Semi-Supervised Learning Using Prior Probabilities and EM

Rebecca Bruce
University of North Carolina at Asheville
bruce@cs.unca.edu

Abstract

This paper presents a Bayesian approach to semi-supervised learning for the purpose of developing classifiers. In this approach, the prior distribution of the parameters of a probability model is formulated from a small amount of tagged data via a stochastic inferencing process and expressed in terms of pseudo-counts. These pseudo-counts together with the counts in the untagged data form the posterior distribution which is used to estimate the model parameters via the EM algorithm. This procedure is used to develop classifiers for three different data sets. In all data sets, the average performance of classifiers developed using priors is superior to the performance of comparable classifiers developed without using priors.

1 Introduction

Recently, there have been several new learning algorithms that use a combination of tagged and untagged data; this is referred to as *semi-supervised* learning. The motivation for semi-supervised learning is that tagged data is expensive while untagged data can usually be acquired cheaply. The idea behind semi-supervised learning is to exploit the tagged data to acquire information about the problem and then use that information to guide learning from the untagged data.

This paper presents a Bayesian approach to semi-supervised learning. In this approach, the parameters of a probability model are estimated using Bayesian techniques and then used to perform classification. The prior probability distribution, which is expressed in term of *pseudo-counts*, is formulated from the tagged data via a stochastic inference process. Intuitively, the inference process starts with the counts in a small tagged data sample and adds new counts that are similar to those that would be expected in a larger tagged data sample from the same population. The pseudo-counts forming the prior distribution are combined with the counts in the untagged data to form the posterior distribution and then the model parameters are estimated from the counts in the posterior distribution using the EM algorithm. This paper presents the theoretical background for a Bayesian framework utilizing pseudo-counts and demonstrates that it is an effective approach for using a small amount of tagged data in combination with untagged data. Further, the algorithm presented in section 4 for generating prior distributions expressed as pseudo-counts from a small sample of tagged data is shown to produce classifiers with high accuracy across a range of data sets.

This paper is organized as follows. We begin with a discussion of related work in Section 2. Section 3 presents theoretical background for this approach, section 4 describes the procedure used to formulate a prior probability distribution of pseudo-counts, section 5 presents the experimental setup, and section 6 presents and discusses the experimental results.

2 Related Work

Many different approaches to semi-supervised learning have been studied recently. *Bootstrapping* methodologies are a type of semi-supervised learning procedure. Typically, bootstrapping consists of iterative clustering in which the initial or seed set of tagged data is enlarged on each iteration. In each iteration, tags are assigned to the unlabeled data via a classification system trained on the existing tagged data. The most reliable of these tags is retained and added to the tagged data used by the classification system in the next iteration. Recent examples of bootstrapping methodologies can be found in Riloff & Jones (1999), Jones et al. (1999), and Agichtein et al. (2000).

Co-training (Blum & Mitchell 1998) can also be considered a type of bootstrapping. In co-training, two classifiers are defined using distinct feature sets. Co-training then consists of iteratively using the output of one classifier to train the other. Variations of co-training have been applied to many application areas. Collins and Singer (1999) apply several variations of co-training to the name-entity recognition problem. Goldman & Zhou (2000) perform co-training using overlapping feature sets but distinct classification algorithms. This is also discussed in Nigam & Ghani (2000b). Blum & Mitchell (1998) have applied co-training to classifying web pages, and Yarowsky (1995) describes a co-training approach to word-sense disambiguation.

The approach discussed in this paper is most similar to the body of work described in Nigam et al. (2000) and Nigam et al. (1998). In these works, the EM algorithm is used to estimate the parameters of the Naive Bayes model from both tagged and untagged data. Here, the EM algorithm is also used to estimate model parameters from both tagged and untagged data. But, in contrast to the work by Nigam et al., the tagged data is used to formulate an informative prior distribution of pseudo-counts, and these pseudo-counts are combined with the counts in the untagged data to formulate the posterior distribution for the model parameters. Once the posterior distribution is formed, this approach is again similar to that described in Nigam et al. (2000) in that the model parameters are estimated from the posterior distribution using the EM algorithm.

Both Nigam & Ghani (2000a) and McCallum & Nigam (1998) present extensions to semi-supervised learning using the EM algorithm that are not discussed in this paper. There are also other learning paradigms in which semi-supervised algorithms have been developed. These include *active learning* (e.g., Dagan & Engelson 1995, Liere & Tadepalli 1997), and classification via support vector machines (e.g., Joachims 1999, Zhang 1999).

3 Background

3.1 The Likelihood Function

This section presents the theoretical basis for a Bayesian approach using pseudo-counts. We consider the special case of discrete data modeled using decomposable models. Further, we describe the use of the EM algorithm for estimating model parameters under these circumstances.

In this section, disambiguation of the meaning of "interest" is used as an example. For simplicity, assume that each instance of "interest" is described by the values of three random variables: S representing the sense tag, and $F1$ and $F2$ representing contextual features. If there are 6 senses of "interest" and 20 possible values for each of the contextual features, then there are 2,400 ($6 \times 20 \times 20$) possible configurations of these variables.

