Modeling Word Burstiness Using the Dirichlet Distribution

Rasmus E. Madsen REM@IMM.DTU.DK
Department of Informatics and Mathematical Modelling, Technical University of Denmark

David Kauchak DKAUCHAK@CS.UCSD.EDU
Charles Elkan ELKAN@CS.UCSD.EDU
Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92092-0144

Abstract

Multinomial distributions are often used to model text documents. However, they do not capture well the phenomenon that words in a document tend to appear in bursts: if a word appears once, it is more likely to appear again. In this paper, we propose the Dirichlet compound multinomial model (DCM) as an alternative to the multinomial. The DCM model has one additional degree of freedom, which allows it to capture burstiness. We show experimentally that the DCM is substantially better than the multinomial at modeling text data, measured by perplexity. We also show using three standard document collections that the DCM leads to better classification than the multinomial model. DCM performance is comparable to that obtained with multiple heuristic changes to the multinomial model.

1. Introduction

Document classification is the task of identifying what topic(s) a document concerns. Generative approaches to classification are popular since they are relatively easy to interpret and can be trained quickly. With these approaches, the key problem is to develop a probabilistic model that represents the data well. Unfortunately, for text classification too little attention has been devoted to this task. Instead, a generic multinomial model is typically used.

Recent work (Rennie et al., 2003) has pointed out a number of deficiencies of the multinomial model, and suggested heuristics to improve its performance. In this paper, we follow an alternative path. We present a different probabilistic model that, without any heuristic changes, is far better suited for representing a class of text documents.

As most researchers do, we represent an individual document as a vector of word counts (Salton et al., 1975). This bag-of-words representation loses semantic information, but it simplifies further processing. The usual next simplification is the assumption that documents are generated by repeatedly drawing words from a fixed distribution. Under this assumption, word emissions are independent given the class, i.e. the naive Bayes property holds. This property is not valid (Lewis, 1998), but naive Bayes models remain popular (McCallum & Nigam, 1998; Sebastiani, 2002) because they are fast and easy to implement, they can be fitted even with limited training data, and they do yield accurate classification when heuristics are applied (Jones, 1972; Rennie et al., 2003).

The central problem with the naive Bayes assumption is that words tend to appear in bursts, as opposed to being emitted independently (Church & Gale, 1995; Katz, 1996). Rennie et al. (2003) address this issue by log-normalizing counts, reducing the impact of burstiness on the likelihood of a document. Teevan and Karger (2003) empirically search for a model that fits documents well within an exponential family of models, while Jansche (2003) proposes a zero-inflated mixture model.

In this paper we go further. We show that the multinomial model is appropriate for common words, but not for other words. The distributions of counts produced by multinomials are fundamentally different from the count distributions of natural text. Zipf's law (Zipf, 1949) states that the probability p_i of occurrence of an event follows a power law $p_i \approx i^{-a}$, where i is the rank of the event and a is a parameter. The most famous

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

example of Zipf's law is that the frequency of an English word, as a function of the word's rank, follows a power law with exponent close to minus one.

We propose to model a collection of text documents with a Dirichlet distribution (Minka, 2003). The Dirichlet distribution can be interpreted in two ways for this purpose, either as a bag-of-scaled-documents or as a bag-of-bags-of-words. We show below that the latter approach works well.

Dirichlet distributions have been used previously to model text, but our approach is fundamentally different. In the LDA approach (Blei et al., 2003) the Dirichlet is a distribution over topics, while each topic is modeled in the usual way as a multinomial distribution over words. In our approach, each topic, i.e. each class of documents, is modeled in a novel way by a Dirichlet distribution instead of by a multinomial. Our approach is therefore complementary to the LDA and related approaches.

2. Multinomial modeling of text

When using a multinomial distribution to model text, the multinomial specifies the probability of observing a given vector of word counts, where the probability θ_w of emitting word w is subject to the constraints $\sum_w \theta_w = 1$ and $\theta_w > 0$ for all w. The probability of a document x represented as a vector of counts x_w is

$$p(x|\theta) = \frac{n!}{\prod_{w=1}^{W} x_w!} \prod_{w=1}^{W} \theta_w^{x_w}$$

where x_w is the number of times word w appears in the document, W is the size of the vocabulary, and $n = \sum x_w$.

