents of the article. For example, if an article is truly about a popular sports team and its players, classification of this article should rank it as related to other articles about the same topic. The problem of topic classification of news data is complex, and there are trade offs depending on which method is chosen. The problem addressed will be deciding upon the most appropriate method to achieve valid topic clustering or classification within the text data.

The first consideration to address is reduction of noise. Before the news data is analyzed, so called ‘noisy words’ such as ‘and’, ‘the’, and other words which can be used in many contexts must be filtered so they do not skew the results of the data. There are various techniques for this, but discussion of stop word removal is outside the scope of this paper. In the case of this project, most of it is written using Python and

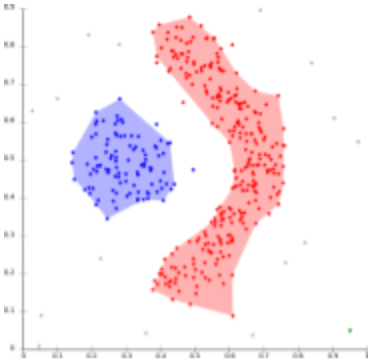
Python libraries. Because of this, removal of stop words will be a pre-processing step before topic classification. There exists an NLTK package which can be used to remove stop words from the text [1]. Due to ease of use, this will be the method chosen to filter out noisy words from the data. Another valuable tool in preprocessing is stemming. Stemming algorithms remove words such as training, trained, and trains can all be replaced by their root train [2]. This also helps reduce the size and feature set of the data before processing further, increasing the efficiency of the classification by reducing the size of the data and removing unnecessary tenses and forms of root words.

The first technique to discuss is clustering. This process, at a high level, involves creating clusters of data, where each cluster represents a grouping of similar articles based upon their keywords[3]. One important feature of clustering data is that the data sorts itself out based upon measures of similarity between the data. There are several different types of clustering. The three main types are connectivity based clustering, centroid-based, and distribution based. Connectivity based clustering is based on the idea that those items more closely related will be closer together in distance. Centroid clustering involves representing clusters by a centroid, meaning a vector which represents the cluster as a whole. Then comparing these vectors, the similarity can be measure by how close in distance the representative centroids are [4]. A popular implementation of this technique is K means. Generally, centroid clustering requires choosing K clusters in advance,



Graphical Representation of K-means clustering [10]

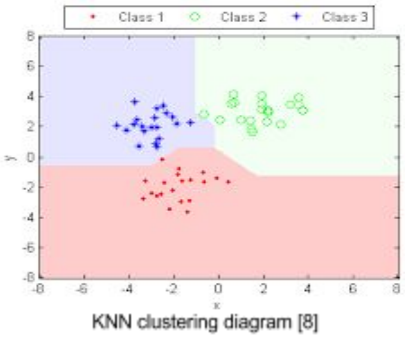
which can be considered one its drawbacks. Finally, another clustering method is Distribution-based, which organizes data points within clusters by their most likely placement within a distribution. Those data points most likely to appear within the same distribution are clustered. This method is generally less suited to text data, and more suited to statistical data, so it does not seem appropriate to the problem at hand. Of these, it appears that a variant of K-means, called bisecting K-means, is most appropriate for text clustering, and performs better than hierarchical clustering (connectivity based)[5]. In light of this, it appears bisecting k-means is the best solution to the problem. However, this technique is unsupervised learning and does not allow for choice of classification, that could be a potential drawback. Supervised learning introduces the element of choice into this process.



Density Based Clustering Diagram [9]

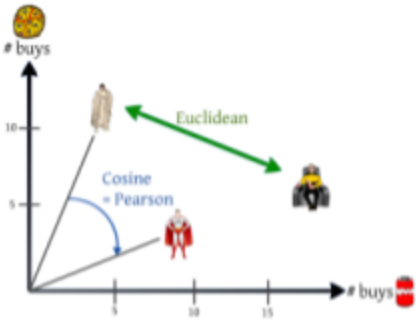
The supervised learning technique most suited to what has been described, is K nearest neighbors. This technique is similar to k-means in that it classifies based upon how similar the data point is to other data points, and place in a class accordingly. The advantage offered with KNN that k-means lacks in this case is that k-means randomly assigns clusters, where as KNN is more used to classify. This classifies the data based upon how similar it is to the training data (or weighted vectors from pre-determined tf-idf values from chosen topic documents) provided initially. Indeed, supervised learning could yield poor results if the initial training data is weak, and clustering does not have this drawback. Below, a diagram of KNN classified data can be seen. Notice that those

in close proximity are considered similar, and groupings of similar data points create class boundaries.



