Statistical Models for Web Searches: The naive Bayes approach

Stella Maris Salvatierra
Universidad de Navarra
Pamplona, Spain

Automation of diverse sorts, from commercial bank transactions and grocery shopping to government processes and activities has lead to the creation and storage of huge databases, often requiring terabytes of storage. Automated processes involving the Internet, such as web searches, must handle large amounts of data, drawing on databases that are growing exponentially every day. These databases not only include large number of individuals, but also large number of features or variables per individual or basic unit.

Statistical methods for classification and discrimination are especially popular tools for analyzing data in such settings. Web search queries, for example, automatically generate two populations: one constituted by those web pages that belong to (or agree with) the given query and another one containing all remaining pages. We might have more than two populations, depending of the specific goal. The goal of this research is to get statistical tools to assist in WWW databases.

This work is motivated by the fact that the World Wide Web contains millions of pages, but understands nothing by itself. To extract information from the web we need tools and these inevitable have statistical components. For example, given a query, we would like to find the web-based relevant documents. Usually searches based on “key words” produce a huge number of documents most of, them being irrelevant for the particular query. Of course, by looking at any particular page we usually know what it is. A typical web page has structure and organization, and its layout ties the words together in a stylized that may carry considerable information. Web documents contain words, graphs, hyperlinks and even sound, but everything follows an organization. This structure allows us to classify it according to different classification schemes. Web pages do not come with labels because we look at them, we can implicitly process the organizational information and syntax, and then we can label them. This happens because each page was written by humans for other humans to read visually. Our goal is to do this task of reading and classifying automatically, without using the human interaction to convert everything we see into a “label”. We need to convert all the information that we need to classify the page into something numeric. Note that the data are highly noisy and the main difficulty is that the information that we need to satisfy the query is not uniquely determined. Technically speaking, the covariates or ``discriminators" are not uniquely defined. For example, what are the relevant covariates we would need in order to identify a list of all the research projects at Carnegie Mellon University? And there is an extra point to be taking into account: Pages come written in HTML language.

One approach to solve the classification problem might be to develop a trainable system that can be taught to extract various type of information by automatically browsing the web. A first attempt to achieve this goal is a system with the following two inputs: 1) A specification of the classes and relations of interest (called an "ontology"). The classes are given, for example, by: "student," "faculty," "research project," etc. and the relations are, for example, "student of," "advisor of," etc., and 2) Training samples that describe instances of the classes and the relations. The training samples are called "labeled" data or "labeled examples" because we can give them labels reflecting the fact that we know from which class they come. Given such an ontology and a set of training samples, the system attempts to learn a general procedure for extracting new instances of these classes and relations from the web. Of course, there are also assumptions to be made with respect to the definitions of the ontology classes and relationships.

Several open research questions arise out of the context of this particular class of classification problems. A very small dataset is used as a test to develop new techniques. We reduce the entire Web to a small database consisting of web pages and hyperlinks drawn from the WWW files of several Computer Science Departments including those of University of Texas at Austin, Cornell University, University of Washington, University of Wisconsin and Carnegie Mellon University. The data were provided by the Text-Learning research group of the School of Computer Science, Carnegie Mellon University. This simplification allows us to understand and interpret the results easily at this early stage of the problem formulation.

We simplify the problem and use only the text that appears on the page. We are focused on the naive Bayes classifier. "Naive Bayes" became very popular for web searches. Intuitively speaking, the approach considers that all the possible words of the web pages of the universe compose a "Bag Of Words" and each page is made by randomly drawing a given number of words. The model underlying is incorrect (because it assumes independence of words counts within WWW pages) but the results of its use remain impressive.

When the cost of labeling data to get training samples is high, but unlabeled data can be obtained easily, one might think about applying algorithms such as EM. These algorithms are used to label the unlabeled data and use both labeled and unlabeled data to build the classifier. We show the performance of "naive Bayes" when different amount of unlabeled data are incorporated to the model. The idea is to use the labeled data to estimate the parameters, label the unlabeled ones using the maximum posterior probability of the group given the data, and iterate until convergence. After convergence is reached, the final parameter estimates are used to build the classifier. Another important issue is sampling mechanism used to draw the unlabeled data, which clearly affects the estimation of the probability of the group at each iteration of any algorithm and we have also done empirical work about this point.

Key words: Naive Bayes, Discriminant analysis, Web searches, Classification, Text learning.