

Table of Contents

Background.....	2
Environment.....	2
Discovering new boards.....	2
The transformation process.....	2
Solution	2
Bayes theorem.....	3
Bayes in practice as a classifier	3
Other similar solutions.....	4
Requirements.....	5
Design Specification.....	5
Observations.....	5
Classifier.....	5
Training.....	5

Background

The background for this project is that Integrasco A/S needed to develop and implement a solution to identify discussion board types like phpbb and vbulletin, given a random downloaded discussion board.

Environment

The classification of discussion boards are mainly intended to work in a crawler environment, to identify a new discussion board. The classification could also be useful in the difficult task to automate the transformation of the threads and posts, from xhtml to a more friendly format (e.g. XML).

Discovering new boards

There are millions of discussion boards located on world wide web, and to find them can be quite difficult in some cases. The discussion boards have very often a small target groups, often consisting of enthusiasts. These enthusiasts form their own community, where they discuss ideas and opinions of their subject of interest. Their subjects are also in many cases about specific products, and topics related to them. These discussion boards are important to gather information from, since many of the product specific enthusiast are often first to identify or complain about product flaws etc[1]. Enthusiast boards is often difficult to find, without first hand knowledge of the community. The classification of a given board type, would be great help in a crawling situation, since it would also identify that it is a discussion board, and not a random online newspaper or something else.

The transformation process

The transformation process, into a more friendly format, mainly consists of parsing Xquery templates, that almost fit each discussion board type with some deviations. The manual transformation process is a quite tedious job and consist of trial and error working, and a lot of testing. The testing is done manually by a down scaled agent running on the developer machine. The tests are validating the the output from the xquery parser after strict rules. This ensures that the data gathered by the crawlers is fitting the data models defined by the production system. These tests are naturally only a brief, since the 24/7 production systems gather a lot more data than a test system could do.

The transformation process consume a lot of time, and is a limitation when Integrasco need a new boards fast, to satisfy the needs of different customers. It is desirable to create a automated process to create the transformation documents. An essential part of this automated process, could be identifying the discussion board type before creating the transformation document. This process could in fact be a Learning Automata solution, that tests parts of a xquery template against a already classified board, and getting a response in form of a penalty or a reward from a validator environment. The output could then be a finished xquery document that fits the specific board pretty good.

Solution

We choose to utilize the power and simpleness of the naïve bayesian classifier to archive a solution to this problem.

Bayes theorem

Bayes' theorem (also known as Bayes' rule or Bayes' law) is a mathematical formula used for calculating conditional probabilities.

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

In this formula A is the hypothesis and B is the observation.

- $P(A|B)$ = Probability of the hypothesis (A) given a observation (B)
- $P(B|A)$ = Probability of the observation (B) given the hypothesis (A)
- $P(A)$ = Probability of the hypothesis
- $P(B)$ = Probability of observation

[8]

Bayes in practice as a classifier

A classifier that uses the Bayes theorem in practice is the Navie Bayes classifier. The naïve bayes classifier is very simple, but it can outperform more sophisticated methods.[9]

Figure 1:



A simple example [9] that we can use to try to explain the naïve bayes classifier, is a simple scenario when we have a collection of two types of knobs. One part coloured green and one type coloured red.

When a new knob is added to the collection, we can try with the help of the naïve bayes classifier, to guess with good accuracy which colour the new knob has.

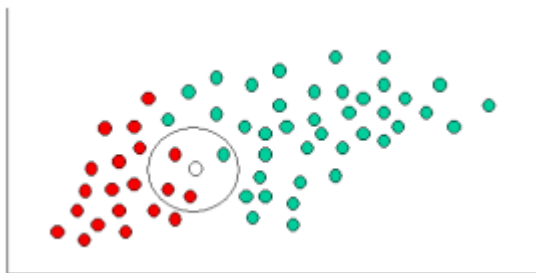
From figure 1 you can calculate, that since there are twice as many greens as reds, the new knob will probably green rather than red. This is called a prior probability, because it is built on former knowledge. The prior $P(A1)$ or $P(\text{green})$ is calculated number of greens divided by total number of knobs. The same is for $P(\text{red})$ or $P(A2)$.

$$P(\text{green}) \text{ or } P(A1) = 40/60 = 0.667$$

$$P(\text{red}) \text{ or } P(A2) = 20/60 = 0.333$$

Now we have calculated a the prior probabilities we can try to classify the new knob called X. We first selects a random cluster (the circle in figure 2) with a predefined number of knobs around the unclassified knob. Then count the number of green and red knobs inside this cluster.

