CSE 5243: Introduction to Data Mining

Assignment 1

Vaibhav Devekar
Akshay Nikam

Goal of the Assignment

This assignment aims at preprocessing a given rudimentary data to form structured datasets and feature vectors for further processing.

Feature Vectors

We created several feature vectors from the Reuters articles:

1. Data Matrix:
   Data matrix is a two dimensional structured dataset, one dimension (Y) of which identifies each Reuters article while the other (X) has the extracted wordlist. Each entry *(x, y)* in the data matrix is the number of times the word *(x)* has appeared in the article *(y)*.
2. Transaction Matrix:
   Transaction matrix is a two dimensional dataset which provides a list of words appeared in each article.
3. Document Frequency and Inverse Document Frequency:
   This dataset specifies the number of articles a word has appeared in and also the inverse of this value.

Logic of the program

Iterate through each file:
    Remove invalid or interfering characters and feed file to parser.
    For each article from parsed data:
        Store article id, topics list, places list.
        For each word in text:
            Apply stemming on the each word.
            Add word to frequency list for that article.
        Update the Document Frequency for all words in the article.

Sort the Document Frequency by the value(frequency).
Output Document Frequency and Inverse Document Frequency.
Trim by 0.75% based on maximum frequency.
Remove stopwords from trimmed word list.
Output final word list.
Output data matrix.
Output transaction matrix.

Input Parsing and Document Frequency

We parse the input Reuters articles one by one and create a dictionary containing following items for each of them:

1. Frequency dictionary: dictionary of words appeared in the article body and title as keys and the number of times they occur in the article and title as their values
2. Topics: List of topics in the article
3. Places: Places the article refers to in <PLACES> tag

Data is only stored for those articles that have either title or body. We ignore the articles that lack both these parameters as it does not make sense to include them without any useful information. As we parse through the input articles, we also record the number of articles a word appears in. Note that we only consider pure alphabetic words i.e. no numbers. This creates a ***Document Frequency*** map. We also compute inverse document frequency as:

$$IDF_i = \log(\frac{N}{DF_i})$$

Where N = total number of documents,

DF = Document Frequency

DF and IDF help to identify a word's importance or share in the feature vector.

Stemming and Lemmatization

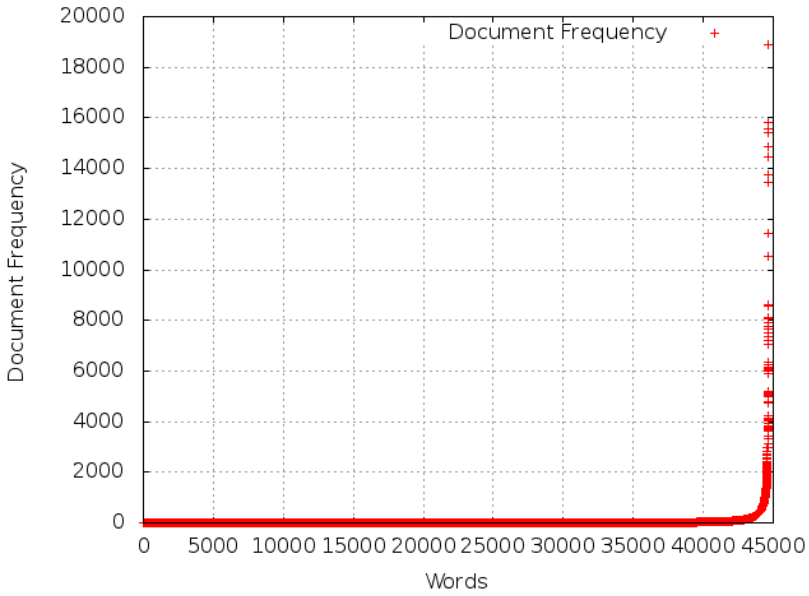
Stemming is a process of grouping together words that come from the same root. Using NLTK stemming, we find out root of each word and use it instead of the original word appeared in the article. This way, we count different forms of the word as the same word and do not lose on the frequency of these words. The idea being applied behind the usage of these words is captured instead of the individual words which is more effective in terms of natural language analysis.

We perform this step before all other words shortlisting steps because different forms of a single stem will not be lost due to low individual frequency of stemming were done in later stages.

We also tried lemmatization and though it was as effective as lemmatization in reducing number of words, it is not guaranteed that it is effective in detecting root word without context it appears in. As a result, we preferred stemming over lemmatization.

Trimming wordlist based on thresholds

We sort the words (from this point onwards, we refer to the word stems as words) by the frequency they appear in all the documents and remove the words that appear too many times or too few times. Words that occur too many times are seldom important words that are a deciding factor for classification of articles. They tend to be the common words used in natural languages as grammatical constructs without proper meaning w.r.t. the subject of the article. Also, the words that appear too few times are not going to be good decision factors for classification. It is best to get rid of them and maintain those words that occur with a moderate frequency. We identified threshold as 0.75% of the maximum document frequency and we remove words from both sides that are within this threshold.



Stopwords filtering

Stopwords are the words that appear very frequently in the natural language and often do not have any interesting meaning from the knowledge discovery perspective. Such words should not be considered for data analysis as they can hardly be useful in determining the class of an article. We provide a custom list of stopwords (taken from NLTK) along with the submission which we use to filter the words from the Reuters articles.

Class labels and weighted words

We identify words in <TOPICS> tags as class labels for the articles along with the words in <PLACES> tags, and maintain them separately from the wordlist. These are recorded in the end of the data matrix. We provide more weight to the words that appear in the titles as they hold more importance with respect to that article. This is the final word list that we use in constructing the feature vectors.

Data Matrix

From the reduced wordlist, we compute the data matrix in a two-dimensional data space where words, topics and places are placed along X dimension in that order and articles ids along Y. Each value is the number of times the corresponding word has appeared in the article. For topics and places, these values will be 0 or 1 since they can appear only once in an article.

Transaction Matrix

From the data matrix generated in previous step, we compute the transaction matrix which is a list of articles with each article specifying the list of words, topics and places that appear in that article.

Individual Contribution

For this assignment, we team programmed and worked together on all parts of the assignment.