Classification of Cookie Warnings

A comparison of Naïve Bayes, Decision Trees, C4.5 and Maximum Entropy

Elbert Fliek, s1917188, e.j.fliek@student.rug.nl,
M.A. Wiering*

February 5, 2016

Abstract

The European Union dictates that visitors of websites should be warned about the placement of cookies on their computer. Web crawlers should not include cookie warnings in their searchable database. This paper compares the application of several classification algorithms, namely Naïve Bayes, Decision trees, C4.5 and Maximum Entropy, to the identifcation of cookie warnings. The results show that text classification is an effective tool in identifying cookie warnings.

1 Introduction

Cookies are small text files stored on a computer when requested by a visited website. Cookies' main uses are saving visitor information and identifying users. Cookies have become an important tool used to track users.
The EU e-Privacy Directive (2002) dictates that visitors must consent to the usage of cookies by a website. In 2012, the EU countries implemented this directive in the national law. Henceforth websites are required to warn users about their usage of cookies. Figure 1 shows an example of such a cookie warning, as used by the BBC.

These cookie warnings have become a problem for search engines. Web crawlers are used by a search engine to gather the information on websites. To ensure that search results are as relevant as possible, web crawlers should only store the main content of a webpage. Commenly discarded parts of a webpage include advertisement and menu bars. Cookie warnings are not relevant content to search engines, therefore they should also not be included in the searchable text.
This identifier was built to be used within the search engine created by OpenIndex, a Dutch firm specialised in building search engines for a diverse number of companies.
Algorithms are used to identify which parts of a webpage are important, and which can be discarded. Common methods of identifying irrelevant parts of a website look at variables such as text to hyperlink ratios within a block of text. Identification of cookie warnings is difficult for web crawlers because of their similarity to relevant content.
Because cookie warnings commonly use similar wordings, text classification techniques may prove to be very effective. This thesis will discuss and compare several classification algorithms, namely Naïve Bayes (Duda and Hart (1973)), Decision Trees (Quinlan (1986)), C4.5 (Quinlan (1993)), and Maximum Entropy

*University of Groningen, Department of Artificial Intelligence

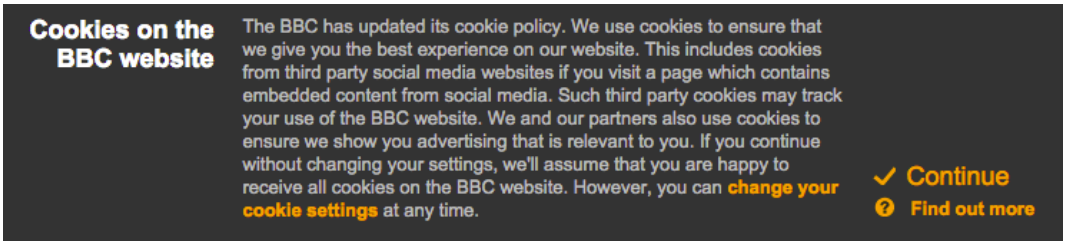

Figure 1: Example of a cookie warning (source: *http://www.bbc.co.uk*)

(Berger, Della Pietra, and Della Pietra (1996)), when applied to the identification of cookie warnings. The frequency of each word is used as a feature. This approach should be fruitful as the narrow subject of cookie warnings results in the usage of similar wording in each warning.
Section two gives a theoretical background and previous findings of each classification algorithm. Section three is dedicated to the data collection and implementation. Readers who are only interested in the performance of each algorithm can skip to section four and five where the results are presented and discussed.

2 Related literature

This section describes the Naïve Bayes, Maximum Entropy, ID3 and C4.5 algorithms. Each subsection will contain a theoretical background as well as previous emperical results.

