A New Probabilistic Retrieval Model Based on the Dirichlet Compound Multinomial Distribution

Zuobing Xu
School of Engineering
University of California
Santa Cruz, CA, USA, 95064
zbxu@soe.ucsc.edu

Ram Akella
School of Engineering
University of California
Santa Cruz, CA, USA, 95064
akella@soe.ucsc.edu

ABSTRACT

The classical probabilistic models attempt to capture the Ad hoc information retrieval problem within a rigorous probabilistic framework. It has long been recognized that the primary obstacle to effective performance of the probabilistic models is the need to estimate a relevance model. The Dirichlet compound multinomial (DCM) distribution , which relies on hierarchical Bayesian modeling techniques, or the Polya Urn scheme, is a more appropriate generative model than the traditional multinomial distribution for text documents. We explore a new probabilistic model based on the DCM distribution, which enables efficient retrieval and accurate ranking. Because the DCM distribution captures the dependency of repetitive word occurrences, the new probabilistic model is able to model the concavity of the score function more effectively. To avoid the empirical tuning of retrieval parameters, we design several parameter estimation algorithms to automatically set model parameters. Additionally, we propose a pseudo-relevance feedback algorithm based on the latent mixture modeling of the Dirichlet compound multinomial distribution to further improve retrieval accuracy. Finally, our experiments show that both the baseline probabilistic retrieval algorithm based on the DCM distribution and the corresponding pseudo-relevance feedback algorithm outperform the existing language modeling systems on several TREC retrieval tasks.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms, Theory

1. INTRODUCTION

The classical probabilistic retrieval model [16, 13] of information retrieval has received recognition for being theoretically well founded. For the probabilistic retrieval models, we estimate two probabilistic models for each query: relevant class and non-relevant class. The probability ranking principle [16] suggests ranking documents by the log-odds ratio of being observed in the relevant class against the non-relevant class. Robertson [16] has proved that ranking documents by the odds of being generated by the relevant class against non-relevant class optimizes the retrieval performance under the word independence condition.

The problem of effectively estimating the relevant and non-relevant models remains a major obstacle in the practical applications of the probabilistic retrieval models. Various approaches for the estimation of relevance models have been considered in previous literature. The Binary independence retrieval model [13] treats each document as a binary vector over the vocabulary space and assumes independence between words. The 2-Poisson model [4] treats the term frequency as a mixture of 2-Poisson distributions, but ignores document length. Robertson and Walker [14] approximate the 2-Poisson model to account for several influential variables, including document length. The classical probabilistic retrieval models face the major challenge of effectively estimating the relevance model to take into account the variables influencing retrieval performance, and resort to different approximation techniques to model the relevant class.

However, the perceived limitation of the probabilistic retrieval model led to the development of the language models [12, 3, 6]. These language modeling approaches focus on effective estimation techniques for document modeling, and have been shown excellent retrieval accuracy and efficient implementation in practice. Language modeling approaches treat each document in the collection as a unique model and the query as strings of text randomly sampled from these models. The ranking is based on the probability of the query being generated by the document distribution. The Unigram language model is based on the multinomial distribution; several smoothing techniques [20] have been developed to avoid zero probabilities of non-occurring words.

A major limitation of the language models has been the lack of a clear connection with the explicit modeling of relevance. This gap has occasioned effort to relate these two models [7, 8]. Lafferty and Zhai [7] have demonstrated the probability equivalence of the language model to the probabilistic retrieval model under some very strong assumptions, which may or may not hold in practice. Language modeling approaches apply query expansion to incorporate information from (pseudo) relevance feedback documents [19],

SIGIR'08, July 20–24, 2008, Singapore.

while the probabilistic retrieval approaches treat (pseudo) relevance feedback as model adjustment. Thus, probabilistic approaches based on document generation have the advantage of being able to inherently improve the estimation of the probabilistic models by exploiting both the positive and negative feedback information. Considering the advantages of the probabilistic model, Lavrenko and Croft [8] proposed a novel technique for estimating the relevance model from the top ranked retrieved documents by combining language modeling estimation techniques with probabilistic model framework. Nevertheless, they essentially mod-el the pseudo relevance feedback process, and their model relies on a baseline language modeling approach.

Lewis [9] has pointed out the connection between the probabilistic retrieval model in information retrieval and the Naive Bayes classification model in machine learning. He also discussed the fact that the multinomial distribution performs well in the Naive Bayes classification in the context of text classification, but very poorly in the context of ranking documents in a search engine. *Consequently, an open research problem that remains to be solved is the following: "What is a reasonable distributional model to be applied in probabilistic retrieval models."* The multinomial distribution assumes word independence, and cannot capture word burstiness: the phenomenon that if a word appears once, it is more likely to appear again. In contrast, the Dirichlet compound multinomial (DCM) distribution [10, 1] has been shown effective in accommodating word burstiness, and achieves better performance in text classification and text clustering. Relying upon hierarchical Bayesian modeling, the DCM distribution integrates out the parameters of the multinomial distribution. The DCM distribution is also motivated by the Polya urn scheme [5], which intuitively explains how it captures word burstiness. The classical probabilistic model makes strong word independence assumptions, and thus has an incorrect model of relevant documents [7]. *The DCM distribution relaxes the independence assumption by accounting for dependence among repetitive word occurrences, and this is a better distribution for probabilistic models.* In our paper, we apply the DCM distribution as the generative source the new probabilistic model. Our model significantly simplifies the resulting form of the score function by taking the log-odds ratio, and the new score function is very efficient to implement in practice.

