The Smoothed-Dirichlet distribution: A new building block for generative topic models

Ramesh Nallapati NMRAMESH@CS.UMASS.EDU
University of Massachusetts, Amherst, MA 01002, USA

Thomas Minka and Stephen Robertson {MINKA,SER}@MICROSOFT.COM
Microsoft Research, Cambridge, U.K.

Abstract

In this work, we present the Smoothed Dirichlet (SD) distribution, as an alternative distribution to the multinomial and the more recent Dirichlet-Compound-Multinomial (DCM) distributions as a basic building block for generative topical models for text. We show that this distribution is as simple to estimate as the multinomial and as effective in capturing term occurrence statistics as the DCM, thus combining the most desirable properties of these two distributions into one unit. We also argue that the particular form of KL-divergence ranking function used successfully in information retrieval performs well for the simple reason that it corresponds to log-likelihood w.r.t. the Smoothed-Dirichlet distribution, a better generative model of text. We compared various generative distributions for text on the task of text classification and found that besides outperforming the multinomial, SD is also significantly better than the DCM and ordinary Dirichlet distributions. Therefore it deserves serious consideration as a new building block in generative models for text.

1. Introduction

Generative topical models for classification or clustering of textual data rely on a base distribution to generate text. Choosing the right distribution that fits empirical data distribution accurately is critical for optimal performance of these models. In the past, several distributions such as Multiple-Bernoulli (Robertson & Jones, 1976) and mixture of Poissons (Robertson et al., 1981) were considered as potential generators of text.

In the recent past, the multinomial distribution has become the *de facto* distribution for generative models of text since it is not only effective, but is also very simple and easy to estimate and draw inference about. Models ranging from the simple naive-Bayes classifier (McCallum & Nigam, 1998) to the more complex topical mixture models such as the LDA (Blei et al., 2002) use the multinomial distribution, shown below as the basic building block to generate documents:

$$Pr(\mathbf{f}|\bar{\theta}, L) = \frac{L!}{\prod_{j=1}^{V} f_j!} \prod_{j=1}^{V} \theta_j^{f_j} \qquad (1)$$

where $L = \sum_j f_j$ is the document length, $\mathbf{f} = \{f_1, \cdots, f_V\}$ is the counts-vector representation of the document and f_j is the raw-count of occurrence of the j^{th} word in the document, V is the vocabulary size and $\bar{\theta}$ is the parameter vector of the multinomial. The generative process of the multinomial consists of L repeated i.i.d. samplings of words from the distribution to obtain the vector $\mathbf{f}$ as shown in figure 1(a).

Recent work has found that the multinomial hugely under-predicts the heavy tail or burstiness behavior[1] of term occurrence (Teevan & Karger, 2003; Rennie et al., 2003). To mitigate this problem, (Madsen et al., 2005) proposed the Dirichlet Compound Multinomial (DCM) distribution for text and showed that it models word burstiness better than the multinomial distribution. They also achieved improvements over the multinomial in the text classification task. The DCM distribution uses the same counts-vector representation $\mathbf{f}$ for documents but its generative process consists of sampling a multinomial distribution from a Dirichlet prior from which the document is sampled as shown in figure 1(b). Given the parameters $\bar{\alpha}$ of the prior, the probability of the counts-vector $\mathbf{f}$ is given by:

$$\begin{aligned} Pr(\mathbf{f}|\bar{\alpha}) &= \int Pr(\mathbf{f}|\bar{\theta}) Pr(\bar{\theta}|\bar{\alpha}) d\bar{\theta} \\ &= \frac{L!}{\prod_{j=1}^{V} f_j!} \frac{\Gamma(\sum_j \alpha_j)}{\Gamma(\sum_j (\alpha_j + f_j))} \frac{\prod_j \Gamma(\alpha_j + f_j)}{\prod_j \Gamma(\alpha_j)} \qquad (2) \end{aligned}$$

where Γ is the Gamma function. Despite impressive gains in performance compared to the multinomial, the

[1] words are much more likely to occur in a text once they have occurred once

CIIR Technical Report Do not distribute.

downside of DCM is the non-availability of a closed-form solution. It requires computationally expensive iterative techniques to estimate its parameters (Minka, 2003) and as a result may not be very attractive for many information retrieval related tasks where quick response to the user is of utmost importance.

One of the main motivations of the current work is to identify a distribution that is at least as effective as the DCM in capturing term occurrence statistics but as simple as the multinomial in estimation and inference. The rest of the paper is organized as follows. In section 2 we describe the new SD distribution and its approximation and show that its estimation corresponds to a simple geometric averaging of term statistics and inference is equivalent to the standard form of KL-divergence used in IR. In section 3, we empirically demonstrate that the SD fits data much better than the multinomial and as well as the DCM. In section 4, we argue that SD distribution justifies KL-divergence by virtue of it being a better model of text. Our experiments in text classification reported in section 5, show that SD is an effective replacement for the multinomial and DCM distributions. We end the paper in section 6 with a discussion on future work.

