Available online at www.sciencedirect.com
ScienceDirect
Information Processing and Management 44 (2008) 1076–1085
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Fast exact maximum likelihood estimation for mixture of language model

Yi Zhang [a,*,1], Wei Xu [b]

[a] School of Engineering, University of California Santa Cruz, 1156 High Street, Santa Cruz, CA, USA
[b] NEC Lab America, 10080 North Wolfe Road, Suite SW3-350, Cupertino, CA, USA

Received 27 June 2007; received in revised form 26 November 2007; accepted 11 December 2007
Available online 29 January 2008

Abstract

Language modeling is an effective and theoretically attractive probabilistic framework for text information retrieval. The basic idea of this approach is to estimate a language model of a given document (or document set), and then do retrieval or classification based on this model. A common language modeling approach assumes the data D is generated from a mixture of several language models. The core problem is to find the maximum likelihood estimation of one language model mixture, given the fixed mixture weights and the other language model mixture. The EM algorithm is usually used to find the solution.

In this paper, we proof that an exact maximum likelihood estimation of the unknown mixture component exists and can be calculated using the new algorithm we proposed. We further improve the algorithm and provide an efficient algorithm of $O(k)$ complexity to find the exact solution, where k is the number of words occurring at least once in data D. Furthermore, we proof the probabilities of many words are exactly zeros, and the MLE estimation is implemented as a feature selection technique explicitly.

Keywords: Information retrieval; Language models; Mixture model

1. Introduction

A statistical language model is a probabilistic mechanism for generating text. Over the last several years, the language modeling approach has become an important research topic in the information retrieval community due to its theoretical attractiveness and practical effectiveness. In particular, the mixture of language modeling approach has been applied to various information retrieval tasks such as relevance feedback (Hiemstra, Robertson, & Zaragoza, 2004; Kraaij, Pohlmann, & Hiemstra, 1999; Zhai & Lafferty, 2001), context sensitive

* Corresponding author. Tel.: +1 831 459 4549; fax: +1 831 459 4826.
E-mail addresses: yiz@soe.ucsc.edu (Y. Zhang), xw@sv.nec-labs.com (W. Xu).
[1] Part of the work was carried out when the author was at Carnegie Mellon University.

0306-4573/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2007.12.003

language model smoothing (Zhou, Hu, Zhang, Lin, & Song, 2006), and novelty detection (Zhang, Callan, & Minka, 2002).

In all these IR applications, we assume a sequence of text data D is generated from a language model r, which is a linear combination of two multinomial language models p and q

$$r = \alpha p + \beta q \tag{1}$$

where α and β are interpolation weights that sum to 1. As described later in Section 2, the observed frequency of each word in the data, the language model p, α and β are given in most of the problem settings. EM algorithm is usually used to find the unknown mixture component q that maximizes the likelihood of observed data D.

In this paper, we derive an exact solution and propose a fast algorithm of $O(k)$ complexity to find the exact solution, where k is the number of unique words occurring at least once in D. We verify the theoretical results with experiments that find topic language models for relevance feedback. As expected, the EM algorithm converges to the result calculated directly by our algorithm. Besides, the exact solution reveals some interesting properties of the mixture language model, and this paper also discusses these properties.

The rest of the paper is organized as follows. Section 2 provides more details about the mixture modeling approach and how it is used for relevance feedback and novelty detection. Section 3 describes a commonly used EM approach for finding the unknown mixture component. Section 4 proof that an exact maximum likelihood estimation exists and Section 5 presents a new fast algorithm of $O(k)$ complexity to find the exact solution. Section 6 presents some experimental results that demonstrate the efficiency and effectiveness of the proposed algorithm. Section 7 concludes with further discussion about the implication of the exact solution and fast algorithm.

2. Mixture language model

The research reported in this paper is motivated by the following two important information retrieval tasks where the mixture language model was successfully used.

2.1. Mixture language model for relevance feedback

Relevance feedback uses the user feedback about whether or not some documents are relevant to a query to refine the query model. A search engine can achieve significant amount of performance increase through relevance feedback (Lavrenko & Croft, 2001; Miller, Leek, & Schwartz, 1999).