The optimal setting of the retrieval parameters is usually achieved by practical tuning. Since both the query and document collection vary in practical retrieval scenarios, it is unrealistic to tune parameters for different retrieval tasks. Applying already tuned parameters for another retrieval tasks may not perform consistently well. Therefore, automatically setting of retrieval parameters is critical to accommodating various retrieval tasks. Zhai and Lafferty [21] have derived a general two-stage parameter estimation method for the language model. We propose several parameter estimation approaches that explicitly capture the different impacts of document collection and query on the optimal settings of retrieval parameters. We first estimate the non-relevant model by fitting a DCM distribution to the whole document collection by using three approaches, and then estimate the query interpolation parameter which controls the relevance model generation.

Pseudo-relevance feedback has been demonstrated to be one of the most efficient approaches to improve retrieve accuracy. Pseudo-relevance feedback assumes that the top ranked retrieved documents are relevant, and reformulates the query representation by using these documents. Various pseudo-relevance feedback algorithms have been applied to several retrieval models. For the vector space model approaches, the original query is expanded with the centroid of feedback documents [15]. For the language model approaches, Zhai and Lafferty [19] have proposed several model based feedback algorithms, which expand the original query representation by the relevant topic terms from the feedback documents. In this paper, we design a pseudo-relevance feedback algorithm based on a mixture of two DCM distributions: feedback relevant model and collection model. We resort to the EM algorithm to estimate the feedback relevant model and enrich the original relevant model without feedback. We also propose several insightful improvements on the EM algorithm to find a better local optimum.

In summary, this paper focuses on an effective probabilistic model which applies advanced text document modeling and estimation techniques. The main contributions of this paper are three fold:

1. A formal probabilistic retrieval model based on the DCM distribution, with the associated analysis.
2. Several approaches that effectively estimate parameters of the proposed DCM probabilistic model.
3. A pseudo-relevance feedback algorithm based on the latent mixture modeling of feedback documents.

2. PROBABILISTIC RETRIEVAL MODEL BASED ON THE DCM DISTRIBUTION

2.1 Motivation

In this section, we analyze the probabilistic retrieval model based on the multinomial distribution to shed some light on the intuition of using the DCM distribution. In the language modeling framework, documents are modeled as the multinomial distributions capturing the word frequency occurrence within the documents. We model the relevant model and non-relevant model in the probabilistic retrieval model as two multinomial distributions. We define the parameters of relevant and non-relevant document language model as θ_R and θ_N. The probability of document d_l generated by relevant class is defined as the multinomial distribution:

$$P(d_l|\theta_R) = \frac{n(d_l)!}{\prod_m c(w_m,d_l)!} \prod_{m=1}^{V} (\theta_R^m)^{c(w_m,d_l)} \quad (1)$$

In the above equation, $c(w_m, d_l)$ is the term frequency of word w_m in document d_l; $n(d_l) = \sum_m c(w_m, d_l)$ is the length of the document d_l; V is vocabulary size. The Non-relevant model $P(d_l|\theta_N)$ is defined in the same way. The score function of the probabilistic retrieval model based on the multinomial distribution can be derived from taking the log-odds ratio of two multinomial distributions.

$$Score^{MN}(d_l) = \log \frac{P(d_l|\theta_R)}{P(d_l|\theta_N)} = \sum_m c(w_m,d_l) \log \frac{\theta_R^m}{\theta_N^m} \quad (2)$$

Since the relevant class contains the query information, θ_R^m is larger than θ_N^m for any word w_m occurring in the query. Consequently, $Score^{MN}$ increases linearly with term frequency $c(w_m, d_l)$. However, the change in the relevance score caused

by increasing term frequency from 1 to 2 should be larger than that caused by increasing term frequency from 2 to 3. In another word, score function is a concave function of term frequency such that score increase rate decreases with term frequency. Therefore, the probabilistic model based on the multinomial distribution violates the concavity constraint of the score function. The classical probabilistic model [13] uses binary indexing to avoid this linearity of term frequency in the score function. Another drawback of the score function (2) is the inability to capture document length.

Consequently, the multinomial distribution is not an appropriate distribution for the probabilistic model. Because the multinomial distribution assumes the independence of the word repetitive occurrences, it results in a score function which incorporates undesired linearity in term frequency. *To capture the concave property and penalize document length in the score function, a more appropriate distribution should be able to model the dependency of word repetitive occurrences (burstiness) that is if a word appears once, it is more likely to appear again.* The Dirichlet compound multinomial (DCM) distribution [11, 10], which is motivated by the Polya urn scheme, is able to capture word burstiness, and thus better addresses the need to capture score function concavity and document length.

2.2 A Detailed Description of the Model

In the Bayesian hierarchical modeling framework, the Dirichlet distribution is a commonly used conjugate prior distribution of the multinomial distribution. The Dirichlet distribution for the parameters of the relevant class is

$$P(\theta_R|\beta_R) = \frac{\Gamma(S_R)}{\prod_{m=1}^{V}\Gamma(\beta_R^m)}\prod_{m=1}^{V}\theta_R^{m(\beta_R^m-1)} \tag{3}$$

where θ_R denotes the parameters of the multinomial distribution; β_R^m denotes the parameters of the Dirichlet distribution; $S_R = \sum_{m=1}^{V}\beta_R^m$. We define $p(\theta_N|\beta_N)$ similarly as in Equation (3).

