A Comparison of Event Models for Naive Bayes Text Classification

Andrew McCallum[‡†]
mccallum@jprc.com

[‡]Just Research
4616 Henry Street
Pittsburgh, PA 15213

Kamal Nigam[†]
knigam@cs.cmu.edu

[†]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Recent work in text categorization has used two different first-order probabilistic models for classification, both of which make the *naive Bayes assumption.* Some use a multi-dimensional Bernoulli model, that is, a Bayesian Network with no dependencies between words and binary word counts [*e.g.* Larkey and Croft 1996; Koller and Sahami 1997]. Others use a multinomial model, that is, a uni-gram language model with integer word counts [*e.g.* Lewis and Gale 1994; Mitchell 1997].

This paper aims to clarify the confusion by describing the differences and details of these two models, and by empirically comparing their classification performance on five text corpora. We find that the multi-Bernoulli performs best with small vocabulary sizes, but that the multinomial performs best at larger vocabulary sizes—an average of 27% better than the binomial model at any vocabulary size.

Introduction

Simple Bayesian classifiers have been gaining popularity lately, and have been found to perform surprisingly well [Friedman *et al.* 1997; Friedman 1997; Sahami 1996; Langley *et al.* 1992]. These probabilistic approaches begin by making strong assumptions about how the data is generated, and posit a probabilistic model that embodies these assumptions; then they use a collection of labeled training examples to estimate the parameters of the generative model. Classification on new examples is performed with Bayes rule by selecting the class that is most likely to have generated the example.

The *naive Bayes* classifier is the simplest of these models, in that it assumes that all attributes of the examples are independent of each other given the context of the class. This is the so-called "naive Bayes assumption." While the assumption is clearly false in most real-world tasks, naive Bayes often performs classification very well. This paradox is explained by the fact that classification estimation is only a function of the sign (in binary cases) of the function estimation; the function approximation can still be poor while classification accuracy remains high [Domingos and Pazzani 1997; Friedman 1997]. Because of the independence assumption, the parameters for each attribute can be learned separately, and this greatly simplifies learning, especially when the number of attributes is large.

Document classification is just such a domain with a large number of attributes. The attributes of the examples to be classified are words, and the number of different words can be quite large indeed. While some simple document classification tasks can be accurately performed with vocabulary sizes less than one hundred, many complex tasks on real-world data from the Web, UseNet and newswire articles do best with vocabulary sizes in the thousands. Naive Bayes has been successfully applied to document classification in many research efforts; (references are below).

Despite its popularity, there has been some confusion in the document classification community about the "naive Bayes" classifier because there are two *different* generative models in common use, both of which make the "naive Bayes assumption." Both are sometimes called "naive Bayes" by their practitioners.

One model specifies that a document is represented by a vector of binary attributes indicating which words occur and do not occur in the document. The number of times a word occurs is not captured. When calculating the probability of a document, one multiplies the probability of all the attribute values, including the probability of non-occurrence for words that do not occur in the document. Here we can understand the document to be the "event," and the absence or presence of words to be attributes of the event. This describes a multi-variate Bernoulli distribution; we will call this the *multi-Bernoulli* event model. This approach is more traditional in the field of Bayesian networks, and is generally traditional in tasks that have a relatively small number of fixed attributes, such as the UCI repository data sets. The approach has been used for text classification by numerous people [Robertson and Sparck-Jones 1976; Lewis 1992; Kalt and Croft 1996; Larkey and Croft 1996; Koller and Sahami 1997].

The second model specifies that a document is represented by the set of word occurrences from the document. As above, the order of the words is lost, however, the number of occurrences of each word in the document is captured. When calculating the probability of a document, one multiplies the probability of the words that occur. Here we can understand the individual word occurrences to be the "events" and the document to be the collection of word events. We call this the *multinomial* event model. This approach is more traditional in statistical language modeling for speech recognition, where it would be called a "uni-gram language model." This approach has also been used for text classification by numerous people [Lewis and Gale 1994; Kalt and Croft 1996; Joachims 1997; Li and Yamanishi 1997; Mitchell 1997; Nigam *et al.* 1998; McCallum *et al.* 1998].