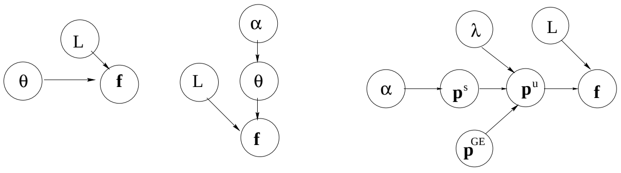
2. Smoothed Dirichlet distribution

We here describe the generative process of the Smoothed Dirichlet distribution. The rationale for this process is discussed below in sub-section 2.1. As shown in figure 1(c), we first generate a smoothed proportions vector $\mathbf{p}^s$, from the SD distribution and unsmooth it to get the raw proportions $\mathbf{p}^u$ as follows:

$$\mathbf{p}^u \stackrel{\text{def}}{=} (\mathbf{p}^s - (1-\lambda)\mathbf{p}^{GE})/\lambda \tag{3}$$

where $\mathbf{p}^{GE}$ is the proportions of words in general English and $0 < \lambda < 1$ is a smoothing parameter. The unsmoothed proportions $\mathbf{p}^u$ are then converted into a bag of words $\mathbf{f}$ given the document length L, using the relation $\mathbf{f} = \text{int}(L\mathbf{p}^u)$ where int() is a function that returns the nearest integer-vector to its real-vector argument. Only the generation of $\mathbf{p}^s$ is probabilistic and its conversion to unsmoothed proportions $\mathbf{p}^u$ and then to bag of words $\mathbf{f}$ is completely deterministic. Hence the probability of generating a counts vector $\mathbf{f}$ under SD distribution is same as that of generating the smoothed proportions vector $\mathbf{p}^s$ given by:

$$Pr(\mathbf{f}|\mathbf{p}^{GE}, \lambda, L, \bar{\alpha}) = Pr(\mathbf{p}^s|\bar{\alpha}) = \frac{1}{Z^{SD}(\bar{\alpha})}\prod_{j=1}^{V} p_j^{s^{\alpha_j - 1}} \tag{4}$$



(a) Multinomial (b) DCM (c) Smoothed Dirichlet

Figure 1. Graphical representation of the distributions

where $\bar{\alpha}$ is the parameter vector of the smoothed-Dirichlet distribution and Z^{SD} is the SD-normalizer that guarantees the probabilities add up to 1. From an inference perspective, given a counts-vector representation $\mathbf{f}$ of a document, estimating its probability under SD is follows: we first get a raw proportions representation of the document using the relation $\mathbf{p}^u = \mathbf{f}/L$ and then get a smoothed proportions representation using the inverse of relation (3), *i.e.*,

$$\mathbf{p}^s = \lambda\mathbf{p}^u + (1-\lambda)\mathbf{p}^{GE} \tag{5}$$

and then compute its probability under the SD distribution as given by (4). Thus, $\mathbf{p}^u$ corresponds to raw-counts in a document normalized by document length L and $\mathbf{p}^s$ corresponds to a mixture of $\mathbf{p}^u$ and general English proportions $\mathbf{p}^{GE}$ with λ as the mixing weight.

2.1. Rationale

The reason we generate the smoothed document representation $\mathbf{p}^s$ and not the raw-proportions $\mathbf{p}^u$ directly is to avoid assigning zero probability to any document: the raw-proportions $\mathbf{p}^u$ of a document is typically a sparse vector with many zeros in it and as such, if we replace $\mathbf{p}^s$ with $\mathbf{p}^u$ in (4), we end up with a zero probability for almost all documents.

Notice that the functional form of the SD distribution defined in (4) is same as the ordinary Dirichlet distribution (Minka, 2003). One may argue that we could use the ordinary Dirichlet distribution to generate the smoothed proportions $\mathbf{p}^s$ instead of defining a new distribution. However, the Dirichlet distribution is incorrect for smoothed proportions because it assigns probability mass to the entire simplex $\Delta = \{\mathbf{p} \mid \forall_j p_j > 0; \sum_j p_j = 1\}$ while smoothed proportions occupy only a subset Δ^s of the simplex. To illustrate this phenomenon, we generated 1000 documents of varying lengths uniformly at random using a vocabulary of size 3, converted them to raw-proportions $\mathbf{p}^u$, smoothed them with $\mathbf{p}^{GE}$ estimated from the entire document set, and plotted the smoothed-proportions $\mathbf{p}^s$ vectors in figure 2. The leftmost plot represents the unsmoothed proportion vectors $\mathbf{p}^u$ corresponding to $\lambda = 1$. As shown in the plot, the documents cover the whole simplex Δ when not smoothed. But as we increase the degree of smoothing, the new domain Δ^s spanned by the smoothed documents gets compressed towards the centroid. The compressed domain Δ^s in figure 2 corresponds to the set of all feasible values of $\mathbf{p}^s$ that guarantee that the corresponding $\mathbf{p}^u$'s as de-

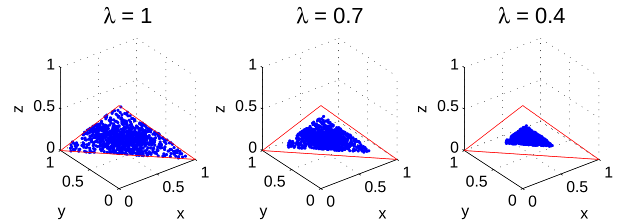


Figure 2. Domain of smoothed proportions Δ^s for various degrees of smoothing: dots are smoothed-proportions vectors $\mathbf{p}^s$ and the triangular boundary is the 3-D simplex.

fined in (3) lie in the simplex Δ. Hence, the Dirichlet normalizer, that considers the whole simplex Δ as its domain, shown in (6), is clearly incorrect given our smoothed document representation.

$$Z(\bar{\alpha}) = \int_{\Delta} \prod_j p_j^{\alpha_j - 1} d\mathbf{p} = \frac{\prod_j \Gamma(\alpha_j)}{\Gamma(\sum_j \alpha_j)} \tag{6}$$

Despite this flaw, Dirichlet can still be considered as an approximate distribution for smoothed proportions and we do compare its performance with SD in the following sections. The SD distribution rectifies this flaw in the Dirichlet distribution by defining a normalizer that assigns the probability mass only to the new compressed domain Δ^s.