Let $x_{(s_i,f1_j,f2_k)}$ represent the frequency (i.e., the count) and $\theta_{(s_i,f1_j,f2_k)}$ the probability of a specific configuration of the 3 variables. The counts, $(x_{(s_1,f1_1,f2_1)},\ldots,x_{(s_6,f1_{20},f2_{20})})$, have a *multinomial distribution* with 2401 parameters, $(N,\theta_{(s_1,f1_1,f2_1)},\ldots,\theta_{(s_6,f1_{20},f2_{20})})$, where N is the size of the data sample:

$$P(x|\theta) \quad \propto \quad \prod_{i=1}^{6}\prod_{j=1}^{20}\prod_{k=1}^{20}\theta_{(s_i,f1_j,f2_k)}^{x_{(s_i,f1_j,f2_k)}}$$

where $\sum_{i=1}^{6}\sum_{j=1}^{20}\sum_{k=1}^{20}\theta_{(s_i,f1_j,f2_k)} = 1$.

The equation above is the sampling distribution; it as called the *likelihood function* when regarded as a function of θ for fixed counts, x. The values of $\theta_{(s_i,f1_j,f2_k)}$ that maximize this function are referred to as the *maximum likelihood estimates* of θ (Mood et al. 1974): $\hat{\theta}_{(s_i,f1_j,f2_k)} = \frac{x_{(s_i,f1_j,f2_k)}}{N}$.

Probability models may be formed that represent the likelihood function as a product of conditional distributions resulting from conditional independence constraints. Decomposable models (Wermuth & Lauritzen 1983; Darroch et al. 1980) are such models. If, for example, we assume that $F1$ and $F2$ are conditionally independent given the value of S, then the likelihood function can be re-parameterized[1]. In other words, we can express each $\theta_{(s_i,f1_j,f2_k)}$ as the product of parameters describing the marginal distributions of just the interdependent variables.

$$\theta_{(s_i,f1_j,f2_k)} = \frac{\theta_{(s_i,f1_j)} \times \theta_{(s_i,f2_k)}}{\theta_{(s_i)}}$$

where:

$$\theta_{(s_i,f1_j)} = \sum_k \theta_{(s_i,f1_j,f2_k)} = \sum_k P(s_i,f1_j,f2_k)$$

$$\theta_{(s_i,f2_k)} = \sum_j \theta_{(s_i,f1_j,f2_k)} = \sum_j P(s_i,f1_j,f2_k)$$

$$\theta_{(s_i)} = \sum_{j,k} \theta_{(s_i,f1_j,f2_k)} = \sum_{j,k} P(s_i,f1_j,f2_k)$$

The theory supporting the formulation of decomposable models also specifies the *sufficient statistics* (functions $x_{(s_i,f1_j,f2_k)}$) for estimating the new model parameters. These sufficient statistics are the sample counts from the highest-order marginals distributions composed of only interdependent variables. Maximum likelihood estimates of these parameters can be formulated directly from the sufficient statistics as shown below for the current example:

$$\hat{\theta}_{(s_i,f1_j)} = \frac{x_{(s_i,f1_j)}}{N} = \frac{\sum_k x_{(s_i,f1_j,f2_k)}}{N}$$

$$\hat{\theta}_{(s_i,f2_k)} = \frac{x_{(s_i,f2_k)}}{N} = \frac{\sum_j x_{(s_i,f1_j,f2_k)}}{N}$$

$$\hat{\theta}_{(s_i)} = \frac{x_{(s_i)}}{N} = \frac{\sum_{j,k} x_{(s_i,f1_j,f2_k)}}{N}$$

The existence of this kind of closed-form expression for the model parameters in terms of their sufficient statistics is a unique property of decomposable models (Whittaker 1990; Pearl 1988). Familiar examples of decomposable models are the Naive Bayes (the model formulated in section 3.1) and n-gram models. For a more complete definition of decomposable models see Bruce & Wiebe 1999; Whittaker 1990; Pearl 1988; Wermuth & Lauritzen 1983.

[1] $F1$ and $F2$ are conditionally independent given S if $P(f1_j|f2_k,s_i) = P(f1_j|s_i)$.

3.2 The Likelihood Function and the Prior Distribution

Using notation introduced in the previous section, the conjugate prior distribution for data having a multinomial sampling distribution is:

$$P(\theta|\alpha) \propto \prod_{i=1}^{6}\prod_{j=1}^{20}\prod_{k=1}^{20} \theta_{(s_i,f1_j,f2_k)}^{\alpha_{(s_i,f1_j,f2_k)}-1}$$

where the θ's must be non-negative and $\sum_{i=1}^{6}\sum_{j=1}^{20}\sum_{k=1}^{20}\theta_{(s_i,f1_j,f2_k)} = 1$.