Hierarchical Bayesian modeling treats the generation of a document in the following way: A sample is drawn from the Dirichlet distribution to generate a multinomial distribution, and then a document is generated by the multinomial distribution. This hierarchical Bayesian model is called the Dirichlet compound multinomial (DCM) distribution [11, 10, 1]. In the DCM distribution, the actual parameters are the Dirichlet parameters β_R^m and β_N^m, because the multinomial parameters θ_R^m and θ_N^m have been integrated out. The DCM distribution for the relevant class is defined below.

$$\begin{aligned}
P(d_l|\beta_R) &= \int P(d_l|\theta_R)P(\theta_R|\beta_R)d\theta_R \\
&= \frac{n(d_l)!}{\prod_m c(w_m,d_l)!}\frac{\Gamma(S_R)}{\prod_m\Gamma(\beta_R^m)}\int_0^1\prod_m\theta_R^{m(\beta_R^m-1+c(m,d_l))}d\theta_R^m \\
&= \frac{n(d_l)!}{\prod_m c(w_m,d_l)!}\frac{\Gamma(S_R)}{\Gamma(S_R+n(d_l))}\prod_m\frac{\Gamma(c(w_m,d_l)+\beta_R^m)}{\Gamma(\beta_R^m)}
\end{aligned} \tag{4}$$

The first line of the above equation is derived by directly multiplying the Dirichlet distribution (3) with the multinomial distribution (1). The third line of the above equation is derived by treating $\int_0^1\prod_m\theta_R^{m(\beta_R^m+c(w_m,d_l)-1)}d\theta_R^m$ as an unnormalized version of the Dirichlet distribution with parameters $\beta_R^m + c(w_m,d_l)$. We can derive the DCM distribution for the non-relevant class $p(d_l|\beta_N)$ in the same way. Equation (4) looks fairly daunting, yet it can be significantly simplified by taking the log-odds ratio.

The Bayesian hierarchical modeling perspective does not provide very intuitive insights on how the DCM distribution captures word burstiness. The DCM distribution, however, also arises naturally from the Polya urn scheme [5], which explains the intuition. The Polya (DCM) distribution models the following scenario: Consider an urn filled with colored balls, with one color for each word in the vocabulary, thus, the generation of a document can be simulated as drawing color balls from the urn. The multinomial distribution models the standard draw with replacement scheme, and the Polya (DCM) distribution models the draw with one additional replacement scheme. Consequently, following the second scheme, words that have already been drawn are more likely to be drawn again, which is word burstiness.

The classical probability ranking principle [16] suggests ranking the documents by the log-odds ratio of their probabilities of being generated by the relevant class against the non-relevant class. Using the DCM distribution (4), we can rank documents by the following score function:

$$\begin{aligned}
Score(d_l) &= \log\frac{\frac{\Gamma(S_R)}{\Gamma(S_R+n(d_l))}\prod_m\frac{\Gamma(c(w_m,d_l)+\beta_R^m)}{\Gamma(\beta_R^m)}}{\frac{\Gamma(S_N)}{\Gamma(S_N+n(d_l))}\prod_m\frac{\Gamma(c(w_m,d_l)+\beta_N^m)}{\Gamma(\beta_N^m)}} \\
&= \sum_{m:c(w_m,d_l)>0}\sum_{i=0}^{c(w_m,d_l)-1}\log\frac{\beta_R^m+i}{\beta_N^m+i} \\
&\quad - \sum_{i=0}^{n(d_l)-1}\log\frac{S_R+i}{S_N+i}
\end{aligned} \tag{5}$$

The first line of the above equation is derived by canceling the common term $n(d_l)!/\prod_m c(w_m,d_l)!$ in the denominator and numerator. The second line of the above equation is derived by noticing that $\Gamma(s+n)/\Gamma(s) = \prod_{i=0}^{n-1}s+i$. The score function (5) consists of two components: the first term depends on all the words occurring in the documents and the second term depends on the document length $n(d_l)$. Therefore, the complexity of scoring all the documents in the collection depends on the total term occurrences (including repetition) in the collection. Nevertheless, appropriate initialization will significantly reduce the computational complexity, and we will further analyze the computation issue below.

In the information retrieval tasks, little information regarding the user's retrieval intent is available. Constructing the relevant model and non-relevant model without any relevance feedback information is a challenging problem in implementing the classical probabilistic model. User query and collection distribution are the only information available to the retrieval system. Intuitively, the initial parameters of the relevant model should capture the information in the query. Because there is limited amount of text contained in query, smoothing becomes extremely critical. The smoothing method involves a linear combination of the non-relevant model and the query, and is similar to the document smoothing approaches in the language model [20].

$$\beta_R^m = \beta_N^m + \gamma\cdot c(w_m,q) \tag{6}$$

where γ controls the degree of smoothing in the relevant model; $c(w_m,q)$ is the term frequency of word w_m in query

q. Based on this initialization, a word not occurring in the query has the same parameter in the relevant model as the non-relevant model. By plugging Equation (6) into Equation(5), *we obtain the final score function of the DCM retrieval algorithm.*

$$Score(d_l) = \sum_{\substack{m;c(w_m,d_l)>0 \\ c(w_m,q)>0}} \sum_{i=0}^{c(w_m,d_l)-1} \log(1+\frac{\gamma \cdot c(w_m,q)}{\beta_N^m + i}) - \sum_{i=0}^{n(d_l)-1} \log(1+\frac{\gamma \cdot n(q)}{S_N+i}) \quad (7)$$

where $n(q)$ is the length of query q. The above score function consists of two components: The first term depends on the frequency of word co-occurring in the query and the document (that is term frequency); the second term depends on the query length and the document length. The first term of the score function increases with the term frequency, and the score increase rate is larger for smaller term frequencies because of the concavity of the log function. This is consistent with the intuition that repeated occurrences of query terms in the document have less impact on the relevance than their first occurrence, which is the basic term frequency constraint in [2]. The second term decreases with document length. This indicates that if two documents contain the same number of query terms, the shorter document is more likely to be relevant because it contains fewer non-relevant terms. This intuition also agrees with the length normalization constraint in [2].

