References

Literature

1. 2006, Department of Mathematics University of Patras, Greece & Department of Computer Science & Technology, University of Peloponnese, Logitboost of Multinomial Bayesian Classifier for Text - S. Kotsiantis1, E. Athanasopoulou2, and P. Pintelas3

2. 2004, 701 First Avenue, Sunnyvale, California 94089 Document Preprocessing For Naive Bayes Classification and Clustering with Mixture of Multinomials - Dmitry Pavlov, Ramnath Balasubramanyan, Byron Dom, Shyam Kapur, Jignashu Parikh Yahoo Inc.

3. November 2006, Ieee Tranasactions on knowledge and data engineering, Some Effective Techniques for Naive Bayes Text Classification - Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng

4. 2007, Survey of Text Mining: Clustering, Classification, and Retrieval, Second Edition - Michael W. Berry and Malu Castellanos

5. 2008, Cambridge University Press, Naive Bayes text classification

6. 2010, Massachusetts Institute of Technology, Introduction to Machine Learning Second Edition, Ethem Alpaydn The MIT Press Cambridge, Massachusetts London, England

7. 2002, John Benjamin B.V.,Natural languaage processing for online applications Text retrieval, Extracton and Categorization - Peter Jackson and Isabelle Moulinier

8. Institute of Information Theory and Automation, Department of Pattern Recognition, Academy of Sciences of the Czech Republic, Prague, Czech Republic; The University of Economics, Faculty of Management, Prague, Czech Republic; Czech Technical University, Faculty of Electrical Engineering; Feature Selection using Improved Mutual Information for Text Classification - Jana Novovicov, Anton Maly, and Pavel Pudil

9. 2010, School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213-3702, USA, A Comparative Study on Feature Selection in Text Categorization - Yiming Yang and Jan O. Pedersen

Text and references

There is a wide variety of document classification mechanisms using quite different approaches. These approaches handle an input data in their own specific way which means choosing the most appropriate method for each of the tasks. One can see preprocessing input data as the first task of the categorisation algorithms. We will discuss this topic in the next paragraph. Right then, certain techniques except Naive Bayes classifier will be issued. Moreover, their advantages and disadvantages as well as their specifics will be argued.

Data pre-processing

The process of text retrieval usually begins with indexing an input document. Every word in the document takes an index or the position to simplify its handling. This attempt for document well handling is called Vector space document representation. However, one of the greatest problems that this representation does not solve by itself, is a high dimensionality of vectors.

Before the problem could be solved, the document can be pre-processed to eliminate some of the vocabulary parts that are not necessarily essential in the text. These include stop words(conjunctions, prepositions ...) and stemming words(words like train, trained, training ...). To work the high dimensionality problem out, a feature text representation is used instead. However, it is not such a different approach as the transformation of the text into the vector makes the basis here as well. So, this technique adds some functions to reduce the dimensionality of vectors. First one, best individual features uses evaluation function that is applied to a single word. Scoring individual words can be performed by some measures like term frequency, document frequency, mutual information ... The conclusion of these individual features is processed and the top features are selected afterwards. On the other hand, the subset feature selection functions firstly select the top scoring word and then add one word at a time to fullfil the number of words. Another, quite a polular approach is a feature transformation which does not measure words' weights and processes the top features but compacts the vocabulary based on feature occurrencies. Its aim is to learn discriminative transformation matrix in order to reduce initial vector space.

The big advantage of feature vector representation is that it could be used by both, instance-based and model-based classifiers. However, this method does not capture all important structural information. Therefore, it is not

convenient enough for web documents.

There exists one other way to represent a document that statistically outperforms the others and also satisfies the categorisation of web pages. It is a recently developed graph based document representation, using the k-nearest neighbour classification algorithm. Although it presented much better performance on web documents, the problem is, that the eager, model based classifiers, cannot use this method directly. So, there seems to appear new hybrid text classification methods combining both the vector and the k-nearest neighbour approach.

Classification vector models

Even though my decision for classifier is Naive Bayes, there are still ways how to approach the classifier. The ways differs mainly in vocabulary vector and the information they consist of. Older way considers input as a binary feature vector representing whether a current word is present or absent in the vocabulary. This is called multivariate Naive Bayes. Although this way is the closest one to the native Naive Bayes classifier, it still lacks the ability to utilize term frequencies in the document.

