Facultad de Ciencias Económicas y Empresariales
Universidad de Navarra

Working Paper nº 06/02

Using unlabeled data to improve classification in the naive Bayes approach: Application to web searches

Stella Maris Salvatierra

Facultad de Ciencias Económicas y Empresariales
Universidad de Navarra

Using unlabeled data to improve classification
in the naive Bayes approach: Application to web searches
Stella M. Salvatierra
Working Paper No. 06/02
October 2002
JEL Codes: C11, C13, C15, C49

ABSTRACT

This paper introduces a method to build a classifier based on labeled and unlabeled data. We set up the EM algorithm steps for the particular case of the naive Bayes approach and show empirical work for the restricted web page database. Original contributions includes the application of the EM algorithm to simulated data in order to see the behavior of the algorithm for different numbers of labeled and unlabeled data, and to study the effect of the sampling mechanism for the unlabeled data on the results.

Stella M. Salvatierra
Universidad de Navarra
Facultad de Ciencias Económicas y Empresariales
Departamento de Métodos Cuantitativos
31080 Pamplona
ssalvat@unav.es

USING UNLABELED DATA TO IMPROVE CLASSIFICATION IN THE NAIVE BAYES APPROACH: APPLICATION TO WEB SEARCHES

Stella M. Salvatierra[1]
Universidad de Navarra
Spain
ssalvat@unav.es

1 Introduction

The World Wide Web contains millions of pages, but understands nothing by itself. To extract information from the web we need tools and these inevitably 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.

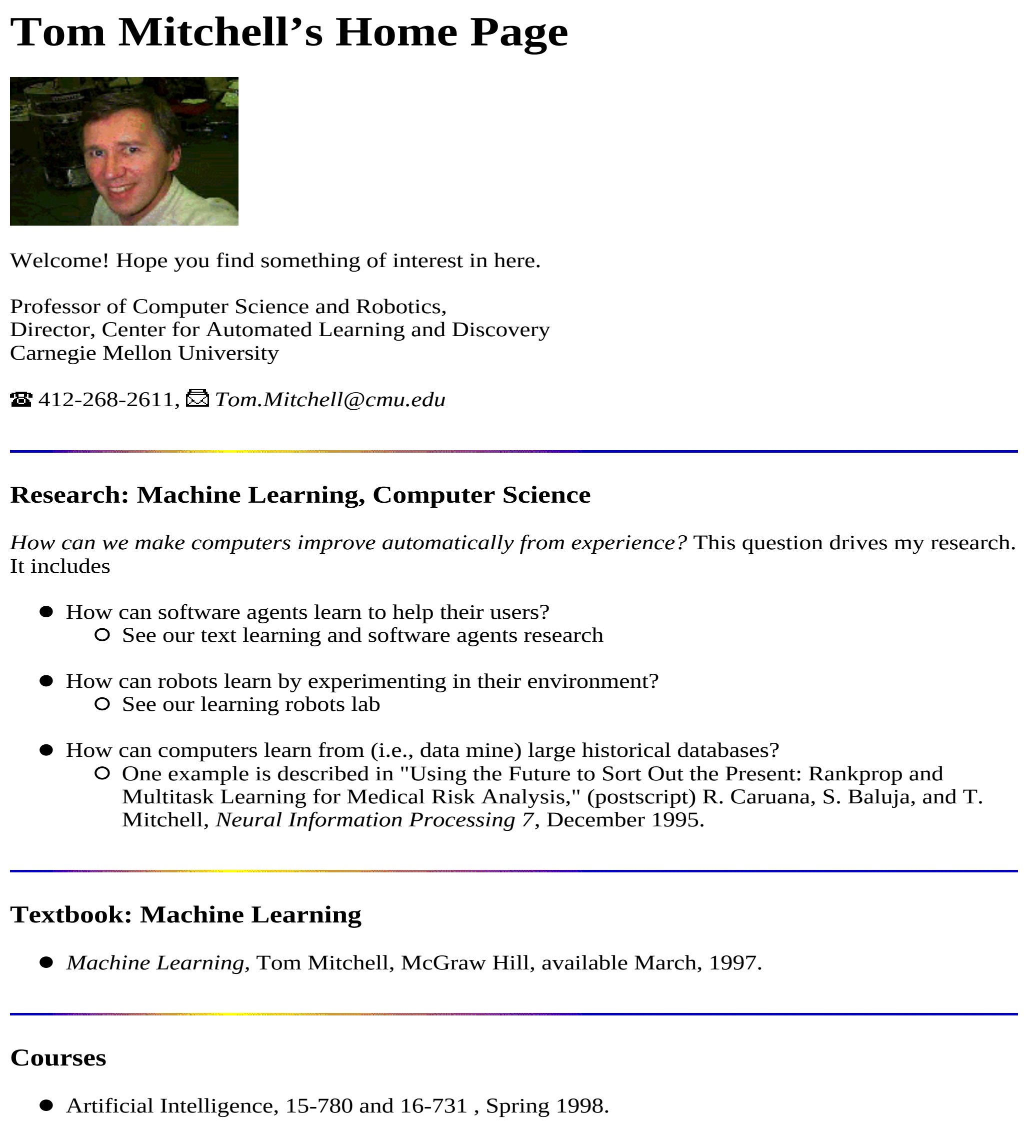
Statistical methods for classification and discrimination are especially popular tools for analyzing data in such settings. Web search queries, for example, automatically generate two populations: one constituted by those web pages that belong to (or agree with) the given query and another one containing all remaining pages.