The multinomial distribution is different for each different document length n. This is not a problem when learning the parameters; it is possible to generalize over documents with different lengths. The maximum likelihood parameter estimates $\hat{\theta}$ are

$$\hat{\theta}_w = \frac{\sum_{d=1}^{D} x_{dw}}{\sum_{w'=1}^{W} \sum_{d=1}^{D} x_{dw'}}$$

where d is the document number and D is the number of documents. These estimates depend only on the fraction of times a given word appears in the entire corpus.

When the multinomial model is used to generate a document, the distribution of the number of emissions (i.e. count) of an individual word is a binomial:

$$p(x_w|\theta) = \binom{n}{x_w} \theta_w^{x_w} (1 - \theta_w)^{n - x_w}. \qquad (1)$$

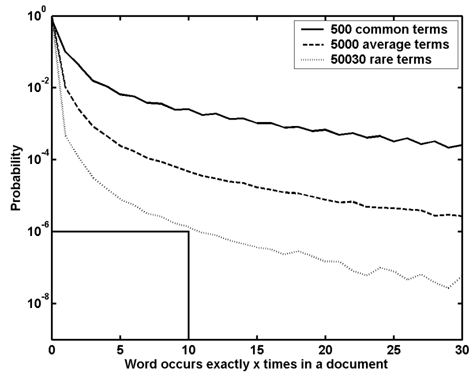


Figure 1. Count probabilities of common, average and rare words in the industry sector corpus. The figure shows, for example, that the probability a given rare word occurs exactly 10 times in a document is 10^{-6}. A ripple effect is seen for common words because no vocabulary pruning is done, so certain HTML keywords such as "font" or "table" occur an even number of times in beginning and ending tags.

3. The burstiness phenomenon

The term "burstiness" (Church & Gale, 1995; Katz, 1996) describes the behavior of a rare word appearing many times in a single document. Because of the large number of possible words, most words do not appear in a given document. However, if a word does appear once, it is much more likely to appear again, i.e. words appear in bursts. To illustrate this behavior, the probability that a given word occurs in a document exactly x times is shown in Figure 1 for the industry sector corpus. Words have been split into three categories based on how often they appear in the corpus. The categories are "common," "average," and "rare." The common words are the 500 most frequent words; they represent 1% of the words in the vocabulary and 71% of the emissions. The average words are the next 5000 most common words; they represent 10% of the vocabulary and 21% of the emissions. The rare words are the rest of the vocabulary (50,030 words) and account for 8% of the emissions.

A few things should be noted about Figure 1. Not surprisingly, common words are more probable than average words which are more probable than rare words. Interestingly, though, the curves for the three categories of words are close to parallel and have similar decay rates. Even though average and rare words are less likely to appear, once a word has appeared, the probability that it will occur multiple times is similar across all words.

Equation (1) shows that it is unlikely under the multi-

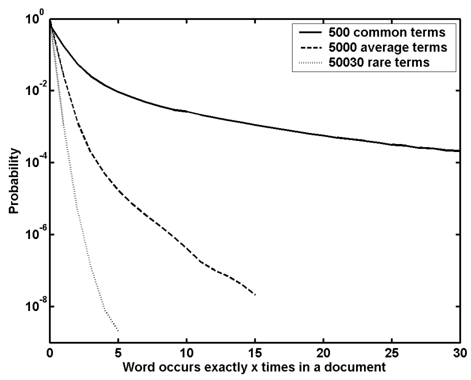


Figure 2. Count probabilities for a maximum likelihood multinomial model, trained with the industry sector corpus.

