



Classification of web-based discussions using Naive Bayes

by

Ekaterina Soukhikh

Supervisor: Morten Goodwin Olsen and Leiming Chen

Project report for IKT407 in Autumn 2006

based on report template version 3.0 (2006)

Agder University College
Faculty of Engineering and Science

Grimstad, 26 November 2006

Status: <Draft>

Keywords: Naïve Bayes, text classification, forum, mobile phone

Abstract: Automatic text classification is the task of assigning an unknown text a class of a set of classes and one important domain for machine learning. Automatic text classifiers are becoming more important with the growing amount of data present in our society. The naive Bayesian learning algorithm has proven to be a good choice in this area (as for example for spam filtering). This paper describes testing of efficiency of an automatic text classifier with naive Bayesian learning for classification of messages in mobile related forums.

Version Control

Version¹	Status²	Date³	Change⁴	Author⁵
V1	draft	17.11.2006	-	me
V2	draft	23.11.2006	Further work on chapters	me
V3	final	26.11.2006	Finishing discussion and conclusion chapters	me

1 **Version** indicates the version number starting at 0.1 for the first draft and 1.0 for the first review version.

2 **Status** is DRAFT, REVIEW or FINAL

3 **Date** is given in ISO format: yyyy-mm-dd

4 **Change** describes the changes carried out since the previous version

5 **Author** is the one who did the change

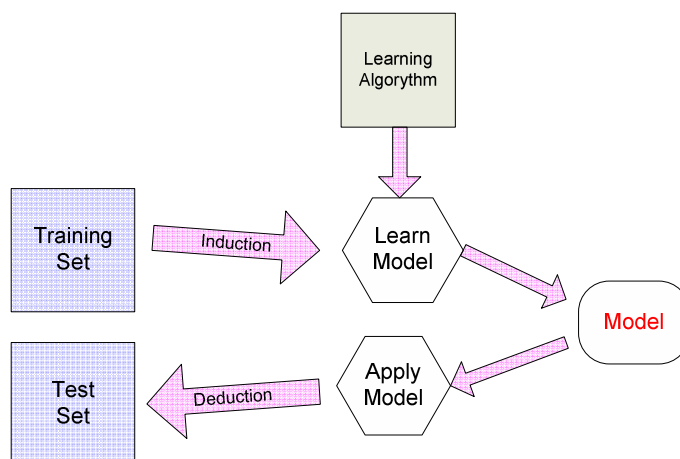
Table of Contents

1	Introduction.....	4
1.1	Acknowledgements.....	4
1.2	Report outline.....	5
2	Problem description.....	6
3	Background.....	7
3.1	Bayes rule.....	7
3.2	Bayesian Classification.....	7
3.3	Other ways.....	8
3.4	Existing tools.....	8
4	Solution.....	10
4.1	Definition of the objects of study.....	10
4.2	Theorem about their relationships.....	10
4.3	Proof of the relationships.....	11
4.4	Interpretation of results.....	14
5	Discussion.....	15
5.1	Research area speciality.....	15
5.2	Naïve Bayes approach limitations.....	15
5.3	Further development.....	16
6	Conclusion.....	17
	Appendices.....	18
	Appendix 1 Glossary & Abbreviations.....	18
	Appendix 2 References.....	18
	Appendix 3 List of attachments.....	18

1 Introduction

Automatic text classifiers are becoming more important with the growing amount of data present in our society. Automatic text classification is the task of assigning an unknown text to one class from a predefined set of classes. Assignment should of course be made as accurately as possible. Spam filters and news clipping services are two instances of well known text classifiers. This field has become one of most important domain for machine learning. An amount of different document classification techniques exists nowadays.

Next model illustrates the task of text classification using learning algorithms.



Training set is a collection of records similar to those that will be tested. Each record contains a set of attributes; one of the attributes defines the class. By analyzing the training set, a model for class attribute as a function of the values of other attributes is found. A test set is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Among the algorithms that are used for learning classification today is Support Vector Machine (which is really the set of related linear mechanism), Nearest Neighbor and Naïve Bayes. This is the last one that will be described in this project.