2.1 Naïve Bayes

Naïve Bayes is one of the oldest classification algorithms. It is based on the work by Duda and Hart (1973). Naïve Bayes uses Bayes' Rule. We can state Bayes' Rule as:

$$P(Y|X_i) = \frac{P(X_i|Y)P(Y)}{P(X_i)} \qquad (2.1)$$

Where Y is the class and X_i is a feature. $P(Y|X_i)$ is known as a conditional probability. It is hard to estimate $P(Y|X_i)$ directly, therefore Bayes' rule is used. Friedman, Geiger, and Goldszmidt (1997) describe the Naïve Bayes classifier as an algorithm that calculates the conditional probability of attributes X_i. When the conditional probabilities are known, the classification algorithm then uses these conditional probabilities to calculate the likelihood that an example belongs to a certain class.
An important assumption in Naïve Bayes is the conditional independance, the assumption that all features are independent for a given class, given the class label:

$$P(f_1, f_2..f_n|Y) = \pi_i P(f_i|y) \qquad (2.2)$$

Once the most common classification algorithm, Naïve Bayes has gone out of fashion. Criticism comes from theoretical as well as empirical observations. Some argue that Naïve Bayes solves a different problem than the actual problem ($P(X_i|Y)$ instead of directly $P(Y|X_i)$), and given a choice one should always solve the problem directly, which other algorithms can do. A quote often used is from Vladimir Vapnik: *"One should solve the classification problem directly and never solve a more general problem as an intermediate step"*. Another area of discussion is the assumption of conditional independence. It is evident that in most real world applications, such as word frequencies in texts, this assumption will not hold. Using the cookie warnings as an example: if a block is a cookie warning, the pres-

ence of the words *cookie* and *accept* are highly correlated. One could argue that this would make Naïve Bayes unsuitable for the task at hand. However, as Domingos and Pazzani (1997) show, Naïve Bayes can still offer excellent performance on tasks with highly correlated features.
Another critique is that in some cases other algorithms outperform Naïve Bayes. Ng and Jordan (2002) shows that as training sets get larger logistic regression outperforms Naïve Bayes. They do however note that Naïve Bayes is better on smaller datasets.
Even though newer, more complicated algorithms, have been introduced it still offers a performance that rivals the latest classifiers. As Langley, Iba, and Thompson (1992) show, Naïve Bayes can perform as well as or better than algorithms such as C4.5.

2.2 Maximum Entropy

Maximum entropy has a long history. References go back as far as Herotodus and the Bible (Jaynes (1991)). Berger et al. (1996) describe the maximum entropy model as a model that is "consistent with all the facts, but otherwise as uniform as possible". This leads to the conclusion that when one does not have information to compare two probabilities, they should be considered the same.

The parameters of the model are then iteratively changed to maximize H, which is equal to maximizing entropy.
This can be understood more easily with an example. When one looks at a case where a sample belongs to class A, B, or C. The following must hold:

$$1 = p(A) + p(B) + p(C)$$

Of course there is an infinite number of possibilities to satisfy this constraint. One could assume $P(A) = 1$, thus always predicting class A. This is obviously a very strong assumption, with no empirical reason behind it. The result is that more information is included in the model than is actually known. A more uniform solution would be:

$$P(A) = \frac{1}{3}$$
$$P(B) = \frac{1}{3}$$
$$P(C) = \frac{1}{3}$$

This solution does not contain any information that was not fed into the model. Now it is also told that $P(A)+P(B) = \frac{1}{3}$, then we can adapt the model to:

$$P(A) = \frac{1}{6}$$
$$P(B) = \frac{1}{6}$$
$$P(C) = \frac{2}{3}$$

By repeating this until all information is used, the model will represent every bit of known information, but no more than that.
There is a strong mathematical background supporting Maximum Entropy. Berger et al. (1996), for example, show that the model with the maximum entropy is the same as the model that maximizes the likelihood of the training sample. The full mathematical derrivation is beyond the scope of this thesis, interested readers are refered to their paper (Berger et al. (1996)). An advantage of maximum entropy compared to Naïve Bayes is the lack of an assumption of statistical independence, as was shown by Langley et al. (1992). A disadvantage for some applications is the need for more samples to train the classifier.

2.3 Decision Trees

A decision tree is a representation of a model that can classify to which class a new input sample belongs. It takes the form of a tree. Each node in the tree is a test for an attribute of the input data, that has a finite number of answers. When an example is presented to the model the test at the top node is done. The result of this test determines the path down the tree. At each leaf (final) node a class is predicted. Figure 2 shows an example of a decision tree. Here the classification problem is deciding if an umbrella should be brought with the weather as input variables.
There are multiple methods to create such a tree. This paper will examine two methods developed by Ross Quinlan, the ID3 method and the C4.5 method.

