The Smoothed-Dirichlet distribution: Explaining KL-divergence based ranking in Information Retrieval

Ramesh Nallapati
nmramesh@cs.umass.edu
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA

Thomas Minka, Hugo Zaragoza, Stephen Robertson
{minka,zaragoza,ser}@microsoft.com
Microsoft Research
7 J J Thomson Ave
Cambridge CB3 0FB, UK

Abstract

In this work[1], we analyze the popular KL-divergence ranking function in information retrieval. We uncover the generative distribution, namely the Smoothed Dirichlet distribution, underlying this ranking function and show that this distribution captures term occurrence distribution much better than the multinomial, thus offering, for the first time, a reason behind the success of the KL-divergence ranking function. We present theoretically motivated approximations to the distribution that lead to a closed form maximum likelihood solution, much like the multinomial, making it ideal for online IR tasks. We use the new distribution to construct a new, well-motivated ad-hoc retrieval algorithm. Our experiments show that this algorithm performs at least as well as similar algorithms that employ cross-entropy ranking. It also provides additional flexibility, e.g. in handling scenarios like a mixture of true and pseudo relevance feedback, due to a consistent generative framework.

1 Introduction

Ad-hoc retrieval is one of the important tasks of information retrieval in which the user's information need is typically expressed in the form of a key-word query, in response to which, the system is expected to return a ranked list of textual documents in decreasing order of relevance. It is natural to think of ad-hoc retrieval as a classification problem in which documents are classified into one of 'relevant' and 'non-relevant' classes w.r.t. the query. In this view, the task is similar to document classification.

In line with this view, a few generative classifiers were considered for ad-hoc retrieval in the past. One of the first among them is the Binary Independence Retrieval model (Robertson and Jones, 1976) which used the Multiple-Bernoulli distribution as the generator of documents. However, this distribution considers only the presence or absence of a term in a document and ignores term-frequency information which is a useful indicator of relevance. To rectify this problem, a mixture of Poissons (Robertson et al., 1981) was proposed, but it did not show any significant improvement in performance. However, an approximation of this distribution has resulted in the famous BM25 model (Robertson and Walker, 1994), which is considered as a standard baseline in IR. In (McCallum and Nigam, 1998), the multinomial distribution was proposed as an alternative to multiple Bernoulli as it models term frequency information. They showed that the multinomial betters the performance of multiple-Bernoulli on the task of text classification. The document log-likelihood w.r.t. the multinomial distribution is shown below.

$$\log Pr(D|\boldsymbol{\theta}^Q) \quad \propto \quad \sum_{j=1}^{V} f_j^D \log \theta_j^Q \qquad (1)$$

where V is the vocabulary size, f_j^D is the raw-count of the j^{th} word in the document D and $\boldsymbol{\theta}^Q$ is the multinomial distribution of the query's topic. The multinomial, however, was not as successful as other vector-space based models in the ad-hoc retrieval task (Teevan, 2001). The inferior performance of multinomial is explained by the observation that multinomial distribution is not a good fit to textual data as it hugely under-predicts heavy-tail behavior or burstiness of term-occurrence (Teevan and Karger, 2003; Rennie et al., 2003; Madsen et al., 2005).

It is however, interesting to note that the new class of language models for information retrieval

[1]CIIR technical report. Please do not cite or distribute.

(Ponte and Croft, 1998; Lafferty and Zhai, 2001) that achieve state-of-the-art performance employ the same multinomial distribution to model documents and queries, but they use a completely different ranking function namely, the negative KL-divergence $-KL(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D)$, which in the IR context, is rank-equivalent to negative cross-entropy $-CE(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D)$ as shown below.

$$-KL(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D) = -\sum_{j=1}^{V} \theta_j^Q \log \frac{\theta_j^Q}{\theta_j^D}$$

$$\stackrel{rank}{=} -CE(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D) = \sum_{j=1}^{V} \theta_j^Q \log \theta_j^D \quad (2)$$

where $\boldsymbol{\theta}^D$ is the multinomial distribution representing the document's topic, called the document language model.

On comparing the ranking functions in (2) and (1), it is evident that both have the same general form $\sum_j A_j \log B_j$, but the roles of variables A and B are interchanged: while in (2), variables A and B correspond to query and document respectively, in (1) it is the exact opposite. One could have defined a cross-entropy ranking function as $CE(\boldsymbol{\theta}^D||\boldsymbol{\theta}^Q) = -\sum_{j=1}^{V} \theta_j^D \log \theta_j^Q$ which would make it equivalent to the multinomial log-likelihood of the document in (1). But there is empirical evidence that ranking functions of the form $-CE(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D)$ perform better than the form $-CE(\boldsymbol{\theta}^D||\boldsymbol{\theta}^Q)$, using the same values of parameters (Lavrenko, 2004). However, no theoretical reasoning is yet available either for why cross-entropy is a good ranking function or for why one particular form works better than the other. One of the main motivations of the present work is to understand the reason behind the superior performance of $-CE(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D)$.