The naive Bayesian learning algorithm has proven to be a good choice in classification area (as for example for spam filtering). This paper describes testing of efficiency of an automatic text classifier with naive Bayesian learning on classification of messages in mobile related forums.

1.1 Acknowledgements

Originally the idea was to implement the Naive Bayes algorithm in java by my self, but while I was searching the internet about related projects, I found one implementation that seemed to be perfect for my research, and I decided to concentrate on testing this one on different test data. That's why my special thanks to Markus Forsberg and Kenneth Wilhelmsson, from The School of Mathematics and Systems Engineering, Växjö University [1].

1.2 Report outline

- Problem description

This chapter describes the problem assignment

- Background

Contains some general information about research topic and tools that were used

- Solution

Theory description, testing methods, analyzing of testing results

- Discussion

Discuss findings from the previous chapter

- Conclusion

Conclude if the Naive Bayes is useful for text classification under this specific research outlines

2 Problem description

Naive Bayes learning algorithm is widely used nowadays for text classification. Examples can be previously mentioned spam filtering. The other example is emotion modelling. [2]

My projects assignment is to investigate whether the Naive Bayes algorithm is applicable for classifying of web-based discussions. I will look at classification efficiency in three next categories:

- Forum topics
- Statement's language recognition
- Positive/negative statements

In this project the focus is only on showing a proof of usefulness of the Naive Bayes classifier with regards to web based discussion because this is the specific research area that was interesting for the external supervisor. More specified mobile related web discussions are used as giving data set, with the same reason.

I will not present any accurate statistical data on research, that in such case had to be much wider, but just conclude the general impression on efficiency and usefulness of the algorithm for such purposes.

3 Background

3.1 Bayes rule

Bayes' theorem is named after the Reverend Thomas Bayes (1702–1761), who studied computing of a distribution for the parameter of a binomial distribution (to use modern terminology). His friend, Richard Price, edited and presented the work in 1763, after Bayes' death, as *An Essay towards solving a Problem in the Doctrine of Chances*. Pierre-Simon Laplace replicated and extended these results in an essay of 1774, apparently unaware of Bayes' work. [3]

Bayes rule is a technique to estimate the likelihood of a property given the set of data as evidence or input.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{\sum_i P(B|A_i) \cdot P(A_i)}$$

In this formula A is a hypothesis and B is an observable event.

$P(A|B)$ is the posterior probability, and $P(A)$ is prior probability associated with hypothesis A.

$P(B_i)$ is the probability of the occurrence of data value A, and the $P(B|A_i)$ is the conditional probability that, given a hypothesis, a tuple satisfies it.

The other way to write the Bayes rule is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

The power of Bayes' rule is that in many situations where we want to compute $P(A|B)$ it turns out that it is difficult to do so directly, yet we might have direct information about $P(B|A)$. Bayes' rule enables us to compute $P(A|B)$ in terms of $P(B|A)$.

3.2 Bayesian Classification

Naïve Bayes classification scheme assumes that the contribution of all attributes is independent, and each contributes equally to the classification problem. Assumption about independency is clearly often wrong in reality, and that's why the algorithm got the name "naïve" or even sometimes "idiot's". Despite this it seems to work well in practice.

By analyzing the condition of each "independent" attribute, a conditional probability is determined. A classification is made by combining the impact that the different attributes have on the prediction to be made.

The approach has several advantages:

- It is easy to implement and use
- Only one scan of the training data is required
- Handles missing values easily

Previous mentioned assumption of independency of attributes is considered to be its biggest disadvantage. The other is that the accuracy of the method can be gained in expense of its efficiency, for example in cases when words used as attributes. This is because big sized

vocabularies are hard to handle, but the bigger they are, the better theoretically the accuracy rate will get.

Bayesian text classification can be implemented in (at least) 2 different ways, either using Binary Independence Model or Multinomial model. The main difference between those two is that the binary independence model assumes conditional independence between attributes (words) and gives positive and negative evidence equal weight. The multinomial model relies only on positive evidence (occurrence of the words) and ignores word frequency. [4] There is a number of researches and testing that have been performed on this area, and several articles can be found on internet that are proving the multinomial model to work better. See references for further reading.[5]

Model used for testing on this particular paper doesn't take in consideration negative presence of words and then it is closer to the multinomial model. I will explain more about this method in the chapter "Solution" while describing how it works on text classification my case.