Zhai and Lafferty (2001) proposes a model based approach for relevance feedback by assuming that each relevant document is generated by a mixture of a query topic model q and a collection language model p. Each word in a relevant document is picked up using either the query topic model $q(w \mid \theta)$ or the collection language model $p(w \mid C)$. The probability of generating a set of relevant documents $D = \{d_1, d_2, \ldots, d_J\}$ is

$$\begin{aligned} P(D \mid \theta) &= \prod_{j=1}^{J} P(d_j \mid \theta) \\ &= \prod_{j=1}^{J} \prod_{i=1}^{k} (\alpha q(w_i \mid \theta) + \beta p(w_i \mid C))^{tf_{i,j}} \\ &= \prod_{i=1}^{k} (\alpha q(w_i \mid \theta) + \beta p(w_i \mid C))^{f_i} \end{aligned}$$

where i is the index for words, j is the index for documents, J is the number of documents in D, k is the number of unique words occurring at least once in $D, tf_{i,j}$ is the term frequency of word w_i in document d_j, and $f_i = \Sigma_{j=1}^{J} tf_{i,j}$ is the frequency of word w_i in document set D. α and β are given, and $\alpha + \beta = 1$.

We can treat the maximum likelihood estimator (MLE) of the collection language model p as already known. Usually it is calculated directly as

$$\hat{p}_i = \widehat{P}(w_i \mid p) = \frac{cf_i}{\sum_j cf_j}$$

where cf_i is the number of times word w_i occurs in the collection.

The EM algorithm was used to find the query model $q(w \mid \theta)$. When estimating the query model, the system "purifies" the document by eliminating some background noise, which is explained by the collection language model $p(w \mid C)$. Thus the estimated query model will be concentrated on words that are common in relevant documents, but not very common in the collection.

After finding the query model $q(w \mid \theta)$, the retrieval system interpolates it with the original query model to obtain an updated query model. The KL-divergence between the updated query model and a document language model is used as the relevance score for a document. Experimental results show that the mixture language modeling approach works better than Rocchio method in terms of precision.

2.2. Mixture language model for novelty detection

Another emerging research problem in IR is novelty detection, which focuses on estimating the novelty and redundancy of relevant documents while searching or filtering. In the task of detecting novel relevant documents (Zhang et al., 2002) and sentences (Allan, Wade, & Bolivar, 2003), researchers have compared several methods. One algorithm that performs well is based on the following novel mixture model of how each relevant document is generated.

As shown in Fig. 1, we assume each relevant document is generated by the mixture of three language models: A general English language model θ_E, a user-specific topic model θ_T, and a document-specific language model θ_{d_core}. Each word w_i in the document is generated by each of the three language models with probability λ_E, λ_T and λ_{d_core}, respectively:

$$P(w_i \mid \theta_E, \theta_T, \theta_{d\_core}, \lambda_E, \lambda_T, \lambda_{d\_core}) = \lambda_E P(w_i \mid \theta_E) + \lambda_T P(w_i \mid \theta_T) + \lambda_{d\_core} P(w_i \mid \theta_{d\_core})$$

where $\lambda_E + \lambda_T + \lambda_{d\_core} = 1$.

For example, if the user is interested in "Star Wars", words such as "is" and "the" probably come from the general English model θ_E. Words such as "star" and "wars" probably come from the topic model θ_T. For a