This paper aims to clarify the confusion between these two approaches by explaining both models in detail. We present an extensive empirical comparison on five corpora, including Web pages, UseNet articles and Reuters newswire articles. Our results indicate that the binomial model sometimes performs better than the multinomial at small vocabulary sizes. However, the multinomial usually out-performs the binomial at large vocabulary sizes, and almost always beats the binomial when vocabulary size is chosen optimally for both. While sometimes the difference in performance is not great, on average across data sets, the multinomial incurs 27% less classification error than the multi-Bernoulli.

Probabilistic Framework of Naive Bayes

This section presents the generative model for both cases of the naive Bayes classifier. First we explain the mechanisms they have in common, then, where the event models diverge, the assumptions and formulations of each are presented.

We approach the task of text classification in a Bayesian learning framework. We assume that the text data was generated by a parametric model, and use training data to calculate estimates of the model parameters. Then, armed with these estimates, we classify new test documents by using Bayes rule to turn the generative model around and calculate the posterior probability that a class would have generated the test document in question. Classification then becomes a simple matter of selecting the most probable class.

Both scenarios assume that text documents are generated by a mixture model parameterized by θ. The mixture model consists of mixture components $c_j \in \mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$. Each component is parameterized by θ_j, which are disjoint subsets of θ. Thus a document, d_i, is created by (1) selecting a component according to the priors, $\mathrm{P}(c_j|\theta)$, then (2) having the mixture component generate a document according to its own parameters, with distribution $\mathrm{P}(d_i|c_j;\theta_j)$. We can characterize the likelihood of a document with a sum of total probability over all mixture components:

$$\mathrm{P}(d_i|\theta) = \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta_j). \quad (1)$$

Each document has a class label. We assume that there is a one-to-one correspondence between classes and mixture model components, and thus use c_j to indicate both the jth mixture component and the jth class.[1] In this setting, (supervised learning from labeled training examples), the typically "hidden" indicator variables for a mixture model are provided as these class labels.

Multi-Bernoulli Model

In the multi-Bernoulli event model, a document is a binary vector over the space of words. Given a vocabulary V, each dimension of the space t, $t = \{1, \ldots, |V|\}$, corresponds to word w_t from the vocabulary. Dimension t of the vector for document d_i is written B_{it} and is either 0 or 1, indicating whether word w_t occurs at least once in the document. With such a document representation, we make the naive Bayes assumption: that the probability of each word occurring in a document is independent of the occurrence of other words in a document. Then, the probability of a document given its class from Equation 1 is simply the product of the probability of the attribute value over all word attributes:

$$\mathrm{P}(d_i|c_j;\theta_j) = \prod_{t=1}^{|V|}(B_{it}\mathrm{P}(w_t|c_j;\theta_j) + (1 - B_{it})(1 - \mathrm{P}(w_t|c_j;\theta_j))). \quad (2)$$

Thus given a generating component, a document can be seen as a collection of multiple independent Bernoulli experiments, one for each word in the vocabulary, with the probabilities for each of these word events defined by each component, $\mathrm{P}(w_t|c_j;\theta_j)$. This is equivalent to viewing a document as the result of a Bayes net, where the existence or absence of each word is dependent only on the class of the document.

Given a set of labeled training documents, $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$, learning a probabilistic classification model corresponds to estimating each of these class conditional word probabilities. These probabilities for the parameterization of a mixture component such that $\theta_{jt} = \mathrm{P}(w_t|c_j;\theta_j)$, where $0 \leq \theta_{jt} \leq 1$. We can calculate Bayes optimal estimates for these probabilities, which are maximum likelihood estimates, straightforward counting of events, supplemented by simple 'smoothing' that primes each word's count with a count of one to avoid probabilities of zero [Vapnik 1982]. We define $\mathrm{P}(c_j|d_i) = \{0, 1\}$ as given by the document's

[1] In a more general setting, this one-to-one correspondence can be avoided [Li and Yamanishi 1997; Nigam *et al.* 1998].