3.3 Other ways

Some other algorithms are known to be used in text classification.

One common and easy to use technique for classification is KNN (k-Nearest Neighbour) and is absolute worth mentioning. When using KNN classification happens by examine K items near item to be classified, and placing the new item into a class which the most number of close items has. Another common classification method is the hidden Markov model. This method relies on computing the probabilities going from one state to another state.

In text classification with hidden Markov model, the text is viewed as a temporal series, with each new character having a certain probability of appearing next in the sequence based on the language and what the model has seen previously. [6]

Both this two approaches are used with good results for language recognition or author identification.

Naïve Bayes model was selected for investigation in this assignment because of the request from the external supervisor, so it was never really the choice between this one and other models.

3.4 Existing tools

The Java implementation of the Naive Bayes classification that will be used for testing consists of two programs, one trainer and one runner.

The training data consists of two files, input file and stop file.

Input file `input.txt` contains a list of pairs: the file name and the category. Below is a sample input of a set of texts each tagged with a class. This input file was used in test 1:

```
nokia.txt Nokia
ericsson.txt Ericsson
motorola.txt Motorola
sprint.txt Sprint
docomo.txt DoCoMo
```

The other is the stopfile `stopwords.txt`, which contains a list of words that should not be considered. List of English stop words that has been used in tests was founded on [7] The same source is used for finding the other languages stop words for test 2.

In windows trainer can be started from command line with the next line, where `input.txt` and `stopwords.txt` are 2 files that were mentioned before and that now where placed in the same directory with `.jar` files:

```
java -jar trainer.jar input.txt stopwords.txt
```

After training a new `bayesian.data` file is created, and this file has to be placed into the `runner.jar` archive, if one is planning to use this newly trained data in the next testing.

Test is run with the next line, where `nokia.test1.txt` is the name of the file that one wants to test:

```
java -jar runner.jar nokiatest1.txt
```

More about this implementation is in the next part of the report.

4 Solution

4.1 Definition of the objects of study

When Naïve Bayes is used for text classification, the question will be “What is the probability that a given document D belongs to a given class C ?” Using Baye’s Rule, the formula will be:

$$p(\text{Class}|\text{Document}) = \frac{p(\text{Class})p(\text{Document}|\text{Class})}{p(\text{Document})}$$

$P(\text{Document})$ is a constant divider common to every calculation, and can be disregarded. In testing just the words are taken into consideration (not white spaces, for example). Then the probability that document belongs to a specific class is a product of the conditional probabilities for each attribute value. In this case such attribute values will be words.

$$p(\text{Class}|\text{Document}) = p(\text{Class}) \prod_i p(\text{Word}_i|\text{Class})$$

During the training, the application finds words that are presented more often and creates “vocabulary” for each category. When finding probability of a test document, only those words that presented in the vocabulary will be taken into consideration.

For each word appearing in the vocabulary, the conditional probability is estimated by the following formula, where c_w is the number of times the word occurs in the category, c_{cat} the number of words in the category, and c_{voc} is the size of the vocabulary.

$$\frac{1+c_w}{c_{cat}+c_{voc}}$$

More information about the calculation of probability and a description of some specific problems about it can be found at [1].

4.2 Theorem about their relationships

Project approach consists of following steps:

- Decide test alternatives and categories / classes for each test
- Gather sample data for all the training and tests from different web forums

The samples for training and test data should’ve been presented by supervisor, but at the end I had to gather it by myself. It took some time, and I can’t really assure that this is exactly the kind of data that the supervisor had in mind.

- To run the application on training and then on test samples.

Third party application will then:

- Read an input file to get the files’ and the categories’ list for training
- Read in each text file into a Vector, removing ”stop words”

- Mapping files, assigning document's class (category)
- Analyze training data arrays and find probabilities for each documents class - $P(\text{Class})$

If each category presented with one text file, then the probability assumed to be the same for all classes and estimated for each class (category) as $= 100\% / \text{number of classes}$.