The distribution above is the *Dirichlet* and it is used to express *prior* knowledge of the values of the model parameters. The prior distribution and the likelihood function can be combined using Bayes law to form the *posterior distribution*. If the prior distribution is Dirichlet with parameters α, as shown above, then the posterior distribution is Dirichlet with parameters $x_{(s_i,f1_j,f2_k)} + \alpha_{(s_i,f1_j,f2_k)}$ (Bishop et al. 1975):

The prior distribution can be interpreted as containing information equivalent to $N_\alpha = \sum_{i=1}^{6}\sum_{j=1}^{20}\sum_{k=1}^{20}\alpha_{(s_i,f1_j,f2_k)}$ *pseudo-counts* with $\alpha_{(s_i,f1_j,f2_k)}$ pseudo-counts for the configuration of variables $s_i, f1_j, f2_k$. In keeping with this view, the prior mean of $\theta_{(s_i,f1_j,f2_k)}$ is $\frac{\alpha_{(s_i,f1_j,f2_k)}}{N_\alpha}$, and the posterior mean of $\theta_{(s_i,f1_j,f2_k)}$ is:

$$\theta_{(s_i,f1_j,f2_k)} = \frac{x_{(s_i,f1_j,f2_k)} + \alpha_{(s_i,f1_j,f2_k)}}{N + N_\alpha}$$

Hence, the posterior mean of $\theta_{(s_i,f1_j,f2_k)}$ is a weighted average of the prior mean of $\theta_{(s_i,f1_j,f2_k)}$ and the maximum likelihood estimate of $\theta_{(s_i,f1_j,f2_k)}$ with weights proportional to the observed sample size (N) and the "prior sample size" (N_α).

It is convenient to choose the prior distribution by choosing the values of N_α and $\frac{\alpha_{(s_i,f1_j,f2_k)}}{N_\alpha}$ (the prior mean of $\theta_{(s_i,f1_j,f2_k)}$). The prior means should be chosen to give the most likely multinomial distribution, and the prior sample size should be chosen to represent the strength of the belief in the accuracy of the prior means in representing the true multinomial distribution.

Dawid and Lauritzen (1993) have shown that prior distributions for the parameters of decomposable models may be derived from the vector α (the prior for the full joint distribution of variables $S, F1, F2$). For the decomposable model introduced in section 3.1, the prior distributions for the model parameters can be formulated in terms of $\alpha_{(s_i,f1_j,f2_k)}$ as follows:

$$\alpha_{(s_i,f1_j)} = \sum_k \alpha_{(s_i,f1_j,f2_k)}$$

$$\alpha_{(s_i,f2_k)} = \sum_j \alpha_{(s_i,f1_j,f2_k)}$$

$$\alpha_{(s_i)} = \sum_{j,k} \alpha_{(s_i,f1_j,f2_k)}$$

3.3 EM Algorithm

The EM algorithm for the exponential family of probabilistic models (of which decomposable models are a subset) was introduced in Dempster et al. (1977) as a procedure for estimating the parameter values that maximize the likelihood function when there are missing data. In this section, we use the example introduced in section 3.1 and assume that the values of S are missing.

There are two steps in the EM algorithm, expectation (E-step) and maximization (M-step). The E-step calculates the expected values of the sufficient statistics given the current parameter estimates. The M-step makes maximum likelihood estimates of the parameters given the estimated values of the sufficient statistics. Starting with a randomly generated set of parameter estimates, these steps alternate until the parameter estimates in iteration $k-1$ and k differ by less than ϵ or until a predefined maximum number of iterations has been exceeded.

For decomposable models, the formulation of the E-step is considerably simplified (Lauritzen 1995). For example, for the model introduced in section 3.1, the EM algorithm proceeds as follows:

---

1. randomly initialize all $\theta_{(s_i,f1_j)}$ and $\theta_{(s_i,f2_k)}$, and set $k = 1$
2. E-step: $x_{(s_i,f1_j)} = \frac{\theta_{(s_i,f1_j)}}{\theta_{f1_j}} \times x_{f1_j}$
   $x_{(s_i,f2_k)} = \frac{\theta_{(s_i,f2_k)}}{\theta_{f2_k}} \times x_{f2_k}$
3. M-step: re–estimate $\theta_{(s_i,f1_j)} = \frac{x_{(s_i,f1_j)}}{N}$
   $\theta_{(s_i,f2_k)} = \frac{x_{(s_i,f2_k)}}{N}$
4. $k = k + 1$
5. go to step 2 if the parameter estimates from iteration k and $k-1$ differ by more than ϵ and $k < \beta$ where β is some pre-defined stopping point.

The procedure above is appropriate for maximizing the likelihood function, but the EM algorithm can also be used to maximize the posterior distribution. In this case, the algorithm is modified to include the pseudo-counts forming the prior distribution. The E-step computes:

$$count_{(s_i,f1_j)} = \alpha_{(s_i,f1_j)} + \frac{\theta_{(s_i,f1_j)}}{\theta_{f1_j}} \times x_{f1_j}$$

$$count_{(s_i,f2_k)} = \alpha_{(s_i,f2_k)} + \frac{\theta_{(s_i,f2_k)}}{\theta_{f2_k}} \times x_{f2_k}$$

and the M-step computes new parameter estimates based on the latest estimate of the expected counts as follows:

$$\theta_{(s_i,f1_j)} = \frac{x_{(s_i,f1_j)} + \alpha_{(s_i,f1_j)}}{N + N_\alpha}$$

$$\theta_{(s_i,f2_k)} = \frac{x_{(s_i,f2_k)} + \alpha_{(s_i,f2_k)}}{N + N_\alpha}$$

The EM algorithm is guaranteed to converge (Dempster et al. 1977) under normal conditions[2]. However, if the data is sparse and skewed, or if estimates of the posterior mean do not exist, i.e., $x_{(s_i,f1_j,f2_k)}+\alpha_{(s_i,f1_j,f2_k)} < 1$ for some (i,j,k), then the EM algorithm is not guaranteed to be well behaved.