class label. Then, the estimate of the probability of word w_t in class c_j is:

$$\hat{\theta}_{jt} = \mathrm{P}(w_t|c_j;\theta_j) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} B_{it}\mathrm{P}(c_j|d_i)}{2 + \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}. \quad (3)$$

The class prior parameters, θ_{0j}, are set by the maximum likelihood estimate:

$$\hat{\theta}_{0j} = \mathrm{P}(c_j|\hat{\theta}) = \frac{\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{D}|}. \quad (4)$$

Note that this model does not capture the number of times each word occurs, and that it explicitly includes the non-occurrence probability of words that do not appear in the document.

Multinomial Model

In contrast to the multi-Bernoulli event model, the multinomial model captures word frequency information in documents. As a motivator, consider our tokenization of the Reuters newswire articles, in which all strings of digits are mapped to a common token. Since every news article is dated, and thus has a number, the number token in the multi-Bernoulli event model is uninformative. However, news articles about earnings tend to have a lot of numbers compared to general news articles. Thus, capturing frequency information of this token can help classification.

In the multinomial model, a document is an ordered sequence of word events, drawn from the same vocabulary V. We assume that the lengths of documents are evenly distributed, and are independent of class.[2] We again make a similar naive Bayes assumption: that the probability of each word event in a document is independent of the word's context and position in the document. Thus, each document d_i is drawn from a multinomial distribution of words with as many independent trials as the number of words in d_i. This yields the familiar "bag of words" representation for documents. We define N_{it} to be the count of the number of times word w_t occurs in document d_i. Then, the probability of a document given its class from Equation 1 is just the multinomial distribution:

$$\mathrm{P}(d_i|c_j;\theta_j) = \mathrm{P}(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{\mathrm{P}(w_t|c_j;\theta_j)^{N_{it}}}{N_{it}!}. \quad (5)$$

Each class parameterization θ_j defines the underlying distribution that is used to select each word for a document from that class. For each class θ_j is composed of probabilities for each word, such that $\theta_{jt} = \mathrm{P}(w_t|c_j;\theta_j)$, where $0 \leq \theta_{jt} \leq 1$ and $\sum_t \theta_{jt} = 1$.

Again, we can calculate Bayes optimal estimates for these parameters from a set of labeled training data. Here, the estimate of the probability of word w_t in class c_j is:

$$\hat{\theta}_{jt} = \mathrm{P}(w_t|c_j;\hat{\theta}_j) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N_{it}\mathrm{P}(c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N_{is}\mathrm{P}(c_j|d_i)}. \quad (6)$$

The class prior parameters are calculated as before according to Equation 4.

Classification

Given estimates of these parameters calculated from the training documents, classification can be performed on test documents by calculating the posterior probability of each class given the evidence of the test document, and selecting the class with the highest probability. We formulate this by applying Bayes rule:

$$\mathrm{P}(c_j|d_i;\hat{\theta}) = \frac{\mathrm{P}(c_j|\hat{\theta})\mathrm{P}(d_i|c_j;\hat{\theta}_j)}{\mathrm{P}(d_i|\hat{\theta})}. \quad (7)$$

The right hand side may be expanded by first substituting using Equations 1 and 4. Then the expansion of individual terms for this equation are dependent on the event model used. Use Equations 2 and 3 for the multi-Bernoulli event model. Use Equations 5 and 6 for the multinomial

Experimental Results

This section provides empirical evidence that the multinomial event model usually performs better than the multi-Bernoulli. The results are based on five different data sets.[3]

Data Sets and Protocol

The web pages pointed to by *Yahoo!* 'Science' hierarchy were gathered in July 1997. The web pages are divided into 95 disjoint classes containing 13589 pages as result of coalescing classes of hierarchy depth greater than two, and removing those classes with fewer than 40 documents. After tokenizing as above and removing stopwords and words that occur only once, the corpus has a vocabulary size of 44383 [McCallum *et al.* 1998].

