Comparison of Classification Methods by Using the Reuters Database

Author: Gabor Kecskemeti
Supervisor: dr. Laszlo Kovacs (University of Miskolc, Department of Information Technology)

Introduction

In this paper we have focused on a frequently used data mining technique: document classification. This technique's goal is to categorize elements. This categorization is based on a sample set, which was filled with special manually categorized elements. During the preparation methods supervised learning could be used, such as back propagation and Hopfield neural networks, inductive learning methods (like decision trees, case based reasoning), etc.

Let we start with defining the basic principles of classification:
Class is a user defined concept. $O_i \in \{o_1 \dots o_C\}$
Entity(ε_i) is an element of the training or test document set. Actually this is a text object (a document), so it can be separated into sentences, and words.
Attribute is a parameter of an entity. First of all the attributes of a document must be defined, for which several possibilities exist. Here are some examples: the number of paragraphs, the number of expressions, the instance of expressions, and an expression implicates another, etc. In general document classification techniques attributes mean the expression of entities. Entities comprise several attributes. In order to be able to make use of attributes, the most important ones must be underlined and emphasized. The more attributes we have the bigger chance they cause problems. All of the classification algorithms highly depend on the number of attributes. If uncertain expressions are left in the attribute set, this will significantly increase the time required by the analysis and the classification. Later in this paper the essence of emphasizing attributes will be discussed in details.

The classification techniques' common input parameter is the fact set (Ω). This set consists of attributes (w) of certain entities (ε). Every element of the fact set has a special attribute which describes the pre-classification (O) that was made by an expert or a maintainer. The input can be described with the following formula, if the number of attributes is W, the number of entities is E, and the number of possible classes is C:

$$\Omega = \{\{\varepsilon_1, O_1\} \dots \{\varepsilon_i, O_i\} \dots \{\varepsilon_E, O_E\}\} \quad where\ \varepsilon_i = \{w_{i1} \dots w_{iW}\} and\ O_i \in \{o_1 \dots o_C\}$$

1. formula

Bayesian Classification

Two classification methods have been selected for comparison in our project: ID3 and Bayesian classification. Among them the Bayesian classification algorithm is the simpler, which is based on a simple conditional probability.

The naive Bayesian classification assumes the independence of the attributes to simplify the calculations needed by learning the fact set. This method solves the optimization problem in the 2nd formula.

$$P(O_\varepsilon|\varepsilon) \Rightarrow \max$$

$$\textit{With the Bayes theorem}: P(O_\varepsilon|\varepsilon)P(\varepsilon) = P(\varepsilon|O_\varepsilon)P(O_\varepsilon) \quad \textit{So}:$$

$$\frac{P(\varepsilon|O_\varepsilon)P(O_\varepsilon)}{P(\varepsilon)} \Rightarrow \max, \quad P(\varepsilon|O_\varepsilon)P(O_\varepsilon) \Rightarrow \max$$

$$\textit{With the assumption}: P(\varepsilon_i|O_c) = \prod_j P(w_{ij}|O_c) \quad \textit{where } P(w_{ij}|O_c) = \frac{N_{cj}}{N_j}$$

2. formula

Where N_{cj} is the **j**th attribute's occurrence in class **c**, and N_j is the **j**th attribute's occurrence in every case.

ID3 Classification

Decision tree

The ID3 Classification technique is based on decision trees. The decision tree is a method for knowledge representation. It was developed in the 60s. With the use of an attribute value set it can determine the class of a test entity. The decision tree is a cycle free graph which has nodes as attributes to support decisions. The branches of the tree represent a precedence relationship between the nodes. The weight of a branch is an element of the attribute value set of the branch's parent node.
The attributes are nodes with at least two children, because an attribute has got as many branches as the cardinality of the value set of the actual attribute. The root of the tree is the common ancestor attribute, from where the classification can be started. The last building block of the tree is the class nodes. In every relation the class is only a child, so it is a leaf of the tree in every case.

This tree can be used with the following method to classify an entity: First a decision has to be made on each attribute's actual value. Then the next node must be reached along the branch with this value. If this node is an attribute, this procedure must be repeated. If it is a class, the decision tree's level of information storage is reached. This class is the decision what has to be made when a sample with these attribute values is in question. The usage of the

tree demonstrates that a rule set can be built for each class, which enables the effective usage of this technique in a program. This rule set is based on the if-then layout.