2.2. SD normalizer and its approximator

When we use smoothed representation for documents, the integral in (6) should span only over the compressed domain Δ^s that contains all the smoothed documents, as given by the following expression:

$$\Delta^s = \{\mathbf{p}^s\} = \{\lambda \mathbf{p}^u + (1-\lambda)\mathbf{p}^{GE} \mid \mathbf{p}^u \in \Delta\} \tag{7}$$

The above equation is a transform for $\mathbf{p}^s$ from its domain Δ^s into Δ. Exploiting this mapping, we can define the exact analytical form of the normalizer for smoothed documents Z^{SD} in Δ as:

$$\begin{aligned} Z^{SD} &= \int_{\Delta^s} \prod_{j=1}^{V} (p_j^s)^{\alpha_j - 1} d\mathbf{p}^s \\ &= \int_{\Delta} \prod_{j=1}^{V} \{\lambda p_j^u + (1-\lambda) p_j^{GE}\}^{\alpha_j - 1} \lambda d\mathbf{p}^u \end{aligned} \tag{8}$$

For fixed values of λ and $\mathbf{p}^{GE}$, Z^{SD} can be transformed to an incomplete integral of the multi-variate Beta function. However, this has no straight-forward analytic solution. In the reminder of this subsection, we will focus on developing a theoretically motivated approximation Z_a^{SD} for the SD normalizer of (8).

Figure 3(a) compares Z^{SD} with the Dirichlet normalizer Z of (6) for a simple case where the vocabulary size V is 2, *i.e.*, $\bar{\alpha} = \{\alpha_1, \alpha_2\}$. We imposed the condition that $\alpha_1 + \alpha_2 = 1$ and used $\lambda = 0.2$ and $\{p_1^{GE}, p_2^{GE}\} = \{0.5, 0.5\}$. The plot shows the value of Z^{SD} for various values of α_1 computed using the incomplete two-variate Beta function implementation of *Matlab*. Notice that Z^{SD} tends to finite values at the boundaries while Z, the Dirichlet normalizer is unbounded. We would like to define Z_a^{SD}, an approximation to Z^{SD} such that it not only shows similar behavior to Z^{SD}, but is also analytically tractable. Taking cue from the functional form of the Dirichlet normalizer Z in (6), we define Z_a^{SD} as:

$$Z_a^{SD}(\bar{\alpha}) = \prod_j \Gamma_a(\alpha_j) \ / \ (\Gamma_a(\sum_j \alpha_j)) \tag{9}$$

where $\Gamma_a(\alpha)$ is an approximation to $\Gamma(\alpha)$. Now all that remains is to choose a functional form for $\Gamma_a(\alpha)$ such that Z_a^{SD} closely approximates the SD normalizer Z^{SD} of (8). We turn to the Stirling's approximation of the Gamma function (Abramowitz & Stegun, 1972), shown in (10) for guidance.

$$\Gamma(\alpha) \approx e^{-\alpha} \alpha^{\alpha - 1/2} \sqrt{2\pi} (1 + \frac{1}{12\alpha} + O(\frac{1}{\alpha^2})) \tag{10}$$

Figure 3(b) plots the Γ function and its Stirling approximation which shows that $\Gamma(\alpha) \to \infty$ in the limit as $\alpha \to 0$. Inspecting (6), it is apparent that this behavior of the Γ function is responsible for the unboundedness of Dirichlet normalizer at small values of α. Since our exact computation in low dimensions shows that the Smoothed Dirichlet normalizer Z^{SD} is bounded as $\alpha \to 0$, we need a bounded approximator of Γ. An easy way to define this approximation is to ignore the terms in Stirling's approximation that make it unbounded and redefine it as:

$$\Gamma_a(\alpha) \overset{\text{def}}{=} e^{-\alpha} \alpha^{\alpha} \tag{11}$$

While there are several ways to define a bounded approximation, we chose an approximation that is not only mathematically simple, but also yields a closed form solution to maximum likelihood estimation as we will show later. The approximate function Γ_a is compared to the exact function Γ again in figure 3(b). Note that the approximate function yields bounded values at low values of α but closely mimics the exact function at larger values. Combining (9) and (11), we have:

$$Z_a^{SD}(\bar{\alpha}) = \frac{\prod_j e^{-\alpha_j} \alpha_j^{\alpha_j}}{e^{-\sum_j \alpha_j} (\sum_j \alpha_j)^{\sum_j \alpha_j}} = \frac{\prod_j \alpha_j^{\alpha_j}}{S^S} \tag{12}$$

where $S = \sum_j \alpha_j$. The approximation in (12) is independent of λ and $\mathbf{p}^{GE}$ which is clearly an oversimplification of the exact SD normalizer Z^{SD} in (8). However our plot of the approximate SD normalizer Z_a^{SD} in figure 3(a) shows that it behaves very similar to Z^{SD}. Our new approximate Smoothed Dirichlet distribution can now be defined as:

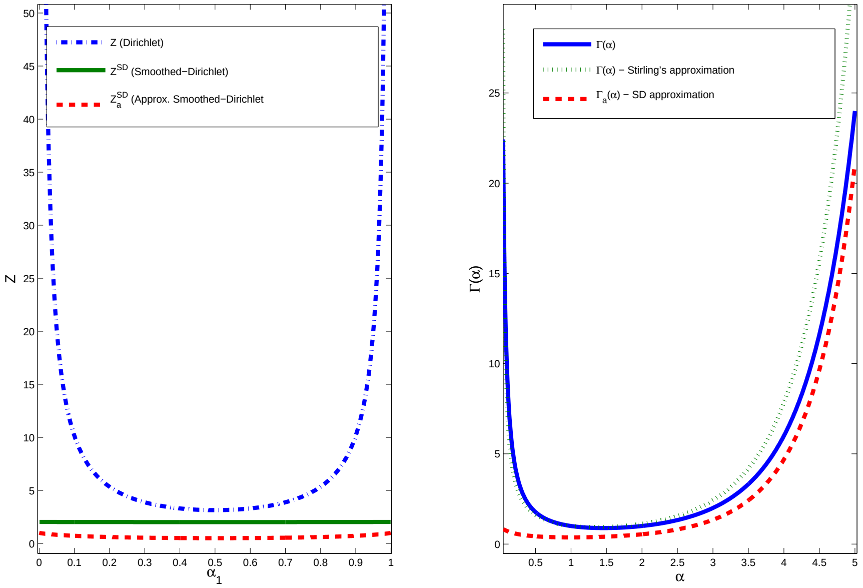


Figure 3. (a) Comparison of the normalizers (b) Gamma function and its approximators

$$Pr_a(\mathbf{f}|\mathbf{p}^{GE},\lambda,L,\bar{\alpha}) = Pr_a(\mathbf{p}^s|\bar{\alpha}) = \frac{S^S}{\prod_j \alpha_j^{\alpha_j}} \prod_j p_j^{s^{\alpha_j - 1}} \tag{13}$$

Henceforth, we will refer to the approximate SD distribution as the SD distribution for convenience. The subscript in Pr_a helps remind us that it is an approximate probability density function.

2.3. Estimation

Given a set of N documents $\{\mathbf{p}_1^s, \cdots, \mathbf{p}_N^s\}$ where each $\mathbf{p}_i^s$ is a smoothed-proportion vector representation of the i^{th} document, the maximum likelihood estimates (MLE) of the SD parameter vector $\bar{\alpha}$ are given by the values that maximize the Smoothed-Dirichlet likelihood-function shown in (13). Differentiating the log-likelihood function for N documents with respect to each α_j with an additional Lagrange multiplier term with the constraint that $\sum \alpha_j = S$ and equating to zero gives us the following closed-form solution for $\bar{\alpha}$

$$\bar{\alpha} = \{\prod_{i=1}^{N} \mathbf{p}_i^s\}^{\frac{1}{N}}/Z \tag{14}$$

Here, Z is a normalizer that ensures $\sum_j \alpha_j = S$. We consider S a free parameter that scales individual α_j's proportionately. It is easy to verify that the second derivative of the log-likelihood function is always less than zero guaranteeing convexity of the log-likelihood function and thereby the global optimality of the MLE solution. Thus, the SD distribution provides a closed form solution for training where our estimates of $\bar{\alpha}$ are simply normalized geometric averages of the smoothed proportions of words in training documents.

2.4. Inference

In a multi-class classification task, inference consists of finding the best class C_{best} for a given test document $D \equiv \mathbf{p}^s$. The best class is decided using the Bayes' rule as shown in table 1, where step (2) follows from (13) and in step (3) we assume that all classes have the same value for S and a uniform prior π_C. It is interesting to note that our inference mechanism chooses the class C whose parameter vector $\bar{\alpha}^C$ is closest to the document's feature vector $\mathbf{p}^s$ in terms of the KL-divergence, which is a distance metric between two probability distributions. In our case, the parameter vector $\bar{\alpha}$ can be considered a probability distribution over the vocabulary in the special case when $S = 1$.

3. Data analysis

We used a Porter-stemmed but not stopped version of Reuters-21578 corpus for our experiments. Similar to the work of Madsen *et al* (Madsen et al., 2005),we sorted words based on their frequency of occurrence in the collection and grouped them into three categories, W_h, the high-frequency words, comprising the top 1% of the vocabulary and about 70% of the word occurrences, W_m, medium-frequency words, comprising the next 4% of the vocabulary and accounting for 20% of the occurrences and W_l, consisting of the remaining 95% low-frequency words comprising only 10% of occurrences. We pooled within-document counts f of all words from each category in the entire collection and computed category-specific empirical distributions of proportions $Pr(f|W_h), Pr(f|W_m)$ and $Pr(f|W_l)$.