The Industry Sector hierarchy, made available by *Market Guide Inc.* (www.marketguide.com) consists of company web pages classified in a hierarchy of industry sectors [McCallum *et al.* 1998]. There are 6440 web pages partitioned into the 71 classes that are two levels deep in the hierarchy. In tokenizing the data we

[2] Previous formalizations of the multinomial model have omitted the term for document length. It is necessary because document length specifies one parameter for the multinomial. Additionally, the assumption that document length contains no class information is a simplification only. In practice document length is class dependent, and a more general formalization should make it so.

[3] These data sets are available on the Internet. See http://www.cs.cmu.edu/~textlearning and http://www.research.att.com/~lewis.

do not stem. After removing tokens that occur only once or on a stoplist, the corpus has a vocabulary of size 29964.

The Newsgroups data set, collected by Ken Lang, contains about 20,000 articles evenly divided among 20 UseNet discussion groups [Joachims 1997]. Many of the categories fall into confusable clusters; for example, five of them are `comp.*` discussion groups, and three of them discuss religion. When tokenizing this data, we skip the UseNet headers (thereby discarding the subject line); tokens are formed from contiguous alphabetic characters with no stemming. The resulting vocabulary, after removing words that occur only once or on a stoplist, has 42191 words.

The WebKB data set [Craven *et al.* 1998] contains web pages gathered from university computer science departments. The pages are divided into seven categories: student, faculty, staff, course, project, department and other. In this paper, we use the four most populous non-other categories: student, faculty, course and project, all together containing 4199 pages. We did not use stemming or a stoplist; we found that using a stoplist actually hurt performance because, for example, "my" is the fourth-ranked word by information gain, and is an excellent indicator of a student homepage. The resulting vocabulary has 23830 words.

The 'ModApte' train/test split of the Reuters 21578 Distribution 1.0 data set consists of 12902 Reuters newswire articles in 135 overlapping topic categories. Following several other studies [Joachims 1998; Liere and Tadepalli 1997] we build binary classifiers for each of the 10 most populous classes. We ignore words on a stoplist, but do not use stemming. The resulting vocabulary has 19371 words.

For all data sets except Reuters, naive Bayes is performed with randomly selected train-test splits. The Industry Sector and Newsgroups data sets use five trials with 20% of the data held out for testing; Yahoo uses five trials with a 30% test data, and WebKB uses ten trials with a 30% test data. Results are reported as average classification accuracy across trials. In all experiments with multiple trials graphs show small error bars twice the width of the standard error; however they are often hard to see since they are often quite narrow. For Reuters, results on the Mod-Apte test set are shown as precision-recall breakeven points, a standard information retrieval measure for binary classification.

When reducing the vocabulary size, feature selection is done by selecting words that have highest mutual information with the class variable. This method works well with text and has been used often [Yang and Pederson 1997]. In all previous work of which we are aware, this was done by calculating the mutual information between the (1) class of a document and (2) the absence or presence of a word in the document, *i.e.* using a document event model, the multi-Bernoulli. We experimented with this method, as well as an event model that corresponds to the multinomial: calculating the mutual information between (1) the class of the document from which a word occurrence was drawn, and (2) a random variable over all word occurrences. Here the word occurrences are the events. Our preliminary experiments comparing these two feature selection methods on the Newsgroups data set with the multinomial event model showed little difference in classification accuracy. The results reported in this paper use the feature selection event model that corresponds to the event model used for classification.

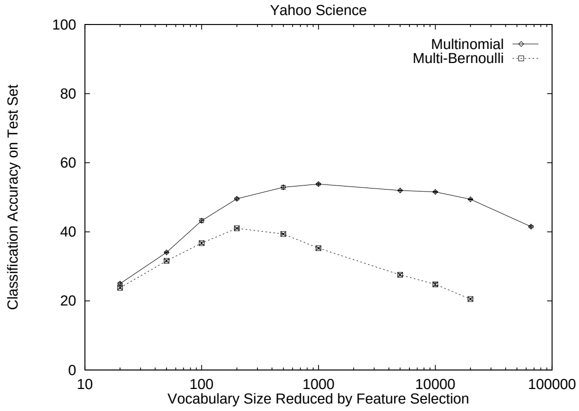

Figure 1: A comparison of event models for different vocabulary sizes on the Yahoo data set. Note that the multi-Bernoulli performs best with a small vocabulary and that the multinomial performs best with a larger vocabulary. The multinomial achieves higher accuracy overall.