Of course, by looking at any particular page we usually know precisely what it is and how to classify it. Let us begin by looking a typical web page. See Figure 1.

We know that Figure 1 corresponds to a Faculty member's home page. A typical web page has structure and organization, and its layout ties the words together in a stylized manner that may carry considerable information. Documents contain words, graphs, hyperlinks and even form 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, but when 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

[1] Special thanks are due to Stephen Fienberg, whose comments helped in the formulation of this work. The author also thanks to the Text Learning group of Carnegie Mellon University for helpful discussions.

at the University of Navarra?



Tom Mitchell's Home Page

Welcome! Hope you find something of interest in here.

Professor of Computer Science and Robotics,
Director, Center for Automated Learning and Discovery
Carnegie Mellon University

☎ 412-268-2611, 🖂 *Tom.Mitchell@cmu.edu*

Research: Machine Learning, Computer Science

How can we make computers improve automatically from experience? This question drives my research. It includes

- How can software agents learn to help their users?
  - See our text learning and software agents research
- How can robots learn by experimenting in their environment?
  - See our learning robots lab
- How can computers learn from (i.e., data mine) large historical databases?
  - One example is described in "Using the Future to Sort Out the Present: Rankprop and Multitask Learning for Medical Risk Analysis," (postscript) R. Caruana, S. Baluja, and T. Mitchell, *Neural Information Processing 7*, December 1995.

Textbook: Machine Learning

- *Machine Learning,* Tom Mitchell, McGraw Hill, available March, 1997.

Courses

- Artificial Intelligence, 15-780 and 16-731 , Spring 1998.

Figure 1: Example of a typical home page

One approach to solve the classification problem might be to develop a trainable system that can be taught to extract various types of information by automatically browsing the web (see Craven et al., 1998). 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 (referred to as an "ontology" in the Computer Science literature). 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 based on a visual inspection. 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.

The CMU CS text-learning research group has already created a system which, for a restricted web database, is able to answer questions like: "give me all the professors who taught course xx" (the questions are written in symbolic form). The system learns from the training samples and is not only able to give the name of the professors but also to give additional information that may be of interest, such that "students of those professors." The method that has been used by researchers in this group is based on a "naive Bayes" classifier involving word counts and became very popular for web searches (see Craven et al., 1998; Domingos and Pazzani, 1997; McCallum and Nigam, 1998; Blum and Mitchell, 1998). 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 underlying model is incorrect (because it assumes independence of word counts within WWW pages) but the results of its use remain impressive. The cited research group has also developed other models incorporating the words that appear in the hyperlinks but all of them use the naive Bayes approach as a basic classifier plus different combinations (see, for example, Blum and Mitchell, 1998).

Often the cost of labeling data to produce training samples is high, but unlabeled data can be obtained easily. In such circumstances, one might think about applying algorithms such as EM in order to infer the values of the "missing" labels and then use both labeled and unlabeled data to build the classifier. This work introduces a way to build a classifier based on both labeled and unlabeled data. We set up the EM algorithm steps for the particular case of the multinomial version of the naive Bayes approach and show empirical work for a restricted web page database. Original contribution includes the application of EM to simulated data in order to see the behavior of the algorithm for different number of labeled and unlabeled data, and to study the effect of the sampling mechanism of the unlabeled data on the results.

2 The classification problem

Let $(Y, \mathbf{X})$ denote a pair, where: Y is a categorical random variable taking values in $\{1, \cdots, K\}$ with probabilities $\{q_1, \cdots, q_K\}$; $\mathbf{X} = (X_1, \cdots, X_p)$ is a vector of p covariates; $f_i(\mathbf{x})$ is the probability density function of group i, for $i = 1, \cdots, K$. Given $\mathbf{X} = \mathbf{x}$, the posterior probability that the entity comes from group i is given by: $\tau_i(\mathbf{X}) = P(Y = i|\mathbf{X}) = f_i(\mathbf{x})q_i/\left(\sum_{j=1}^{K} f_j(\mathbf{x})q_j\right)$.

We have an entity $(Y, \mathbf{X})$ where Y is not observed, and want to allocate this individual in one of K possible groups. In other words, we wish to predict Y based on the value of $\mathbf{X}$. Let c_{ij} be the cost of allocation when an entity of group j is assigned to group i. Without loss of generality, we can assume the $0 - 1$ loss function, i.e., $c_{ii} = 0$ and $c_{ij} = 1$ for $i \neq j$ (see McLachlan, 1992). The optimal rule (denoted by $r_0(\mathbf{x})$) is the one that minimizes the conditional risk at each $\mathbf{x}$. In decision theory, that optimal rule is referred to as the Bayes rule (see McLachlan, 1992). In our case, the Bayes rule is given by

$$r_0(\mathbf{x}) = i \qquad \text{if} \qquad \tau_i(\mathbf{x}) \geq \tau_j(\mathbf{x}) \qquad\qquad j = 1, \cdots, K, \quad j \neq i.$$