We did maximum likelihood estimation of the parame-

$$\begin{array}{lll}
C_{best} & = \arg\max_C \log Pr_a(C|D, \bar{\alpha}^C) = \arg\max_C \log Pr_a(D|\bar{\alpha}^C, C)\pi_C & \text{step (1)} \\
 & = \arg\max_C \{S \log S - \sum_j \{\alpha_j^C \log \alpha_j^C - (\alpha_j^C - 1) \log p_j^s\} + \log \pi_C\} & \text{step (2)} \\
 & = \arg\max_C \{- \sum_j \alpha_j^C \log(\alpha_j^C / p_j^s)\} = \arg\max_C \{-KL(\bar{\alpha}^C || \mathbf{p^s})\} & \text{step (3)}
\end{array}$$

Table 1. Inference in SD: $\bar{\alpha}^C$ are the parameters of SD distribution of class C, π_C is the prior probability of class C.

ters of Multinomial, DCM, Dirichlet and SD distributions using the entire collection. For Dirichlet and SD, we fixed the value of the smoothing parameter λ at 0.9. To train the Dirichlet and DCM distributions, we used iterative techniques to estimate the mean, keeping the precision S at constant, as described in (Minka, 2003) using the *fastfit*[2] toolkit.

In case of multinomial and DCM distributions, the probability that a word w_j occurs at count f in a document of length L, $Pr(f|L, \theta_j)$ is given by their marginals, which are the binomial and the beta-binomial respectively. To compute the probability that it occurs at count f in any document, $Pr(f|\theta_j)$, we marginalize the distribution over the document length using the relation $Pr(f|\theta_j) = \sum_L Pr(f|\theta_j, L)P(L)$ where we estimated $Pr(L)$ from the corpus.

The Dirichlet marginal for a single variable is a Beta distribution. We assume that the marginal of the SD distribution has the same parametric form:

$$Pr_a(p_j|\bar{\alpha}) = \frac{S^S}{\alpha_j^{\alpha_j}(S - \alpha_j)^{S - \alpha_j}} (p_j^s)^{\alpha_j - 1}(1 - p_j^s)^{S - \alpha_j - 1} \tag{15}$$

For these distributions, the probability that a word w_j occurs at count f is given by $Pr(f|\alpha_j) = \sum_L Pr(p^u = f/L|\alpha_j, L)Pr(L)$. Next, for each distribution, we evaluated the category-specific probabilities by averaging word probabilities in each set and normalizing them over different values of f. We also tuned the value of the free-parameter S in DCM, Dirichlet and SD distributions until their plots were as close a visual-fit as possible to the empirical distributions. We caution that since we did not use any objective function to optimize the plots, they are only for illustration purposes. Figure 4 compares the predictions of each distribution with the empirical distributions for each category. The data plots corresponding to empirical distribution exhibit a heavy tail on all three categories W_h, W_m and W_l as noticed by earlier researchers (Rennie et al., 2003; Madsen et al., 2005). The multinomial distribution predicts the high frequency words well while grossly under-predicting the medium and low frequency words. The Dirichlet and SD distributions fit the data much better than the multinomial on all three sets, validating our choice of the particular functional form to model text. The plots also show that the DCM distribution is a good a fit to data as shown by Madsen *et al* (Madsen et al., 2005). While it is unclear which of SD and DCM is a better fit, the advantage of SD lies in its simplicity in estimation while achieving a good fit at the same time.

4. Relation to IR ranking functions

Language models for information retrieval use the multinomial distribution to model topics, but they differ from the typical multinomial models used in text classification mainly in the way the ranking function is defined. Given an estimate of the query's multinomial distribution $\bar{\theta}^Q$, a natural ranking function would be the document's log-likelihood w.r.t. the query, $\log Pr(\mathbf{f}|\bar{\theta}^Q, L)$ as defined in (1), which can be shown proportional to $-CE(\mathbf{p}^s||\bar{\theta}^Q)$ where $\mathbf{p}^s$ is the smoothed-proportions representation of the document. However, language models employ its assymetric counterpart, namely $-CE(\bar{\theta}^Q||\mathbf{p}^s)$ as the ranking function (Lafferty & Zhai, 2001), which is proportional to the query's log-likelihood with respect to the document's multinomial model $\mathbf{p}^s$. There is empirical evidence that ranking functions of the form $-CE(\bar{\theta}^Q||\mathbf{p}^s)$ perform better than the form $-CE(\mathbf{p}^s||\bar{\theta}^Q)$, using the same values of parameters (Lavrenko, 2004), but no theoretical justification has been offered.

Notice that log-likelihood of a document w.r.t. the SD distribution given a query's parameters $\bar{\alpha}^Q$ is proportional to $-KL(\bar{\alpha}^Q||\mathbf{p}^s)$ as shown in step (3) of table 1. This is rank-equivalent to $-CE(\bar{\alpha}^Q||\mathbf{p}^s)$ [3], which is equivalent to the ranking function of language models. Thus our work offers an explanation why $-CE(\bar{\theta}^Q||\mathbf{p}^s)$ performs better than $-CE(\mathbf{p}^s||\bar{\theta}^Q)$: the latter corresponds to an underlying multinomial distribution, while the former corresponds to SD distribution, a better fit to textual data, as shown in section 3. Language models, although based on multinomial distribution, manage state-of-the-art performance by simply using a ranking function based on a better modeler of text. In this work, we have removed this inconsistency by uncovering the underlying distribution that corresponds to the successful cross-entropy ranking function.