Now we analyze the computational efficiency of the score function (7). *The first term of the score function depends only on the terms co-occurring in the query and document, and consequently inverted indexing can significantly speed up the computation. The second term of this score function can be pre-computed, because this term only requires information on query length and document length. Thus, this model is very efficient to implement for large scale dataset.*

2.3 Estimation of the Non-relevant Model

In this section, we propose three approaches to estimate the non-relevant model from the document collection. The first approach is to fit a DCM distribution to the collection, and then compute the maximum likelihood estimate (MLE). To reduce the computation, the second approach is to fit an approximated DCM distribution to the collection, and then compute the MLE. The third approach is to fit the DCM distribution to the collection, and then compute the Leave-one-out estimate.

2.3.1 MLE based on the DCM Distribution

Applying the maximum likelihood estimator based on the DCM distribution is the most straight forward approach to estimate the non-relevant model. We can not derive a 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 vector by computing the gradient of the DCM log likelihood. The maximum likelihood estimate [10, 11] can be computed using the fixed point iteration.

$$\beta_N^{m\,new} = \beta_N^m \frac{\sum_i \Psi(c(w_m,d_i)+\beta_N^m) - \Psi(c(w_m,d_i))}{\sum_i \Psi(n(d_i)+S_N) - \Psi(n(d_i))} \quad (8)$$

where Digamma function Ψ is defined as $\Psi(\beta) = \frac{d}{d\beta}\log\Gamma(\beta)$. A detailed proof is given in [11].

2.3.2 MLE based on the EDCM Distribution

Although the estimation procedure can be done off-line, the above formulation is still very inefficient when the collection size is large. The DCM distribution can be approximated by the EDCM distribution [1] to reduce the computation, when the dimension is very large. The EDCM distribution is defined as

$$p(d_l|\beta_N) = \frac{n(d_l)!\Gamma(S_N)}{\Gamma(S_N+n(d_l))}\prod_m \frac{\beta_N^m}{c(w_m,d_l)} \quad (9)$$

Based on the EDCM distribution, we can derive the maximum likelihood estimate by the following steps.

$$S_N = \frac{\sum_i \sum_m I(c(w_m,d_i))}{\sum_i \Psi(S_N+n(d_i)) - N\Psi(S_N)}$$
$$\beta_N^m = \frac{\sum_i I(c(w_m,d_i))}{\sum_i \Psi(S_N+n(d_i)) - N\Psi(S_N)} \quad (10)$$

We can use the fixed point iteration to calculate S_N based on the first Equation, and calculate β_N^m based on the second Equation after the fixed iteration algorithm converges. The calculation of the maximum likelihood estimate for the EDCM distribution is more efficient than that of the DCM distribution, because only S_N is calculated in the fixed point iteration in the MLE of the EDCM distribution, while all the β_R^m are calculated in that of the DCM distribution.

2.3.3 LLO based on the DCM Distribution

An alternative to estimate the non-relevant model is to maximize the leave-one-out (LLO) likelihood [21, 11] instead of the true likelihood. The LLO likelihood, based on the cross validation criterion, is the product of the probability of each word given the distributional model constructed by the remaining data with the target word excluded. The LLO log-likelihood [11] for the DCM distribution is

$$l(\mu|C) = \sum_{i=1}^{N}\sum_m c(w_m,d_i)\log\left(\frac{c(w_m,d_i)-1+\beta_N^m}{n(d_i)-1+S_N}\right) \quad (11)$$

In the above equation, $\frac{c(w_m,d_i)-1+\beta_N^m}{n(d_i)-1+S_N}$ is the predictive probability of observing outcome $c(w_m,d_i)$ given the remaining data. This probability is derived from

$$p(d_i|d_i\backslash w_m,\beta_N) = p(d_i|\beta)/p(d_i\backslash w_m|\beta_N) \quad (12)$$

where $d_i\backslash w_m$ represents document d_i without one occurrence of word w_m. Equation (11) does not involve any special functions, so it is very efficient to implement. After taking the first derivative, a convergent fixed-point iteration can be used to solve the above optimization problem.

$$\beta_N^{m\,new} = \beta_N^m \frac{\sum_i \frac{c(w_m,d_i)}{c(w_m,d_i)-1+\beta_N^m}}{\sum_i \frac{n(d_i)}{n(d_i)-1+S_N}} \quad (13)$$

2.4 Estimation of the Relevant Model

Manually tuning the free parameters to accommodate different retrieval tasks dominates much of the research in information retrieval. Automatically estimating the retrieval parameters improves the robustness of the retrieval system significantly. Because γ is query dependent, we process the

estimation of interpolation parameter γ online after the user sends a query. Thus, the computational requirement in estimating the parameter γ is more demanding than in estimating the non-relevant model. Here we apply the EDCM distribution given by Equation (9) to expedite the estimation of the parameter γ. In order to estimate the parameter γ, we approximate the relevant model space by the set of documents which contain all the terms in the query. Thus, we maximize the log likelihood of the set of documents over parameter γ with the EDCM distribution. By plugging Equation (6) into Equation (9), we get

$$\begin{aligned}\hat{\gamma} &= \arg\max_{\gamma} \log \prod_{d_i \in \mathcal{C}} n(d_i)! \frac{\Gamma(S_N + \gamma n(q))}{\Gamma(S_N + n(d_i) + \gamma n(q))} \\ &\times \prod_{m:c(w_m,d_i)\geq 1, d_i \in \mathcal{C}} \frac{\beta_N^m + \gamma c(w_m, q)}{c(w_m, d_i)} \end{aligned} \quad (14)$$

where $\mathcal{C}$ indicates the set of documents containing all the terms in the query. A closed-form solution is not feasible for the above optimization problem. In such a case, gradient descent approaches provide an alternative avenue for estimating parameter γ. Zhai and Lafferty [21] used the whole collection as the relevant space in the second stage estimation, and thus their model requires expensive computation.