2.3.1 ID3

ID3 was developed by Quinlan at the university of Sydney. See Quinlan (1986). It is, as was discussed above, a method for building a decision tree. The algorithm works as follows:

1. The dataset is presented to the algorithm.
2. For each attribute the information gain from testing for that attribute is calculated. How important is an attribute in finding the correct class?
3. The attribute with the highest information gain is selected and the dataset is split into subsets according to the possible values for the attribute.
4. The subsets go to step one, unless the problem is solved (all samples in the subset have the same class), unsolveable with this data (no attributes left to classify further), or if the subset is empty.

The possible information gain is measured in entropy. Entropy is defined as follows:

$$Entropy(set) = -\sum_{x \in X} p(x) log_2 p(x) \qquad (2.3)$$

Where $p(x)$ is the proportion of elements in class x to the elements in the whole set.
The gain is the difference in entropy before and after a split on an attribute. When the set is perfectly classified the entropy is 0, there is no more possible information gain.

2.3.2 C4.5

The second decision tree building algorithm used will be C4.5. In its current form, C4.5 was introduced by Quinlan (1993). It extends ID3. As Hssina, Merbouha, Ezzikouri, and Erritali (2014) note, ID3 is overly sensitive to features with a large number of possible values. Since our data will consist of word occurrence frequencies, it can contain quite a few different values for each attribute. By using C4.5 as described by Quinlan (1993), this problem is reduced.

3 Method

3.1 Dataset

The first step in training the algorithms is obtaining a suitable training set. As the purpose of this thesis is comparing classification algorithms, as well as providing a practical solution for OpenIndex (it must work on the websites and cookie warnings they encounter), the chosen data should be comparable to the websites that OpenIndex encounters in its daily activities. Their advice was to start with the websites of every dutch municipality and then move on to each page that is linked to from those websites.
The Nutch web crawler was used to crawl the

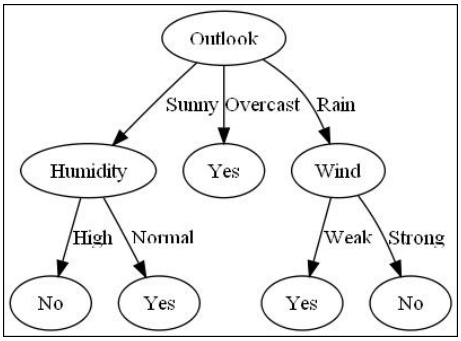

Figure 2: Decision tree example for classifying the necessity of an umbrella

web. Nutch is an open source web crawler developed by Apache. It was originally unvieled in September 2004.
After crawling every page was checked if they contained the word *cookie*. It was assumed that pages not containing *cookie* did not contain a cookie warning.
The resulting webpages are then divided into blocks of continous text. This resulted in a total of 69219 blocks. Each block is then manually marked as being (part of) a cookie warning, the result is 11101 blocks that contain (part of) a cookie warning and 58118 blocks that did not contain a cookie warning. Of those 58118 blocks, 1268 blocks contained text that was relevant to the web crawler. These blocks ranged from information about edible cookies to news articles about cookies, these type of results should be included in the searchable database, not discarded as irrelevant. The results are summarized in table 1.

Table 1: Amount of blocks

Type of block	N
Cookie warning	11101
Non-cookie warning	58118
Total	69219

3.2 Our research implementation

As our research aims to produce a working solution for a company, using an existing Java package that will be maintained for the forseeable future is a good choice. MALLET is such a package (McCallum (2002)). It is designed as a natural language processing package with classification at its core. For the specifics of the implementation of each algorithm the interested reader is urged to read their documentation and code.

3.3 Train and test split

To cross-validate the results, the data will be split into two random subsets. The training set will be 45528 examples. The test set will be 23691 examples.

4 Results

This section contains the results of running each classification algorithm on the gathered data.