5. Text Classification

We used the 20 Newsgroups, Reuters-21578 and Industry-Sector corpora. All three collections used in our experiments were stopped and stemmed using Porter stemmer. For any of the collections or models,

[2] http://research.microsoft.com/~minka/software/fastfit

[3] $KL(\bar{\alpha}^Q||\mathbf{p}^s) = -H(\bar{\alpha}^Q) + CE(\bar{\alpha}^Q||\mathbf{p}^s)$ and the entropy term $H(\bar{\alpha}^Q)$ is independent of documents

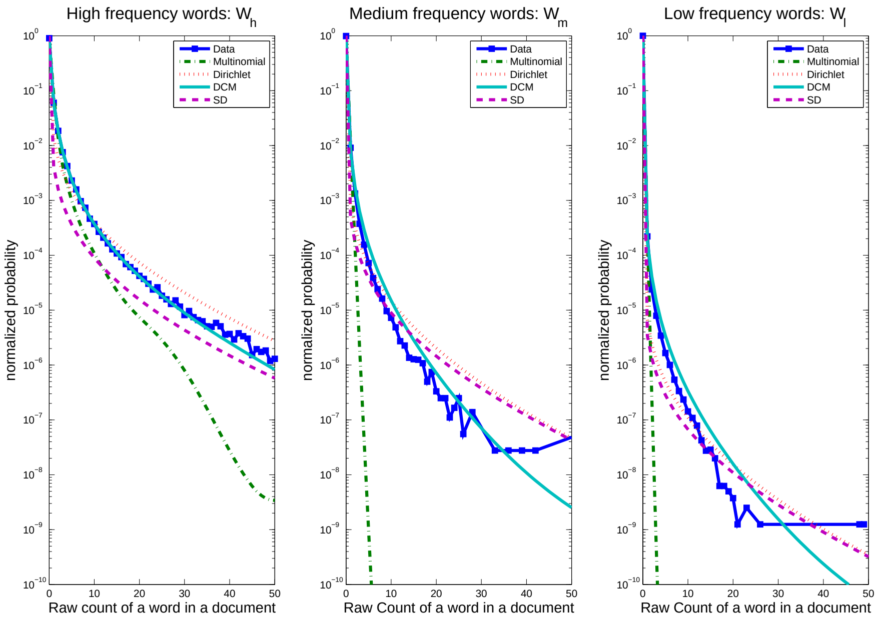


Figure 4. Comparison of predicted and empirical distributions

we did not do any feature selection, as we consider it a separate problem altogether. We indexed the collection using *Lemur*[4] toolkit, version 3.0. We performed all our experiments on *Matlab* using the document-term matrix obtained from *Lemur*'s output.

The version of the 20 Newsgroups collection we used has 18,828 documents and 20 classes. The Industry-sector corpus has 9569 documents and 104 classes. Since documents in this collection are web pages, we used the HTML parser of *Lemur* to pre-process the documents. We randomly split documents in each class into train-test subsets at a ratio of 80:20 on both of these collections. We repeated this process 25 times to obtain as many versions of train-test splits to experiment on. We used the Mod-Apte (Apte et al., 1994) split of the Reuters-21578 collection that consists of 12,902 documents and a predefined train-test split. We used only 10 most popular classes for our experiments as done in (McCallum & Nigam, 1998).

Additionally, to facilitate learning the values of free parameters, for each of the three collections, we randomly picked one of the training sets (it is unique in case of Reuters) and further randomly split them class-wise, in the same ratio as the corresponding train-test split, into sub-training and validation sets.

In case of Industry Sector and 20 Newsgroups data, documents are uniquely labeled. Hence we built multi-way classifiers and used the standard classification accuracy as evaluation metric, which is defined as the percentage of test documents that are correctly labeled. In case of the Reuters collection, documents can belong to multiple classes. Following several other authors(McCallum & Nigam, 1998; Madsen et al., 2005; Rennie et al., 2003), we built one-versus-all classifiers for all classes. For each of these classes C, we then ranked all the test documents in the decreasing order of posterior log-odds ratio and Macro-averaged Break Even Precision (BEP) as our evaluation measure.

5.1. Experiments

Maximum likelihood parameter estimates (MLE) of any model are typically smoothed to avoid zeros that may result when there are words in the test set that are never seen in the training set. For the multinomial, we used two kinds of smoothing as shown below:

$$\text{Laplacian:} \quad \theta_j^C = \frac{\sum_{i \in C} f_{ij} + \delta}{\sum_{i \in C} \sum_j f_{ij} + V\delta} \quad (16)$$

$$\text{Jelinek-Mercer:} \quad \theta_j^C = \lambda \theta_j^{MLE} + (1-\lambda) p_j^{GE} (17)$$

where θ_j^C is the multinomial parameter of the j-th word in class C, λ and δ are free parameters. p_j^{GE} is estimated as follows:

$$p_j^{GE} = \frac{\sum_i f_{ij} + \beta}{\sum_i \sum_j f_{ij} + V\beta} \quad (18)$$

where the index i ranges only over the entire set of training documents and β is another free parameter. Laplacian smoothing shown in (16) is more common in text classification research while Jelinek-Mercer (JM) smoothing shown in (17) is popular in IR and is shown to boost performance (Zhai & Lafferty, 2004). For the DCM model, smoothing is done as follows:

$$\alpha_j^C = S \frac{\alpha_j^{MLE} + \delta}{\sum_j \alpha_j^{MLE} + V\delta} \quad (19)$$

