An Improved Hierarchical Bayesian Model of Language for Document Classification

Ben Allison
Department of Computer Science
University of Sheffield
UK
ben@dcs.shef.ac.uk

Abstract

This paper addresses the fundamental problem of document classification, and we focus attention on classification problems where the classes are mutually exclusive. In the course of the paper we advocate an approximate sampling distribution for word counts in documents, and demonstrate the model's capacity to outperform both the simple multinomial and more recently proposed extensions on the classification task. We also compare the classifiers to a linear SVM, and show that provided certain conditions are met, the new model allows performance which exceeds that of the SVM and attains amongst the very best published results on the Newsgroups classification task.

1 Introduction

Document classification is one of the key technologies in the emerging digital world: as the amount of textual information existing in electronic form increases exponentially, reliable automatic methods to sift through the haystack and pluck out the occasional needle are almost a necessity.

Previous comparative studies of different classifiers (for example, (Yang and Liu, 1999; Joachims, 1998; Rennie et al., 2003; Dumais et al., 1998)) have consistently shown linear Support Vector Machines to be the most appropriate method. *Generative* probabilistic classifiers, often represented by the multinomial classifier, have in these same studies performed poorly, and this empirical evidence has been bolstered by theoretical arguments (Lasserre et al., 2006).

In this paper we revisit the theme of generative classifiers for mutually exclusive classification problems, but consider classifiers employing more complex models of language; as a starting point we consider recent work (Madsen et al., 2005) which relaxes some of the multinomial assumptions. We continue and expand upon the theme of that work, but identify some weaknesses both in its theoretical motivations and practical applications. We demonstrate a new approximate model which overcomes some of these concerns, and demonstrate substantial improvements that such a model achieves on four classification tasks, three of which are standard and one of which is a newly created task. We also show the new model to be highly competitive to an SVM where the previous models are not.

§2 of the paper describes previous work which has sought a probabilistic model of language and its application to document classification. §3 describes the models we consider in this paper, and gives details of parameter estimation. §4 describes our evaluation of the models, and §5 presents the results of this evaluation. §6 explores reasons for the observed results, and finally §7 ends with some concluding remarks.

2 Related Work

The problem of finding an appropriate and tractable model for language is one which has been studied in many different areas. In many cases, the first (and often only) model is one in which counts of words are modelled as binomial– or Poisson–distributed random variables. However, the use of such distributions entails an implicit assumption

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 25–32
Manchester, August 2008

that the occurrence of words is the result of a fixed number of independent trials—draws from a "bag of words"—where on each trial the probability of success is constant.

Several authors, among them (Church and Gale, 1995; Katz, 1996), observe empirically such models are not always accurate predictors of actual word behaviour. This moves them to suggest distributions for word counts where the underlying probability varies between documents; thus the expected behaviour of a word in a new document is a combination of predictions for all possible probabilities. Other authors (Jansche, 2003; Eyheramendy et al., 2003; Lowe, 1999) use these same ideas to classify documents on the basis of subsets of vocabulary, in the first and third cases with encouraging results using small subsets (in the second case, the performance of the model is shown to be poor compared to the multinomial).

When one moves to consider counts of all words in some vocabulary, the proper distribution of the whole vector of word counts is multinomial. (Madsen et al., 2005) apply the same idea as for the single word (binomial) case to the multinomial, using the most convenient form of distribution to represent the way the vector of multinomial probabilities varies between documents, and report encouraging results compared to the simple multinomial. However, we show that the use of the most mathematically convenient distribution to describe the way the vector of probabilities varies entails some unwarranted and undesirable assumptions. This paper will first describe those assumptions, and then describe an approximate technique for overcoming the assumptions. We show that, combined with some alterations to estimation, the models lead to a classifier able to outperform both the multinomial classifier and a linear SVM.

3 Probabilistic Models of Language for Document Classification

In this section, we briefly describe the use of a generative model of language as applied to the problem of document classification, and also how we estimate all relevant parameters for the work which follows.