If $\tau_i(\mathbf{x}) = \tau_j(\mathbf{x})$ for some (i, j), then $r_0(\mathbf{x})$ is not uniquely defined, and the entity can be assigned arbitrarily to one of the groups for which $\tau_i(\mathbf{x})$ takes its maximum value. If $P[\tau_i(\mathbf{x}) = \tau_j(\mathbf{x})] = 0$ for $i \neq j = 1, \cdots, K$, then $r_0(\mathbf{x})$ is unique for almost all $\mathbf{x}$ relative to the underlying measure ν on $\Re^p$ appropriate for $f_{\mathbf{X}}(\mathbf{x}) = \sum_{i=1}^{K} f_i(\mathbf{x})$ (See McLachlan, 1992).

Note that we have assumed that the group prior probabilities are known and that the group-conditional densities are completely specified. This is unlikely to be the case in practice. When the parameters are unknown, we need to have entities of known origin on which $\mathbf{X}$ has been recorded. In other words, we need a sample $T = \{(Y_1, \mathbf{X}_1), \cdots, (Y_n, \mathbf{X}_n)\}$, usually referred to as a *training sample*. Unknown parameters are often estimated from the training sample T and plugged into the decision rule. Let $\mathbf{q} = (q_1, \cdots, q_K)$ be the unknown vectors of prior probabilities, and $\theta = (\theta_1, \cdots, \theta_K)$ be the unknown parameters of the group-conditional densities. Provided that $\hat{\theta}_i$ is a consistent estimator of θ_i ($i = 1, \cdots, K$) and $f_i(\mathbf{x}; \theta_\mathbf{i})$ is continuous in θ_i ($i = 1, \cdots, K$), r_0 is a Bayes-consistent rule in the sense that its risk, conditional on $(\hat{\mathbf{q}}, \hat{\theta})$, converges in probability to that of the Bayes rule, as n goes to infinity. See McLachlan (1992).

3 The naive Bayes approach

The naive Bayes approach assumes that each group conditional density is given by the product of its marginal densities. The marginal densities can be any specific densities. The independence assumption is clearly not valid most of the time, but the approach has demonstrated empirical success (see, for example, Langley, Iba, and Thompson, 1992).

We use the naive Bayes approach for the following particular situation of a finite discrete sample space. This is the multinomial naive Bayes approach; however, in all that follows we call it simply the naive Bayes approach. Assume that our sample space, S, is composed of the single events $\{a_1, \cdots, a_p\}$, with each of the a_i distinct. Suppose that p_i is the conditional probability of feature i under population G, i.e., $\pi_i = p(a_i|G)$. The naive Bayes approach considers that each individual corresponds to N independent realizations of S. Thus, the probability of a given individual I under group G is proportional to: $P(I|G) \propto \prod_{j=1}^{p} \pi_j^{x_j}$, where x_j denotes the frequency of feature a_j in individual I. Then, instead of reporting the N events that constitute each individual, we report only the frequency of the elements of S.

That is, according to the notation of Section 2, we have the following:

- q_l is the prior probability that an observation comes from population l, that is, $q_l = P(Y_i = l)$, for $i = 1, \ldots, n$.
- $\mathbf{X_i}|Y_i = l$ has multinomial distribution with parameters $(N_i, \boldsymbol{\pi}_l)$, where $N_i = \sum_{j=1}^{p} X_{ij}$ and $\boldsymbol{\pi}_l = (\pi_{l1}, \ldots, \pi_l p)$ ($l = 1, \cdots, K$).

In practice, the parameter $\boldsymbol{\theta}$ is unknown, and we use the training data *T* to estimate or "learn" a classification rule $y(\mathbf{x}|T)$ for future prediction. The usual way to do this is by using the training data to get an estimate $\hat{P}(Y = l|\mathbf{x}, T)$ of $P(Y = l|\mathbf{x}, T)$. When $\hat{P}(Y = l|\mathbf{x}, T) \neq P(Y = l|\mathbf{x}, T)$, the rule may be different from the Bayes rule and the proposed rule may not achieve the minimum Bayes risk (see, for example, McLachlan, 1992, or Flury, 1997).

4 EM algorithm

Often the cost of labeling data to produce training samples is high, but unlabeled data can be obtained easily. In such circumstances, one might think about applying algorithms such as EM in order to infer the values of the "missing" labels and then use both labeled and unlabeled data to build the classifier. For example, empirical results in Nigam et al. (2000), McCallum and Nigam (1998) or Mitchell (1999), show the overall accuracy improvement reached after incorporating unlabeled data through the EM algorithm. The application of EM to the naive Bayes approach is easy because the loglikelihood function is a linear function of the sufficient statistic and also explicit parameter estimates can be found at each iteration step. 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. The EM method for the classification problem is as follows. Assume that we have observed the values of

$$\mathbf{O} = \{(Y_1, \mathbf{X}_1), \cdots, (Y_m, \mathbf{X}_m), \mathbf{X}_{m+1}, \cdots, \mathbf{X}_n\}$$

and that the values of $Y_{m+1}, \cdots, Y_n$ are missing "at random". Denote the unknown parameters as $\boldsymbol{\theta} = (q_1, \cdots, q_K, \boldsymbol{\pi}_1, \cdots, \boldsymbol{\pi}_K)$, where $\boldsymbol{\pi}_i = (\pi_{i1}, \cdots, \pi_{ip})$.