4 Formulating Prior Probabilities using Simulated Pseudo-Counts

As we have seen in the previous sections, prior probabilities expressed as pseudo-counts are easy to use when estimating the values of model parameters, even in the presence of missing data. But, how is it possible to simultaneously generate a large number of consistent pseudo-counts that provide reasonable estimates of the model parameters? In this paper, a *semi-Bayes* (Bishop 1975) approach is taken in that the prior distribution is estimated based on the data themselves.

The procedure is fully automatic and it makes use of an algorithm developed by Patefield (1981) for simulating the multiple hypergeometric distribution. The multiple hypergeometric distribution describes the counts that arise in bi-variate data where the two variables (of arbitrary range) are statistically independent of each other, and the marginal totals for each variable are fixed. Such data can be represented in a 2-dimensional *contingency table*. In the case of a 2-dimensional contingency table, the marginal totals are the row and column totals and the internal cell counts have a multiple hypergeometric distribution. It is important to note that the distribution of counts for two variables that are conditionally independent (such as $F1$ and $F2$ in the example from section 3.1) can also be described by the multiple hypergeometric distribution. In this case, the 2-dimensional contingency table represents the distribution of the two variables given the fixed values of those variables in the conditioning set (S in the example from section 3.1).

Intuitively, the procedure for formulating priors takes a small amount of fully tagged data and adds to it new counts that are similar to the counts we would expect to find in a larger data sample drawn from the same population. Data generation is a sequential procedure in which each pair of variables, in turn, is treated as conditionally independent given the values of the remaining variables. Patefield's algorithm is used to generate 2-dimensional tables of counts for those variables where the marginal totals in each table are proportional to the distribution of the variable values in the tagged data sample. The exact number of 2-dimensional tables generated is fixed in advance, and only those tables of counts that are distributionally most similar to the actual tagged data are kept; these counts are the pseudo-counts generated by the procedure.

For the example introduced in section 3.1, the algorithm is as follows. Generate all pairs of variables, $(S,F1)$, $(S,F2)$, $(F1,F2)$. For each pair of variables do the following letting X represent the current pair and Y represent the remaining variables (e.g., on the first iteration, $X=(S,F1)$ and $Y=F2$).

- For each combination of the values of the variables in Y found in the tagged data, do the following:
  1. Calculate the G^2 value measuring the fit of the model for conditional independence between the variables in X given the values of the variables in Y using tagged data. Call this G^2 value G^2_{data}
  2. Define the row and column totals for a 2-dimensional table representing the values of the variables in X. For $X=(S,F1)$, the row totals are defined as follows:
     $\sum_{row_1} = k \times (\beta + \sum_{(j,k)} x_{(s_1,f1_j,f2_k)})$
     $\sum_{row_2} = k \times (\beta + \sum_{(j,k)} x_{(s_2,f1_j,f2_k)})$
     ...
     $\sum_{row_6} = k \times (\beta + \sum_{(j,k)} x_{(s_6,f1_j,f2_k)})$
     And, the column totals are defined as:
     $\sum_{col_1} = k \times (\beta + \sum_{(i,k)} x_{(s_i,f1_1,f2_k)})$
     $\sum_{col_2} = k \times (\beta + \sum_{(i,k)} x_{(s_i,f1_2,f2_k)})$
     ...
     $\sum_{col_{20}} = k \times (\beta + \sum_{(i,k)} x_{(s_i,f1_{20},f2_k)})$
     where k is some constant and β is a default count set equal to 6 in these experiments.
  3. Use Patefield's algorithm to generate η 2-dimensional tables having the row and column totals defined above. η is a pre-defined constant set equal to 50 in these experiments.
  4. For each of the η tables generated above, calculate the G^2 value measuring the fit of the model for conditional independence (between the variables in X given the values of the variables in Y) using the data in the table.
  5. Save only γ% of the η tables, where the tables saved have G^2 values that are the closest to G^2_{data} (i.e., the G^2 value of the actual data). γ is a pre-defined constant set equal to 10% in these experiments.
  6. Add the counts in the tables saved above to the set of pseudo-counts representing the prior distribution.

The final step in the algorithm is to scale the counts from the tagged data and combine them with the pseudo-counts generated as described above. The scaling factor is chosen to assure that the observed data have more influence on the formulation of the model parameters estimates than do the simulated pseudo-counts.

5 The Experiments

5.1 The Data

Three different data set are used in this evaluation. The first data set is the same as that described in Bruce & Wiebe (1994). The data, distributed by the CLR (ftp site: clr.nmsu.edu), consists of 2,369 sentences from the

[2]The EM algorithm is only guaranteed to find a local maximum.

ACL/DCI Wall Street Journal corpus that contain the noun "interest". Each instance of "interest" is hand-tagged with one of the six senses defined in the Longman Dictionary of Contemporary English (LDOCE). The distribution of senses is presented in Table 1. Each sentence containing interest is represented by the features described in Table 2.

