Profiling and classification of scientific documents with SAS Text Miner™

Ulrich Reincke
SAS Germany
Ulrich.Reincke@ger.sas.com

Abstract

The automatic classification of documents into categories is an increasingly important task. As in life sciences scientific document collections continue to grow at exponential growth rates, the task of retrieving and classifying the appropriate documents by hand can become unmanageable. In fact it is impossible to follow a scientific field by manual methods.

SAS Institute (www.sas.com) and the European Molecular Biology Laboratory (EMBL)/ the ELM Consortium (http://elm.eu.org) are cooperating on the development of a text mining-application for the automated identification and ranking of scientific articles. The so-called topic scoring engine is based on the SAS Text Miner. The topic scoring engine identifies documents with similar contents and creates search-profiles which will comply with the congruencies of the documents. This paper will provide an overview over the key features of the topic scoring engine.

1.2 Introduction

Text categorization techniques have traditionally determined a subset of terms that are most diagnostic of particular categories and then tried to predict the categories using the weighted frequencies of each of those terms in each document. We will refer to this technique as the truncation approach (since only a subset of terms are used). This approach is subject to several deficiencies:

1. It does not take into account terms that are highly correlated with each other, such as synonyms. As a result, it is very important to employ a useful stemming algorithm, as well.
2. Documents are rated close to each other only according to co-occurrence of terms. Documents may be semantically similar to each other while having very few of the truncated terms in common. Most of these terms only occur in a small percentage of the documents.
3. The words used need to be recomputed for each category of interest.

These problems present themselves also for text retrieval; as a result it has become de rigueur to use a reduced-dimensionality vector-space model when retrieving documents using search terms. In the vector-space model, vectors in a multi-dimensional space can represent both documents and terms. To determine which documents match retrieval terms, Latent Semantic Analysis [3] is used to find the nearest documents in that space to the search terms.

The use of a reduced-dimensionality normalized vector-space model to represent documents in multi-dimensional space can be useful for both classification and categorization of text documents, particularly in the context of categorization approaches that are based on Euclidean distances between documents, such as Discriminant and Regression Analysis.

In this paper, we will use the singular value decomposition (SVD) technique for projecting documents into a k-dimensional subspace using Enterprise Miner for Text. Different techniques for weighting the terms for both approaches are complemented.

1.3 Background

The Enterprise Miner for Text uses a vector space model [2] for representing the collection of documents. In this approach, documents are represented as vectors of length n, where n is the number of unique terms that are indexed in the collection. The vector for each document is typically very sparse because few of the terms in the collection as a whole are contained in any one given document. The entries in the vector are the frequency that each term occurs in that document. If m is the number of documents in the collection, we now have an n by m matrix A that represents the document collection. Typically, the matrix is oriented with the rows representing terms and the columns representing documents.

2.1 Weightings

Generally the entries in the term-document frequency matrix A are adjusted by a weighting factor [2]. These weightings can be critically important for developing a good categorization model.

Without taking into consideration category-specific information, there are two types of weightings that can be applied to the frequency matrix; local weights (or cell weights) and global weights (or term weights). Local weights are created by applying a function to the entry in the cell of the term-document frequency matrix. Global weights are functions of the rows of the term-document frequency matrix. As a result, local weights only deal with the frequency of a given term within a given document, while global weights are functions of how the term is spread out across the collection.

It is generally beneficial to assign the words that occur frequently, but in relatively few documents, a high weight. The documents that contain those terms will be easy to set apart from the rest of the collection. On the other hand, terms that occur in every document should receive a low weight because of their inability to discriminate between documents.

Local Weightings

Enterprise Miner for Text offers two variations of local weights (in addition to not using a local weight at all). The binary local weight sets every entry in the frequency matrix to a 1 or a 0. In this case, the number of times the term occurred is not considered important. Only information about whether the term did or did not appear in the document is retained.

A less drastic approach is the log weighting. For this local weight, each entry is operated on by the log function. Large frequencies are dampened but they still contribute more to the model than terms that only occurred once.

Global Weightings

Global weights are functions of how the word is distributed throughout the collection as a whole. These weightings are used to help determine which terms have the most discriminating power. Generally, the best terms are those that occur frequently, but only in a few documents.

Besides the option of not applying a global weight, Enterprise Miner for Text offers 4 weighting schemes:

1. Entropy – This setting actually calculates a scaled version of 1-Entropy so that the highest weight goes to terms that occur infrequently in the document collection as a whole, but frequently in a few documents.
2. Inverse Document Frequency (IDF) – Dividing by the document frequency is another approach that emphasizes terms that occur in few documents.
3. Global Frequency Times Inverse Document Frequency – Magnifies the inverse document frequency by multiplying by the global frequency.
4. Normal – Scales the frequency. Entries are proportional to the entry in the term-document frequency matrix.