This is the method that is used in all tests in this report. Originally it is possible to assign different probabilities for each document class. As for example, to assume from the beginning that in general 70% of all the responses are positive and just 30% are negative. This can be done after deeper analysing and predicting of a data that is going to be tested.

- Find how often specific words occurs for each class $p(\text{Word}|\text{Class})$, creating the vocabularies, after removing most common words (stop words list) and most seldom words
- Analyze data samples

After testing:

- Find right/wrong ratio after a number of repetition of similar tests
- Make the conclusion about possibility to use naive Bayes for text classification of web discussions

4.3 Proof of the relationships

4.3.1 Test 1 - Forum categories

The first test goes on to classify forum categories into predefined groups. Test data was collected from the address given by supervisor. [8]

5 sub forums have been selected for testing.

Nokia
Ericsson
Motorola
Sprint
DoCoMo

It was gathered 20 "random" (20 newest) messages from different discussion threads under each sub forums. After first look at the forums content some words were added to the stop words file, such as thanks, thank, guy, guys and help. The input file was created as it showed in 3.1 and then the trainer application was be run.

The result is constructed vocabulary with 573 words.

For category Nokia - 92 entries
For category Ericsson - 134 entries
For category DoCoMo -156 entries
For category Sprint -190 entries
For category Motorola - 189 entries
bayesian.data file with the trained information was created.

Because one file with data was created for each category, the probability for each of five categories is 0,2 (20%). As the messages for training was chosen randomly, there is no wonder that vocabularies for each category varies in sizes from 92 to 189. This variation can also help to see if and how the vocabulary size affects the accuracy ratio.

At first, preliminary test was made, where small samples from learning data are used as tests samples. This test cannot be taking into consideration in final decision, as it doesn't represent the real situation.

Forum topics:	Pre-tests topics				
	nokia	ericsson	motorolla	sprint	docomo
Nokia	1	0	0	0	0
Ericsson	0	1	0	0	0
Motorola	0	0	1	0	0
Sprint	0	0	0	1	0
DoCoMo	0	0	0	0	1

In all the cases tests showed accurate results, and this showing that the implementation has learned, is working and ready for further testing. Zeroes in table are approximate values, real accurate data from all the tests can be found in attachment 1. "1" represents there a 100% probability.

At the first category test, the application showed right results with probability from 83% to 100% for all the categories except Ericsson, when it guessed Motorola. Strangely enough, then it was pretty sure that it wasn't Ericsson (0%).

Forum topics:	Estimated probability:				
	nokia	ericsson	motorolla	sprint	docomo
Nokia	100%	0	0	0	0
Ericsson	8%	0	91,2%	0	0
Motorola	0	0	100%	0	0
Sprint	0	0	2,4%	83,6%	13,9%
DoCoMo	0	0	0	0	100%

But the further retesting showed some worse results, application struggled to categorize messages from both from Nokia, Ericsson and Motorola discussions topics. The size of the vocabularies didn't seem to make any difference, probably because the variety of sizes was not so big. I will nevertheless conclude in this case that Naïve Bayes is usable for text categorizations, under some limitations that are explained more in a Discussion chapter.

4.3.2 Test 2 – Languages

Three languages were selected for testing at first, English, German and Norwegian. The languages were chosen with next criteria:

- It had to be European languages, to avoid the problems with Unicode sings that can not be stored in plain text documents. There also were some problems to find the stop words lists for some other languages.
- They had to be alike, to test the Naïve Bayes without confusing the approach by the languages that are pretty similar.

All the training and test messages were gathered on the mobile related forums, different sets were used for training and testing.

The test showed surprisingly good result as in all 9 tests the application managed to the language right with 99-100% certainty, even though some of the messages were short. The results are not presented in the report because they are not much alike then those that were gotten a bit later, while testing 5 different languages.

Then the idea to run the application on the 4-th and unknown for it language (Swedish in this case) came to mind.

	English	Norwegian	German
Swedish test 1	0,03%	0,02%	99,9%
Swedish test 2	0	99,9%	0
Swedish test 3	56,9%	32,1%	11%

The test showed that the application had really no idea what language it was, as it showed different results each time. At first look it seems surprisingly that it managed to guess on English and German side, than only Norwegian, when one would think that Norwegian and Swedish are more alike then English and Swedish, for example. But one shouldn't forget that the testing was performed on mobile-related forums, when many English (international) words have been used. Taking that into consideration, a result doesn't seem so odd any more.