It turns out that the well-performing cross-entropy ranking function $-CE(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D)$ in (2) corresponds to ranking using the log-likelihood assigned by a document generative model (as in (1)) using a Dirichlet distribution instead of the Multinomial as shown below:

$$\log Pr(\boldsymbol{\theta}^D|\boldsymbol{\alpha}^Q) \propto \sum_j \alpha_j^Q \log \theta_j^D \quad (3)$$

where $\boldsymbol{\alpha}^Q$ are the parameters of the Dirichlet distribution corresponding to the query's topic and $\boldsymbol{\theta}^D$ is the multinomial distribution of the document's topic. Comparing (2) and (3), we know that they have the same form $\sum_j Q_j \log D_j$. We hypothesize that the superior performance of $-CE(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D)$ and its correspondence to the Dirichlet distribution indicates that the Dirichlet could be a better modeler of text than the multinomial.

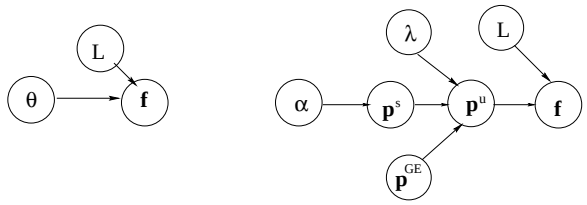
This intuition led us to explore the applicability of the Dirichlet distribution as a potential replacement for the multinomial in a generative classifier for information retrieval. The Dirichlet distribution has never been used as a generator of text, but has been extensively used as a prior to the multinomial in several topical models (Blei et al., 2002; Y.W. Teh and Blei, 2004). In (Madsen et al., 2005) the Dirichlet-Compound-Multinomial (DCM) distribution was used to model text, where the Dirichlet acts as an empirical prior to the multinomial. They showed that it models term-burstiness better than the multinomial and also demonstrated its effectiveness in text classification. However, the likelihood of a document w.r.t. the DCM does not correspond to the cross-entropy ranking function. Additionally, this distribution requires iterative gradient descent techniques for maximum likelihood parameter estimation and as such is not very attractive for IR tasks that require a very quick response to the user.

2 Smoothed Dirichlet (SD) distribution

We here describe the generative process of the Smoothed Dirichlet distribution. The rationale for this process is discussed in section 2.1. As shown in figure 1(b), we first generate a smoothed document model $\boldsymbol{\theta}^D$ from the SD distribution and unsmooth it to get the raw proportions $\widehat{\boldsymbol{\theta}}^D$ as follows:

$$\widehat{\boldsymbol{\theta}}^D \stackrel{\text{def}}{=} (\boldsymbol{\theta}^D - (1-\lambda)\boldsymbol{\theta}^{GE})/\lambda \quad (4)$$

where $\boldsymbol{\theta}^{GE}$ is the general English multinomial distribution and $0 < \lambda < 1$ is a smoothing parameter. The unsmoothed proportions $\widehat{\boldsymbol{\theta}}^D$ are then converted into a bag of words $\mathbf{f}$ given the document length L, using the relation $\mathbf{f} = \text{int}(L\widehat{\boldsymbol{\theta}}^D)$ where int() is a function that returns the nearest integer-vector to its real-vector argument. Only the generation of $\boldsymbol{\theta}^D$ is probabilistic and its conversion to unsmoothed proportions $\widehat{\boldsymbol{\theta}}^D$ 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 document model $\boldsymbol{\theta}^D$ given by:



(a) Multinomial (b) Smoothed Dirichlet

Figure 1: Graphical representation of document generation using SD distribution

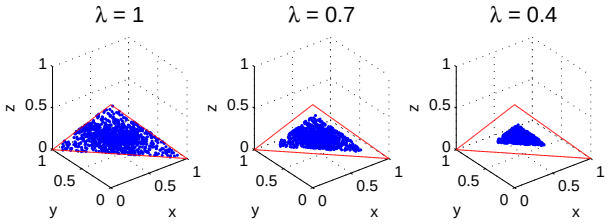


Figure 2: Domain of smoothed document models Δ^s for various degrees of smoothing: dots are smoothed-document models $\boldsymbol{\theta}^D$ and the triangular boundary is the 3-D simplex Δ.