LDOCE Sense	Total Sample	Test Sample
sense 1: readiness to give attention	361 (15%)	90 (15%)
sense 2: quality of causing attention to be given	11 (<1%)	2 (<1%)
sense 3: activity, subject, etc., which one gives time and attention to	66 (3%)	16 (3%)
sense 4: advantage, advancement, or favor	178 (8%)	48 (8%)
sense 5: a share (in a company, business, etc.)	500 (21%)	122 (20%)
sense 6: money paid for the use of money	1253 (53%)	322 (54%)

Table 1: Distribution of Sense Tags

Symbol	Description	Possible Values
e	suffix of interest	"s" or *null*
r1	word 1 position right of interest	all words $r1$ where $f_{r1} > 5$ or *null*
r1	word 2 positions right of interest	all words $r2$ where $f_{r2} > 5$ or *null*
l1	word 1 position left of interest	all words $l1$ where $f_{l1} > 5$ or *null*
l1	word 2 positions left of interest	all words $l2$ where $f_{l2} > 5$ or *null*

Table 2: "interest" Data Features

The remaining two data sets were taken from the UCI Machine Learning Repository (http://www.ics.uci.edu/∼mlearn/MLRepository.html). The first of these, named *Internet Advertisements*, contains the descriptions of 3,279 images taken from Internet pages. Each image is given a binary classification indicating that it is either an advertisement ("ad") or not ("nonad"). The classification distribution is 2821 nonads, and 458 ads.

Each image in the ad data set is represented by 1558 features. The features encode the geometry of the image as well as phrases occurring in the URL, the image's URL and alt text, the anchor text, and words occurring near the anchor text. Of the 1558 features, 3 are continuous and the others are binary. To reduce the computation time required for the analysis, only the classification feature and the 23 most frequently occurring ($f > 140$) binary features are used in this study. A list of the specific features used in this study can be found in the on-line appendix (http://www.cs.unca.edu/∼bruce/textbeyond/appendix).

The final data set, named *Spambase Database*, describes 4,601 e-mail messages. Each message is classified as being either "spam" (i.e., unsolicited commercial e-mail) or "nonspam" and is represented by 57 continuous features. Most of the features indicate whether a particular word or character is frequently occurring in the e-mail. There are also three "run-length" features measuring the length of sequences of consecutive capital letters. The ratio of "spam" to "nonspam" messages is 1813 (39.4%) to 2788 (60.6%).

Again, to reduce computation time, only 12 of the 57 features plus the classification feature are used in this study. Additionally, all features are converted to a discrete-valued representation in order to allow the formulation of discrete priors as described in this paper. The conversion is based on the deviation of each feature from its average value as described in Table 3.

Fea. #	Description	Possible Values, where $\alpha_{(i,<item>)} = f_{avg_{<item>}} +/- (i \times f_{std_{<item>}})$
f16 f24 f27	$f_{<word>} = f_{free}$ $f_{<word>} = f_{money}$ $f_{<word>} = f_{george}$	if $f_{<word>}$ bounded by $\alpha_{(1,<word>)}$ then 0 else 1
f49 f50 f51 f52 f54	$f_{<char>} = f_{;}$ $f_{<char>} = f_{)}$ $f_{<char>} = f_{]}$ $f_{<char>} = f_{!}$ $f_{<char>} = f_{\#}$	if $f_{<char>}$ bounded by $\alpha_{(1/4,<char>)}$ then 0 else if $f_{<char>}$ bounded by $\alpha_{(1/2,<char>)}$ then 1 else if $f_{<char>}$ bounded by $\alpha_{(3/4,<char>)}$ then 2 else 3
f55 f56 f57	$f_{CL<val>} = f_{CL<avg>}$ $f_{CL<val>} = f_{CL<max>}$ $f_{CL<val>} = f_{CL<tot>}$ (where CL stands for Capital Letter run length)	if $f_{CL<val>}$ bounded by $\alpha_{(1/4,CL<val>)}$ then 0 else if $f_{CL<val>}$ bounded by $\alpha_{(1/2,CL<val>)}$ then 1 else if $f_{CL<val>}$ bounded by $\alpha_{(3/4,CL<val>)}$ then 2 else 3

Table 3: Spam Data Features

5.2 Generating a Range of Models

For each data set, a range of models with varying complexities are considered. These models are generated via a model search (e.g., Pedersen & Bruce 1997; Wermuth & Lauritzen 1983) using tagged data. The search starts with the saturated model, the model in which all variables are considered to be interdependent, and ends with the Naive Bayes model. At each step of the search, a new conditional independence constraint is introduced to produce a new model. The exact conditional test (Pedersen et al.1996; Kreiner 1987), a kind of significance test, is used to select the most appropriate conditional independence constraint to introduce at each step of the search.

For the six variables in the "interest" data, a total of 11 models are generated during the search; each of these models is used in this study. For the ads data set, 253 models are generated during the search. Of these 253 models, a sample consisting of every 35th model generated by the search is selected for use in the study. For the spam data set, 66 models are generated during the search. Again, a sample of these models formed by selecting every 8th model generated by the search is used in the study. The models selected for each data set are presented in the on-line appendix (http://www.cs.unca.edu/∼bruce/textbeyond/appendix).