In terms of notation, we use $\tilde{c}$ to represent a random variable and c to represent an outcome. We use roman letters for observed or observable quantities and greek letters for unobservables (i.e. parameters). We write $\tilde{c} \sim \varphi(c)$ to mean that $\tilde{c}$ has probability density (discrete or continuous) $\varphi(c)$, and write $p(c)$ as shorthand for $p(\tilde{c} = c)$. Finally, we make no explicit distinction in notation between univariate and multivariate quantities; however, we use θ_j to refer to the j-th component of the vector θ.

We consider documents to be represented as vectors of count–valued random variables such that $d = \{d_1 ... d_v\}$. For classification, interest centres on the conditional distribution of the class variable, given such a document. Where documents are to be assigned to one class only (as in the case of this paper), this class is judged to be the most probable class. For generative classifiers such as those considered here, the posterior distribution of interest is modelled from the joint distribution of class and document; thus if $\tilde{c}$ is a variable representing class and $\tilde{d}$ is a vector of word counts, then:

$$p(c|d) \propto p(c) \cdot p(d|c) \tag{1}$$

For the purposes of this work we also assume a uniform prior on $\tilde{c}$, meaning the ultimate decision is on the basis of the document alone.

Multinomial Sampling Model

A natural way to model the distribution of counts is to let $p(d|c)$ be distributed multinomially, as proposed in (Guthrie et al., 1994; McCallum and Nigam, 1998) amongst others. The multinomial model assumes that documents are the result of repeated trials, where on each trial a word is selected at random, and the probability of selecting the j-th word from class c is θ_{cj}. However, in general we will not use the subscript c – we estimate one set of parameters for each possible class.

Using multinomial sampling, the term $p(d|c)$ has distribution:

$$p_{multinomial}(d|\theta) = \frac{\left(\sum_j d_j\right)!}{\prod_j (d_j!)} \prod_j \theta_j^{d_j} \tag{2}$$

A simple Bayes estimator for θ can be obtained by taking the prior for θ as a Dirichlet distribution, in which case the posterior is also Dirichlet. Denote the total training data for the class in question as $\mathcal{D} = \{(d_{11} ... d_{1v}) ... (d_{k1} ... d_{kv})\}$ (that is, counts of each of v words in k documents). Then if $p(\theta) \sim Dirichlet(\alpha_1 ... \alpha_v)$, the mean of $p(\theta|\mathcal{D})$ for the j-th component of θ (which is the estimate we use) is:

$$\hat{\theta_j} = \mathrm{E}[\theta_j|\mathcal{D}] = \frac{\alpha_j + n_j}{\sum_j \alpha_j + n_\bullet} \quad (3)$$

where the n_j are the sufficient statistics $\sum_i n_{ij}$, and $n_\bullet$ is $\sum_j n_j$. We follow common practice and use the standard reference Dirichlet prior, which is uniform on θ, such that $\alpha_j = 1$ for all j.

3.1 Hierarchical Sampling Models

In contrast to the model above, a hierarchical sampling model assumes that $\tilde{\theta}$ varies between documents, and has distribution which depends upon parameters η. This allows for a more realistic model, letting the probabilities of using words vary between documents subject only to some general trend.

For example, consider documents about politics: some will discuss the current British Prime Minister, Gordon Brown. In these documents, the probability of using the word *brown* (assuming case normalisation) may be relatively high. Other politics articles may discuss US politics, for example, or the UN, French elections, and so on, and these articles may have a much lower probability of using the word *brown*: perhaps just the occasional reference to the Prime Minister. A hierarchical model attempts to model the way this probability varies between documents in the politics class.

Starting with the joint distribution $p(\theta, d|\eta)$ and averaging over all possible values that θ may take in the new document gives:

$$p(d|\eta) = \int p(\theta|\eta)p(d|\theta)\, d\theta \quad (4)$$

where integration is understood to be over the entire range of possible θ. Intuitively, this allows $\tilde{\theta}$ to vary between documents subject to the restriction that $\tilde{\theta} \sim p(\theta|\eta)$, and the probability of observing a document is the average of its probability for all possible θ, weighted by $p(\theta|\eta)$. The sampling process is 1) θ is first sampled from $p(\theta|\eta)$ and then 2) d is sampled from $p(d|\theta)$, leading to the hierarchical name for such models.