[4] http://www.lemurproject.org

For the Dirichlet and SD distributions, we already use smoothed document proportions as shown in (5), so we do not expect any zeros in our parameter estimates. For DCM and Dirichlet, we consider S as a free parameter. In SD, the value of S does not influence inference or learning. Hence we fix $S = 1$, allowing us to treat the SD parameter vector $\bar{\alpha}$ as a probability distribution over the vocabulary.

To learn the optimal values of the free parameters of the models, we did maximum-likelihood training on the sub-training set first and then performed a simple hill-climbing on the domain of the free parameters until the evaluation criterion is optimized on the validation set. We then performed regular maximum likelihood training and testing on all train-test splits, fixing the free parameters at these optimal values. On Industry sector and 20 Newsgroups corpora, we performed statistical significance tests using two-tailed paired T-test at a confidence level of 95%.

We also tested a variant of the SD inference formula of table 1, which we call SD-CE, as described below. We noted in section 4 that $KL(\bar{\alpha}^C||\mathbf{p}^s)$ and $CE(\bar{\alpha}^C||\mathbf{p}^s)$ are equivalent in a setting where documents are ranked w.r.t. one class, as in the Reuters collection in our experiments. However in case of choosing the best class for each document as in the 20 Newsgroups and Industry-sector data sets, they are no longer equivalent. In such cases, KL-divergence minimization as in step (3) of table 1 amounts to maximizing the entropy $H(\bar{\alpha}^C)$ besides minimizing the cross entropy $H(\bar{\alpha}||\mathbf{p}^s)$. We believe maximization of entropy of the model is a consequence of our modeling assumptions that is not necessarily desirable in a labeling setting. Hence in SD-CE, we considered only the cross-entropy term $CE(\bar{\alpha}^C||\mathbf{p}^s)$ for inference, rest being the same as SD.

Additionally, we tested a linear SVM as a standard discriminative baseline using a one-versus-all SVM^{light} toolkit for Reuters and SVM^{multiclass} toolkit for the other two data-sets (Joachims, 1999). As features, we used normalized TF-IDF weights defined by tf× $\log((N+1)/(n+0.5))$ where tf is the raw count of a term in a document, N is the total number of training documents and n is the number of training documents the term occurs in. We used the parameter C that represents trade-off between margin maximization and training error as a free parameter during training.

Table 2 presents the results of our experiments on all three datasets. Our experiments show that SD outperforms the Laplace smoothed multinomial, the DCM and ordinary Dirichlet on all collections. On the two collections on which we could do significance tests, the difference with the nearest model is found to be statistically significant. However, the JM smoothed multinomial improves on the Laplace smoothed one, as observed in IR experiments (Zhai & Lafferty, 2004), in two of the three collections. On Industry-Sector corpus, the improvement is remarkable and it outperforms the SD distribution by a statistically significant margin too. These results are in line with those of (McCallum et al., 1998) wherein the authors performed a similar smoothing in a hierarchical classification setting which they called shrinkage. Note that our approximation to the SD inference, SD-CE outperforms all distributions including JM smoothed Multinomial and on all collections justifying our intuition behind the modified inference formula in SD-CE. It also illustrates the subtle differences between ranking and labeling and suggests treating the problems differently. The results also show that the SD distribution performs better than the linear SVM baseline on 2 of the 3 datasets confirming its effectiveness as a classifier. We hasten to add that it is possible to further boost the performance of SVMs by defining better features or by doing good feature selection. The main aim of our experiments is not to outperform the best classifier but to demonstrate the effectiveness of the SD distribution as an elegant and effective distribution for text. We would also like to emphasize that besides performance, another attractive property of the SD distribution is its relatively quick training owing to its closed form MLE solution: SD takes at least an order of magnitude less computational time than DCM and Dirichlet and the SVM models and almost the same time as the multinomial, while performing at least as well as any of these models.

Comparing with results from other work, we note that our multinomial results agree quite closely with the results in (McCallum & Nigam, 1998) on all three collections. Our SVM results on 20 Newsgroups agree very well with the SVM baseline in (Rennie et al., 2003). Our results are slightly lower on Industry sector (our 88.20% vs. their 93.4%) while higher on Reuters (our 79.24% vs. their 69.4%). The difference in Reuters is primarily because we used top 10 classes while they used 90 classes. Our SVM results on Reuters are slightly lower than those reported in (Joachims, 1998) (our 76.21% vs. their 82.51% in Macro-BEP). For SVM features, we computed IDF values from only the training documents to make for a fair comparison with the generative distributions that used smoothing only with the training documents. It is not clear how IDF is computed in (Rennie et al., 2003) and (Joachims, 1998). Further, our preprocessing and indexing resulted in significantly higher number of unique tokens on all collections than in (Joachims, 1998), making