$$Pr(\mathbf{f}|\boldsymbol{\theta}^{GE}, \lambda, L, \boldsymbol{\alpha}) = Pr(\boldsymbol{\theta}^D|\boldsymbol{\alpha}) = \frac{1}{Z^{SD}} \prod_{j=1}^{V} {\theta_j^D}^{\alpha_j - 1} \quad (5)$$

where $\boldsymbol{\alpha}$ is the parameter vector of the smoothed-Dirichlet distribution and Z^{SD} is the SD-normalizer. 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 $\widehat{\boldsymbol{\theta}}^D = \mathbf{f}/L$ and then get a smoothed document model using the inverse of relation (4):

$$\boldsymbol{\theta}^D = \lambda \widehat{\boldsymbol{\theta}}^D + (1-\lambda)\boldsymbol{\theta}^{GE} \quad (6)$$

and then compute its probability under the SD distribution as given by (5). In the rest of the paper, we use $\widehat{\boldsymbol{\theta}}^D$ to represent raw-proportions in a document and $\boldsymbol{\theta}^D$ to represent a smoothed model.

2.1 Rationale

The reason we generate the smoothed document representation $\boldsymbol{\theta}^D$ and not the raw-proportions $\widehat{\boldsymbol{\theta}}^D$ directly is to avoid assigning zero probability to any document: the raw-proportions $\widehat{\boldsymbol{\theta}}^D$ of a document is typically a sparse vector with many zeros in it and as such, if we replace $\boldsymbol{\theta}^D$ with $\widehat{\boldsymbol{\theta}}^D$ in (5), we end up with a zero probability for almost all documents.

Notice that the functional form of the SD distribution defined in (5) is same as the ordinary Dirichlet distribution (Minka, 2003). One may argue that we could use the ordinary Dirichlet distribution to generate the smoothed document model $\boldsymbol{\theta}^D$ instead of defining a new distribution. However, the Dirichlet distribution is incorrect for smoothed proportions because it assigns probability mass to the entire simplex $\Delta = \{\boldsymbol{\theta} \mid \forall_j \theta_j > 0; \sum_j \theta_j = 1\}$ while smoothed models 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 $\widehat{\boldsymbol{\theta}}^D$, smoothed them with $\boldsymbol{\theta}^{GE}$ estimated from the entire document set, and plotted the document models $\boldsymbol{\theta}^D$ in figure 2. The leftmost plot represents the unsmoothed proportion vectors $\widehat{\boldsymbol{\theta}}^D$ 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 document models gets compressed towards the centroid. From the generative perspective, restricting the domain of $\boldsymbol{\theta}^D$ is necessary to ensure that the raw-proportions vectors $\widehat{\boldsymbol{\theta}}^D$ generated using the definition in (4) lie on the multinomial simplex Δ. The compressed domain Δ^s in figure 2 corresponds to the set of all feasible values of $\boldsymbol{\theta}^D$ that guarantee meaningful values for $\widehat{\boldsymbol{\theta}}^D$. Hence, the Dirichlet normalizer, that considers the whole simplex Δ as its domain, as defined below in (7), is clearly incorrect given our smoothed document representation.

$$Z(\boldsymbol{\alpha}) = \int_{\Delta} \prod_j \theta_j^{\alpha_j - 1} d\boldsymbol{\theta}^D = \frac{\prod_j \Gamma(\alpha_j)}{\Gamma(\sum_j \alpha_j)} \quad (7)$$

The SD distribution rectifies this flaw by defining a normalizer that assigns the probability mass only to the new compressed domain Δ^s.

2.2 SD normalizer and its approximator

The compressed domain Δ^s is given by:

$$\Delta^s = \{\boldsymbol{\theta}^D\} = \{\lambda \widehat{\boldsymbol{\theta}}^D + (1-\lambda)\boldsymbol{\theta}^{GE} \mid \widehat{\boldsymbol{\theta}}^D \in \Delta\} \quad (8)$$

The above equation is a transform for $\boldsymbol{\theta}^D$ 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 terms of the regular simplex domain Δ as:

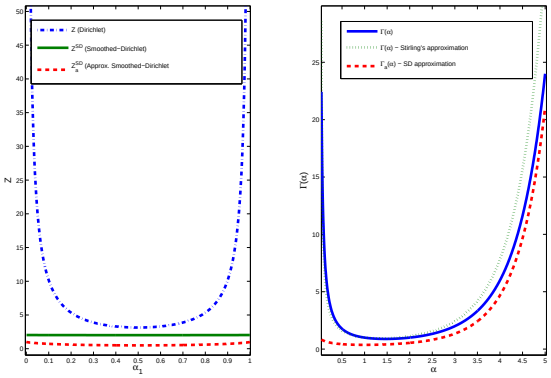


Figure 3: (a) Comparison of the normalizers (b) Gamma function and its approximators

$$Z^{SD} = \int_{\Delta^s} \prod_{j=1}^{V} (\theta_j)^{\alpha_j - 1} d\boldsymbol{\theta}^D \quad (9)$$

$$= \int_{\Delta} \prod_{j=1}^{V} \left\{ \lambda \widehat{\theta}_j + (1-\lambda)\theta_j^{GE} \right\}^{\alpha_j - 1} (\lambda d\widehat{\boldsymbol{\theta}}^D) \quad (10)$$

For fixed values of λ and $\boldsymbol{\theta}^{GE}$, 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 for the SD distribution.