In further testing the 2 other languages were added to the training data: Danish and Swedish. The motivation to choose this two was exactly opposite to the one from first part of language test. The idea was to check if the approach will be confused with the languages that are so much alike to each other. The application was re-trained with 2 new languages, and then retested on the former 3 language samples, and also on 2 new.

At first the Danish language got some false results, as 2 of 3 times it was detected to be Norwegian. That could be assumed to be probable as those two languages are much alike. But the problem presumably was that the Danish vocabulary happened to be much smaller than all the others. As the messages from forums was picked quite incidental (20 last messages as before), the Danish messages were much shorter for some reason. After the Danish vocabulary grew to the size of the others, results became different. In any case, it is important to keep in mind that the possibility of recognition of similar languages can be limited. One cannot expect the 100% hit accuracy.

3 tests for each language have been run, and the average is estimated. This caused results that presented in the table.

	English	Norwegian	German	Danish	Swedish
English test	99,8%	0	0	0	0,03%
Norwegian test	0	99,9%	0	0,03%	0
German test	0	0	100%	0	0
Danish test	0	0	0	99,9%	0
Swedish test	0	0	0	0	100%

Results in the table are approximate, the accurate results can be found in the attachment 2. In general I will conclude that test showed good results. With the limited amount of predefined classes and accurate collected training data, Naïve Bayes gives well hit accuracy and is applicable for such categorizing.

4.3.3 Test 3 – “Emotional”

Emotional test goes on recognising if one particular statement in the forum was negative or positive. Positives and negatives replies on different topics were gathered as samples for this test. The forum discussions about different Nokia’s models were used. As in tests before, here is just the approximate data; the accurate results can be found in the attachment 3. All the sources that are used for gathering data can also be found listed there.

In this test I tried to vary the test samples in size and also in degree of enthusiasm / negativity of expressions.

Tests	Positive probability	Negative probability
negative expression	100%	0
negative expression	0,3%	99,7%
positive	100%	0
Neutral positive (big size)	95,5%	4,5%
Positive (middle size)	3,6%	96,4%
Positive (short message)	100%	0

This test didn’t come to be as successful as previous two. Application seems to get right emotional spirit of short one-sided messages, either really positive or really negative, but struggles to understand neutral or two-sided messages, and also bigger ones that contain the description of what is good or bad in one particular phone model rather than just describing the feelings of the author.

4.4 Interpretation of results

As one can see tests showed pretty good results, and the conclusion about the usefulness of the chosen approach under chosen test conditions can be made, as no one expected the method to be 100% secure at the starting point.

Nevertheless there are some assumptions and limitations that have been seen or can be predicted, and it will be further discussed in the Discussion chapter.

5 Discussion

5.1 Research area speciality

Naive Bayes is widely used for text classification, as it is known to be a simple but efficient algorithm.

But classification of the web forums (or mobile forums in this case) varies from the spam classification and has some own distinctive specialities. For the first, the replies on one topic are often short and sometimes also are not directly related to the topic itself, or even posted to a wrong forum. This makes it difficult to collect the training data at the first, and also makes it harder to recognize the topic just looking on one specific message (reply) from it, although training data was accurate collected.

As it is about language recognition, this can also be tricky with the mobile related forums, as many words that are used there are international (mobile brands, models, software, or some features names). When a forum question or a reply consists just of those words and “stop words”, it makes it impossible to recognize the language, naturally.

In those bought cases can be an idea to use for testing a number of messages from the same forum/topic instead of one. This increases the size of testing sample and gives the testing mechanism more data applicable for classification.

5.2 Naïve Bayes approach limitations

As one can see from results of the testing, it works fine for language recognition, worse for text categorization, and not really so good for emotional detection, at least not in this implementation.