Building up the tree

A decision tree can be built up with the ID3 algorithm which was developed in the late 1970s. When making a decision using the tree our aim is to get as near as possible to the probable solution. This is the main goal of the ID3 algorithm, which constructs the tree on a way that the attribute having maximal gain will be chosen from the available attribute set. The **gain** is the entropy fall of the learning set when a specific attribute is chosen. (This will result in the most homogeneous division of the learning set.) This attribute will be assigned to the next node. The next attribute set will be the same as before, except for the selected attribute. At least the algorithm divides the learning set based on the selected attribute's value set. Then the algorithm uses recursion to search for the best sub-trees of the learning set, since the actually divided learning set or the actual attribute set can not identify a particular class.

Attribute reduction

Prior to using a classification algorithm, its inputs must be prepared. This preparation creates a small attribute set from the large database with full of possible attributes, for example expressions. This technique is called essence emphasizing. The number or the instance of expressions is easily collectable, but it is hard to manage the huge number of expressions, so the available expressions must be filtered out before they become to attributes. This pre-filtering is called relevance analysis. So an importance value is added to every expression, this value depends on the actual expression's class or the learning set. Then only those expressions will be used which have greater relevance value than a minimal reference. In the 1st chart 4 types of relevance calculations are defined.

Relevance calculation method	Relevance calculation formula for the i^{th} expression.	Cost function
Document based	$\omega_{i1} = D(P_j(w_i \mid O_j))$	$O(W_{\max}E + W_{\max}C + W_{\max}) \Rightarrow O(EW_{\max})$
Local	$\omega_{i2} = \max_k \frac{N_{ki}}{\sum_j N_{ji}}$	$O\left(\sum_{i=0}^{W_{\max}}(W_{\max} - i) + W_{\max}E + W_{\max}C\right) \Rightarrow O(W_{\max}^2)$
Global	$\omega_{i3} = \frac{\sum_j N_{ji}}{\sum_k \sum_l N_{kl}}$	$O(W_{\max}^2 C + W_{\max}E) \Rightarrow O(CW_{\max}^2)$
TFIDF	$TF_i = P(w_i \mid \varepsilon_j) \quad IDF_i = \log(E/N_i + 1)$ $\omega_{i4} = TF_i * IDF_i$	$O(4W_{\max})$

1. chart

Where ω_{i1} is the **i**th expression document-based importance value, and $w_i|O_j$ is the **i**th expression occurrence in the **j**th class. Here N_{ji} is the **i**th expression's occurrence in the **j**th class, ω_{i2} is the **i**th expression's local relevance level, and at last ω_{i3} is the **i**th expression's global relevance level.

The lastly reviewed method in the 1st chart is the most widespread; therefore its efficiency is compared to the others.

These methods need statistically representative expressions. Every expression can be transformed to fulfil this condition with a thesaurus. An expression with a relatively rare occurrence can be generalised using the thesaurus, so that it will be relevant.

Implementation

The problems described above were solved with a program which has 3 main parts. These parts have unique interfaces, which suit to each phase of data mining. So a newly developed method can be added to it later easily.

The first part is the loader. This part is the service of the learning and testing entities. Thus it is the source of the information.
The second part is the dimension reduction. It uses the relevancy values to rank the expressions in the learning set. Although these routines can calculate the importance values for expressions, the algorithm to find expressions in the documents is more complicated, therefore it is not detailed in this paper. In these routines only words will be used. For example the a-priori algorithm can be used for detecting the frequent word sets.
The third part is the classifier. To reach its aim it needs a helper class which describes the facts. Its implementers' cost function described in the 2nd chart.

Classification method	Learning time	Classification time
ID3	$O(EW^2)$	O(W)
Bayesian	$O(EW + CW + C) \Rightarrow O(EW)$	$O(W + CW + C) \Rightarrow O(CW)$

2. chart

Results

The Reuters database consists of 2586 (E_{max}) documents. There are 123 (C) classes, so an entity could have more than one classification. The number of appeared words in the whole entity set is 24527 (W_{max}). This document set was used to be the test and the learning set of the classification.