Figure 3(a) compares Z^{SD} with the Dirichlet normalizer Z of (7) for a simple case where the vocabulary size V is 2, *i.e.*, $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2\}$. We imposed the condition that $\alpha_1 + \alpha_2 = 1$ and used $\lambda = 0.2$ and $\left\{\theta_1^{GE}, \theta_2^{GE}\right\} = \{0.5, 0.5\}$. The plot shows the value of normalizer for various values of α_1. We computed the exact value of Z^{SD} using the incomplete two-variate Beta function implementation of *Matlab*. Notice that Z^{SD} tends to finite values at the boundaries while the normalizer for Dirichlet distribution 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 (7), we define Z_a^{SD} as:

$$Z_a^{SD}(\boldsymbol{\alpha}) = \prod_j \Gamma_a(\alpha_j) \;/\; (\Gamma_a(\sum_j \alpha_j)) \quad (11)$$

where $\Gamma_a(\alpha)$ is an approximation to the Gamma function. 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 (10). We turn to the Stirling's approximation of the Gamma function (Abramowitz and Stegun, 1972), shown in (12) for guidance.

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

Figure 3(b) plots the Gamma function and its Stirling approximation which shows the Gamma function yields unbounded values in the limit as $\alpha \to 0$. Inspecting (7), it is apparent that the unboundedness of Dirichlet normalizer Z results from the unboundedness of Γ at small values of α. Since our exact computation in low dimensions shows that the Smoothed Dirichlet normalizer Z^{SD} is actually bounded as $\alpha \to 0$, we need a bounded approximator of the Gamma function. An easy way to define this approximation is to ignore the terms in Stirling's approximation that make it unbounded and redefine it as:

$$\Gamma_a(\alpha) \stackrel{\text{def}}{=} e^{-\alpha}\alpha^{\alpha} \quad (13)$$

The approximate Gamma function Γ_a is compared to the exact Gamma function 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 (11) and (13), we have:

$$Z_a^{SD}(\boldsymbol{\alpha}) = \frac{\prod_j e^{-\alpha_j}\alpha_j^{\alpha_j}}{e^{-\sum_j \alpha_j}(\sum_j \alpha_j)^{\sum_j \alpha_j}} = \frac{\prod_j \alpha_j^{\alpha_j}}{S^S} \quad (14)$$

where $S = \sum_j \alpha_j$. The approximation in (14) is independent of λ and $\boldsymbol{\theta}^{GE}$ which is clearly an oversimplification of the exact SD normalizer Z^{SD} in (10). However our plot of the approximate SD normalizer Z_a^{SD} in figure 3(a) shows that it behaves very similar to Z^{SD}. The approximate Smoothed Dirichlet distribution can now be defined as:

$$Pr_a(\boldsymbol{\theta}^D|\boldsymbol{\alpha}) = \frac{S^S}{\prod_j \alpha_j^{\alpha_j}} \prod_j \theta_j^{\alpha_j - 1} \quad (15)$$

Henceforth, we will refer to the approximate SD distribution as the SD distribution for convenience.

3 An SD based Generative model for IR

As described in the introduction, we consider IR as a problem of classifying documents into two classes $R = 1$ and $R = 0$ representing relevant and non-relevant classes respectively with corresponding SD parameters $\boldsymbol{\alpha}^R$ and $\boldsymbol{\alpha}^N$. For simplicity, we assume that both the classes have the same precision $S = \sum_j \alpha_j^R = \sum_j \alpha_j^N$, which is considered a free-parameter of the model. We fix the parameters of

the non-relevant class $\boldsymbol{\alpha}^N$ proportional to the general English proportions as $\boldsymbol{\alpha}^N = S\boldsymbol{\theta}^{GE}$. We use the Expectation Maximization algorithm to estimate the parameters of the relevant class $\boldsymbol{\alpha}^R$ from the query as well as a combination of true-feedback and pseudo-feedback documents.