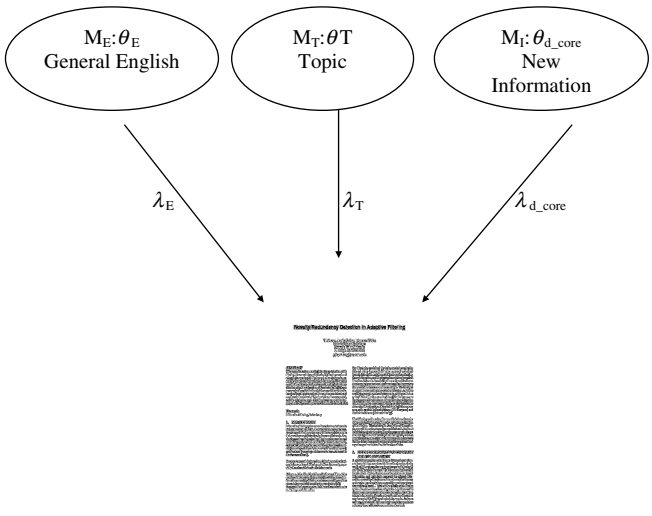

Fig. 1. A mixture model for generating relevant documents.

document with the title "Martin Marietta Is Given Contract For Star Wars Site", the words "Martin" and "Marietta" are more likely to be generated from the new document model θ_{d_core}. θ_T is the core information of a topic, while θ_{d_core} is the core information of a particular relevant document.

In the task of adaptive information filtering, (Zhang et al., 2002) first estimates an approximation of θ_{T_core} based on all labeled relevant documents the user has seen. For a new document to be considered, $\lambda_E, \lambda_T, \lambda_{d\_core}, \theta_{T\_core}, \theta_E$ were fixed. The document core model $\theta^*_{d\_core}$ is estimated to maximizes the likelihood of the document.

$$\theta^*_{d\_core} = \text{argmax}_{\theta_d} P(d \mid \theta_E, \theta_T, \theta_{d\_core}, \lambda_E, \lambda_T, \lambda_{d\_core})$$

We can assume a document with a θ_{d_core} that is quite different from that of the relevant documents the user has read before is more likely to be novel. With the document information model θ_{d_core} that describes new information covered by a document d, a similarity measure between two documents based on the core information contained in each document can be defined as

$$R(d_t \mid d_i) = KL(\theta_{d_t\_core}, \theta_{d_i\_core})$$

If a document is not similar to any of the documents that the user has seen before, it is considered novel.

One major step here is estimating the document core model $\theta^*_{d\_core}$. Let $p = \lambda_E\theta_E + \lambda_T\theta_T, \alpha = 1 - \lambda_{d\_core}$, $\beta = \lambda_{d\_core}, q = \theta_{d\_core}$. Finding $\theta^*_{d\_core}$ is equivalent to the problem described in Eq. (1).

3. Existing solution: EM algorithm

In the examples introduced in Section 2, the system observes a sequence of data D that is assumed to be generated from a linear combination of two multinomial models p and q. We formalize this as

$$r_i = \alpha p_i + \beta q_i \tag{2}$$

where α and β are interpolation weights that sum to 1.0. The log-likelihood of data is

$$\text{LL} = \sum_i f_i \log(r_i) = \sum_i f_i \log(\alpha p_i + \beta q_i) \tag{3}$$

where f_i is the observed frequency of word w_i in the data. Depending on the particular task, q can be query model, document model or topic model.

The traditional way to find q is using the EM algorithm. Each EM iteration is as follows:

$$f^{(n)}_{i,q} = \frac{\beta q^{(n)}_i}{\alpha p_i + \beta q^{(n)}_i}$$

$$q^{(n+1)}_i = \frac{\sum_d f_i * f^{(n)}_{i,q}}{\sum_j \sum_d f_j * f^{(n)}_{j,q}}$$

where n is the iteration number.

Although this solution is used often, it has several weaknesses.

- EM can be computationally expensive, because it is a greedy search algorithm. This expense discourages the use of the language modeling approach for the adhoc retrieval task in environments where computational efficiency is important, for example, in a large scale Web search engine. In such an environment one must choose between speed and accuracy.
- EM can only provide an approximately optimal result. The greater the desired accuracy, the greater the computational cost because of the iterative nature of the algorithm.

4. New exact solution

Instead of using EM, we found that an exact solution exists for the mixture modeling problem. We derive the exact solution in this section based on Lagrange multiplier, a natural method for finding the maximum of a function of several variables subject to one or more constraints.