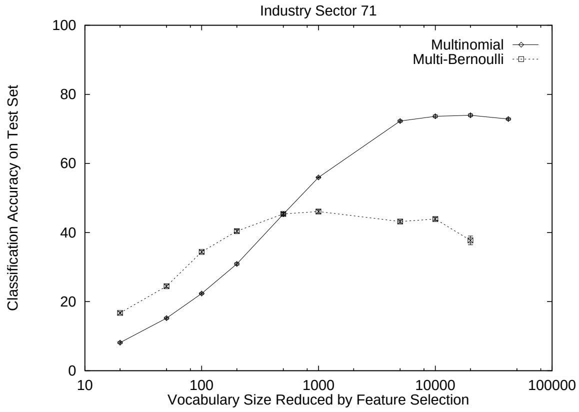

Figure 2: A comparison of event models for different vocabulary sizes on the Industry Sector data set. Note the same trends as seen in the previous figure.

Results

Figure 1 shows results on the Yahoo data set. The multinomial event model reaches a maximum of 54% accuracy at a vocabulary size of 1000 words. The multi-Bernoulli event model reaches a maximum of 41% accuracy with only 200 words. Note that the

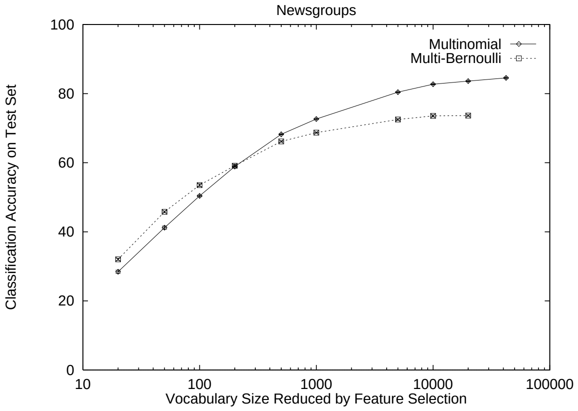

Figure 3: A comparison of event models for different vocabulary sizes on the Newsgroups data set. Here, both data sets perform best at the full vocabulary, but multinomial achieves higher accuracy.

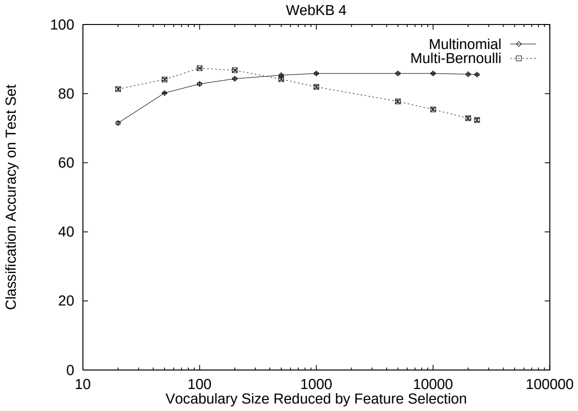

Figure 4: A comparison of event models for different vocabulary sizes on the WebKB data set. Here the two event models achieve nearly equivalent accuracies, but the multi-Bernoulli achieves this with a smaller vocabulary.

multi-Bernoulli shows its best results at a smaller vocabulary than the multinomial, and that the multinomial has best performance at a larger vocabulary size. The same pattern is seen in the Industry Sector data set, displayed in Figure 2. Here, multinomial has the highest accuracy of 74% at 20000 words, and multi-Bernoulli is best with 46% accuracy at 1000 words.