3. PSEUDO RELEVANCE FEEDBACK

A natural way to estimate the relevant model from pseudo relevance feedback documents is to assume that the feedback documents are directly generated by the relevant DCM distribution. In addition, the feedback documents also include general English words besides the words relevant to the search topic. Therefore, a more reasonable model would be a mixture model that generates a feedback document by mixing the query topic model with a collection language model. We define two latent generative model components based on the DCM distributions: z_{FR} and z_N. z_{FR} is the feedback relevant model variable, which represents terms occurring in the feedback documents and pertinent to the user's search intent. z_N is the collection model variable, which represents the general English words occurring frequently in the whole collection. The parameters of the z_N are consistent with the parameters of the non-relevant class β_m^N, which can be estimated by one of the three approaches discussed in Section 2.3. Thus, a document is generated by picking a word either from the feedback relevant model z_{FR} or the collection model z_N. The goal of this algorithm is to estimate the feedback relevant model z_{FR} and use the most frequently occurring terms in z_{FR} to enrich the original relevant model.

In order to speed up the computation, we employ the approximated DCM distribution in Equation(9) as the underlying generative sources. Thus, The log likelihood function of the feedback document is

$$\begin{aligned}\ell &= \sum_{d_i \in \mathcal{F}} \log \sum_{k \in \{FR, N\}} P(z_k|d_i)\Gamma(S_k) \\ &- \sum_{d_i \in \mathcal{F}} \log \sum_{k \in \{FR, N\}} P(z_k|d_i)\Gamma(S_k + n(d_i)) \\ &+ \sum_{d_i \in \mathcal{F}} \sum_{m:c(w_m,d_i)\geq 1} \log \sum_k P(z_k|d_i)\beta_k^m - \log c(w_m, d_i) \end{aligned} \quad (15)$$

where $\mathcal{F}$ is the feedback documents set. We fix the background collection model z_N, and apply the EM algorithm to estimate the z_{FR}. The detailed EM step is listed in Table 1. In the E-step, we calculate $P(z_k|d_i, w_m)$, the probability of term w_m in document d_i belongs to generative model z_k. In the M-step, we use $P(z_k|d_i, w_m)$ to calculate both the probability $P(z_k|d_i)$ and the relevant feedback model distribution parameters S_{FR} and β_{FR}^m.

The simple mixture model in [19] fixes the mixing coefficients $P(z_k|d_i)$ across all the feedback documents, even though some feedback documents presumably have more noise than others. Without fixing the mixing coefficients, the EM algorithm will converge to the local optimum where $P(z_{FR}|d_i) = 1$. To address this problem, we propose three improvements to the EM algorithm. These are the critical elements which lead to improvement of the algorithmic performance.

First, the traditional language modeling approach uses the original collection model $p(w_m|C)$, whose parameters are estimated from the whole collection. *Since the feedback documents set $\mathcal{F}$ contains fewer terms than the whole collection, directly applying the collection model results in $\sum_{m:c(w_m,d_i)\neq 0}^{d_i \in \mathcal{F}} P(w_m|z_N) << 1$. Consequently, the underestimated background collection model will cause the EM algorithm to converge to $P(z_{FR}|d_i) = 1$.* Zhai and Lafferty [19] noticed this convergence problem, but they did not explicitly point out the underlying reason. Instead, they solved the problem by fixing $P(z_{FR}|d_i)$ and $P(z_N|d_i)$. More recently, Tao and Zhai [17] addressed this problem by using early stopping to avoid converging to $P(z_{FR}|d_i) = 1$. In contrast to their Ad hoc approach, we use the *reduced collection estimate*, where we only count the word occurring in the feedback documents $S_N^{Reduced} = \sum_{m:c(w_m,d_i)\neq 0}^{d_i \in \mathcal{F}} \beta_N^m$ and $P(w_m|z_N) = \beta_N^m / S_N^{Reduced}$. Thus, we avoid fixing $P(z_k|d_i)$, and the EM algorithm still converges to a desirable local optimum. Moreover, updating the mixing coefficients $P(z_k|d_i)$ helps to converge to the local optimum quickly. Thus, this algorithm is very efficient and requires fewer EM iterations than the simple mixture model.

Second, *a deterministic annealing procedure [18] allows the EM algorithm to find better local optimum of the likelihood function.* Deterministic annealing is also proposed for the EM clustering based on the approximated DCM distribution in [1], which shows that deterministic annealing leads to a substantial better results. We replace the original Expectation step with

$$P(z_k|d_i, w_m) = \frac{(P(z_k|d_i)P(w_m|z_k))^T}{\sum_k (P(z_k|d_i)P(w_m|z_k))^T} \quad (16)$$

where T is a temperature parameter. In each iteration, we decrease $T \rightarrow \eta T$ until the EM steps converge. The parameter η is a large value close to one, and we set $\eta = 0.96$ in the experiments. This shows that the effect of T is to dampen the posterior probabilities such that they will get closer to the uniform distribution with decreasing T.