Dirichlet Compound Multinomial Sampling Model

(Madsen et al., 2005) suggest a form of (4) where $p(\theta|\eta)$ is Dirichlet-distributed, leading to a *Dirichlet–Compound–Multinomial* sampling distribution. The main benefit of this assumption is that the integral of (4) can be obtained in closed form. Thus $p(d|\alpha)$ (using the standard α notation for Dirichlet parameters) has distribution:

$$p_{DCM}(d|\alpha) = \frac{\left(\sum_j d_j\right)!}{\prod_j (d_j!)} \times \frac{\Gamma\left(\sum_j \alpha_j\right)}{\Gamma\left(\sum_j d_j + \alpha_j\right)} \times \prod_j \frac{\Gamma(\alpha_j + d_j)}{\Gamma(\alpha_j)} \quad (5)$$

Maximum likelihood estimates for the α are difficult to obtain, since the likelihood for α is a function which must be maximised for all components simultaneously, leading some authors to use approximate distributions to improve the tractability of maximum likelihood estimation (Elkan, 2006). In contrast, we reparameterise the Dirichlet compound multinomial, and estimate some of the parameters in closed form.

We reparameterise the model in terms of μ and λ – μ is a vector of length v, and λ is a constant which reflects the variance of θ. Under this parametrisation, $\alpha_j = \lambda\mu_j$. The estimate we use for μ_j is simply:

$$\hat{\mu_j} = \frac{n_j}{n_\bullet} \quad (6)$$

where n_j and $n_\bullet$ are defined above. This simply matches the first moment about the mean of the distribution with the first moment about the mean of the sample. Once again letting:

$$\mathcal{D} = \{d_1 ... d_k\} = \{(d_{11} ... d_{1v}) ... (d_{k1} ... d_{kv})\}$$

denote the training data such that the d_i are individual document vectors and d_{ij} are counts of the j-th word in the i-th document, the likelihood for λ is:

$$\mathcal{L}(\lambda) = \prod_i \frac{\Gamma(\sum_j \lambda\mu_j)}{\Gamma(\sum_j d_{ij} + \lambda\mu_j)} \prod_j \frac{\Gamma(\lambda\mu_j + d_{ij})}{\Gamma(\lambda\mu_j)} \quad (7)$$

This is a one–dimensional function, and as such is much more simple to maximise using standard optimisation techniques, for example as in (Minka, 2000).

As before, however, simple maximum likelihood estimates alone are not sufficient: if a word fails to appear at all in $\mathcal{D}$, the corresponding μ_j will be zero, in which case the distribution is improper. The theoretically sound solution would be

to incorporate a prior on either α or (under our parameterisation) μ; however, this would lead to high computational cost as the resulting posterior would be complicated to work with. (Madsen et al., 2005) instead set each $\hat{\alpha_j}$ as the maximum likelihood estimate plus some ϵ, in some ways echoing the estimation of θ for the multinomial model. Unfortunately, unlike a prior this strategy has the same effect regardless of the amount of training data available, whereas any true prior would have diminishing effect as the amount of training data increased. Instead, we supplement actual training data with a pseudo–document in which every word occurs once (note this is quite different to setting $\epsilon = 1$); this echoes the effect of a true prior on μ, but without the computational burden.

A Joint Beta-Binomial Sampling Model

Despite its apparent convenience and theoretical well–foundedness, the Dirichlet compound multinomial model has one serious drawback, which is emphasised by the reparameterisation. Under the Dirichlet, there is a functional dependence between the expected value of θ_j, μ_j and its variance, where the relationship is regulated by the constant λ. Thus two words whose μ_j are the same will also have the same variance in the θ_j. This is of concern since different words have different patterns of use – to use a popular turn of phrase, some words are more "bursty" than others (see (Church and Gale, 1995) for examples). In practice, we may hope to model different words as having the same expected value, but drastically different variances – unfortunately, this is not possible using the Dirichlet model.

The difficulty with switching to a different model is the evaluation of the integral in (4). The integral is in fact in many thousands of dimensions, and even if it were possible to evaluate such an integral numerically, the process would be exceptionally slow.