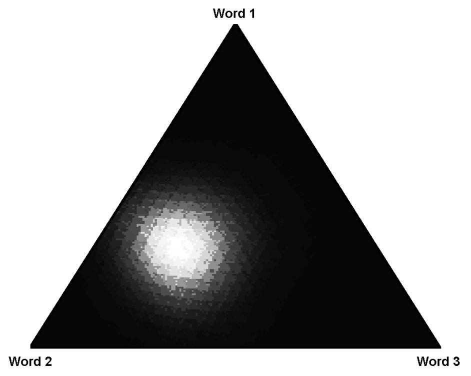


Figure 3. Simplex of possible count vectors using the multinomial bag-of-words model with parameters $0.4375, 0.25, 0.3125$ and $n = 50$.

nomial model for a word to occur many times in a document, because the single word count distribution decays exponentially. Figure 2 shows the average word count probabilities from ten synthetic corpora generated from a multinomial model trained on the industry sector corpus. Each synthetic corpus was generated so its documents have the same length distribution as documents in the industry sector corpus.

The multinomial captures the burstiness of common words, but the burstiness of average and rare words is not modeled correctly. This is a major deficiency in the multinomial model since rare and average words represent 99% of the vocabulary and 29% of emissions and, more importantly, these words are key features for classification. An explanation for this behavior is that the common words are more likely to satisfy the independence assumption, since many of the common words are non-content, function words. The rare and average words are information-carrying words, making them more likely to appear if they have already appeared in a document.

Figure 3 shows the simplex of possible count vectors for three words when a multinomial model is used and the sum of the counts is $n = 50$. All the probability mass is close to the most likely counts, so burstiness is not likely. If bursts were likely, then the probability mass would be on the edges and corners of the simplex.

4. Dirichlet modeling of text

The Dirichlet distribution is a probability density function over distributions. It is defined as

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w\right)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \prod_{w=1}^{W} \theta_w^{\alpha_w - 1}$$

where θ is a vector in the W-dimensional probability simplex, i.e. $\sum_w \theta_w = 1$. The α vector entries are the parameters of the Dirichlet.

When modeling text, the θ vector represents a document, making the model have the form $data^{parameter}$. This form makes Dirichlet models qualitatively similar to Zipf distributions, where the parameter is an exponent. In contrast, the multinomial model has the form $parameter^{data}$. By rewriting the Dirichlet distribution in the exponential family form

$$\begin{aligned} \log p(\theta|\alpha) &= \sum_{w=1}^{W} (\alpha_w - 1) \log \theta_w \\ &+ \log \Gamma(\sum_{w=1}^{W} \alpha_w) - \sum_{w=1}^{W} \log \Gamma(\alpha_w) \end{aligned}$$

we see that the log transform of the data is naturally considered. This is again in contrast to the multinomial in exponential form.

In the bag-of-words representation, documents are vectors of word counts. The Dirichlet distribution is a distribution not over count vectors but over probability vectors. There are multiple ways to represent a document as a probability vector. The obvious choice is to let θ be a scaled version of the document vector. We can view this approach as drawing a scaled bag of words (representing one document) from the Dirichlet bag-of-scaled-documents.

The difficulty with this approach is that document vectors are sparse in nature, i.e. each document tends to contain only a small subset of the vocabulary, resulting in many of the entries being zero. Since the Dirichlet likelihood for a probability vector is zero if the vector contains any zeros, smoothing of the training data is required before a Dirichlet distribution can be estimated. In practice, this results in an over-smoothed distribution, where all the rare words have about the same probability of appearing in all the classes. Since the rare words contain most of the discriminative information, this model is not useful for document classification.

A better approach is hierarchical: let the count vector for each document be generated by a multinomial distribution whose parameters are generated by the Dirichlet distribution. This model is called the Dirichlet compound multinomial (DCM) (Minka, 2003) and can be understood as a bag-of-bags-of-words. To generate a document using the DCM, a sample is first drawn from the Dirichlet to get a multinomial distribution, then words are iteratively drawn for the document based on the multinomial distribution.

