General text classifier
(Project documentation)

Erik Lux

29^{th} August 2011

Technical documentation

General I

Classification library is implemented as a set of java packages working together. The parts are all exported as a jar file. To use the library, one has to set a build path of their project to this external jar file and import the main package *NaiveBayes*. Then, there are accessible two general methods for the library in use.
Method *train* to give a classifier certain knowledge about the documents classified later. The main method parameters contain string file, name of the file with a training text or plain string with a training text. Document type, the library supports a plain text files, html content, emails or can be applied with a XML document at some points of classification. Another parameter is the category name, one wants to train. If the category exists the text is added to this set. If not a new category is created internaly.
Method *classify* enables document caterorisation. The parameters are the document, the type of the text input, and an integer number, representing which of feature selectors should be used. There is a local variable, specifying the number of features in the feature vector.

Main package

This namespace provides a general accessibility to the library by one default class. It imports almost every other package and deals with it.

Data types package

A set of categories is implemented as a category vector, where the category is a new data type class. Each category has a name, its length and a vector of documents. And a few setters and getters to add, rename or change the category content.
The document is a data type too. It contains a property vector, applicable as a metadata storage, where each different piece of metadata information is one property. And a vocabulary vector which is internaly a hashtable of string words and the number of their occurence in the document text.

Plain text package

A document conversion is provided here. Each class represents a different type of conversion tool. The html, email and xml conversion tools are included.The package is freely extensible by other conversion tools due to user's choice. Or any better or more suitable method converting the text. Moreover, a metadata parser class is implemented, extracting metadata from html or email documents when neccessary.

Vector conversion package

The removal of redundant words and their stemming can be procceded by the package. Two classes are available, stopwords and stemming words. The input text is parsed according to the current stopword file deleting all their occurence. A stopword file can be easily changed or updated by the appropriate setter from the default class in the main package.
Stemming a word means converting a word into the most basic format, which can differ a lot in many situation. A freely distributed Porter stemming algorithm slightly adjusted is the basic of the stemming here.

Serialize package

The current instance of the category set is always serialised in a file specified by an internal value. All of this is obtained by the package using serializing and deserializing method. The serialisation is built in a way to be used as little as possible as it takes time. Surely, it might be used after each single serialisation process if neccessary.

Feature vector package

Afterall, finding the top vector features tends to be the most difficult and the most peculiar part of all. This namespace includes several of the BIF(Best individual feature) methods for achieving this in its own class. Term frequency, Document frequency, mutual information, information gain as well as χ^2 statistic. They can be used separately or together.
Furthermore, the package offers one of SFS(Subset feature selection) method using the pre-calculeted mutual information. The class contains one method implementing an interface, specifying the parameter and return types. As parameters, a categories current instance is needed and a vocabulary vector of the classifying document. The methods returns a java vector - feature vector.

Vector classifier

The namespace provides two methods, using current categories instance and vocabulary feature vector firstly to caculate conditional probabilities and secondly to find the most probable class for the classifying document. The document can be then optionaly used for training if expected.

Extension package

It provides an independent graphical interface structure to build a plugin application with the library project functionality.
!!!Still working here.!!!