Third, the simple mixture model [19] does not involve the original query in any way during the EM iteration. In stead, it interpolates the estimated feedback model with original query model by using a fixed interpolation coefficient. A query regularization approach was proposed in [17] for language model based relevance feedback to reduce the deficiency. We apply a query regularization approach similar to [17]. In this algorithm, we treat query q as a relevant document occurring λ times in the M-step of the algorithm. Therefore, the parameter λ controls the relative weight we

Table 1: Detailed Expectation and Maximization Step

1. E Step:
   $P(z_k|d_i, w_m) = \frac{(P(z_k|d_i)P(w_m|z_k))^T}{\sum_k (P(z_k|d_i)P(w_m|z_k))^T} \quad k \in \{FR, N\}$
2. M Step:
   $P(z_k|d_i) = \frac{\sum_m P(z_k|d_i,w_m)\mathbb{1}(c(w_m,d_i)\geq 1)}{\sum_m \mathbb{1}(c(w_m,d_i)\geq 1)} \quad k \in \{FR, N\}$
   $S_{FR} = \frac{\sum_{i=1}^N \sum_m P(z_{FR}|d_i,w_m)\mathbb{1}(c(w_m,d_i)\geq 1)+\lambda n(q)}{\sum_{i=1}^N \psi(S_{FR}+n(d_i))P(z_{FR}|d_i)+\psi(S_{FR}+n(q))\lambda-\sum_{i=1}^N \psi(S_{FR})P(z_{FR}|d_i)-\psi(S_{FR})\lambda}$
   $P(w_m|z_{FR}) = \frac{\beta_k^m}{S_{FR}} = \frac{\sum_{i=1}^N \mathbb{1}(c(w_m,d_i)\geq 1)P(z_{FR}|d_i,w_m)+\lambda c(w_m,q)}{\sum_m \sum_{i=1}^N \mathbb{1}(c(w_m,d_i)\geq 1)P(z_{FR}|d_i,w_m)+\lambda n(q)}$
3. Annealing: lower temperature by setting $T \leftarrow \eta T$.
4. Iterate between E-step and M-step until $|\ell_{new} - \ell_{old}| < \epsilon$.

add the original query to the feedback relevant model. In the experiments, we will show how the parameter λ influences the retrieval performance. After the algorithm converges, we interpolate β_{FR} with β_N to obtain the expanded query. We scale down the value of the largest β_R^m to the same value of the parameter γ in Equation (6) by multiplying $\frac{\gamma}{\max_m \beta_{FR}^m}$.

$$\beta_R^{m\,new} = \beta_{FR}^m \times \frac{\gamma}{\max_m \beta_{FR}^m} + \beta_N^m \tag{17}$$

4. EXPERIMENTS

4.1 Experimental Datasets and Procedure

To evaluate our DCM retrieval algorithm and its pseudo-relevance feedback algorithm described in the previous sections, we experimented with three TREC datasets. The first one is the TREC 2003 HARD track, which uses part of the AQUAINT dataset plus two additional datasets (Congressional Record (CR) and Federal Register (FR)). We do not have the additional datasets in the TREC 2003 HARD track. Our results are still comparable to other published TREC 2003 HARD results, although the data are a little different. The second one is the TREC 7 dataset, which contains data from the TREC Disk 4 and 5 (excludes Congressional Record). The third one is the TREC 8 dataset. For all these tracks, we use the topic titles as queries on all the 50 topics, because they are closer to the actual queries used in real applications. Data pre-processing is standard: terms were stemmed using the Porter Stemming and stop words were removed by using standard stop word list.

We employed the Lemur Toolkit as our retrieval system. To measure the performance of the retrieval algorithms, we used three standard Ad hoc retrieval measures: (1) Mean Average Precision (MAP), which is calculated as the average of the precision after each relevant document is retrieved, reflects the overall retrieval accuracy. (2) Precision at 10 documents (Pr@10): this measure gives us the precision for the first 10 documents.

4.2 Effectiveness of the DCM Retrieval Model

To evaluate the effectiveness of our DCM retrieval algorithms, We experimented with 6 variants of the DCM retrieval algorithm, and the notations are shown in Table 2. we compared the DCM retrieval algorithms with the probabilistic model based on the multinomial distribution(MN), the Dirichlet prior smoothing language model (DP) [20] and the two stage smoothing retrieval model (TS) [21].

In order to obtain a fair comparison, we pursued 5-fold cross-validation on the DP algorithm, the MN algorithm and

Table 2: Notations of different variants of the DCM retrieval algorithms

	Tuned γ	**E**stimated γ
MLE of **F**ull DCM	**DCM-F-T**	**DCM-F-E**
MLE of **A**pprox. DCM	**DCM-A-T**	**DCM-A-E**
LOO of Full DCM	**DCM-L-T**	**DCM-L-E**

DCM retrieval algorithm with tuned parameters (DCM-F-T, DCM-A-T, DCM-L-T), and then compared their cross-validation performance (CVP) with the DCM retrieval algorithms with automatic parameter estimation DCM-F-E, DCM-A-E, DCM-L-E and the TS algorithm (these four algorithms are parameter free). For the probabilistic model based on multinomial distribution, we use collection multinomial model as non-relevant model, and interpolate the query multinomial model with collection multinomial model to generate the relevant model. We separated 50 queries into 5 parts, where each part contains 10 queries. For the kth set of queries, we trained the parameters to optimize the retrieval performance for the other 4 sets of queries, and used this set of the parameters to test on kth ($k = 1, 2, 3, 4, 5$) set of queries to obtain the retrieval performance measure for kth part. The cross-validation performance is the average performance on the 5 test query sets.