Although we talk about the parameters θ, the only actual parameters for a DCM are the α vector entries. The likelihood of a document of length n is an integral over θ vectors weighted by a Dirichlet distribution:

$$\begin{aligned} p(x|\alpha) &= \int_\theta p(x|\theta)p(\theta|\alpha)d\theta \\ &= \int_\theta \frac{n!}{\prod_{w=1}^{W} x_w!}\left(\prod_{w=1}^{W}\theta_w^{x_w}\right)\frac{\Gamma\left(\sum_{w=1}^{W}\alpha_w\right)}{\prod_{w=1}^{W}\Gamma(\alpha_w)}\prod_{w=1}^{W}\theta_w^{\alpha_w-1}d\theta \\ &= \frac{n!}{\prod_{w=1}^{W} x_w!}\frac{\Gamma\left(\sum_{w=1}^{W}\alpha_w\right)}{\prod_{w=1}^{W}\Gamma(\alpha_w)}\int_\theta\prod_{w=1}^{W}\theta_w^{\alpha_w+x_w-1}d\theta \\ &= \frac{n!}{\prod_{w} x_w!}\frac{\Gamma\left(\sum_{w=1}^{W}\alpha_w\right)}{\Gamma\left(\sum_{w=1}^{W}x_w+\alpha_w\right)}\prod_{w=1}^{W}\frac{\Gamma(x_w+\alpha_w)}{\Gamma(\alpha_w)}. \end{aligned} \quad (2)$$

The last step of equation (2) is obtained by noticing that $\prod_{w=1}^{W}\theta_w^{x_w}$ combined with $\prod_{w=1}^{W}\theta_w^{\alpha_w-1}$ is the unnormalized version of the Dirichlet distribution $p(\theta|\alpha+x)$, and using the fact that $\int p(\theta|\alpha)d\theta = 1$.

There exists no closed-form solution for the maximum likelihood parameter values for the DCM model. An iterative gradient descent optimization method can be used to estimate the α vector by computing the gradient of the DCM log likelihood. Two bound inequations are used with the gradient, leading to the update

$$\alpha_w^{new} = \alpha_w \frac{\sum_{d=1}^{D}\Psi(x_{dw}+\alpha_w)-\Psi(\alpha_w)}{\sum_{d=1}^{D}\Psi(x_{dw}+\sum_{w'=1}^{W}\alpha_{w'})-\Psi(\sum_{w'=1}^{W}\alpha_{w'})}$$

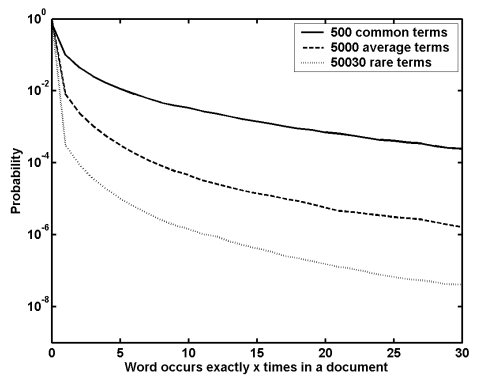


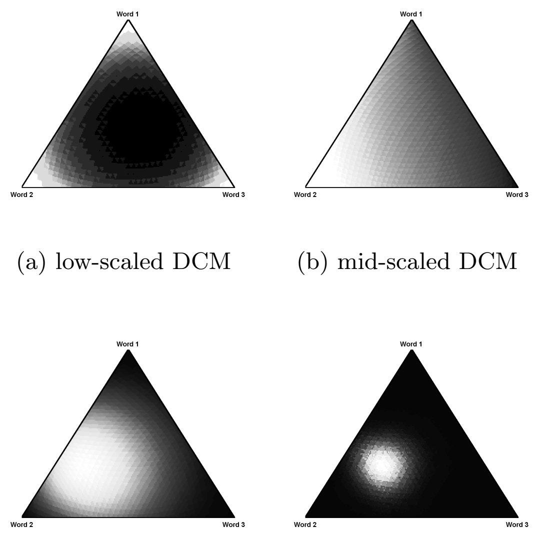
Figure 4. Count probabilities for a maximum likelihood DCM model, trained with the industry sector corpus.

where the digamma function Ψ is defined as $\Psi(\alpha) = \frac{d}{d\alpha}\log\Gamma(\alpha)$. For more information see Minka (2003). After maximum-likelihood fitting, we smooth a DCM model by adding a small constant to each parameter α_w. This constant equals 0.01 times the smallest non-zero fitted α_w.