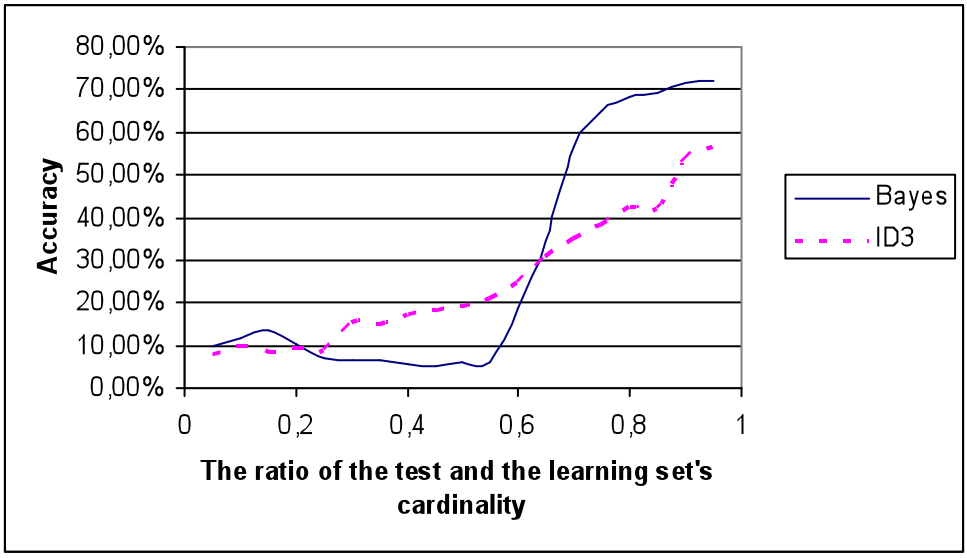
The 3rd chart demonstrates the results when using the Reuters database with 100 learning entities to construct a decision tree. This contains the Accuracy rating (AR.) of an algorithm,

and the complexity (C.) of the decision tree which means the number of leaves(lvs) of the tree.

Number of attributes	Local relevance calculation method		Global		TFIDF		Document based	
	AR.	C.	AR.	C.	AR.	C.	AR.	C.
20	3.66%	2 lvs	0.48%	4 lvs	1.99%	287 lvs	6.55%	1019 lvs
100	3.66%	2 lvs	3.72%	7 lvs	13.37%	6388 lvs	7.11%	7522 lvs

3. chart

The 1st figure compares the ID3 and the Bayesian classification algorithm:



1. figure

Summary

The usage of the learning tree is reasonable only if the learning set is relatively small and we want to classify great number of entities. The ID3 algorithm can not solve the problems with high C correctly. We can see this speciality in the comparison of the Reuters and the Origo accuracy rates at 50% and lower learning set levels. It makes decisions in a fast way, but when a new entity is to be added to the learning set, the whole learning procedure has to be repeated, which is time consuming.

The Bayesian classification method can easily learn new classes, and its accuracy will grow with the expansion of the learning set. The learning phase can be performed in smaller steps, because the two sets generated during the previous learning phase are easily alterable.

Similarly ID3, the Bayesian algorithm throws out the attributes which do not add more information. With a too high number of attributes neither of the implemented algorithms can do the learning in expected time. The ID3 classifier's learning time is increasing dramatically when some attributes are added. The measured data in chart 4 confirms the $O(W^2)$ calculation:

Number of attributes used	Learning time used by the ID3 class
20	0.879 s
100	26 s
200	118 s

4. chart

Bibliography

1. Istvan Futo: Mesterséges Intelligencia (Aula, 1999)
2. Paul Gestwicki: ID3: History, Implementations, and Applications (http://citeseer.nj.nec.com/gestwicki97id.html ,1997)
3. Laszlo Kovacs: Document Clustering using Concept Lattice and Attribute Thesaurus
4. Thorsten Joachims: A probabilistic analysis of the Rocchio Algorithm with TFIDF for Text Categorization (http://citeseer.nj.nec.com/54920.html , 1997)