# model param.	all data (2,124 t 2,369 ut)	all tagged (2,124 t)	all untagged (2,369 ut)	mix 1+noninf. (708 t, 2,369 ut)	mix 1 (708 t, 2,369 ut)	mix 2 (472 t, 2,369 ut)	mix 3 (236 t, 2,369 ut)	mix 4 (118 t, 2,369 ut)
10,810,800	64%	66%	23%	66%	52%	48%	39%	30%
5,821,200	64%	66%	21%	66%	51%	48%	39%	30%
570,240	65%	71%	33%	62%	56%	50%	43%	34%
381,240	66%	71%	27%	63%	57%	50%	41%	32%
362,880	65%	71%	30%	62%	57%	50%	45%	34%
215,460	66%	72%	33%	62%	59%	55%	49%	41%
22,950	70%	78%	32%	60%	59%	54%	48%	39%
16,086	74%	80%	47%	66%	64%	58%	58%	47%
10,278	77%	84%	33%	64%	69%	66%	61%	49%
9,750	79%	84%	39%	64%	70%	66%	61%	52%
780 (N.B.)	81%	84%	47%	68%	69%	60%	61%	53%
Avg.	70%	75%	33%	64%	60%	55%	50%	40%

Figure 1: Classification Accuracy without Semi-Bayes Priors for "interest" Data

# model param.	all data (2,943 t 3,279 ut)	all tagged (2,943 t)	all untagged (3,279 ut)	mix 1+noninf. (327 t, 3,279 ut)	mix 1 (327 t, 3,279 ut)	mix 2 (163 t, 3,279 ut)	mix 3 (81 t, 3,279 ut)	mix 4 (40 t, 3,279 ut)	mix 5 (20 t, 3,279 ut)	mix 6 (10 t, 3,279 ut)
16,777,216	94%	92%	80%	89%	90%	88%	87%	86%	84%	84%
892,936	94%	91%	85%	91%	91%	89%	86%	86%	85%	85%
93,200	94%	90%	85%	91%	91%	90%	89%	87%	85%	85%
11,800	94%	92%	82%	92%	91%	89%	86%	87%	84%	86%
1,344	94%	92%	82%	92%	93%	88%	88%	86%	81%	83%
236	94%	88%	93%	93%	94%	93%	93%	93%	93%	93%
92 (N.B.)	72%	88%	69%	70%	70%	70%	70%	70%	70%	70%
Avg.	91%	90%	82%	88%	89%	87%	86%	85%	83%	84%

Figure 2: Classification Accuracy without Semi-Bayes Priors for Ad Data

6 Results

In this section, our purpose is to compare the performance of a wide range of classifiers formulated using semi-Bayes priors to comparable classifiers formulated without semi-Bayes priors. The formulation of a classifier for a particular data set depends on three factors: (1) the model, (2) the amount of tagged and untagged data used during parameter estimation, and (3) whether semi-Bayes priors are used during parameter estimation. The models are those developed during the model search described in section 5.2 above; each model is identified by the number of parameters it contains. All classifiers for a particular data set are tested on the same test set. One tenth of the available data in each data set is randomly selected to form that test set. In each data set, the data is organized into 3 overlapping groups: (1) all data (including the test set) with tags removed, referred to as the *untagged* data or *ut*; (2) the *test set* as defined above; and (3) varying amounts of tagged data randomly selected from the total data set minus the test set; this is referred to as *tagged* data or *t*.

We begin by discussing classifiers developed without using prior distributions. For each data set, 3 benchmark classifiers are formulated as follows: (1) via the EM algorithm using the untagged data in combination with all tagged data; (2) using all tagged data to formulate maximum likelihood estimates of the model parameters; and (3) via the EM algorithm using only the untagged data. The first classifier is the upper bound for classification using the EM algorithm without priors on a data set. The third classifier is the lower bound. The second classifier allows us to contrast semi-supervised learning with supervised learning. Figures 1, 2 and 3 present the performance of classifiers developed without using semi-Bayes prior distributions, these include the benchmark classifiers plus several classifiers developed using the EM algorithm and varying amounts of tagged data in combination with the untagged data. For each data set, there is also a classifier developed using an noninformative prior distribution obtained by setting all $\theta_{(s_i, f1_j, f2_k)} = 1$. In figures 1, 2 through 3, this classifier is denoted by the addition of the text "+noninf." to the column heading stating the amount of tagged and untagged data used in forming the classifier.

Figures 4, 5 and 6 present the performance of comparable classifiers developed using semi-Bayes priors. In developing these classifiers, the tagged data is used to formulate priors via the algorithm in section 4 as opposed to being input directly to the EM algorithm. Below is a list of some of the empirical observations that can be made from Figures 1 through 6.

The performance of classifiers developed using semi-Bayes priors is equal to or better than the performance of comparable classifiers developed using noninformative priors.

Due to space limitations, classification results using noninformative priors are presented for only one of the data mixes from each of figures 4, 5 and 6, but semi-Bayes priors compare favorably to noninformative priors for all data mixes investigated in these experiments. It is also interesting to note that use of noninformative priors degrades classifier performance on the spam data.

The average performance of classifiers developed using semi-Bayes priors is consistently better than the average performance of comparable classifiers developed without priors.