Figure 2 shows that the multinomial is unable to correctly model the burstiness of natural text. Figure 4 shows the probability of a term appearing multiple times in a document under the DCM model. The experimental design is similar to that of Figure 2. The DCM can model burstiness for all word types. The curves for the three categories of words are close to parallel, like in the original data (Figure 1).

A Dirichlet has the same number of parameters as a multinomial, so it is not obvious how the DCM can model burstiness. The reason is that the multinomial parameters are constrained to sum to one, unlike the DCM parameters, so the DCM has one extra degree of freedom. The smaller the α parameters are, the more the emission of words is bursty. Figure 5 shows the simplex of count probabilities given by equation (2) for three words with $n = 50$, for different α vectors. When the parameters are small, most probability mass is located near the corners of the simplex. When the parameters are large, most mass is near the center of the simplex, modeling word counts that are not bursty. As the parameters tend to infinity, the DCM model approaches equivalence with a multinomial model.

Since a DCM has only a single extra degree of freedom compared to a multinomial, it cannot model the individual burstiness of each word. This inflexibility is a trade-off between the expressiveness of the model and its learnability with limited training data.



(a) low-scaled DCM (b) mid-scaled DCM

(c) high-scaled DCM (d) multinomial

Figure 5. Three word probability simplices with DCM parameters (a) 0.44, 0.25, 0.31 (b) 1.32, 0.75, 0.93 and (c) 3.94, 2.25, 2.81 and multinomial (d) 0.44, 0.25, 0.31.

5. Experiments

We use three standard corpora: the so-called industry sector, 20 newsgroups and Reuters-21578 document collections. We compare DCM models with multinomial models, as well as with heuristically modified versions of the multinomial method, which perform as well as discriminative methods, in particular SVM methods (Rennie et al., 2003). We have made every effort to reproduce previous results in order to ensure that a fair comparison is made.

Documents are preprocessed and count vectors are extracted using the Rainbow toolbox (McCallum, 1996). The 500 most common words are removed from the vocabulary to ensure that our results are comparable with previous results. The Dirichlet toolbox (Minka, 2003) is used to estimate the parameters of the DCM model.

When using DCM or multinomial models for classification, we apply Bayes' rule $p(y|x) = p(x|y)p(y)/p(x)$ with a uniform prior $p(y)$ over the classes, following (Rennie et al., 2003). The class y with the highest probability $p(y|x)$ is the predicted label.

5.1. Heuristics to improve multinomial models

Various heuristics have been applied to the multinomial and related models to enhance classification performance (Rennie et al., 2003). We briefly review them here for completeness. The first heuristic is the log-transformation of the data, which has been shown to mitigate problems caused by burstiness. In the tables of results below, L is shorthand for the transformation

$$x_{dw}^{\log} = \log(1 + x_{dw})$$

where all logarithms are natural (base e). A traditional information retrieval heuristic is the term-frequency inverse-document-frequency (TFIDF) transformation, which exists in various forms (Aizawa, 2003). The version used here includes the log-transformation:

$$x_{dw}^{\text{tfidf}} = \log(1 + x_{dw}) \log \frac{D}{\sum_{d'=1}^{D} \delta_{d'w}}$$

where δ_{dw} is 1 if word w is present in document d. After the TFIDF transformation, document vectors are L_2-normalized:

$$x_{dw}^{\text{norm}} = \frac{x_{dw}^{\text{tfidf}}}{\sqrt{\sum_{w'=1}^{W} {x_{dw}^{\text{tfidf}}}^2}}.$$

This makes all document vectors have the same length and therefore the same amount of influence on the model parameters. The combination of TFIDF and L_2-normalization is denoted TW-L below.