For each individual, define a vector $\mathbf{Z_i} = (Z_{i1}, \cdots, Z_{iK})$, such that $Z_{ij} = I_{[Y_i = j]}$. Thus, we can rewrite the observed values as:

$$\mathbf{O} = \{(\mathbf{Z}_1, \mathbf{X}_1), \cdots, (\mathbf{Z}_m, \mathbf{X}_m), \mathbf{X}_{m+1}, \cdots, \mathbf{X}_n\}$$

The log-likelihood function is given by,

$$l(\boldsymbol{\theta}) = \sum_{j=1}^{K} \left(\sum_{i=1}^{n} Z_{ij} \right) \log q_j \; + \sum_{j=1}^{p} \left(\sum_{i=1}^{n} X_{ij} Z_{i1} \right) \log \pi_{1j} \; + \cdots + \sum_{j=1}^{p} \left(\sum_{i=1}^{n} X_{ij} Z_{iK} \right) \log \pi_{Kj}.$$

Note that the log-likelihood function is a linear function of the sufficient statistics $a_{jl} = \sum_{i=1}^{n} X_{ij} Z_{il}$ and $b_l = \sum_{i=1}^{n} Z_{il}$, for $j = 1, \cdots, p$ and $l = 1, \cdots, K$. Then, to get the function $Q(l(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)})) =$

$E(l|\mathbf{O}, \boldsymbol{\theta}^{(r)})$, we only need to calculate $E(Z_{il}|\mathbf{O}, \boldsymbol{\theta})$. This can be easily found by using Bayes theorem, and thus the function Q becomes:

$$Q(l(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)})) = \sum_{l=1}^{K} b_l^{(r)} \log q_l \; + \sum_{j=1}^{p} a_{j1}^{(r)} \log \pi_{1j} \; + \cdots + \sum_{j=1}^{p} a_{jK}^{(r)} \log \pi_{Kj},$$

where

$$\begin{aligned}
b_l^{(r)} &= \sum_{i=1}^{n} e_{il}^{(r)}, \\
a_{jl}^{(r)} &= \sum_{i=1}^{n} X_{ij} e_{il}^{(r)}, \\
e_{il}^{(r)} &= Z_{il} \quad \text{for} \quad i = 1, \cdots, m, \\
e_{il}^{(r)} &= \frac{q_l (\pi_{l1}^{(r-1)})^{X_{i1}} \cdots (\pi_{lp}^{(r-1)})^{X_{ip}}}{\sum_{j=1}^{p} q_j (\pi_{j1}^{(r-1)})^{X_{i1}} \cdots (\pi_{jp}^{(r-1)})^{X_{ip}}} \qquad \text{for} \qquad i = m+1, m+2, \cdots, n.
\end{aligned}$$

We use the labeled data to give initial values to the parameters, and obtain the value of $\boldsymbol{\theta}$ that maximizes $Q(l(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)}))$ explicitly at each iteration step. Specifically, at the $r+1$th iteration step, we set:

$$q_l{}^{(r+1)} = \frac{\sum_{i=1}^{m} Z_{il}}{n} + \frac{1}{n} \sum_{i=m+1}^{n} \frac{q_l{}^{(r)} \pi_{l1}{}^{(r)X_{i1}} \cdots \pi_{lp}{}^{(r)X_{ip}}}{\sum_{j=1}^{k} q_l{}^{(r)} \pi_{j1}{}^{(r)X_{i1}} \cdots \pi_{jp}{}^{(r)X_{ip}}}$$

and

$$\pi_{lj}^{(r+1)} = \frac{\sum_{i=1}^{m} X_{ij} Z_{il} + \sum_{i=m+1}^{n} X_{ij} \frac{q_l{}^{(r)} \pi_{l1}{}^{(r)X_{i1}} \ldots \pi_{lp}{}^{(r)X_{ip}}}{\sum_{j=1}^{k} q_l{}^{(r)} \pi_{j1}{}^{(r)X_{i1}} \ldots \pi_{jp}{}^{(r)X_{ip}}}}{\sum_{i=1}^{m} N_i Z_{il} + \sum_{i=m+1}^{n} N_i \frac{q_l{}^{(r)} \pi_{l1}{}^{(r)X_{i1}} \ldots \pi_{lp}{}^{(r)X_{ip}}}{\sum_{j=1}^{k} q_l{}^{(r)} \pi_{j1}{}^{(r)X_{i1}} \ldots \pi_{jp}{}^{(r)X_{ip}}}}$$

for $j = 1, \cdots, p$ and $l = 1, \cdots, K$. Then we iterate until convergence. The estimates of the parameters converge to the correspondent MLE (see Taner, 1996).