In Table 3, we show the experimental results of these retrieval algorithms and indicate the best performance in bold. From the results, all our DCM retrieval algorithm variants consistently outperform the DP algorithm and the TS algorithm. The MN algorithm performs worst among all the algorithms, because it ignore the concavity of score function and document length. The DCM retrieval algorithms with LLO estimate (DCM-L-T and DCM-L-E) have the best performance among all the variants of the DCM retrieval algorithm. We observe that $S_N^L > S_N^F > S_N^A$, where S_N^L, S_N^F and S_N^A are the sum of non-relevant parameters estimated by the LLO estimator based on the full DCM, the MLE based on the full DCM, and the MLE based on the EDCM respectively. We can rank these three estimation approaches in terms of retrieval accuracy from the most accurate to the least accurate: LLO estimate, MLE based on Full DCM distribution, and MLE based on the approximated DCM distribution. As indicated in [1], the parameter $S_N = \sum_{m=1}^N \beta_N^m$ indicates the burstiness of the distribution. Increasing S_N decreases burstiness and vice versa. Since the non-relevant model captures the information in the large TREC collections, less burstiness is more suitable. The parameter S_N plays the same role as the parameter μ in the Dirichlet prior smoothing language model. The experiments in [20] show

Table 3: Comparison of retrieval algorithms on three TREC datasets. An asterisk (*) beside the DCM-L-T performance value indicates that the performance difference between the DCM retrieval algorithm DCM-L-T and the Dirichlet prior smoothing model DP is statistically significant according to the Wilcoxon signed rank test at the level of 0.05. We perform the same significance test between the DCM retrieval algorithm DCM-L-E and the two stage smoothing model TS.

Topic titles	Eval	DP	MN	TS	DCM-F-T	DCM-A-T	DCM-L-T	DCM-F-E	DCM-A-E	DCM-L-E
HARD 2003	MAP	0.3135	0.1666	0.3170	0.3197	0.3146	0.3210*	0.3188	0.3125	**0.3217***
	Pr@10	0.4993	0.3040	0.5000	0.5100	0.4880	**0.5220***	0.5040	0.4980	0.5080
TREC 7	MAP	0.1857	0.0646	0.1831	0.1850	0.1838	**0.1866***	0.1828	0.1828	0.1858*
	Pr@10	0.4180	0.1340	0.4320	0.4400	0.4260	**0.4460***	0.4300	0.4200	0.4360
TREC 8	MAP	0.2520	0.0721	0.2517	0.2539	0.2538	0.2545*	0.2534	0.2530	**0.2554***
	Pr@10	**0.4540**	0.1440	**0.4540**	0.4520	0.4500	0.4520	0.4480	0.4420	0.4480

that the retrieval algorithm performs constantly well when the parameter μ is large. This indicates why the LLO estimate performs consistently better than other approaches.

4.3 Pseudo-relevance Feedback Algorithm

In this section, we compare the DCM pseudo relevance feedback algorithm (DCM-PR) with language model pseudo relevance feedback algorithms, including the simple mixture model (SMM) [19], the divergence minimization algorithm (DM) [19], and the regularized mixture model (RMM)[17].

The simple mixture model and the divergence minimization model are the standard language model based feedback algorithms [19] with strong performance. The simple mixture model algorithm models the feedback documents as a mixture of feedback topic model and background collection model. It uses the EM algorithm to estimate the feedback topic model and interpolates with the original query model. The divergence minimization algorithm models the relevance feedback in an optimization framework, and tends to minimize the divergence between the feedback topic model and the feedback documents, and at the same time maximize the divergence between the feedback topic model and the background collection model. It also interpolates the original query model with the feedback topic model. The regularized mixture model [17] is the most up to date pseudo-relevance algorithm, which uses query regularization technique and performs an early stopping of the EM iteration.

We use the Dirichlet prior language model with a tuned parameter (DP) as the baseline retrieval model for the simple mixture model (SMM), divergence minimization model (DM) and regularized mixture model (RMM) algorithms. We use the DCM probabilistic retrieval model with the LLO estimate and tuned parameter γ (DCM-L-T) as the baseline retrieval model for the new pseudo relevance feedback algorithm (DCM-PR). We trained these algorithms on TREC 2003 HARD dataset, and set parameter values by optimizing the performance on TREC 2003 HARD dataset. For the SMM and DM algorithms, the tuning parameters are the weighting parameter λ and the interpolation parameter α. We varied both λ and α from 0 to 1.0 with step 0.2. For the TREC 2003 HARD dataset, the SMM algorithm performed best when $\lambda = 0.8$ and $\alpha = 0.8$; the DM algorithm performed best when $\lambda = 0.8$ and $\alpha = 0.4$. The RMM algorithm performed best when $\mu = 30,000$, $\delta = 0.9$ and $\eta = 2$; the DCM-PR algorithm performs best when the query regularization parameter $\lambda = 125$ and annealing damping parameter $\eta = 0.96$. We fixed these parameter settings for all the remaining datasets, and chose the top 20 terms with the largest probabilities in the feedback relevant model for all these algorithms. We compare these algorithms by setting feedback documents size equal to 10 and 30 respectively. The results are shown in Table 4. The DCM-PR algorithm consistently outperforms the SMM, DM and RMM algorithms on all the three datasets in terms of MAP and Pr@10 with significant improvement. The benefits of pseudo relevance feedback decreases as we increase the feedback document size from 10 to 30, because more noise is introduced to the model as more documents are used for pseudo-relevance feedback beyond a limit.

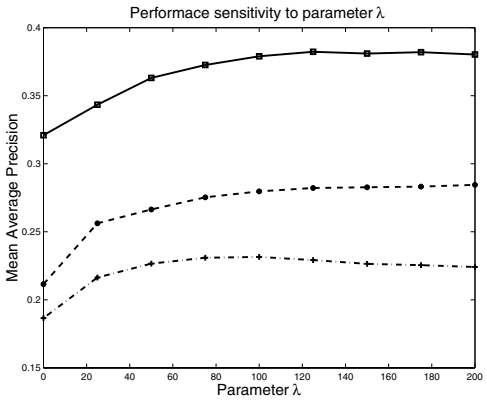

Figure 1: Performance sensitivity to parameter λ.

4.4 Robustness of the Parameter λ

We study the robustness of query regularization parameter λ for the DCM pseudo-relevance feedback algorithm in this section. In the DCM pseudo-relevance feedback algorithm, λ indicates the confidence of the original query model. The larger λ is, the larger weight the original query terms have in the feedback relevant model. In another word, the estimated topic model has a larger impact on the terms in the query if parameter λ is larger.