Two additional heuristics are also applied. The most important is complement modeling (Rennie et al., 2003): the model for a class is trained with all the documents that do not belong to that class. Using all other classes makes more data available for parameter estimation, resulting in better modeling of each class:

$$\hat{\theta}_{kw}^{\text{comp}} = \frac{\sum_{d=1:y_d \neq k}^{D} x_{dw} + \varepsilon}{\sum_{w'=1}^{W} \sum_{d=1:y_d \neq k}^{D} x_{dw'} + \varepsilon}$$

where y_d indicates the class that document d belongs to, and ε is a smoothing constant, which is necessary to prevent probabilities for unseen words from becoming zero. Typically, $\varepsilon = 1$, but this value is often too large, so we also report results below with $\varepsilon = 0.01$. Non-complement models are smoothed similarly.

If documents can belong to more than one class, the usual approach is a one-versus-all-but-one classifier. The complement model is the same as the standard model, in these cases. For this reason, the complement model is defined differently for multi-label problems as an all-versus-all-but-one classifier (Rennie et al., 2003, Appendix A).

Finally, the model parameters are log-normalized:

$$\hat{\theta}_{kw}^{\text{norm}} = \frac{\log \hat{\theta}_{kw}^{\text{comp}}}{\sum_{w'=1}^{W} \log \hat{\theta}_{kw'}^{\text{comp}}}$$

making the influence of common words smaller (Rennie et al., 2003). The letter C below denotes complement modeling combined with log-normalization of parameters.

The heuristics described above, and others commonly used with multinomial models for text, modify both input data (word counts) and distribution parameters. Therefore, they do not give probability distributions that are properly normalized, i.e. that sum to one appropriately.

5.2. Document collections

The industry sector[1] data set contains 9555 documents distributed in 104 classes. The data set has a vocabulary of 55,055 words, and each document contains on average 606 words. The data are split into halves for training and testing. The 20 newsgroups[2] data set contains 18,828 documents belonging to 20 classes. This collection has a vocabulary of 61,298 words with an average document length of 116 words. The data are split into 80/20 fractions for training and testing. In the industry and newsgroup data sets each document belongs to one class only.

The Reuters-21578[3] data set contains 21,578 documents. We use the Mod Apte split which only contains 10,789 documents (Apte et al., 1994), those in the 90 classes with at least one training and one test example. The Mod Apte split uses a predefined set of 7,770 training documents and 3,019 test documents. The documents are multi-labeled and can belong to one or more of the 90 classes. This collection has a vocabulary of 15,996 words and the documents have an average length of 70 words.

5.3. Perplexity results

We start by comparing the perplexity of alternative models on the same test data (Blei et al., 2003). When a document is represented as a vector of word counts, its probability includes a factor $n!/\prod_{w=1}^{W} x_w!$ that measures how many word sequences could generate the same vector of counts. We define perplexity over a set of D documents as

$$\exp\left(\frac{-\sum_{d=1}^{D}\sum_{w=1}^{W}\log p(x_{dw})}{\sum_{d=1}^{D} n_d}\right)$$

[1]`www.cs.umass.edu/~mccallum/code-data.html`

[2]`people.csail.mit.edu/people/jrennie/20Newsgroups`

[3]`kdd.ics.uci.edu`

where $p(x)$ does not include the factor $n_d!/\prod_{w=1}^{W} x_{dw}!$. Perplexity on test data measures how well a model predicts unseen data. A lower value indicates better prediction.

The perplexity measure is calculated for the 20 newsgroups data, with one model trained for each of the 20 classes. The perplexity for multinomial models is 5311 ± 755 versus 2609 ± 382 for DCM models, where both results are means $\pm$ one standard deviation calculated over 10 random splits.

The perplexity improvement with DCM models is highly statistically significant. We cannot compute perplexity results for heuristically modified multinomial models, since the transformed parameters and data no longer define a proper probability distribution that sums to one.

5.4. Classification results

The performance of the models is compared on the industry and newsgroup collections using precision $\frac{TP}{TP+FP}$ to measure the accuracy of classification. Here TP is the number of true positives, FP the number of false positives, and FN the number of false negatives.