We overcome this problem by decomposing the term $p(d|\eta)$ into a product of independent terms of the form $p(d_j|\eta_j)$. A natural way for each of these terms to be distributed is to let the probability $p(d_j|\theta_j)$ be binomial and to let $p(\theta_j|\eta_j)$ be beta–distributed. The probability $p(d_j|\eta_j)$ (where $\eta_j = \{\alpha_j, \beta_j\}$, the parameters of the beta distribution) is then:

$$p_{bb}(d_j|\alpha_j, \beta_j) = \binom{n}{d_j} \frac{\mathrm{B}(d_j + \alpha_j,\, n - d_j + \beta_j)}{\mathrm{B}(\alpha_j, \beta_j)} \tag{8}$$

where $\mathrm{B}(\bullet)$ is the Beta function. The term $p(d|\eta)$ is then simply:

$$p_{beta-binomial}(d|\eta) = \prod_j p(d_j|\eta_j) \tag{9}$$

This allows means and variances for each of the θ_j to be specified separately, but this comes at a price: while the Dirichlet ensures that $\sum_j \theta_j = 1$ for all possible θ, the model above does not. Thus the model is only an approximation to a true model where components of θ have independent means and variances, and the requirements of the multinomial are fulfilled. However, given the inflexibility of the Dirichlet multinomial model, we argue that such a sacrifice is justified.

In order to estimate parameters of the Beta–Binomial model, we take a slight departure from both (Lowe, 1999) and (Jansche, 2003) who have both used a similar model previously for individual words. (Lowe, 1999) uses numerical techniques to find maximum likelihood estimates of the α_j and β_j, which was feasible in that case because of the highly restricted vocabulary and two-classes. (Jansche, 2003) argues exactly this point, and uses moment–matched estimates; our estimation is similar to that, in that we use moment–matching, but different in other regards.

Conventional parameter estimates are affected (in some way or other) by the likelihood function for a parameter, and the likelihood function is such that longer documents exert a greater influence on the overall likelihood for a parameter. That is, we note that if the true binomial parameter θ_{ij} for the j-th word in the i-th document were known, then the most sensible expected value for the distribution over θ_j would be:

$$\mathrm{E}\left[\theta_j\right] = \frac{1}{k} \times \sum_{i=1}^{k} \theta_{ij} \tag{10}$$

Whereas the expected value of conventional method–of–moments estimate is:

$$\mathrm{E}\left[\theta_j\right] = \sum_{i=1}^{k} p\left(\theta_{ij}\right) \times \hat{\theta_{ij}} \tag{11}$$

That is, a weighted mean of the maximum likelihood estimates of each of the θ_{ij}, with weights

given by $p(\theta_{ij})$, i.e. the length of the i-th document. Similar effects would be observed by maximising the likelihood function numerically. This is to our minds undesireable, since we do note believe that longer documents are necessarily more representative of the population of all documents than are shorter ones (indeed, extremely long documents are likeliy to be an oddity), and in any case the goal is to capture variation in the parameters.

This leads us to suggest estimates for parameters such that the expected value of the distribution is as in 10 but with the θ_{ij} (which are unknown) replaced with their maximum likelihood estimates, $\hat{\theta}_{ij}$. We then use these estimates to specify the desired variance, leading to the simultaneous equations:

$$\frac{\alpha_j}{\alpha_j + \beta_j} = \frac{\sum_i \hat{\theta}_{ij}}{k} \quad (12)$$

$$\frac{\alpha_j \beta_j}{(\alpha_j + \beta_j)^2(\alpha_j + \beta_j + 1)} = \frac{\sum_i (\hat{\theta}_{ij} - \mathrm{E}[\theta_j])^2}{k} \quad (13)$$

As before, we supplement actual training documents with a pseudo-document in which every word occurs once to prevent any α_j being zero.

4 Evaluating the Models

This section describes evaluation of the models above on four text classification problems.

The Newsgroups task is to classify postings into one of twenty categories, and uses data originally collected in (Lang, 1995). The task involves a relatively large number of documents (approximately 20,000) with roughly even distribution of messages, giving a very low baseline of approximately 5%.

For the second task, we use a task derived from the Enron mail corpus (Klimt and Yang, 2004), described in (Allison and Guthrie, 2008). Corpus is a nine–way email authorship attribution problem, with 4071 emails (between 174 and 706 emails per author)[1]. The mean length of messages in the corpus is 75 words.