Thus a multinomial model was introduces as an alternative, representing the number frequencies of each term in the document. However, as time passes two serious problems are encounter with a multinomial Naive Bayes classier. First is a rough parameter estimation. In general. the testing data are merged into one big document and the probabilities are calculated from this big document. Well, there comes a consequence that bigger document influences the probabilities more than the smaller one, although that it does not have anything to do with the document importance. Second is that with the insufficient amount of training data the classifier cannot perform well enough as some of the categories are pretty rare to be equipped with the data of certain amount. Here come some nixed or improved techniques to improve the performance of Naive Bayes.

Classification techniques

The most suitable approaches for document classification can be divided into two groups. One could be named model-based classifiers and the other instance-based classifiers.

Model-based classifiers are built upon a mathematical model. Therefore,

their calculation steps use the basic principle of the current model. They generalize the problem and aply the principles.These include classifiers such as Naive Bayes, expectation maximisation, latent semantic indexing, artificial neural network, decision trees ...

On the other side, instance-based learning or memory-based learning is a family of learning algorithms that, instead of performing explicit generalisation, compare new problem instances with instances seen in training data, which have been saved directly into the memory. Instance-based learning is a kind of lazy learning. It is known under the name of instance-based because it makes hypotheses directly from the training set instances. As the main examples for this type of classifiers could be considered k-Nearest neighbour algorithms.

Similar projects

Before introduction of my work, here is a list of projects which have been made, using the same technique as I am planning to implement. Surely, with all their pros and cons, I will try to fully discuss their permanence as well as the preprocessing techniques they use. Furthermore, I will have a look at the classification techniques they adopt.

1. **Data mining software in Java named Weka, using Naive Bayes Classifier**

   Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Moreover, Weka provides access to SQL databases with java connectivity. It is not directly matched with the input tables for categorization process but can be simply transformed into these tables. The area which is still not included in weka algorithms is sequence modeling.
   This cross-platform software provides user except Naive Bayes and other basic algorithms even the implementation of Expectation Maximalisation algorithm or K-means algorithm. It could be easily manipalated by a graphical interface. Simply, this software is a present of text categorization.

Weka homepage

2. Classifier4J

Classifier4J is a Java library designed to do text classification. It comes with an implementation of a Bayesian classifier, and now has some other features, including a text summary facility. It makes the use of vector data reprezentation. Its API could work as a spam filter or blog cl.

Classifier4J homepage

3. A Naive Bayesian Classifier in C# - NClassifier

NClassifier is an open source product. It is a .NET library that supports text classification and text summarization. It is a very extensible library consisting largely of interfaces. It includes, out of the box, an implementation of the Bayesian classification algorithm. The classifier is synched closely with Classifier4J project. For example, the database handling in java is replaced with ADO.Net solution. The future perspective of the project is undetermined. It could at least stay with Classifier4J or, at the most, cut into its own direction.

NClassifier homepage

4. The RDP Classifier - a nave Bayesian classifier

The Ribosomal Database Project (RDP) Classifier, a nave Bayesian classifier, can rapidly and accurately classify bacterial 16S rRNA sequences into the new higher-order taxonomy proposed in Bergey's Taxonomic Outline of the Prokaryotes (2nd ed., release 5.0, Springer-Verlag, New York, NY, 2004). It provides taxonomic assignments from domain to genus, with confidence estimates for each assignment.

The RDP classifier homepage

5. **Mallet**

Mallet is a java-based software for natural language text processing. It provides efficient routines for feature vector conversion. Mallet includes a wide variety of classification tools(Naive Bayes and decision trees techniques for instance). Furthermore, Mallet has got code for evaluating its classification algorithms and its efficiency.

Mallet homepage

6. **jBNC**

It is a java toolkit, providing methods for training, testing and applying Bayesian Network Classifiers. This software was manly tested in artifiacial inteligence and machine learning tasks. It performed well.

jBNC homepage

7. **Orange**

Orange is a component-based data mining and machine learning software suite, featuring friendly yet powerful and flexible visual programming front-end for explorative data analysis and visualization, and Python bindings and libraries for scripting. It includes comprehensive set of components for data preprocessing, feature scoring and filtering, modeling, model evaluation, and exploration techniques. It is implemented in C++ (speed) and Python (flexibility). Its graphical user interface builds upon cross-platform Qt framework. Orange is distributed free under the GPL. Orange includes a component based Naive Bayes Classifier.

Orange Component Naive Bayes Classifier