5 Empirical results

5.1 Application of EM to the web page database

The data[2] consist of a set of pages and hyperlinks drawn from the WWW files of a selected group of Computer Science Departments, including those of the University of Texas at Austin, Cornell

[2] The data were provided by the Text-Learning research group of the School of Computer Science, Carnegie Mellon University

University, University of Washington, University of Wisconsin and Carnegie Mellon University. Let $W = \{w_1, \cdots, w_p\}$ be a vocabulary base consisting of p words. Now, each pair $(Y_i, \mathbf{X}_i)$, $i = 1, \cdots, n$ represents a web page, where $\mathbf{X_i}$ is a p-dimensional random vector with components $(X_{i1}, \ldots, X_{ip})$ and each X_{ij} is the frequency of word w_j on page i, for $i = 1, \cdots, n$, $j = 1, \cdots, p$. Each page is labeled as being in one of the following categories: faculty, student, staff, research project, course or other. There are 4,127 pages. We plan to work with the four most populous ontology classes: "Student," "Research Projects," "Faculty," and "Course." We have a vocabulary base of 57 words. For more details respect to variable selection, see Salvatierra (1999).

The data can be divided in two groups according to the sampling mechanism. One group (approximately 1,000 of them) corresponds to *all* the Computer Science Department pages of University of Texas at Austin, Cornell University, University of Washington, and University of Wisconsin. The second group contains some pages of other Computer Science Departments, including Carnegie Mellon, but they do "not represent" the corresponding populations of web pages in any probabilistic sampling sense. We applied the EM algorithm to a web page classification problem using this restricted database of Computer Science Departments and a polytomous classification (student, project, faculty and course web pages). Different amounts of unlabeled data were added each time.

We have got results when unlabeled data were added without keeping fixed the "true" sampling rates. In other words, unlabeled data were sampled from the second group, where the sampling mechanism was not probabilistic. Figure 2 shows the results for this case when unlabeled data were added without keeping fixed the "true" sampling rates. In other words, unlabeled data were sampled from the second group, where the sampling mechanism was not probabilistic.

Each point in Figure 2 corresponds to an average of 20 runs. The total dataset contains only around 4000 pages, so each run of 1000 labeled documents and 2000 unlabeled ones, contains many repeated pages (i.e. page overlapping). That is the reason why the corresponding variability is small. It would be better to have a bigger dataset. One way to solve the problem is by simulation, and this is what we do in the following sections.

The results shown in Figure 2 correspond to a dimensionality of 57 for each random vector. The computations were done in Splus language and they were extremely slow. It seems that the unlabeled data hurt the accuracy when the number of labeled pages is around 100 or more. For a situation involving only a few labeled pages, the addition of unlabeled data helps the accuracy a little bit, but the difference is within the simulation error (in most of the cases). The case of "Research Project" is somewhat different in the sense that the addition of unlabeled data increases significantly the accuracy. "Research Project" was always poorly classified, even for large sample sizes.

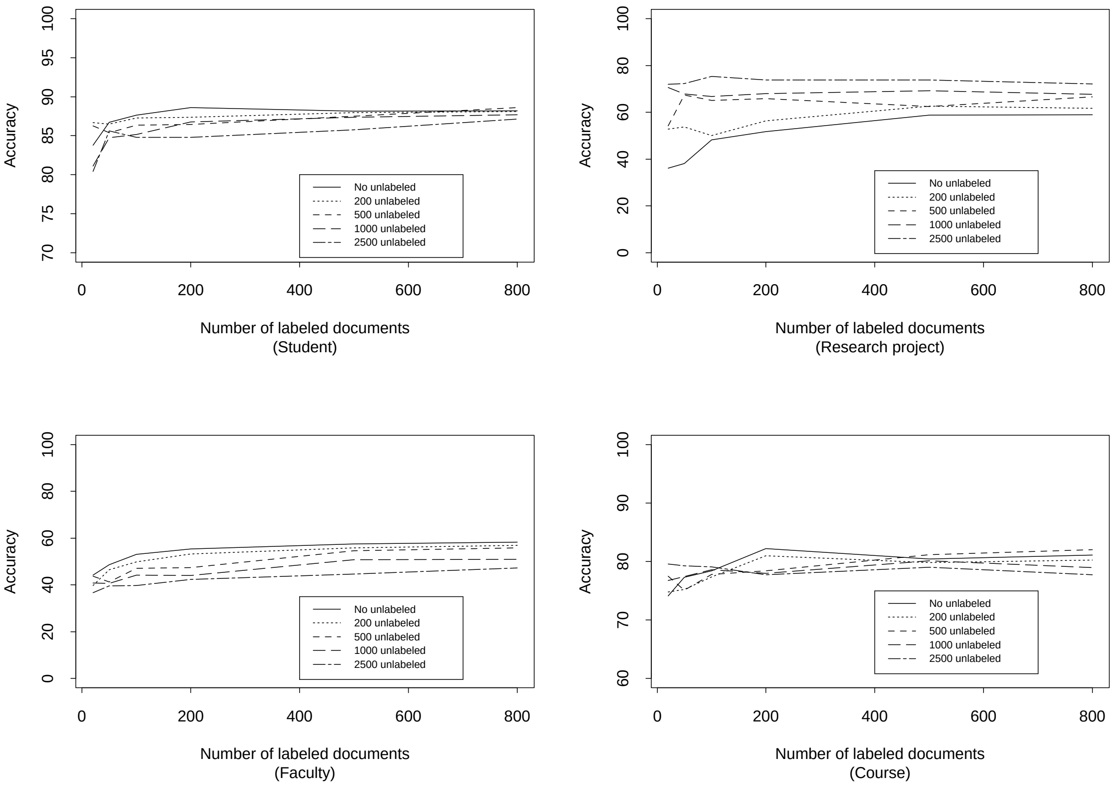


Figure 2: Mean accuracy for the polytomous classification of the web page database