WebKB is a web–page classification task, where the goal is to determine the webpage type of the unseen document. We follow the setup of (McCallum and Nigam, 1998) and many thereafter, and use the four biggest categories, namely `student`, `faculty`, `course` and `project`. The resulting corpus consists of approximately 4,200 webpages.

The SpamAssassin corpus is made available for public use as part of the open-source Apache SpamAssassin Project[2]. It consists of email divided into three categories: `Easy Ham`, which is email unambiguously ham (i.e. not spam), `Hard Ham` which is not spam but shares many traits with spam, and finally `Spam`. The task is to apply these labels to unseen emails. We use the latest version of all datasets, and combine the easy_ham and easy_ham_2 as well as spam and spam_2 sets to form a corpus of just over 6,000 messages.

In all cases, we use 10–fold cross validation to make maximal use of the data, where folds are chosen by random assignment. We define "words" to be contiguous whitespace–delimited alpha–numeric strings, and perform no stemming or stoplisting.

For the purposes of comparison, we also present results using a linear SVM (Joachims, 1999), which we convert to multi–class problems using a one–versus–all strategy shown to be amongst the best performing strategies (Rennie and Rifkin, 2001). We normalise documents to be vectors of unit length, and resolve decision ambiguities by sole means of distance to the hyperplane. We also note that experimentation with non–linear kernels showed no consistent trends, and made very little difference to performance.

5 Results

Table 1 displays results for the three models over the four datasets. We use the simplest measure of classifier performance, accuracy, which is simply the total number of correct decisions over the ten folds, divided by the size of the corpus. In response to a growing unease over the use of significance tests (because they have a tendency to overstate significance, as well as obscure effects of sample size) we provide 95% intervals for accuracy as well as the metric itself. To calculate these, we view accuracy as an (unknown) parameter to a binomial distribution such that the number of correctly classified documents is a binomially distributed random variable. We then calculate the Bayesian interval for the parameter, as described in (Brown et al., 2001), which allows immediate quantification

[1]The corpus is available for download from www.dcs.shef.ac.uk/~ben.

[2]The corpus is available online at http://spamassassin.apache.org/publiccorpus/

of uncertainty in the true accuracy after a limited sample.

As can be seen from the performance figures, no one classifier is totally dominant, although there are obvious and substantial gains in using the Beta-Binomial model on the Newsgroups and Enron tasks when compared to all other models. The SpamAssassin corpus shows the beta–binomial model and the SVM to be considerably better than the other two models, but there is little to choose between them. The WebKB task, however, shows extremely unusual results: the SVM is head and shoulders above other methods, and of the generative approaches the multinomial is clearly superior. In all cases, the Dirichlet model actually performs worse than the multinomial model, in contrast to the observations of (Madsen et al., 2005).

In terms of comparison with other work, we note that the performance of our multinomial model agrees with that in other work, including for example (Rennie et al., 2003; Eyheramendy et al., 2003; Madsen et al., 2005; Jansche, 2003). Our Dirichlet model performs worse than that in (Madsen et al., 2005) (85% here compared to 89% in that work), which we attribute to their experimentation with alternate smoothing ϵ as described in §3.1. We note however that the Beta-Binomial model here still outperforms that work by some considerable margin. Finally, we note that our beta–binomial model outperforms that in (Jansche, 2003), which we attribute mainly to the altered estimate, but also to the partial vocabulary used in that work. In fact, (Jansche, 2003) shows there to be little to separate the beta-binomial and multinomial models for larger vocabularies, in stark contrast to the work here, and this is doubtless due to the parameter estimation.

6 Analysis

One might expect performance of a hierarchical sampling model to eclipse that of the SVM because of the nature of the decision boundary, provided certain conditions are met: the SVM estimates a linear decision boundary, and the multinomial classifier does the same. However, the decision boundaries for the hierarchical classifiers are non–linear, and can represent more complex word behaviour, provided that sufficient data exist to predict it. However, unlike generic non–linear SVMs (which made little difference compared to a linear SVM) the non–linear decision boundary here arises naturally from a model of word behaviour.