This can be verified by comparing the "Avg." rows of comparable tables. With very few exceptions (to be dis-

# model param.	all data (4,140 t 4,601 ut)	all tagged (4,140 t)	all untagged (4,601 ut)	mix 1+noninf. (920 t, 4,601 ut)	mix 1 (920 t, 4,601 ut)	mix 2 (640 t, 4,601 ut)	mix 3 (230 t, 4,601 ut)	mix 4 (115 t, 4,601 ut)	mix 5 (57 t, 4,601 ut)	mix 6 (28 t, 4,601 ut)
4,194,304	81%	78%	56%	70%	76%	73%	70%	67%	64%	61%
656,386	81%	78%	63%	71%	76%	74%	70%	69%	67%	64%
90,176	81%	78%	62%	74%	77%	74%	70%	67%	66%	63%
11,328	80%	79%	56%	75%	77%	71%	69%	61%	60%	59%
3,856	82%	81%	58%	75%	77%	73%	72%	72%	61%	57%
1,104	82%	81%	73%	78%	79%	77%	77%	76%	76%	76%
200	84%	83%	81%	78%	83%	82%	82%	82%	82%	82%
84 (N.B.)	82%	83%	74%	76%	79%	77%	77%	76%	75%	75%
Avg.	82%	80%	65%	75%	78%	75%	73%	71%	69%	67%

Figure 3: Classification Accuracy without Semi-Bayes Priors for Spam Data

# model param.	mix 1 (708 t, 2,369 ut)	mix 2 (472 t, 2,369 ut)	mix 3 (236 t, 2,369 ut)	mix 4 (118 t, 2,369 ut)	mix 5 (59 t, 2,369 ut)	mix 6 (28 t, 2,369 ut)	mix 7 (15 t, 2,369 ut)	mix 8 (8 t, 2,369 ut)
10,810,800	84%	80%	76%	70%	59%	60%	39%	33%
5,821,200	85%	79%	76%	70%	60%	60%	38%	33%
570,240	85%	80%	78%	74%	64%	61%	43%	32%
381,240	85%	80%	77%	72%	64%	59%	42%	32%
362,880	84%	80%	77%	73%	62%	61%	45%	31%
215,460	83%	80%	79%	73%	64%	61%	48%	34%
22,950	83%	80%	77%	75%	68%	64%	50%	41%
16,086	83%	84%	79%	78%	69%	66%	52%	44%
10,278	85%	85%	83%	78%	69%	65%	57%	42%
9,750	85%	83%	82%	78%	69%	65%	57%	44%
780 (N.B.)	83%	82%	81%	78%	69%	66%	58%	48%
Avg.	84%	81%	79%	74%	65%	63%	48%	38%

Figure 4: Classification Accuracy Using Semi-Bayes Priors for "interest" Data

cussed below), the same statement could be made regarding the performance of individual models.

With a small portion of randomly selected tagged data, it is possible to produce a classifier using a semi-Bayes prior that out performs a classifier trained on the entire data set (excluding the test set).

The amount of tagged data required to produce such a classifier varies across data sets. For the "interest" data, only 116 randomly selected tagged instances were required, while for the spam data, 920 tagged instances were needed. Although not apparent in the tables, for each data set, there is a maximum performance that can be achieved using semi-Bayes priors. Once this level of performance is reached, increasing the amount of tagged data does not improve classifier performance.

Semi-Bayes priors reduce the differences in performance among models of varying complexity.

This can be observed by comparing the columns of any 2 comparable tables, but of particularly interest is the case where semi-Bayes priors actually degrade the performance of a model. In both the ad and spam data sets, the second to the last model appears to fit the data exceedingly well. This is especially true in the ad data set where, in addition, the Naive Bayes model seems to have an particularly poor fit. The use of semi-Bayes priors tends to equalize the performance of these model. Note that the degradation of the good model's performance is small while the improvement to the poor model's performance is large.

The effectiveness of Semi-Bayes priors increases when the event space of the variables is large.

This can be seen by comparing the relative performance of the semi-Bayes priors across data sets. Performance is strongest for the "interest" data where the average event space for the variables contains 23 possible values, and weakest for the ad data where all variables are binary. It should also be noted that variables with large event spaces allow more performance variation in general.

Effective semi-Bayes priors can be formulated using very small amounts of randomly selected tagged data.

Note that in all cases, classifiers formulated using semi-Bayes priors exceed the performance of the lower bound classifier.

Semi-supervised learning without the use of priors is not consistently superior to conventional supervised learning.

This fact has also been noted in Nigam et al. (1999).

Acknowledgements

This research was supported by the Office of Naval Research under grant number N00014-95-1-0776.

References

Agichtein, E., Eskin, E. & Gravano, L. (2000). Combining Strategies for Extracting Relations from Text Collections. *Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2000)*, 2000.

Bishop, Y., Fienberg, S., & Holland, P. (1975). *Discrete Multivariate Analysis.* Cambridge, MA: MIT Press.

Blum, A., & Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the 1998 Conference on Computational Learning Theory*, July 1998.

# model param.	mix 1 (327 t, 3,279 ut)	mix 2 (163 t, 3,279 ut)	mix 3 (81 t, 3,279 ut)	mix 4 (40 t, 3,279 ut)	mix 5 (20 t, 3,279 ut)	mix 6 (10 t, 3,279 ut)	mix 7 (5 t, 3,279 ut)
16,777,216	91%	91%	89%	88%	89%	88%	84%
892,936	92%	92%	89%	89%	90%	89%	84%
93,200	92%	91%	88%	89%	89%	90%	87%
11,800	92%	92%	93%	91%	93%	90%	88%
1,344	92%	91%	92%	91%	93%	89%	89%
236	93%	92%	93%	91%	93%	90%	90%
92 (N.B.)	94%	94%	94%	92%	94%	88%	88%
Avg.	92%	92%	91%	90%	92%	89%	87%