#	Dataset →	20 Newsgroups		Industry Sector		Reuters	
	Model (para.) ↓	Opt. Para.	% Accur.	Opt. Para.	% Accur.	Opt. Para.	% BEP
1	Mult-L (δ)	10^{-3}	$87.02_{2,4,7}$	10^{-4}	73.92_{3}	10^{-2}	71.42
2	Mult-JM (λ, β)	$10^{-2}, 10^{-3}$	86.41	$10^{-3}, 10^{-3}$	$84.91_{1,3,4}$	$0.7, 10^{-3}$	72.78
3	DCM (S, δ)	$900, 10^{-4}$	$88.04_{1,2,7}$	$1200, 10^{-3}$	71.01	$1500, 0.1$	72.38
4	Dir (S, λ, β)	$400, 0.1, 10^{-2}$	86.54	$1800, 10^{-1}, 10^{-2}$	$76.97_{1,3}$	$3000, 0.2, 10^{-3}$	74.87
5	SD (λ, β)	$10^{-4}, 0.1$	89.72_{A-6}	$10^{-3}, 10^{-3}$	$80.82_{1,3,4}$	$10^{-2}, 1$	**79.24**
6	SD-CE (λ, β)	$10^{-4}, 10^{-3}$	$\mathbf{90.56}_{A}$	$10^{-5}, 10^{-3}$	86.22_{A-7}	$10^{-2}, 1$	**79.24**
7	SVM (C)	1.0	86.48	1.0	$\mathbf{88.20}_{A}$	20	76.21

Table 2. Performance comparison on the three data sets: Mult-L and Mult-JM correspond to Multinomial with Laplace and Jelinek-Mercer smoothing respectively and Dir is Dirichlet, while the rest have their usual meaning. The symbols in the parentheses in column 2 indicate the free parameters of each distribution. For reproducibility of our experiments, we present the respective optimal parameter settings in each data set in columns titled "Opt. Para.". A subscript $_i$ on an entry in columns 4 and 6 represents that the corresponding model is significantly better than the model whose serial number is i according to a paired 2-tailed T-test at 95% C.I. on the 25 random train-test splits. The notation $_A$ in the subscript implies the model is significantly better than all other models, while $_{A-i}$ indicates the model is better than all but the model numbered i. Bold-face number indicates the best performing model on the corresponding data set.

comparison difficult. However, the trends are quite similar in that, SVM outperforms multinomial distribution (the 20 Newsgroups data being an exception in our case). The work that is most related to ours is that of Madsen *et al* (Madsen et al., 2005). Their results on Reuters are not exactly comparable because they used 90 classes with at least one training and one test document while we used only top 10 classes. On other collections, they used precision as the evaluation metric while we used the more popular classification accuracy. But our results are consistent with theirs in that, in general, DCM is shown to be better performing than the Laplace smoothed multinomial.

6. Future work

Considering the attractive properties of the SD distribution such as better modeling of term-occurrence characteristics and simple closed-form estimation, we hope it will be widely used by researchers in place of multinomial as a basic building block in more complex generative mixture models of text. We would like to emphasize that text classification is just an example application, chosen because it provides a clean and simple test of our new distribution. The effectiveness of the SD distribution, as demonstrated in this application, suggests its utility in other IR tasks, particularly in time critical tasks such as filtering and ad-hoc retrieval where quick training and inference are of utmost importance. As part of future work, we intend to do more experiments with the SD distribution on the IR tasks mentioned above, particularly in a semi-supervised setting, through the EM algorithm.

References

Abramowitz, M., & Stegun, I. A. (1972). *Handbook of mathematical functions, national bureau of standards applied math.series.*

Apte, C., Damerau, F., & Weiss, S. M. (1994). Automated learning of decision rules for text categorization. *ACM Trans. Inf. Syst.*, *12*, 233–251.

Blei, D., Ng, A., & Jordan, M. (2002). Latent dirichlet allocation. *NIPS.*

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. *ECML.*

Joachims, T. (1999). Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning.* MIT-Press.

Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. *SIGIR.* New Orleans, Louisiana, United States.

Lavrenko, V. (2004). A generative theory of relevance. *Ph.D. thesis.*

Madsen, R. E., Kauchak, D., & Elkan, C. (2005). Modeling word burstiness using the dirichlet distribution. *ICML.*

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. *In AAAI-98 Workshop on Learning for Text Categorization.*

McCallum, A. K., Rosenfeld, R., Mitchell, T. M., & Ng, A. Y. (1998). Improving text classification by shrinkage in a hierarchy of classes. *ICML* (pp. 359–367).

Minka, T. P. (2003). Estimating a dirichlet distribution.

Rennie, J., Shih, L., Teevan, J., & Karger, D. (2003). Tackling the poor assumptions of naive bayes text classifiers. *ICML.*

Robertson, S. E., & Jones, K. S. (1976). Relevance weighting of search terms. *JASIS*, *27(3)*, 129–146.

Robertson, S. E., Rijsbergen, C. J. V., & Porter, M. F. (1981). Probabilistic models of indexing and searching. *Information Retrieval Research*, 35–56.

Teevan, J., & Karger, D. R. (2003). Empirical development of an exponential probabilistic model for text retrieval: Using textual analysis to build a better model. *SIGIR*.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, *22*, 179–214.