For the hierarchical models, performance rests on the ability to estimate both the rate of word occurrence θ_j and also the way that this rate varies between documents. To reliably estimate variance (and arguably rate as well) would require words to occur a sufficient number of times. However, this section will demonstrate that two of the datasets have many words which do not occur with sufficient frequency to estimate parameters, and in those two the linear SVM's performance is more comparable.

We present two quantifications of word reuse to support our conclusions. The first are frequency spectra for each of the four corpora, shown in Figure 1. The two more problematic datasets appear in the top of the figure. To generate the charts, we pool all documents from all classes in a each problem, and count the number of words that appear once, twice, and so on. The x axis is the number of times a word occurs, and the y axis the total number of words which have that count.

The WebKB corpus has the large majority of words occurring very few times (the mass of the distribution is concentrated towards the left of the chart), while the SpamAssassin corpus is more reasonable and the Newsgroups corpus has by far the most words which occur with substantial frequency (this correlates perfectly with the relative performances of the classifiers on these datasets). For the Enron corpus, it is somewhat harder to tell, since its size means no words occur with substantial frequency.

We also consider the proportion of all word pairs in a corpus in which the first word is the same as the second word. If a corpus has $n_\bullet$ words total with total counts $n_1 ... n_v$ then the statistic is:

$$r = \frac{1}{\left(n_\bullet(n_\bullet - 1)\right)/2} \sum_i (n_i(n_i - 1))/2. \quad (14)$$

To measure differing tendencies to reuse words, we calculate the r statistic once for each class, and then its mean across all classes in a problem (Table 2). We note that the two corpora on which the hierarchical model dominates have much greater tendency for word reuse, meaning the extra parameters can be esimated with greater accuracy. The SpamAssassin corpus is, by this measure, a harder task, but this is somewhat mitigated by the more even frequency distribution evidenced in Figure 1; on the other hand, the WebKB corpus does

	Newsgroups	Enron Authors	WebKB	SpamAssassin
Multinomial	85.66 ± 0.5	74.55 ± 1.34	85.69 ± 1.06	95.96 ± 0.5
DCM	85.03 ± 0.51	74.43 ± 1.34	82.69 ± 1.15	91.47 ± 0.7
Beta-Bin	**91.65** ± 0.4^{*+}	**83.54** ± 1.14^{*+}	84.81 ± 1.08	97.35 ± 0.4*
SVM	88.8 ± 0.45*	80 ± 1.23*	**92.68** ± 0.79*	**97.65** ± 0.38*

Table 1: Performance of four classifiers on four tasks. Error is 95% interval for accuracy. Bold denotes best performance on a task. * denotes performance superior to multinomial which exceeds posterior uncertainty (i.e. observed performance outside 95% interval). $^{+}$ denotes the same for the SVM

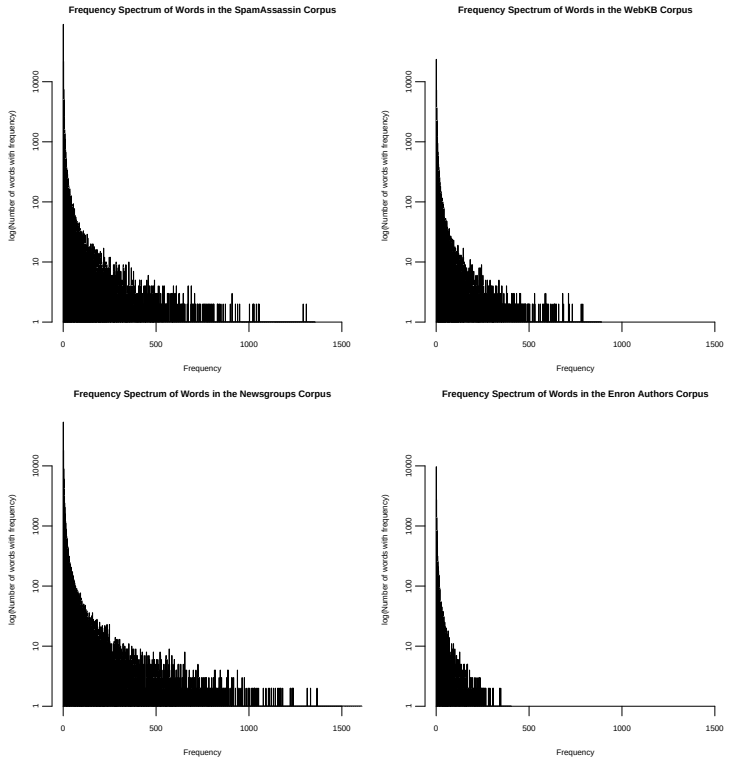

Figure 1: Frequency spectra for the four datasets. y axis is on a logarithmic scale

not look promising for the hierarchical model by either measure.