Figure 3 shows results for the Newsgroups data set. Here, both event models do best at the maximum vocabulary sizes. Multinomial achieves 85% accuracy and multi-Bernoulli achieves 74% accuracy. Previous results in this domain are consistent in that best results were with the full vocabulary [Joachims 1997; Nigam *et al.* 1998]. For the WebKB data, shown in Figure 4, the multi-Bernoulli has marginally higher accuracy than the multinomial, 87% accuracy at 100 words versus 86% accuracy at 5000 words. In ongoing work

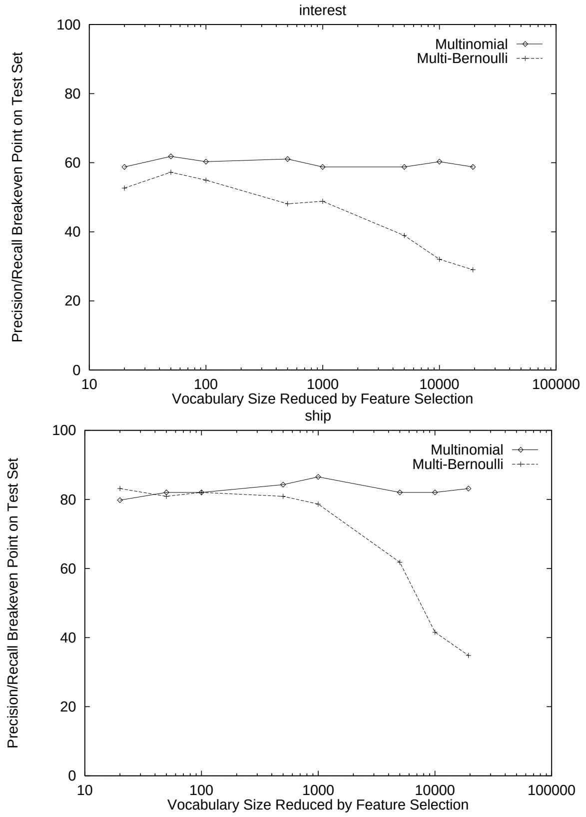

Figure 5: Two of the classification tasks from Reuters. Multinomial event models do an average of 4.8% points better. This domain tends to require smaller vocabularies for best performance. See Figure 6 for the remaining Reuters results.

we are exploring the reasons that this data set shows results different from the others.

Figures 5 and 6 show breakeven point results for the ten Reuters categories. Again, the trends are distinctive. The multi-Bernoulli achieves a slightly higher breakeven point in one case, but on average across categories, its best performance is 4.8 percentage points less than the multinomial. The multi-Bernoulli has a rapid decrease in performance as the vocabulary size grows, where the multinomial performance is more even across vocabulary size. Results by Joachims [1998] found performance was highest in this domain with no feature selection. However, in contrast to our results, this work uses the multi-Bernoulli event model for feature selection and the multinomial for classification. In future work we plan to investigate these feature selection methods more closely because we note that our results our consistently higher than those found in that work.

Discussion

For easy classification tasks, a small vocabulary is sufficient for high performance. The Reuters categorization tasks are examples of these—it is well-known that in several of the categories, high accuracy can be obtained with only a handful of words, sometimes even the single word that is the title of the category [Joachims 1998]. Our results are consistent with this, in that best performance is often achieved with small vocabulary sizes. Many real-world classification tasks do not share this attribute that all documents in a category are about a single narrow subject with limited vocabulary, but instead, a category consists of diverse subject matters with non-overlapping vocabularies. In such tasks large vocabularies are required for adequate classification accuracy. Since our results show that the multi-Bernoulli handles large vocabularies poorly, the multinomial event model is more appropriate for these challenging classification tasks.

It would be incorrect to argue that multi-Bernoulli have the advantage of counting evidence for words that do not occur. Multinomials implicitly encode this information in the probability distributions of words for each class. For example, if the word "professor" is the most likely word for faculty home pages, it will have a large probability for the faculty class, and all other words will be less probable. If the word "professor" does not then occur in a document, that document will be less likely to be a faculty, because the words in that document will be lower frequency words for the faculty class and higher frequency for others.