KNN clustering diagram [8]

KNN solutions require a distance function to measure how similar data points are. It is important to vectorize the text for this reason, and represent it in such a way that the vector accurately represents the text within the document, as without a numerical representation of the document, it is impossible to measure difference in a meaningful way to KNN. One such way to represent the data is TF-IDF, which stands for term frequency-inverse document frequency. TF-IDF is a way to measure how important certain words are in a document, and those weights allow representation of the document in a vector format[6]. Using the TF-IDF weights, a similarity score can be assigned using cosine similarity, which measures the angles between the two vectors to determine how similar they are. Another way to measure similarity between vectors is Euclidian distance. Under experimentation, at high dimension it appears that cosine similarity and euclidean distance perform equally well as metrics for similarity[7]. Because python is the primary language for the project, and



Comparison of Cosine Similarity and Euclidean distance [11]

cosine similarity can be calculated using python libraries already available, cosine similarity will be the choice of similarity measurement. Indeed, these metrics can also be utilized in the previously mentioned clustering techniques to group the data, grouping those most similar by using these metrics.

In the scope of news data, since it is not clear what topics might be important, it seems less appropriate to use supervised techniques to classify the data. Indeed, since training data will not be readily available, and would take time to generate, it will increase the project overhead quite a bit to gather this training data. In light of this, it appears unsupervised clustering using k-means is the optimal solution for this problem. This will allow the data itself to decide upon which clusters are most important, and a subset of these clusters can be chosen as representing certain topics, based upon the keywords contained.

An important consideration in choosing K-means as a technique for clustering the data is that there are already readily available implementations of this algorithm, and it is relatively simple to implement with good accuracy of the results of the data. A well defined algorithm and procedure with existing implementation is important for two reasons. First, it allows for more streamlined development, as it is more manageable to develop software that implements existing techniques. Since time is a factor in solving this problem, it is important to implement a solution that is already proven and functional instead of trying to create the software from scratch based on theoretical concepts. And second, using techniques that have been implemented many times already suggests precedent and validation for the technique's usefulness. Less tested techniques might

not yield as accurate results, or be as straight forward to implement, and k-means overcomes both of those parts of the problem.

Addressing the problem of topic classification in text documents is complex and many considerations must be made. First, it is important to reduce the noise of the dataset in order to prevent bias from the many placeholder words such as ‘the’. Once the dataset has been reduced, using NLTK libraries readily available, classification can begin. Depending on TF-IDF similarity of the document to the training data, KNN classification will place the document in proximity to its closest neighbors, where closest means cosine similarity value that is closest to 1. This classification has precedent, is relatively simple to implement compared to the other techniques, and is an accurate measure at higher levels of dimensionality, which will be achieved with ease due to the multidimensional nature of text. Unsupervised clustering techniques fit all the criteria necessary, but do not allow for classification to compare the data against; rather clustering removes the ability to choose the topics themselves, but in this case that is actually more desired, so that training data is not needed to gather results. K-means clustering will allow classification in later stages based upon the data itself, rather than decisions made by developers in advance. This should result in clusters that are chosen based upon their significance, rather than chosen in advance, which should overall increase the accuracy of the project’s classification of the data at later stages.

Works cited

[1] Information about NLTK python package for removing non-essential words from text
https://pythonspot.com/en/nltk-stop-words/

[2] Discussion of Machine Learning Techniques applied to Text
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.9153&rep=rep1&type=pdf

[3] Book detailing clustering techniques
Cluster analysis, 5th edition
Brian S. Everitt, Sabine Landau, Morven Leese, Daniel Stahl
February 2011, ©2010

[4] Stanford resource for clustering techniques and relevant implementation details
https://web.stanford.edu/class/cs345a/slides/12-clustering.pdf

[5] Discussion of clustering in relation to documents and text
http://glaros.dtc.umn.edu/gkhome/fetch/papers/docclusterKDDTMW00.pdf

[6] Breakdown of K means clustering with text data
https://www.codeproject.com/Articles/439890/Text-Documents-Clustering-using-K-Means-Algorithm

[7] Discussion of cosine similarity and euclidean distance in KNN
http://www.cse.msu.edu/~pramanik/research/papers/2003Papers/sac04.pdf

[8] Image example of KNN
http://www.peteryu.ca/tutorials/matlab/visualize_decision_boundaries

[9] Image of Density based clusters
https://en.wikipedia.org/wiki/Cluster_analysis#/media/File:DBSCAN-density-data.svg

[10] Image of Kmeans
https://en.wikipedia.org/wiki/File:KMeans-Gaussian-data.svg

[11] Image of Euclidean Distance
https://comsysto.wordpress.com/2013/04/03/background-of-collaborative-filtering-with-mahout/