3.1 Ranking: E-step

Given the parameters $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}^R, \boldsymbol{\alpha}^N\}$ of the SD model, we rank the documents $\{\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n\}$ by their posterior probability of relevance $Pr(R = 1|\boldsymbol{\theta}^i, \boldsymbol{\alpha}, \pi^R)$ where π^R is the prior probability of relevance. Incidentally, the posterior probability of relevance is also equal to the expected value of relevance $E[R|\boldsymbol{\theta}^i, \boldsymbol{\alpha}, \pi^R]$ and computing this value corresponds to the E-step of the EM algorithm as shown below:

$$E[R|\boldsymbol{\theta}^i, \boldsymbol{\alpha}, \pi^R] = Pr(R = 1|\boldsymbol{\theta}^i, \boldsymbol{\alpha}, \pi^R) = \frac{\mathcal{L}^i}{1 + \mathcal{L}^i} \quad (16)$$

$$\text{where } \mathcal{L}^i = \frac{Pr(\boldsymbol{\theta}^i|\boldsymbol{\alpha}^R)\pi^R}{Pr(\boldsymbol{\theta}^i|\boldsymbol{\alpha}^N)(1 - \pi^R)} \text{ and} \quad (17)$$

$$\log \mathcal{L}^i = KL(\boldsymbol{\alpha}^N||\boldsymbol{\theta}^i) - KL(\boldsymbol{\alpha}^R||\boldsymbol{\theta}^i) + \log \frac{\pi^R}{1 - \pi^R} \quad (18)$$

$$\stackrel{rank}{=} CE(\boldsymbol{\alpha}^N||\boldsymbol{\theta}^i) - CE(\boldsymbol{\alpha}^R||\boldsymbol{\theta}^i) \quad (19)$$

where $\mathcal{L}^i$ is the likelihood ratio of relevance. Steps (16) and (17) follow directly from Bayes rule while step (19) is obtained by substitution of (15) in (17) and subsequent algebraic manipulation using the assumption that $\sum_j \alpha_j^R = \sum_j \alpha_j^N = S$. Since $E[R|\boldsymbol{\theta}^i, \boldsymbol{\alpha}, \pi^R]$ is a monotonic function of $\mathcal{L}^i$ as shown in (16), it is rank equivalent to $\mathcal{L}^i$. It is also rank-equivalent to $\log \mathcal{L}^i$ which is another monotonic function of $\mathcal{L}^i$.

Notice that the ranking function defined by the smoothed Dirichlet model in (19) is equivalent to the one used in the language modeling approach in (2). Since we use a binary classifier, we have an additional term in $CE(\boldsymbol{\alpha}^N||\boldsymbol{\theta}^i)$ that ensures that the documents whose models are unlike general English proportions are ranked higher. We have thus, uncovered the generative model underlying the cross-entropy ranking function.

3.2 Estimation: M-step

We estimate the parameters of the relevant class $\boldsymbol{\alpha}^R$ from a combination of labeled and unlabeled feedback documents $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n\}$ using the M-step of the EM algorithm whose final expression is given below:

$$\boldsymbol{\alpha}^R = \frac{1}{Z}\{\prod_{i=1}^{n} \boldsymbol{\theta}^{i\,E[R^i]}\}^{\frac{1}{\sum_{i=1}^{n} E[R^i]}} \quad (20)$$

where $E[R^i]$ is short for $E[R|\boldsymbol{\theta}^i, \boldsymbol{\alpha}, \pi^R]$ and Z is a normalizer that ensures $\sum_j \alpha_j^R = S$. Thus, the SD distribution provides a closed form solution for training where our estimates of α_j for term w_j are simply normalized weighted geometric averages of the word's smoothed models in the training documents, where the weights are equal to their respective posterior probabilities of relevance. Note that when the document is explicitly judged relevant by the user (true relevance-feedback), $E[R^i] = 1$ and when the user judgment is not available (pseudo-feedback), $E[R^i]$ is computed using (16).

4 Experiments

4.1 Data Analysis

In this sub-section, we compare empirical term occurrence distribution with that predicted by the multinomial and SD distributions. 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, which consist of the remaining 95% low-frequency words comprising only 10% of occurrences. We pooled together 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 parameters of Multinomial and Smoothed-Dirichlet distributions using all documents in the collection. For SD, we fix the value of the smoothing parameter λ at 0.9 and estimate only $\boldsymbol{\alpha}$. We tuned the free-parameter S of the SD distribution until it achieves the best visual fit w.r.t. the empirical distribution. Figure 4 compares the predictions of each distribution with the empirical distributions for each category.