Figure 5: Classification Accuracy Using Semi-Bayes Priors for Ad Data

# model param.	mix 1 (920 t, 4,601 ut)	mix 2 (640 t, 4,601 ut)	mix 3 (230 t, 4,601 ut)	mix 4 (115 t, 4,601 ut)	mix 5 (57 t, 4,601 ut)	mix 6 (28 t, 4,601 ut)	mix 7 (14 t, 4,601 ut)
4,194,304	82%	79%	79%	79%	75%	74%	70%
656,386	82%	79%	79%	79%	76%	74%	71%
90,176	81%	79%	79%	79%	76%	73%	69%
11,328	82%	79%	80%	80%	79%	75%	73%
3,856	82%	80%	80%	79%	77%	76%	75%
1,104	83%	81%	80%	79%	80%	77%	76%
200	84%	82%	81%	82%	82%	79%	75%
84 (N.B.)	84%	83%	81%	81%	82%	80%	79%
Avg.	83%	80%	80%	80%	78%	76%	74%

Figure 6: Classification Accuracy Using Semi-Bayes Priors for Spam Data

Bruce, R. & Wiebe, J. (1994). Word-Sense Disambiguation using Decomposable Models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, Las Cruces, NM, pp. 139-146.

Bruce, R. & Wiebe, J. (1999) Decomposable Modeling in Natural Language Processing. *Computational Linguistics*, 25(2):195-208.

Collins, M. & Singer, Y. (1999). Unsupervised Models for Named Entity Classification. In *Proceedings of the 1999 Joint Sigdat Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, June 21-22, 1999.

Dagan, I. & Engelson, S. (1995). Committee-based sampling for training probabilistic classifiers. *Proceedings of the 12th International Conference on Machine Learning*, pp.150-157, 1995.

Dawid, A., & Laurtizen, S. (1993). Hypermarkov Laws in the Statistical Analysis of Decomposable Models. *The Annals of Statistics*, 23(3):1272-1317.

Darroch, J., Lauritzen, S., & Speed, T. (1980). Markov Fields and Log-Linear Interaction Models for Contingency Tables. *The Annals of Statistics*, Vol. 8, No. 3, pp. 522-539.

Dempster, A., Laird, N. & Rubin, D. (1977). Maximum Likelihood From Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society B*, 39:1-38.

Goldman, S. & Zhou, Y. (2000). Enhancing Supervised Learning with Unlabeled Data. *Proceedings of the 17th International Conference on Machine Learning*, pp.327–334, 2000.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of the International Conference on Machine Learning*, 1999.

Jones, R., Nigam, K., McCallum, A. & Riloff, E. (1999). Bootstrapping for Text Learning Tasks. *Proceedings of the IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, 1999.

Kreiner, S. (1987). Analysis of Multidimensional Contingency Tables by Exact Conditional Tests: Techniques and Strategies. *Scandinavian Journal of Statistics*, 14:97-112.

Lauritzen, S. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191-201.

Liere, R. & Tadepalli, P. (1997). Active learning with committees for text categorization. *Proceedings of the 14th National Conference on Artificial Intelligence*, pp. 591-596, 1997.

Mood, A., Graybill, F., and Boes, D. (1974). *Introduction to the Theory of Statistics*. New York, N.Y.: McGraw-Hill Book Co.

McCallum, A. and Nigam, K. (1998). Employing EM and Pool-Based Active Learning for Text Classification. *Machine Learning: Proceedings of the Fifteenth International Conference (ICML '98)*, pp. 359-367. 1998.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3), pp. 103-134.

Nigam, K. & Ghani, R. (2000a). Analyzing the Effectiveness and Applicability of Co-training. *Proceeding of the Ninth International Conference on Information and Knowledge Management (CIKM-2000)*. 2000.

Nigam, K. & Ghani, R. (2000b). Understanding the Behavior of Co-training. *Proceeding of the KDD-2000 Workshop on Text Mining.* 2000.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, pp. 792–799, 1998.

Patefield, W. (1981). An Efficient Method of Generating Random R x C Tables with Given Row and Column Totals. *Journal of the Royal Statistical Society C* 30:91-97.

Pearl, Judea (1988). *Probabilistic Reasoning In Intelligent Systems: Neworks of Plausible Inference.* San Mateo, Ca.: Morgan Kaufmann.

Pedersen, T., Kayaalp, M., & Bruce, R. (1996). Significant Lexical Relationships. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, pp. 455-460.

Pedersen, T., Bruce, R., & Wiebe, J.. (1997). Sequential Model Selection for Word Sense Disambiguation. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*, Washington, DC.

Riloff, E. & Jones, R. (1999). Learning Dictionaries for Information Extraction by Multi-level Boot-strapping. *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pp. 1044-1049, 1999.

Wermuth, N. & Lauritzen, S. (1983). Graphical and recursive models for contingency tables. *Biometrika*, 70(3):537-52.

Whittaker, J. (1990) *Graphical Models in Applied Multivariate Statistics.* New York, NY: John Wiley.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Anual Meeting of the Association for Computational Linguistics.* Cambridge, MA. pp. 189-196.

Zang, T. (1999). Some asymptotic results concerning the value of unlabeled data. *Working Notes for the NIPS 99 Workshop on Using Unlabeled Data for Supervised Learning*, 1999.