4.1 Naïve Bayes

Table 2 contains the results for running the Naïve Bayes algorithm on the data. Table 3 shows the accuracy of the classifier on the training and test set. As can be noted, the

Naïve Bayes algorithm performs rather well on the data. Accuracy is high on both the train and the test part of the dataset. However there are still too many false positives (labeled as cookie, while not being a cookie message), for a production system.

Table 2: Confusion matrix for Naïve Bayes

	Label	0	1	total
0	cookie	385	72	457
1	noncookie	756	22478	23234

Table 3: Accuracy for Naïve Bayes

Set	Accuracy
Train	0.976
Test	0.965

4.2 Maximum Entropy

Table 4 contains the results for running the Maximum Entropy algorithm on the data. Table 5 shows the accuracy of the classifier on the training and test set. Performance for the maximum entropy algorithm is near perfect. Only one cookie warning is missed. This is a solid improvement over Naive Bayes.

Table 4: Confusion matrix for Maximum Entropy

	Label	0	1	total
0	cookie	456	1	457
1	noncookie	0	23234	23234

Table 5: Accuracy for Maximum Entropy

Set	Accuracy
Train	1.000
Test	1.000

4.3 Decision Trees

Table 6 contains the results for running the ID3 algorithm on the data. Table 7 shows the accuracy of the classifier on the training and test set. ID3 performs excellent on this task. Each sample is correctly classified in both the training and the test set.

Table 6: Confusion matrix for ID3

	Label	0	1	total
0	cookie	457	0	457
1	noncookie	0	23234	23234

Table 7: Accuracy for ID3

Set	Accuracy
Train	1.000
Test	1.000

4.4 C4.5

Table 8 contains the results for running the C4.5 algorithm on the data. Table 9 shows the accuracy of the classifier on the training and test set. With ID3 already showing a perfect score, it is impossible for C4.5 to be better. With both ID3 and C4.5 being perfectly accurate, we cannot say which performs better.

Table 8: Confusion matrix for C4.5

	Label	0	1	total
0	cookie	457	0	457
1	noncookie	0	23234	23234

Table 9: Accuracy for C4.5

Set	Accuracy
Train	1.000
Test	1.000

5 Conclusion

In conclusion, our research has served two purposes. Proposing a simple natural language processing method (word frequencies) and a classification algorithm to identify cookie warnings.
This was needed because mandatory cookie warnings created problems for web crawlers because they added irrelevant cookie warnings to their indices. The results show that using word frequencies as features for each block of text, four commonly used classification algorithms (Naïve Bayes, Maximum Entropy, ID3 and C4.5) can be used to identify a cookie warning as such.
Naïve Bayes performed reasonably well, with an accuracy of around 0.95, but fell short of the other 3 algorithms.
Maximum entropy made one classification error in a dataset of over 20000 samples.
The decision tree algorithms, ID3 and C4.5, both performed perfectly, each sample was correctly classified. From this test we cannot conclude if C4.5 outperforms ID3. Thus, the proposed solution has proved to be a good tool in eliminating cookie warnings.

References

A.L. Berger, V.J. Della Pietra, and S.A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, March 1996. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=234285.234289`.

P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3): 103–130, 1997.

R. Duda and P. Hart. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997. ISSN 0885-6125. doi: 10.1023/A:1007465528199. URL `http://dx.doi.org/10.1023/A\%3A1007465528199`.

Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, and Mohammed Erritali. A comparative study of decision tree ID3 and C4.5. *International Journal of Advanced Computer Science and Applications*, 4(2), 2014.

E.T. Jaynes. Notes on present status and future prospects. In W.T. Grandy and L.H. Schick, editors, *Maximum Entropy and Bayesian Methods*, volume 43 of *Fundamental Theories of Physics*, pages 1–13. Springer Netherlands, 1991. ISBN 978-94-010-5531-4. doi: 10.1007/978-94-011-3460-6_1. URL `http://dx.doi.org/10.1007/978-94-011-3460-6_1`.

P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *AAAI*, volume 90, pages 223–228, 1992.

A.K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

A.Y. Ng and M.I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841, 2002.

J.R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

J.R. Quinlan. *C4.5: programs for machine learning*, volume 1. Morgan Kaufmann, 1993.