5.2 Applying EM to simulated data

We simulated data for each group, having conditional multinomial distribution given the group label. The parameters where those estimated from the web page database. The steps where the following:

1. Assume there are 4 groups, with probabilities $0.53, 0.08, 0.15, 0.24$.

2. We have observations $(\mathbf{X}, Y)$, where $\mathbf{X}$ is a vector of 57 covariates, Y takes values on $\{1, 2, 3, 4\}$ with probabilities given in Step 1, and the conditional distribution of $\mathbf{X}|Y = j$ is multinomial with parameters $(N, \boldsymbol{\pi}_j)$, for $j = 1, 2, 3, 4$.

3. $\boldsymbol{\pi}_1, \cdots, \boldsymbol{\pi}_4$ correspond to the estimated ones from each group of the web page database.

4. For each observation, the parameter N was sampled from the observed distribution of the page lengths corresponding to each group of the web page database.

5. Simulate $\mathbf{O} = \{(\mathbf{X}_1, Y_1), \cdots, (\mathbf{X}_m, Y_m)\}$ according to Steps 1-4. These values will act as "observed values".

6. Simulate $\{(\mathbf{X}_{m+1}, Y_{m+1}), \cdots, (\mathbf{X}_n, Y_n)\}$ according to Steps 1-4.

7. Use $\{\mathbf{X}_{m+1}, \cdots, \mathbf{X}_n\}$ as unlabeled data.

Figure 3 shows the mean accuracy corresponding to 20 runs, when we added different amounts of data to different number of labeled examples.

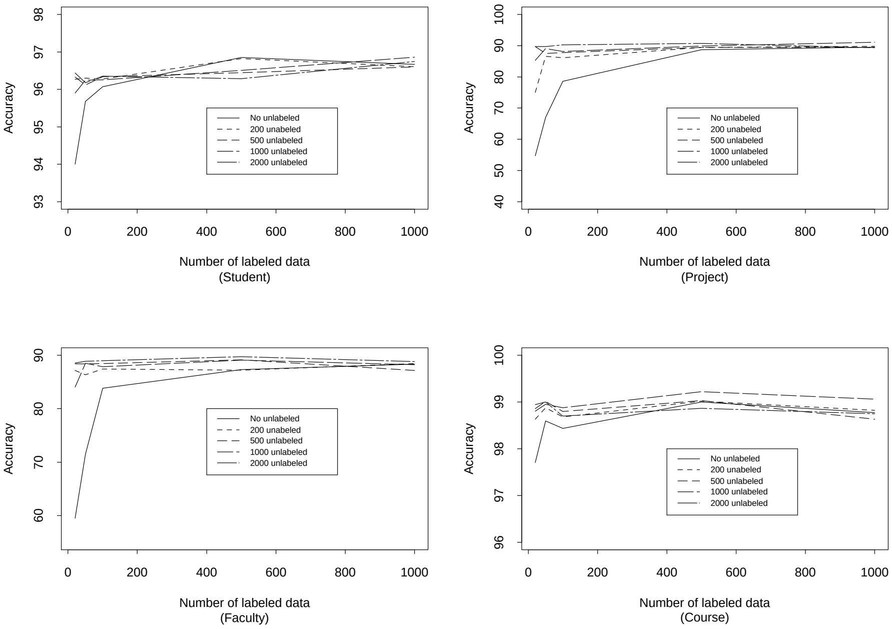


Figure 3: Mean group accuracies for the polytomous classification

For the four cases, we note that the addition of unlabeled data does not help when we have a fairly moderate number of labeled observations. In particular, unlabeled data seems to be useless once we go beyond 100 or 200 labeled observations. It is easy to be convinced that a classifier based on 20 labeled plus 1,000 unlabeled reaches the same accuracy as a classifiers based on 500 labeled. An ANOVA F-test with randomized blocks to compare "500 labeled", "20 labeled plus 1000 unlabeled" and "20 labeled plus 2,000 unlabeled" has a $p-$ value of 0.14.

"Student" and "Course" mean accuracies do not change too much, and they remains approximately constant, even for small training sample sizes. The important change corresponds to the classification of "Research Project" and "Faculty" web pages. The mean accuracy of "Research Projects" with only 20 labeled observations is 54.64 percent, whereas with 20 labeled plus 2,000 unlabeled observations, we reach a mean accuracy of 88.56 percent. The mean accuracy of 500 or 1,000 labeled observations is around 88 percent, so that we can economize on our training sample size. Nevertheless, we must be very careful before reaching a general conclusion at this point, because we have yet to take into account the variability associated with our measure of accuracy.