Another point to consider is that the multinomial event model should be a more accurate classifier for data sets that have a large variance in document length. The multinomial event model naturally handles documents of varying length by incorporating the evidence of each word. The multi-Bernoulli model is somewhat unusual for varying document length in that it is more likely for a word to occur in a long document regardless of the class. Thus, the variance of the classification will be large for documents of varying lengths. The multinomial has the advantage here because its event model more closely models the actual content of a document.

Note that the binomial model takes much more computation at classification time. For example, with a vocabulary size of 20000 (near where Newsgroups and Yahoo do best) classifying 20000 documents with the binomial takes 20 hours on a 200 MHz Pentium, while the multinomial takes only 12 minutes. This is because the multi-Bernoulli must consider every word in the vocabulary, where the multinomial only considers those words that occur in each test document. Even when vocabulary size is adjusted optimally for each event model, the multinomial is still more efficient.

Related Work

Kalt and Croft [1996] previously compared the multinomial model to the "binary independence model," the information retrieval terminology for our multi-Bernoulli model. Their theoretical analysis of the multinomial does not properly address document length assumptions. Their experiments use a single data set with extremely small vocabularies. Also, by normalizing document length for the multinomial, their event model is actually no longer a multinomial.

Conclusions

This paper has compared the theory and practice of two different first-order probabilistic classifiers, both of which make the "naive Bayes assumption." The multinomial model is found to be almost uniformly better than the multi-variate Bernoulli model. The multinomial takes significantly less computation time and almost always performs better. In empirical results on five real-world corpora we find that the binomial model reduces error by an average 27%, sometimes by more than 50%.

In future work we will investigate the role of document length in classification, looking for correspondence between variations in document length and the comparative performance of multi-Bernoulli and multinomial. We will also investigate event models that normalize the word occurrence counts in a document by document length, and work with more complex models that model document length explicitly on a per-class basis.

We also plan experiments with varying amounts of training data because we hypothesize that that optimal vocabulary size may change with the size of the training set.

Acknowledgments

We thank Doug Baker for help formatting the Reuters data set. We thank *Market Guide, Inc.* for permission to use their Industry Sector hierarchy, and Mark Craven for gathering its data from the Web. We thank *Yahoo!* for permission to use their data. We thank Tom Mitchell for helpful and enlightening discussions. This research was supported in part by the Darpa HPKB program under contract F30602-97-1-0215.

References

M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *AAAI-98*, 1998. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/.

P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. *Machine Learning*, 29:103–130, 1997.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29, 1997.

Jerome H. Friedman. On bias, variance, 0/1 - loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.

Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *ICML-97*, 1997.

Thorsten Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In *ECML-98*, 1998.

T. Kalt and W. B. Croft. A new probabilistic model of text classification and retrieval. Technical Report IR-78, University of Massachusetts Center for Intelligent Information Retrieval, 1996. http://ciir.cs.umass.edu/publications/index.shtml.

Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.

Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of bayesian classifiers. In *AAAI-92*, 1992.

Leah S. Larkey and W. Bruce Croft. Combining classifiers in text categorization. In *SIGIR-96*, 1996.

D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of ACM SIGIR Conference*, 1994.

David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR-92*, 1992.

Hang Li and Kenji Yamanishi. Document classification using a finite mixture model. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997.

Ray Liere and Prasad Tadepalli. Active learning with committees for text categorization. In *AAAI-97*, 1997.

Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Ng. Improving text clasification by shrinkage in a hierarchy of classes. In *Submitted to ICML-98*, 1998. http://www.cs.cmu.edu/∼mccallum.

Tom M. Mitchell. *Machine Learning*. WCB/McGraw-Hill, 1997.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Learning to classify text from labeled and unlabeled documents. In *AAAI-98*, 1998. http://www.cs.cmu.edu/∼mccallum.

S. E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

Mehran Sahami. Learning limited dependence bayesian classifiers. In *KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 335–338. AAAI Press, 1996.

V. Vapnik. *Estimations of dependences based on statistical data*. Springer Publisher, 1982.

Yiming Yang and Jan Pederson. Feature selection in statistical learning of text categorization. In *ICML-97*, pages 412–420, 1997.