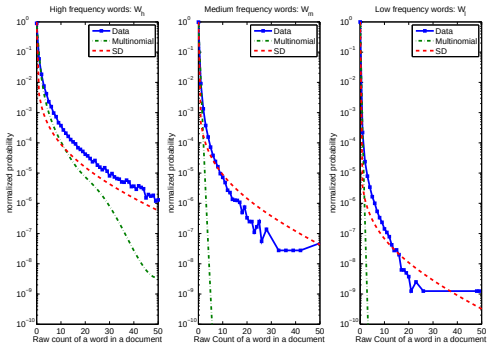


Figure 4: Comparison of predicted and empirical distributions

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 SD distribution fits the data much better than the multinomial on all three sets, showing that SD is a better fit for text than the multinomial distribution.

Coming back to the puzzle we started with in the introduction, our work now offers a simple justification for why $-CE(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D)$ performs better than $-CE(\boldsymbol{\theta}^D||\boldsymbol{\theta}^Q)$: the earlier version corresponds to the SD distribution, while the latter version corresponds to the multinomial. SD distribution is a better fit to textual data than multinomial, hence it is not surprising that the former version should do better. 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 using the same underlying distribution that corresponds to the successful cross-entropy ranking function.

4.2 Ad-hoc Retrieval

In this set of experiments, we compare the performance of the SD model with similar algorithms that use cross-entropy ranking - the query-likelihood model (QL) and the state-of-the-art Relevance model (RM) (**?**). We tested three different scenarios, true relevance feedback (using 2 relevant documents per query), pseudo-relevance feedback (using 25 unlabeled documents) and a combination of both (using 2 relevant and 25 pseudo-relevant documents). We constructed a corpus consisting of all documents from AP88, AP89 and AP90 collections of TREC. We used queries 51-150 for our experiments. Of these queries we ignored 3 queries (queries 63,65,66) that had less than 10 relevant documents for evaluation reasons discussed below. We used the first 49 queries to tune our model's parameters and the last 48 queries (103-150) for testing. For each retrieval scenario, we tested three different query-lengths, short, medium and long. We used titles as short queries, narrative with the description of non-relevant component removed as medium queries and a single relevant document as a long query. On an average, the average length of short queries on the test set is about 4 words, an average medium query is 31 words long and an average long query is about 257 words long. We performed standard stopping and stemming using the Porter stemmer on the entire collection and queries. The collection was indexed using version 3.0 of the *Lemur* tool-kit[2].

To make for a fair evaluation, we sampled 5 relevant documents for each query and isolated them from retrieval as well as evaluation. These documents would only be available for true relevance feedback. Since we used queries that had at least 10 relevant documents, we guarantee that each query has at least 5 relevant documents available for evaluation after isolating the feedback documents. To provide an equal basis for comparison of various retrieval scenarios, we isolated these judged documents from retrieval and evaluation in the pseudo-feedback scenarios too, although we do not provide them for estimation. For pseudo-relevance feedback, we used top ranking documents from the best query-likelihood run. For all the models we experimented with, we used the same relevant and pseudo-relevant documents for feedback to provide an equal basis for comparison.

The Query Likelihood (QL) model uses $-CE(\widehat{\boldsymbol{\theta}}^Q||\boldsymbol{\theta}^D)$ as the ranking function where $\widehat{\boldsymbol{\theta}}^Q$ is the raw proportions of words in the query. $\boldsymbol{\theta}^D$ is computed as in (6) using Dirichlet smoothing where $\lambda = L/(L+\mu)$ and L is the document length and μ is the Dirichlet smoothing parameter. Query likelihood model is not capable of making use of relevance or pseudo-relevance feedback.

[2]http://www-2.cs.cmu.edu/ lemur/

1	$\forall D^i \in RF \cup PF\ P(\boldsymbol{\theta}^i\|Q) \propto Pr(Q\|\boldsymbol{\theta}^i)$
2a	compute $E[\boldsymbol{\theta}\|Q] = \sum_{D^i \in RF \cup PF} \boldsymbol{\theta}^i P(\boldsymbol{\theta}^i\|Q)$
2b	$\forall D^i \in RF$ compute $E[\boldsymbol{\theta}\|D^i] = \boldsymbol{\theta}^i$
3	$\boldsymbol{\theta}^Q = \frac{\sum_{D^i \in RF} E[\boldsymbol{\theta}\|D^i] + E[\boldsymbol{\theta}\|Q]}{\|RF\|+1}$
4	rank document D using $-CE(\boldsymbol{\theta}^Q\|\|\boldsymbol{\theta}^D)$

Table 1: Relevance model: RF is the set of relevant documents and PF is the set of pseudo-relevant documents.