2.2 Dimension Reduction and the SVD

Enterprise Miner for Text can also reduces the dimension by using the singular value decomposition (SVD) of the weighted term-document frequency matrix. As a result, documents are represented as vectors in the best-fit k-dimensional subspace. The similarity of two documents can be assessed by the dot products of the two vectors. In addition the dimensions in the subspace are orthogonal to each other.

Unfortunately, most clustering and predictive modeling algorithms work by segmenting Euclidean distance, so Enterprise Miner for Text also normalizes the document vectors to have a length of one. This essentially placing each one on the unit hyper sphere, so that Euclidean distances between points will directly correspond to the dot products of their vectors.

2.3 Generating the Final Data for the Classification Problem

The Topic Scoring Engine uses these features of text miner to generate in batch 15 different singular value decompositions each based on a different combination of the three local weights with the 5 global weights. In each SVD we calculate up to 100 singular values. Finally we join the document's SVD coordinates of the 15 runs. This allows us to represent each document in a 1500 dimensional space. In addition we have a binary target variable for each topic which indicates if a document belongs to the corresponding topic.

2.4 The Classification Approach

After this exhausting preprocessing we have to cope with the large number of input variables, some of them highly correlated with each other. Remember some of the topics are defined with very few documents. In some cases they are three or four. Including redundant inputs variables however can degrade the analysis by

- destabilizing the parameter estimates
- increasing the risk of overfitting
- confounding interpretation
- increasing computation time
- increasing scoring effort

Thus a combination of correlation analysis and variable clustering was used to reduce the input variables to about 60-100 for each topic.

Correlation Analysis

In a first step the 500 variables with the highest Spearman correlation and Hoeffding correlation coefficient are retained for each topic. The Spearman correlation coefficient was used rather than the Pearson correlation coefficient because it is less sensitive to outliers and nonlinear ties. On the other hand the Hoeffding correlation coefficient can detect a wide range of associations between two variables that are not even monotonically related. Thus this first step allows us to reduce the number of relevant input variables by more or less 60%. Linear related inputs are detected by the Spearman statistic and nonlinear related Inputs are detected by the Hoeffding statistic.

Variable Clustering

The 500 variables that passed the correlation tests will undergo variable clustering. This is closely related to principal component analysis. However, variables clustering has the advantage that it can be used as a means of variable selection and does not use all original input variables. Variable clustering finds groups of variables that are as correlated as possible among themselves and as uncorrelated as possible with variables in other clusters. Having found an appropriate set of clusters one can select from each cluster one variable to represent the cluster. This step has been tuned so that 50 to 60 variables remain for the classification task.

Stepwise Regression

In the last step we have reduced the number of input variables to a manageable number. The final step entails a stepwise linear regression that maximizes the adjusted R-squared statistic for each Topic. It is surprising that in the final model you always find input variables from each of the 15 SVD. This shows that the selected approach has its benefits. Which expert would be able to predict before the analysis which set of local and global weights will provide the best classification results. Last but not least we obtained very positive quantitative classification results on validation data. In some cases we achieved a recall of 88%, precision of 78% and a breakeven of 82%. However, do not forget that the best algorithms fail if they run on bad data. The quantitative results of the topic scoring engine depend also on the documents that the user specify to define and train a topic profile. If there is no common concept behind them, then we should not expect high recall and precision.

3. Conclusion

This paper provided an overview over the functions of the topic scoring engine. The topic scoring engine replaces keyword querying of bibliographic databases such as Pubmed with a structured automated process by means of a "document based retrieval". This will reduce research time while improving the quality of the results. The outstanding feature of the topic scoring engine is that it does not look for pre-defined vocabulary like a search engine. Instead the tool tests with different types of singular value decompositions all possible information resolutions of the concepts underlying the text. Through a complex iteration of correlation analysis, variable clustering and selection an optimum set concepts is generated which enters different types of predictive models to train a search profile for each topic. These profiles are subsequently applied as filters to new publications. This allows the user to seek publications matching these profiles without having to submit complex queries. Furthermore, users can receive weekly or even daily updates about the relevant new publications and research topics. Thus scientific literature research will be rendered much more convenient. Finally the topic scoring engine helps to overcome the barrier of false of mismatching keywords.

References

[1] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, Indexing by latent semantic analysis} *Journal of the American Society for Information Science*, 41, pp. 391-407, 1990.

[2] G. Salton and M. McGill, *Introduction to Modern Information Retrieval,* McGraw-Hill, New York, 1983.