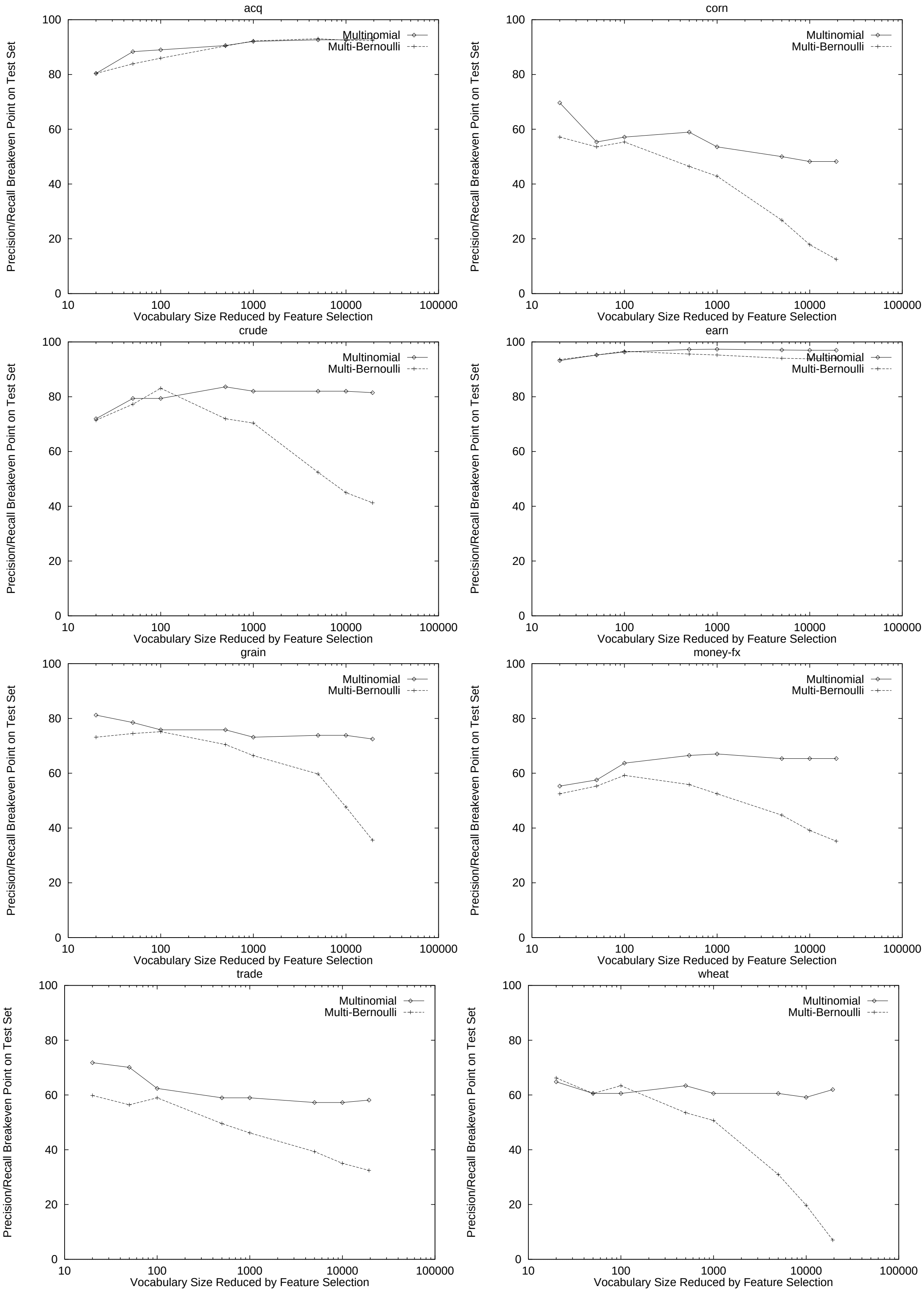

Figure 6: The continuation of the Reuters results from Figure 5.