1	1^{st} M-step from query only: $\boldsymbol{\alpha}^R = S\boldsymbol{\theta}^Q$
2a	1^{st} E-step: $\forall D^i \in PF$ compute F^i
2b	$\forall D^i \in RF \cup Q$ assign weight $G^{-1} = W \max_{D^i \in PF}(F^i)$
3	2^{nd} M-step: Estimate α^R using (24)
4	2^{nd} E-step: rank D^i using new value of F^i

Table 2: SD model

RM on the other hand models both true and pseudo-relevance feedback and the algorithm is shown in table 1, where step (2b) follows from step (2a) by substituting $Q = D^i$ and from the observation that for long queries D^i, the distribution $Pr(\boldsymbol{\theta}|D^i)$ approaches a Dirac-Delta function concentrated at $\boldsymbol{\theta}^i$. The estimate $\boldsymbol{\theta}^Q$ of RM is roughly an arithmetic weighted average of the feedback documents, where the weights are proportional to the query-likelihood of the model.

For the SD model, we first note that the log-likelihood ratio $\mathcal{L}^i$ can be split into document-dependent and document-independent terms as follows:

$$\log \mathcal{L}^i = f(\boldsymbol{\theta}^i, \boldsymbol{\alpha}) + g(\boldsymbol{\theta}^i, \pi_R) \text{ where} \quad (21)$$

$$g(\boldsymbol{\alpha}, \pi_R) = H(\boldsymbol{\alpha}^R) - H(\boldsymbol{\alpha}^N) + \log \frac{\pi^R}{1-\pi^R} \text{ and} \quad (22)$$

$$f(\boldsymbol{\theta}^i, \boldsymbol{\alpha}) = CE(\boldsymbol{\alpha}^N||\boldsymbol{\theta}^i) - CE(\boldsymbol{\alpha}^R||\boldsymbol{\theta}^i) \quad (23)$$

where $H(\boldsymbol{\alpha}) = -\sum_j \alpha_j \log \alpha_j$ is the entropy of the parameter vector. We make the following simplifying assumptions in computing $E[R^i]$: firstly, we noticed that the value of $\mathcal{L}^i$ for most documents is much smaller than 1, owing to the fact that the prior probability of relevance is very low and also the non-relevant distribution better explains most documents. This observation allows us to approximate $E[R^i]$ to $\mathcal{L}^i$ since $\mathcal{L}^i/(\mathcal{L}^i+1) \approx \mathcal{L}^i$ when $\mathcal{L}^i \ll 1$ (see (16)). We substitute this approximation and (21) into (20) to get the following:

$$\boldsymbol{\alpha}^R = \frac{1}{Z}\{\prod_{i \in RF} (\boldsymbol{\theta}^i)^{G^{-1}} \prod_{i \in PF} (\boldsymbol{\theta}^i)^{F^i}\}^{\frac{1}{|RF|G^{-1} + \sum_{i \in PF} F^i}} \quad (24)$$

where G is short for $\exp(g(\boldsymbol{\alpha}, \pi^R))$ and F^i is short for $\exp(f(\boldsymbol{\theta}^i, \boldsymbol{\alpha}))$. Note that F^i and G^{-1} acts as weights of pseudo-relevant and true-relevant documents respectively. It turns out that due to our simplifying assumption $\boldsymbol{\alpha}^N = S\boldsymbol{\theta}^{GE}$, the entropy term $H(\boldsymbol{\alpha}^N)$ dominates the other terms in (22) resulting in a large negative value for g. This in turn, results in a very heavy weight for relevant documents in (24). To discount this effect, we instead use the following intuitive approximation: $G^{-1} = W \max_i(F^i)$ where W is a free-parameter. We restrict the domain of W to $[1, \infty)$ to ensure that relevant documents are always weighted higher than pseudo-relevant ones. Note that we also consider the query Q as a relevant document using its smoothed model $\boldsymbol{\theta}^Q$. The final algorithm of SD model, given these approximations, is given in table 2.

For optimal performance, Relevance model uses different representations for documents in various steps: to compute the query-likelihood in step 1 of table 1, $\boldsymbol{\theta}^{D^i}$ is estimated using Dirichlet smoothing with a smoothing parameter μ, in estimating $E[\boldsymbol{\theta}|Q]$ in step 2(a) using weighted averaging of $\boldsymbol{\theta}^i$, smoothing is done as in (6) with a smoothing parameter λ_{est} and in computing cross entropy $CE(\boldsymbol{\theta}^Q||\boldsymbol{\theta}^D)$ for ranking in step 4, another smoothing parameter λ_{CE} is used to compute $\boldsymbol{\theta}^D$. Based on published results, we set the three parameters to their optimal values at $\mu = 1000, \lambda_{est} = 0.99$ and $\lambda_{CE} = 0.2$ (Lavrenko, 2004).