Figure 2:



We can now calculate the possibilities X given that the colour is green ($P(X|A1)$). This can be calculated as number of greens in the cluster divided by total number of green points. The calculations is done in the same manner for the red knobs.

$$P(X|A1) \text{ or } P(X|\text{green}) = 1/40$$

$$P(X|A2) \text{ or } P(X|\text{red}) = 3/20$$

Combining this with the prior probabilities and multiplying these together, we will get the bayesian approximation of the problem. The calculation goes as follows:

$$P(X|A1) \times P(A1) = 1/60$$

$$P(X|A2) \times P(A2) = 1/20$$

As we see, X is classified as a red knob.

We can translate this problem to ours, and say that each knob is a observation (word or token) and the colours is the discussion board types. The observation X is then a unclassified observation (word or token) taken from a unclassified discussion board template. The cluster of knobs can then be looked upon as whole unclassified discussion board template.

The more abstract mathematical formula of the example is:

Figure 3:

$$p(C) \prod_{i=1}^n p(F_i|C).$$

- $p(C)$ is the prior probability of which colour the unclassified knob is.
- $p(F_i|C)$ is the observation given the colour (board type)
- n is number of observations in the object/circle

Other similar solutions

There have been developed quite few other solutions, implementing the naive bayes classification principles to identify the origin or what generated the given page. One of these projects is created by Svein Arild Myrer, Morten Goodwin Olsen and Tor Oskar Wilhelmsen [2] in the webmining course at HiA (later UiA) in 2003. This project tried to identify which type of authoring tool that the author had use to generate the downloaded page. The project based itself on the Naive Bayesian classifier, and made two implementations. One that looked at tag frequency in combination with author tool specific tag oddities, and another solution that based itself on continuous and discrete tests.

Requirements

To solve this project it is essential with some tools and material to perform the experiment on. Below is a list of the tools and material used.

- Java – The main programming language that is used for this project.
- Jtidy/Tagsoup – Corrects html documents and produces a xhtml document. Tagsoup does intent to change the document tag structure, only to correct markup errors, and not remove unknown tags.[3]
- Xquery and XPath – Fetches the observations that is needed from xhtml to an XML document.[4][5]
- Castor – Castor is an Open Source data binding framework for Java. Castor is, in this project, used to read an XML file with the observations that is made into java objects.[6]
- Junit – For automatic testing purposes of the methods written in java. [7]
- Integrasco Web Crawler – Used to download training and test set for our classifier.
- Training set – For training of the classifier. This set will be used to learn Naive Bayes.
- Test set – This is a set of unknown forums that we will use to conduct our experiment on.

Design Specification

Here it is outlined what is needed to be implemented to solve this project.

Observations

To identify the discussion board type we need to make some observations. The observations is html attributes from the tags. We will extract Class-names, hrefs and links to forum pages.

To make the observations for our classifier, the first page of a discussion board is taken and processed with Jtidy/Tagsoup, to correct the html and produce a clean xhtml document. The XHTML document is parsed against a XQUERY document to get a XML document.

The XML document will be processed by Castor to map it into java objects. The java objects are then cleaned for all unnecessary tokens and stop words will be removed. And then passed to the classifier.

Classifier

For classification a Naïve Bayes classifier will be used. It is the same algorithm that was presented in the lecture “Pattern Classification” by Ole-Christoffer Granmo on the 2007-08-30, only we have chosen to write it in Java and make it more object-oriented.

Training

Naïve Bayes Classifier requires some training data to learn the classifier. The quality of the training set is important. To ensure that the classifier learns efficiently it needs to have a accurate vocabulary. The vocabulary cannot contain tokens that have nothing to do with the given discussion board. To train the classifier we will run it against a training set that is given by Integrasco.

References

- [1] Kilde: (18.10.2007) <http://integrasco.no/main.do?page=services>
- [2] Kilde: (18.10.2007) http://www.eiao.net/webmining/previousprojects/ikt407_deliveries/gruppe1_2003/ProjectReport.pdf
- [3] Kilde: (18.10.2007) <http://ccil.org/~cowan/XML/tagsoup/>
- [4] Kilde: (18.10.2007) <http://www.w3.org/TR/xquery/>
- [5] Kilde: (18.10.2007) <http://www.w3.org/TR/xpath>
- [6] Kilde: (18.10.2007) <http://www.castor.org/xml-mapping.html>
- [7] Kilde: (18.10.2007) <http://www.junit.org/>
- [8] Kilde: (18.10.2007) <http://www.celiagreen.com/charlesmccreery/statistics/bayestutorial.pdf>
- [9] Kilde: (18.10.2007) <http://www.statsoft.com/textbook/stnaiveb.html>