Darrell Nelson II

Text Mining (IST - 736)

09/19/2019

HW02/03: Vectorizing Text Corpus

Introduction

Data mining is a collection of techniques for efficient automated discovery of patterns in large databases. Conventionally, the information mined is built into a model of the semantic structure of the datasets. The model can be utilized for prediction and categorization of new data, or for pattern recognition and topic creation. In recent years the size of databases has increased rapidly, which has led to a growing interest in the development of tools capable in the automatic extraction of knowledge from data. Fields such as marketing, customer relationship management, engineering, medicine, crime analysis, and web mining including many others utilize data mining.

Databases are rich with hidden information, which can be used for intelligent decision making. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or forecasting. Classification is a data mining (machine learning) technique used to predict group membership for data instances. Classification is the task of generalizing known structure to apply to new data while clustering is the task of discovering groups and structures in the data that are in some way or another similar, without using known structures in the data.

Clustering is one of the major data mining tasks and aims to group data objects into meaningful classes such that the similarity of objects within clusters is maximized and the similarity of objects from different clusters is minimized. Cluster analysis is a good way for quick review of data, especially if the objects are classified into many groups. Clustering does not require a prior knowledge of the groups that are formed and the members who must belong to it.

Analysis and Models

About the Data

Small corpus of 6 text files. Three files are homework complaints filled out by the user due to frustration over this assignment. The other three files were created by the professor about dogs. All text files contain less than 50 words and are ~1 KB in size. The complaint files are labeled “Complaint1”, “Complaint2”, and “Complaint3” and the dog files are named “Dog1”, “Dog2”, and “Dog3” respectively. The files were all read into the Spyder (IDE) as one large file. (*Figure 1.*)

```
This is disgusting. I'm having a very hard time with this homework.None of my code works. None of the examples work.
There is nothing online I can find for vectoring twitter data.I'm not sure why this homework is so hard. The Asynch
material is absolutely horrible.A dog is a great pet. Dogs need attention and training. A dog can be a fun companion.
Why get a dog? Because dogs are fun and loving. The best pet is a dog.There are many pet options. Getting a dog is one
pet option. While cats can be fun, dogs are better for sports and exercise.If you get a dog, consider training your
dog. Also remember that other humans exist. So, do not let your dog bark. Train your dog to listen to your commands and
to have fun with you. You dog will appreciate good training.
```

Figure 1. Raw Text Imported into Spyder

Each file was then separated out into its own data instance (row). The filenames were used as row labels and each file was tokenized/vectorized by making all words lowercase and removing punctuations. This created a document term matrix (*Figure 2.*) with each row demarcating a file and words in the corpus constituting a column.

	absolutely	also	and	appreciate	...	work	works	you	your
Complaint1	0	0	0	0	...	0	0	0	0
Complaint2	0	0	0	0	...	1	1	0	0
Complaint3	1	0	0	0	...	0	0	0	0
Dog1	0	0	2	0	...	0	0	0	0
Dog2	0	0	1	0	...	0	0	0	0
Dog3	0	1	1	1	...	0	0	3	4

Figure 2. Term Document Matrix

K-means Clustering

K-means clustering is an unsupervised machine learning algorithm that randomly assigns *k* number of points to be the centroids for each cluster. Each point in the dataset is then assigned a cluster based off its nearest centroid. Once all data points are assigned, the centroid is recalculated based on the average value of all its members. All members are then re-assigned to their nearest centroid and the centroid is in turn recalculated. This process continues until the centroids reach equilibrium. The pitfalls of this technique are that the user must be confident in how many clusters should be created, and the model needs to be run multiple times to try and find the minimum. If not run enough, the model could be showing a local minima solution. Since the initial centroids are random guesses, there is a possibility that they will split one cluster into two or combine clusters when they should be separated out. A good practice to ensure quality results is to run the K-means algorithm multiple times.

Results

The document term matrix was then run through a K-means algorithm with k = 2. Two is the optimal number of clusters as there are only two distinct topics being talked about in the texts. One topic is about the homework, the other is about dogs. The initial K-means results are below:

Complaint1	0
Complaint2	0
Complaint3	0
Dog1	0
Dog2	0
Dog3	1

Table 1. K-means (Initial)

The algorithm had a horrible time deciding which two groups belonged to each other. It put all documents into the first group and the last document into the second. There is a marginal suspicion that it only marked 'Dog3' as being different from the others because it was forced to have 2 groups, and not because it found a distinct difference.

To improve performance, a removal of stopwords was initiated. Stopwords are words that are frequent in language however do not add important details to the sentence. For example, "the", "and", and "a" are stopwords. By removing stopwords the number of unique words in the corpus went from 81 to 53. Then, words were stemmed. Stemming is the removal of prefixes and suffixes and leaving the root word or displaying the lemma. The lemma is the morphological root analysis of a word which reduces words that have the same meaning in a corpus to their original parts (e.g. studies & studying have the lemma: study). This cleaned corpus was then fed through the K-means algorithm once more:

Complaint1	1
Complaint2	1
Complaint3	1
Dog1	0
Dog2	1
Dog3	0

Table 2. K-means (Final)

Here the results show a much better distinction between the Complaint and Dog texts. The model was run 6 times and there appears to be an issue with grouping "Dog2" under the Dog classification. Upon deeper inspection "Dog2" has a low word count for "dog" (1) and "dogs" (1) whereas the other two dog texts have 2 or more counts in one of those words. (*Table 3.*)

```
In [72]: print(CleanDF["dog"])
Complaint1    0
Complaint2    0
Complaint3    0
Dog1          4
Dog2          1
Dog3          5
Name: dog, dtype: int64

In [73]: print(CleanDF["dogs"])
Complaint1    0
Complaint2    0
Complaint3    0
Dog1          2
Dog2          1
Dog3          0
Name: dogs, dtype: int64
```

Table 3. Word Count Inspection

Raw Dog2 Text:

There are many pet options. Getting a dog is one pet option. While cats can be fun, dogs are better for sports and exercise.

Conclusion

Preprocessing is an essential part of text mining and natural language processing. While this example was simple, it showed the power of taking the time to understand and clean the data before jumping into machine learning models. Many times preprocessing is boring and tedious and therefore gets a bad rep, but this exercise proves that the best models are developed when the data is cleaned. As they say, “Garbage In equals Garbage Out”.