7 Conclusion

In this paper, we have advocated the use of a joint beta–binomial distribution for word counts in documents for the purposes of classification. We have shown that this model outperforms classifiers based upon both multinomial and Dirichlet Compound Multinomial distributions for word counts.

We have further made the case that, where corpora are sufficiently large as to warrant it, a generative classifier employing a hierarchical sampling model outperforms a discriminative linear SVM. We attribute this to the capacity of the proposed model to capture aspects of word behaviour beyond a simpler model. However, in cases where the data contain many infrequent words and the tendency to reuse words is relatively low, defaulting to a linear classifier (either the multinomial for a generative classifier, or preferably the linear SVM) increases performance relative to a more complex model, which cannot be fit with sufficient precision.

References

Allison, Ben and Louise Guthrie. 2008. Authorship attribution of e-mail: Comparing classifiers over a new corpus for evaluation. In *Proceedings of LREC'08*.

Brown, Lawrence D., Tony Cai, and Anirban DasGupta. 2001. Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101–117, may.

	Newsgroups	Enron Authors	WebKB	SpamAssassin
Mean r	0.0090	0.0083	0.0047	0.0037

Table 2: Mean r statistic for the four problems

Church, K. and W. Gale. 1995. Poisson mixtures. *Natural Language Engineering*, 1(2):163–190.

Dumais, Susan, John Platt, David Heckerman, and Mehran Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *CIKM '98*, pages 148–155.

Elkan, Charles. 2006. Clustering documents with an exponential-family approximation of the dirichlet compound multinomial distribution. In *Proceedings of the Twenty-Third International Conference on Machine Learning*.

Eyheramendy, S., D. Lewis, and D. Madigan. 2003. The naive bayes model for text categorization. *Artificial Intelligence and Statistics*.

Guthrie, Louise, Elbert Walker, and Joe Guthrie. 1994. Document classification by machine: theory and practice. In *Proceedings COLING '94*, pages 1059–1063.

Jansche, Martin. 2003. Parametric models of linguistic count data. In *ACL '03*, pages 288–295.

Joachims, Thorsten. 1998. Text categorization with support vector machines: learning with many relevant features. In Nédellec, Claire and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142.

Joachims, Thorsten. 1999. Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning*.

Katz, Slava M. 1996. Distribution of content words and phrases in text and language modelling. *Nat. Lang. Eng.*, 2(1):15–59.

Klimt, Bryan and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Proceedings of ECML 2004*, pages 217–226.

Lang, Ken. 1995. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339.

Lasserre, Julia A., Christopher M. Bishop, and Thomas P. Minka. 2006. Principled hybrids of generative and discriminative models. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 87–94.

Lowe, S. 1999. The beta-binomial mixture model and its application to tdt tracking and detection. In *Proceedings of the DARPA Broadcast News Workshop*.

Madsen, Rasmus E., David Kauchak, and Charles Elkan. 2005. Modeling word burstiness using the Dirichlet distribution. In *ICML '05*, pages 545–552.

McCallum, A. and K. Nigam. 1998. A comparison of event models for naïve bayes text classification. In *Proceedings AAAI-98 Workshop on Learning for Text Categorization*.

Minka, Tom. 2000. Estimating a dirichlet distribution. Technical report, Microsoft Research.

Rennie, Jason D. M. and Ryan Rifkin. 2001. Improving multiclass text classification with the Support Vector Machine. Technical report, Massachusetts Insititute of Technology, Artificial Intelligence Laboratory.

Rennie, J., L. Shih, J. Teevan, and D. Karger. 2003. Tackling the poor assumptions of naive bayes text classifiers.

Yang, Y. and X. Liu. 1999. A re-examination of text categorization methods. In *22nd Annual International SIGIR*, pages 42–49, Berkley, August.