In the previous experiments, we set the $\lambda = 125$. We conducted another set of experiments by fixing feedback documents number equal 10 and varying parameter λ. In Figure 1, we plotted the MAP for several values of λ for the DCM pseudo-relevance model. The performance is insensitive to the setting of the parameter λ as long as the prior strength λ is set to be larger than 50. *The performance insensitivity to the parameter λ ensures the robustness of the model.*

5. CONCLUSIONS

The main contribution of this paper is a new retrieval algorithm based on the probabilistic model framework and advanced document modeling and estimation techniques. The probabilistic model framework guarantees the theoreti-

Table 4: Comparison of Pseudo-relevance Feedback Algorithms. An asterisks (*) beside the SMM, DM, and RMM performance value indicates that the performance differences between the DCM-PR algorithm and the SMM, DM, RMM algorithms are statistically significant according to the Wilcoxon signed rank test at the level of 0.05.

Data	Num of docs	Eval	SMM	DM	RMM	DCM-PR	Impr. over SMM	Impr. over DM	Impr. over RMM
TREC 2003	10	MAP	0.3760*	0.3552*	0.3728*	**0.3823**	1.68%	7.63%	2.55%
		P10	0.5160*	0.5140*	0.5280*	**0.5560**	7.75%	8.17%	5.30%
	30	MAP	0.3739	0.3449**	0.3679*	**0.3752**	0.35%	8.79%	1.98%
		P10	0.5260*	0.5200*	0.5200*	**0.5600**	6.46%	7.69%	7.69%
TREC 7	10	MAP	0.2198*	0.2007*	0.2184*	**0.2292**	4.28%	14.2%	4.95%
		Pr@10	0.4060*	**0.4240**	0.4020*	0.4080	0.49%	−3.77%	1.49%
	30	MAP	0.2136*	0.1934*	0.2130*	**0.2285**	6.98%	18.14%	7.28%
		Pr@10	0.4080	**0.4140**	0.4060	**0.4140**	1.47%	0.00%	−1.97%
TREC 8	10	MAP	0.2782	0.2456*	0.2789	**0.2822**	1.44%	14.9%	1.18%
		Pr@10	0.4660*	0.4100*	0.4680*	**0.4760**	2.15%	16.1%	1.71%
	30	MAP	0.2609*	0.2458*	**0.2727**	0.2668	2.26%	8.54%	−2.16%
		Pr@10	0.4560*	0.4200*	0.4520*	**0.4880**	7.02%	16.2%	7.97%

cal rigorousness, and the advanced document modeling and estimation techniques lead to efficient retrieval and accurate ranking. We have proposed applying the DCM distribution as the generative source in the probabilistic model, and this distribution is able to capture the dependency among repetitive word occurrences. To reduce the parameter tuning step, we have proposed several estimation approaches that automatically set the parameters of retrieval ranking function. To further improve the retrieval accuracy, we have proposed a pseudo-relevance feedback algorithm based on the DCM distribution. We have evaluated the algorithm on various TREC datasets. The experimental results show that our algorithm outperforms the existing language model based algorithms significantly.

There are two interesting research directions that will help better understand the role of the DCM retrieval model. First, it would be promising to apply the DCM retrieval model in the relevance feedback context, where the negative feedback documents will help the DCM retrieval algorithm to gain additional benefits. Second, the DCM distribution only accounts for the burstiness among repetitive terms, and ignores burstiness between different but related terms. A new distributional model which is able to capture the burstiness between different terms would be a promising direction to further improve the text retrieval accuracy.

6. ACKNOWLEDGMENTS

We acknowledge support from Cisco, University of California's MICRO program. We also appreciate comments from Charles Elkan, Tom Minka and anonymous reviewers.

7. REFERENCES

[1] C. Elkan. Clustering documents with an exponential family approximation of the dirichlet compound multinomial distribution. In *ICML*, 2006.

[2] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 26th ACM SIGIR conference*, 2004.

[3] F.Song and W.B.Croft. A general language model for information retrieval. In *SIGIR*, 1999.

[4] S. Harter. A probabilistic approach to automatic keyword indexing. *Journal of the American Society for Information Science*, 25(5), 1975.

[5] N. Johnson, S. Kotz, and N. Balakrishnan. *Discrete multivariate distributions*. John Wiley and Sons, 1997.

[6] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th SIGIR*, 2001.

[7] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. *Language Modeling for Information Retrieval, Kluwer International Series on Information Retrieval*, 2003.

[8] V. Lavrenko and W. B. Croft. Relevance-based language models. In *24th SIGIR Conference*, 2001.

[9] D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of 10th European Conference on Machine Learning*, 1998.

[10] R. Madsen, D. Kauchak, and C.Elkan. Modeling word burstiness using the dirichlet distribution. In *Proceedings of 22nd ICML Conference*, 2005.

[11] T. Minka. Estimating a dirichlet distribution. Technical report, Microsoft Research, 2003.

[12] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21th ACM SIGIR Conference*, 1998.

[13] S. Robertson and K. S. Jones. Relevance weighting of search term. *Journal of the American Society for Information Science*, 27, 1976.

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

[15] J. Rocchio. *Relevance feedback in information retrieval. In The Smart System: experiments in automatic document processing*. Prentice Hall, 1971.

[16] S.E.Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33, 1977.

[17] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo relevance feedback. In *Proceedings of the 26th ACM SIGIR conference*, 2006.

[18] N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 1998.

[19] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th CIKM Conference*, 2001.

[20] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *proceedings of SIGIR conference*, 2001.

[21] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *SIGIR*, 2002.