With multi-labeled data, it is necessary to consider both precision and recall $\frac{TP}{TP+FN}$ to get a fair measure of performance. This is the case for the Reuters data. Following previous work, we combine precision and recall by computing the "break-even" point where precision equals recall. This point can be defined using either micro or macro averaging:

$$\begin{aligned} BE_{micro} &= \frac{1}{N}\sum_{k=1}^{K} N_k \frac{TP_k}{TP_k + FP_k} \\ BE_{macro} &= \frac{1}{K}\sum_{k=1}^{K} \frac{TP_k}{TP_k + FP_k} \end{aligned}$$

where K is the number of document classes, N is the number of documents and N_k is the number of documents in class k. It is not always possible to get exactly the same value for precision and recall, so the average between the two measures is used in these cases. Using micro-averaging, every document is considered equally important. The macro-averaging measure penalizes classifiers that have poor performance on documents from rare classes.

We acknowledge that precision and break-even may not be the best measures of the effectiveness of a text classifier (Sebastiani, 2002), but we use these measures here for comparability with previous work. The results in Tables 1 and 2 are averages over 10 random splits (50/50 for industry sector and 80/20 for 20 news-

Table 1. Classification results for the industry sector collection.

METHOD	SMOOTHING ε	PRECISION $\pm \sigma$
M	1	0.600 ± 0.011
L-M	1	0.654 ± 0.009
M	0.01	0.783 ± 0.008
DCM		0.806 ± 0.006
L-M	0.01	0.812 ± 0.005
TW-L-M	1	0.819 ± 0.004
TW-L-M	0.01	0.868 ± 0.005
C-M	1	0.889 ± 0.006
C-M	0.01	0.889 ± 0.004
C-L-M	0.01	0.899 ± 0.005
C-L-M	1	0.912 ± 0.005
C-DCM		0.917 ± 0.004
C-TW-L-M	0.01	0.919 ± 0.005
C-TW-L-M	1	0.921 ± 0.004

Table 2. Classification results for the 20 newsgroups collection.

METHOD	SMOOTHING ε	PRECISION $\pm \sigma$
M	0.01	0.853 ± 0.004
L-M	0.01	0.865 ± 0.005
TW-L-M	0.01	0.876 ± 0.005
C-M	0.01	0.876 ± 0.005
C-L-M	0.01	0.886 ± 0.005
DCM		0.890 ± 0.005
C-DCM		0.892 ± 0.004
C-TW-L-M	0.01	0.893 ± 0.005

groups), shown $\pm$ one standard deviation σ over the 10 splits.

Table 1 shows the performance of the different algorithms on the industry sector data set. Results using multinomial-based methods are similar to those reported by Rennie et al. (2003) and McCallum and Nigam (1998). Smoothing with $\varepsilon = 0.01$ is clearly better than with $\varepsilon = 1$ for non-complement models. The DCM model produces results that are better than the multinomial and the complement-DCM produces results similar to the multinomial with all heuristics applied.

The results in Table 2 are obtained using the 20 newsgroups data. As for the industry sector data, the DCM model outperforms the multinomial. In this corpus, each class is represented by many examples, so complement modeling is not as useful and the DCM and complement-DCM models perform similarly to the best multinomial with heuristics. We only show results with $\varepsilon = 0.01$, because results with $\varepsilon = 1$ are worse.

Table 3. Classification results for the Reuters collection. The third column shows macro break-even, while the last column shows micro break-even.

METHOD	SMOOTHING ε	MACRO BE	MICRO BE
M	1	0.268	0.761
L-M	1	0.303	0.756
DCM		0.359	0.740
TW-L-M	1	0.390	0.768
M	0.01	0.405	0.741
L-M	0.01	0.407	0.759
TW-L-M	0.01	0.456	0.753
C-TW-L-M	0.01	0.560	0.732
C-L-M	0.01	0.562	0.759
C-M	1	0.563	0.759
C-L-M	1	0.594	0.764
C-M	0.01	0.607	0.776
C-DCM		0.624	0.823
C-TW-L-M	1	0.657	0.840

Table 3 shows results on the Reuters corpus, which is special in that documents contain few words, and many classes only contain a few documents. The DCM and C-DCM methods still perform well. Standard deviations are not given since there is a single standard training set/test set split for this corpus.