Language recognition showed to be easiest for the Naïve Bayes, and maybe not without the reasons. There are more differences between languages, then between categories or emotions expressions. And when the words are chosen as independent attributes, the language recognition will be the one most straightforward of 3 classifications. And even when similar languages are chosen for the testing (as Norwegian and Danish), there is still less possibility for one word to be assigned in two vocabularies, then for the other 2 cases. One simple example, rating the mobile phone one can probably write either “I like it” or “I don’t like it”. Then the word “like” will appear in both vocabularies, and it will confuse the algorithm in testing after.

Although tests for category classification didn’t showed best results, I will not exclude the Naïve Bayes as the alternative for such types of classifications. The most problem was in the training and testing data. Then the increasing of vocabularies and the testing data sizes would help a lot in getting better results. Messages that were used for testing were picked “randomly”, but after further analyzing them, I see that in some cases it would’ve impossible even for human to understand the topic / category just from one message, for some of them. And then bad results in some degree are not owed to the lacks of approach but to the lacks of testing methods.

The emotional categorization showed to be the hardest one, although at first it seemed to be most similar to the spam filtration, which is known to be the most common use of naïve bayes, as here are also just 2 classes to divide the information to. The case showed to be more complicated than that, as it not always easy to understand the enthusiasm of the author just by the set of words from the vocabulary.

5.3 Further development

As for the application the future work ideas can be to create a graphical user interface for the application in order to simplify the use of it. Another idea about application's expansion is to create one that will be able to go directly to xml code and get the texts from forums.

Further research about use of Naive Bayes for forums classification can include expansion of stop words with the words that are most common for forums, include slang words and so on. One can also see if it is useful to add some specific words to vocabularies, for example, all the phone's models' names of the brands to each of the category in test 1.

As for the test number 3 I would assume it would help to add not just words, but some signs to vocabularies. People in forums tend to use emotional symbols as "smileys" when expressing positive or negative impressions, for example.

The other approach is perhaps to look at the words in pairs. When it comes to discussing mobile phones, it makes the big difference if it is a big screen / small weight (positive expression) or a big weight / small screen (negative expression) is talked about. It is easy for a human to see that a big screen is positive response about the phone, while the machine doesn't understand it. In this particular case it doesn't make any sense to store all 4 of these words for itself in both vocabularies, but to store the pairs could be useful. As this approach is in some degree in strict with Naïve Bayes' independency of attributes, I would considerate to test some other algorithm for emotional classification to see if it will get better results, for example, previously mentioned hidden Markov model.

6 Conclusion

As it seems to me the Naïve Bayes are best for text classification with the limited amount of documents' classes. It is not really designed for text's class prediction, as it would required to gather the enormous amount of training data (to guarantee that it is similar to the testing data) and storing and handling the big size vocabularies.

The problem of the project anyway was to test if the Naive Bayes is applicable for classification (not prediction) of web discussions and this report is showing that it can definitively be useful for such purposes, under some natural and logical limitations. Because of those limitations, I will not recommend concentrating on just this one algorithm, but rather use it in cooperation with other methods, and/or take the unavoidably margins into consideration, i.e. not to expect the 100% hit accuracy.

Then the conclusion is that Naive Bayes showed to work relatively fine, and can be used for classification of web discussions, especially in cases when:

- It is possible to collect the accurate training data, which is very similar to the data that are going to be tested. Similarity of those two is one of the critical limitations of Naïve Bayes.
- It is possible to pre-define a limited number of classes / categories that tested data have to be sorted into.

Appendices

Appendix 1 Glossary & Abbreviations

Appendix 2 References

[1] The source, application and the report can be found at:
<http://www.cs.chalmers.se/~markus/LangClass/>

[2] <http://www.convo.co.uk/x02/>

[3] [http://en.wikipedia.org/wiki/Bayes' theorem](http://en.wikipedia.org/wiki/Bayes'_theorem)

[4] On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification by Karl-Michael Schneider (University of Passau, 2004)

[5] A Comparison of Event Models for Naive Bayes Text Classification (by Andrew McCallum and Kamal Nigam, Carnegie Mellon University Pittsburgh)

[6] Classification of natural language based on character frequency (Thomas Kerwin, June 7, 2006)

[7] http://en.wikipedia.org/wiki/Stop_words

[8] <http://www.cellphonehacks.com/viewforum.php?f=1>

Appendix 3 List of attachments

- TestData1
- TestData2
- TestData3