In contrast, we use a consistent representation for documents and queries in SD model, where we use parameter λ_D for documents and λ_Q for queries to account for the fact that queries are inherently different from documents. The SD classifier has two additional parameters in W (see step 2b in table 2) and S, which become operational only in case of pseudo and mixed feedback. While W fixes the relative weight of labeled documents w.r.t. the unlabeled ones, the precision S is inversely proportional to the variance of SD distribution and in effect, decides the distribution of weights among the unlabeled documents. We optimize these parameters based on training set of queries.

	Short Queries			Medium Queries			Long Queries		
	2RF	25PF	2RF+25PF	2RF	25PF	2RF+25PF	2RF	25PF	2RF+25PF
QL	19.22			27.87			19.35		
RM	25.80	27.33	30.16	24.97	27.79	31.20	27.31	19.32	27.31
SD	**28.99**	27.35	30.74	**32.55**	**31.56**	33.59	27.81	19.12	27.95

Table 3: Performance comparison of generative retrieval models in various scenarios on AP88-90 corpus and TREC queries 103-150: *2RF* indicates relevance feedback with 2 labeled documents, *25PF* is pseudo-feedback with 25 top-ranking documents from query-likelihood model, *2RF+25PF* indicates a mixture of both scenarios. All numbers are average-precision in %. A Boldface number indicates statistical significance using a 2-tailed paired T-test at 95% C.I., w.r.t. the nearest performing model in the corresponding retrieval scenario.

The results in three different retrieval scenarios for three different types of queries are presented in table 3. The performance of the QL model increases from short queries to medium queries but again drops for long queries. Medium queries have more information than short queries, so the improvement in performance is not surprising. Long queries are whole documents and tend to include a lot of noise, so the query-likelihood model deteriorates. Query-likelihood model does not support feedback of any kind, so for each query type, the performance remains unaltered in different retrieval scenarios.

For short and medium queries, in the scenario of true-relevance feedback, although SD has only two free parameters (λ_Q and λ_D) compared to RM's three (μ, λ_{est} and λ_{CE}), SD is still significantly better than RM. We believe the main reasons are SD's explicit usage of query as a relevant document which helps it to focus the model on the query and also its additional term in the ranking function $CE(\boldsymbol{\alpha}^N||\boldsymbol{\theta}^D)$ as shown in (19) which helps it discount noisy documents. RM includes query only implicitly by conditioning the document models on the query (see step 2a in table 1), so there is a higher chance that it drifts away from the query's topic. However when provided with many pseudo-feedback documents, RM betters its own performance by learning from documents that are close to the query using its nearest neighbor-like weighting scheme. SD model, also improves its performance when provided with pseudo-feedback and is consistently better than RM, although not statistically significant at all times.

In case of long queries, the query is itself not focused, so SD's advantage of explicit modeling of query does not seem to help that much. Both models perform poorly in case of pseudo feedback because of this reason. An interesting observation is that RM's performance in the mixed-feedback remains identical to its performance in true-feedback. This is because of the long query effect as shown in step 2b of table 1: since the query is a long document, conditioning on it gives us back the query's smoothed model, so RM fails to take advantage of pseudo-feedback documents. SD on the other hand, improves its performance from true feedback to mixed feedback, but only marginally.

5 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. The effectiveness of the SD distribution, as demonstrated in ad-hoc retrieval, suggests its utility in other similar IR tasks. We believe it is particularly well suited in time critical tasks such as supervised and unsupervised filtering 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 filtering, particularly in an unsupervised setting, through the EM algorithm.

References

M. Abramowitz and I. A. Stegun. 1972. *Handbook of Mathematical Functions, National Bureau of Standards Applied Math.Series*.

D. Blei, A. Ng, and M. Jordan. 2002. Latent dirichlet allocation. In *NIPS*.

John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *SIGIR*.

Victor Lavrenko. 2004. A generative theory of relevance. In *Ph.D. thesis*.

R. E. Madsen, D. Kauchak, and C. Elkan. 2005. Modeling word burstiness using the dirichlet distribution. In *ICML*.

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

Thomas P. Minka. 2003. Estimating a dirichlet distribution.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281.

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

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

S.E. Robertson and S. Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*.

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

Jaime Teevan and David R. Karger. 2003. Empirical development of an exponential probabilistic model for text retrieval: Using textual analysis to build a better model. In *SIGIR*.

Jaime Teevan. 2001. Improving information retrieval with textual analysis: Bayesian models and beyond. In *Master's Thesis*.

M.J. Beal Y.W. Teh, M.I. Jordan and D.M. Blei. 2004. Hierarchical dirichlet processes. In *Technical Report 653, UC Berkeley Statistics*.