We can evaluate the statistical significance of the differences in performance for the industry sector and 20 newsgroups collections. On these two collections, the DCM model outperforms the standard multinomial and a Student's t-test shows that this difference is extremely significant. The complement-DCM model performs slightly worse than the multinomial model with all heuristics applied. A t-test shows that for both data sets, the differences in performance between the complement-DCM model and C-TW-L-M method are not statistically significant.

6. Discussion

We have argued that the Dirichlet compound multinomial (DCM) model is a more appropriate generative model for text documents than the traditional multinomial model. The reason is that a DCM can model burstiness: the phenomenon that if a word appears once, it is more likely to appear again.

We have shown experimentally that the DCM model performs better than the multinomial model for two standard text mining tasks. First, as measured by perplexity, the DCM models a single collection of documents better. Second, when documents are classified using Bayes' rule using a generative model for each of the alternative classes, accuracy using a DCM model for each class is higher than when using a multinomial model for each class. When the most effective known heuristics are applied in addition, accuracy using multinomial models versus using DCM models is similar.

The DCM model is a generative model for the documents within a class. Given a Dirichlet distribution, a document is not generated directly. Instead, the Dirichlet is used to generate a multinomial; this multinomial is then used to generate the document. Conceptually, different documents within the same class are generated by different multinomials. This procedure allows for diversity within the class. The words that are bursty in a particular document are those that have high probability in the particular multinomial used to generate this document.

A DCM model can represent a topic (i.e. a class of documents) where different documents use alternative terminology. For example, some automotive documents may use the word "hood" while others use the word "bonnet." This within-topic diversity is different from the within-document diversity allowed by latent topic modeling, where each topic is represented by a single multinomial, but each word in a document may be generated by a different topic (Deerwester et al., 1990; Hofmann, 1999; Blei et al., 2003).

We hope that many applications of text modeling in addition to those outlined in this paper will benefit from using DCM models in the future.

Acknowledgments. This paper is based in part on work sponsored by the Air Force Research Laboratory under contract F30602-03-C-0075, performed in conjunction with Lockheed Martin Information Assurance. This research was also supported in part by Fair Isaac, Inc. and by Sun Microsystems, Inc.

References

Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing and Management*, *39*, 45–65.

Apte, C., Damerau, F. J., & Weiss, S. M. (1994). Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, *12*, 233–251.

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Church, K. W., & Gale, W. A. (1995). Poisson mixtures. *Natural Language Engineering*, *1*, 163–190.

Deerwester, S., Dumais, S., Landauer, T., Furnas, G., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41*, 391–407.

Hofmann, T. (1999). Probabilistic latent semantic indexing (PLSI). *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval* (pp. 50–57). Berkeley, California: ACM.

Jansche, M. (2003). Parametric models of linguistic count data. *41st Annual Meeting of the Association for Computational Linguistics* (pp. 288–295). Sapporo, Japan.

Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*, 11–21.

Katz, S. M. (1996). Distribution of content words and phrases in text and language modelling. *Natural Language Engineering*, *2*, 15–59.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. *Proceedings of ECML-98, 10th European Conference on Machine Learning* (pp. 4–15). Chemnitz, Germany: Springer Verlag, Heidelberg, Germany.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive Bayes text classification. *AAAI/ICML-98 Workshop on Learning for Text Categorization* (pp. 41–48). AAAI Press.

McCallum, A. K. (1996). *Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering.* www.cs.cmu.edu/˜mccallum/bow.

Minka, T. (2003). *Estimating a Dirichlet distribution.* www.stat.cmu.edu/˜minka/papers/dirichlet.

Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive Bayes text classifiers. *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 616–623). Washington, D.C., US: Morgan Kaufmann Publishers, San Francisco, US.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*, 613–620.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*, 1–47.

Teevan, J., & Karger, D. R. (2003). Empirical development of an exponential probabilistic model for text retrieval: Using textual analysis to build a better model. *Proceedings of the 26th Annual ACM Conference on Research and Development in Information Retrieval (SIGIR '03)* (pp. 18–25). Toronto, CA.

Zipf, G. (1949). *Human behaviour and the principle of least effort: An introduction to human ecology.* Addison-Wesley.