According to Figure 3, the accuracy seems to be approximately constant when the size of the training samples is 200 or more. Figure 4 shows the accuracy of the polytomous classification for different sizes of the training sample and no unlabeled data. Observe that the accuracy for 500 and

1,000 labeled data seems to be the same. We applied the usual ANOVA F-test for a randomized block design to these data, and it yielded a $p-$value of 0.31, suggesting that the difference is indeed ignorable.

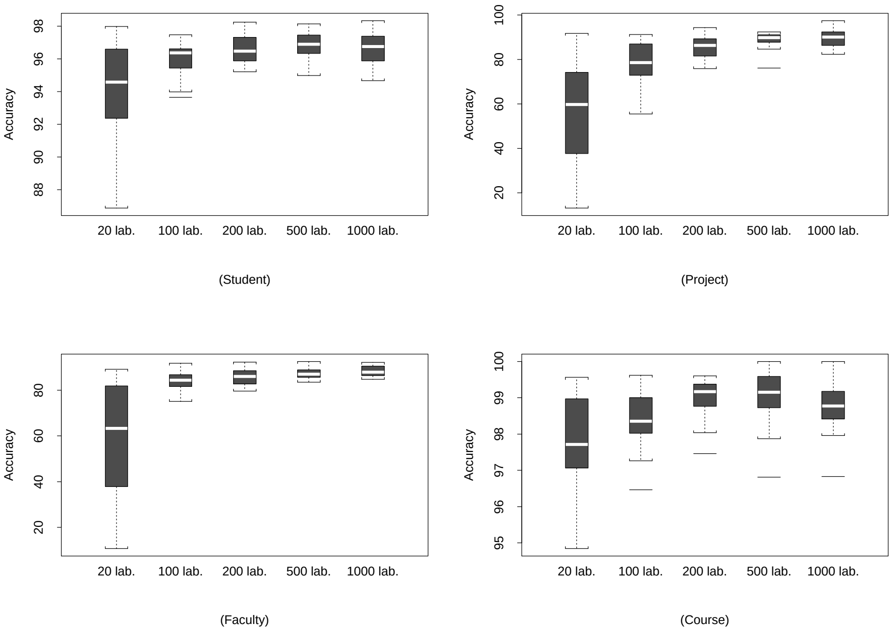


Figure 4: Mean group accuracies for the polytomous classification with no unlabeled data

5.3 Effect of the sampling mechanism of the unlabeled data

In many circumstances we can obtain unlabeled pages easily. The problem is describing the population from which they are drawn. Moreover, the sampling is not always probabilistic, thus groups aree sampled in proportion different from the population of interest. This fact clearly affects the estimation of the probabilities of each group at each iteration step. We would like to explore how the poor estimation of the group probability affects the accuracy of the classifier. Remember that we have the following two sampling schemes in our web page database (see Chapter 2): around 1,000 pages correspond to *all* web pages corresponding to four Computer Sciences Departments; the remaining pages (around 3,000) correspond to some of pages of several CS Departments, but they were drawn according to no probabilistic sampling. The estimated group probabilities from the first scheme are given by: 0.53 (“Student”), 0.08 (“Research Project”), 0.15 (“Faculty”), 0.24 (“Course”). The estimated group probabilities for the second scheme are: 0.34 (“Student”), 0.13 (“Research Project”), 0.31 (“Faculty”), 0.22 (“Course”).

We have simulated data as before, but we sampled the unlabeled data from a mixture of multinomials with group probabilities given by 0.34 (“Student”), 0.13 (“Research Project”), 0.31 (“Faculty”), 0.22 (“Course”). The results are shown in Figure 5.

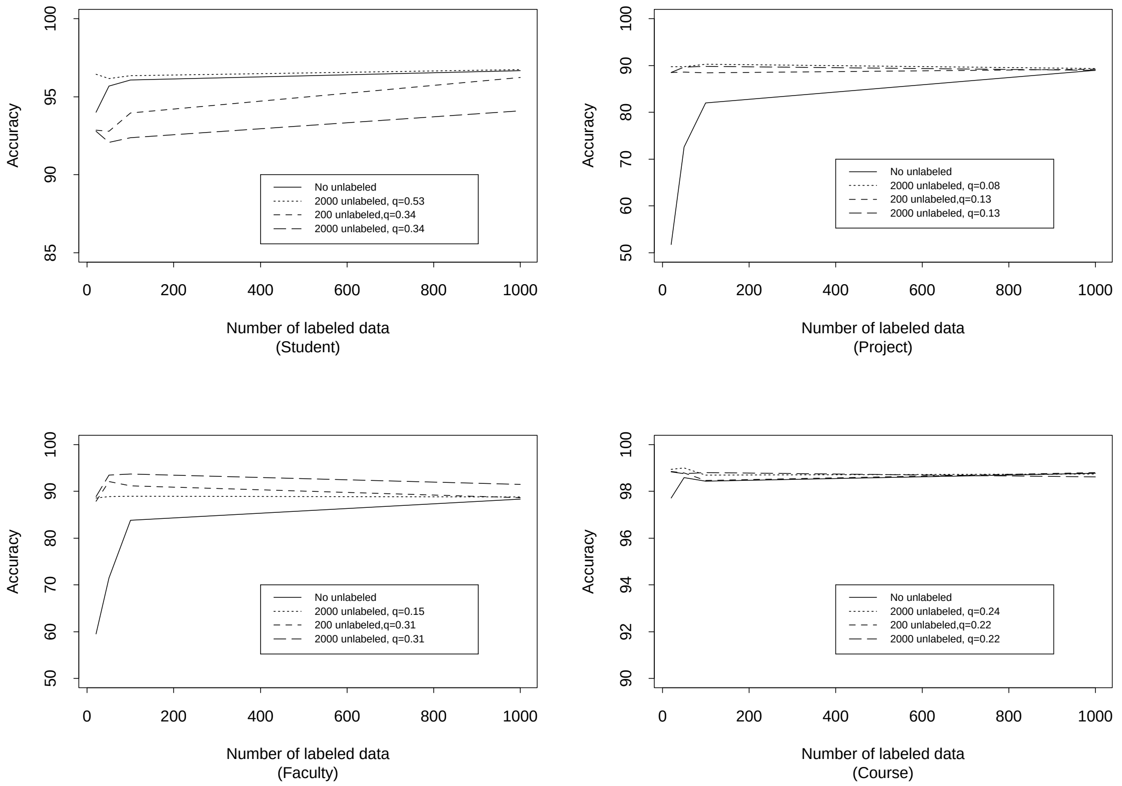


Figure 5: Mean group accuracies for the polytomous classification,
when the sampling of unlabeled data is not probabilistic

Figure 5 shows that “Student” web pages web pages lose some accuracy when the unlabeled data are sampled from a pool where the group probability is lower than the real; however, the change is not significant (around -2 or -3 percent). Of course, the situation is worse as the number of unlabeled pages increases. “Course” accuracy remains approximately constant. “Research Project” and “Faculty” accuracy improve with respect to the correct probabilistic sampling. For “Project” web pages, it does not matter whether the unlabeled data come from probabilistic sampling or not, but we must be aware that the “incorrect” group probability (0.13) is not too far from the true one (0.08). As we noted above, “Faculty” web page improves when the group probability is not the true one, and also this benefit is better as the number of unlabeled items increases.

Note that the “incorrect” group probability for “Faculty” is around twice the true one, whereas the “Faculty” incorrect group probability is 60 percent more that the true one. Also observe that the “Student” incorrect group probability is around 60 percent less than the true one and this group loses accuracy.

6 Conclusions

For this example with 57 feature words we found that the classifier based on 20 labeled plus 1,000 unlabeled observations works as efficiently as one based on 500 labeled observations. Thus we can economize in situations where labelling is expensive as long as we have an appropriate population from which to draw the unlabeled data. Also, this naive Bayes appraoch does not improve its accuracy beyond 500 labeled observations; that is, we will not gain accuracy by adding more and more labeled examples. A classifier that combines labeled and unlabeled data through the EM algorithm, do not improve its accuracy beyond 200 labeled observations.It appears that when the group probability in the unlabeled data is higher than the true one, the corresponding group accuracy increases. The reverse is also true, that is, when the group probability in the unlabeled data is lower than the true one, the corresponding group accuracy decreases.

Several questions arise from the empirical results, like: Is there a maximum amount of labeled data for which the EM algorithm with unlabeled data does not improve beyond this number?; Do the results depend on the percentage of unlabeled documents instead of their absolute value?; How much do the results change when the dimensionality changes?; Does the sampling mechanism of the unlabeled data affect the results? As we indicated in Section 4, the parameter estimates can be obtained explicitly at each iteration step; however, it is hard to see why those estimates should converge in a situation such as this. This is the reason why theoretical answers are difficult to obtain.

References

Blum, A. L. and Mitchell, T. M. (1998), Combining labeled and unlabeled data with co-training, in *Proceedings of the 1998 National Conference on Artificial Intelligence*.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T. M., Nigam, K. P., and Slattery, S. T. (1998), Learning to extract knowledge from the world wide web, Technical Report 98–122, School of Computer Science, Carnegie Mellon University.

Domingos, P. and Pazzani, M. J. (1997), On the optimality of the simple Bayesian classifier under zero-one loss, *Machine Learning* **29**, 103–130.

Flury, B. (1997), *A First Course in Multivariate Statistics*, Springer-Verlag, New York.

Langley, P., Iba, W., and Thompson, K. (1992), An analysis of bayesian classifiers, in *Proceedings of the 1992 National Conference on Artificial Intelligence*.

McCallum, A. and Nigam, K. P. (1998), A comparison of event models for naive Bayes text classification, in *AAAI/ICML-98 Workshop on Learning for Text Categorization*.

McLachlan, G. J. (1992), *Discriminant Analysis and Statistical Pattern Recognition*, Wiley, New York.

Mitchell, T. M. (1999), The role of unlabeled data in supervised learning, in *Proceedings of the Sixth International Colloquium on Cognitive Science, San Sebastian, Spain*.

Nigam, K. P., McCallum, A., Thrun, S., and Mitchell, T. M. (2000), Text classification from labeled and unlabeled documents using EM, *Machine Learning* **39**, 103–134.

Salvatierra, S. M. (1999), Learning text from the web: An application of classification methods, Technical Report 696, Department of Statistics, Carnegie Mellon University.

Tanner, M. (1996), *Tools for Statistical Inference*, Springer-Verlag.