The problem is to find all q_i that maximize the likelihood of observed data D, for given f_i, p_i, α and β, subject to the constraints $\sum_i q_i = 1$ and $q_i \geqslant 0$. This is a constraint optimization problem.

For all the q_i such that $q_i > 0$, apply Lagrange multiplier method and calculate the derivatives of LL (Eq. (3)) with respect to q_i

$$\frac{\partial}{\partial q_i}\left(LL - \lambda\left(\sum_i q_i - 1\right)\right) = \frac{f_i\beta}{\alpha p_i + \beta q_i} - \lambda = 0 \Rightarrow q_i = \frac{f_i}{\lambda} - \frac{\alpha}{\beta}p_i \tag{4}$$

Applying the constraint that $\sum_i q_i = 1$ to all the $q_i > 0$, we have:

$$\sum_i \left(\frac{f_i}{\lambda} - \frac{\alpha}{\beta}p_i\right) = 1 \tag{5}$$

According to Eq. (5), we get:

$$\lambda = \frac{\sum_{i:q_i>0} f_i}{1 + \frac{\alpha}{\beta}\sum_{i:q_i>0} p_i} \tag{6}$$

Substitute λ in Eq. (4) with the right-hand side of Eq. (6), we get

$$q_i = \frac{f_i\left(1 + \frac{\alpha}{\beta}\sum\limits_{j:q_j>0} p_j\right)}{\sum\limits_{j:q_j>0} f_j} - \frac{\alpha}{\beta}p_i = \frac{\alpha}{\beta}f_i\left(\frac{\frac{\beta}{\alpha} + \sum\limits_{j:q_j>0} p_j}{\sum\limits_{j:q_j>0} f_j} - \frac{p_i}{f_i}\right)$$

Statement: If $q_i : i = 1 \cdots k$ maximize the objective function (Eq. (3)), $q_2 > 0$ and $\frac{p_1}{f_1} < \frac{p_2}{f_2}$, then $q_1 > 0$.

Proof. We prove the statement by contradiction.

Let Δq represents a small positive number. If $q_1 > 0$ is false, then $q_1 = 0$, since q_1 is one component of multinomial distribution and cannot be negative. Then we have:

$$\begin{aligned}
&LL(q_1 + \Delta q, q_2 - \Delta q, q_3, \ldots, q_k) \\
&\quad - LL(q_1, q_2, q_3, \ldots, q_k) && (7)\\
&= f_1 \log(\alpha p_1 + \beta(q_1 + \Delta q)) \\
&\quad + f_2 \log(\alpha p_2 + \beta(q_2 - \Delta q)) \\
&\quad - f_1 \log(\alpha p_1 + \beta q_1) \\
&\quad - f_2 \log(\alpha p_2 + \beta q_2) && (8)\\
&\simeq f_1 \frac{\beta\Delta q}{\alpha p_1 + \beta q_1} - f_2 \frac{\beta\Delta q}{\alpha p_2 + \beta q_2} && (9)\\
&> \frac{\beta\Delta q}{\alpha}\left(\frac{f_1}{p_1} - \frac{f_2}{p_2}\right) && (10)\\
&> 0 && (11)
\end{aligned}$$

Eq. (8) to (9) uses the first order Taylor expansion of log function. Eq. (9) to (10) is based on the assumption $q_1 = 0$ and $q_2 > 0$. Eq. (10) to (11) is based on the assumption $\frac{p_1}{f_1} < \frac{p_2}{f_2}$.

So far, we have shown that:

$$LL(q_1 + \Delta q, q_2 - \Delta q, q_3, \ldots, q_k) > LL(q_1, q_2, q_3, \ldots, q_k)$$

Thus $q_i : i = 1 \cdots k$ do not maximize the log likelihood of data (Eq. (3))

Now we have finished the proof of the statement. □

According to the statement, we can claim that the ratio $\frac{p_i}{f_i}$ for all the $q_i > 0$ is higher than those of all the $q_i = 0$. So we can use the following algorithm to find the exact multinomial distribution q that maximize the likelihood of observed data:

Algorithm 0. Sort $\frac{p_i}{f_i}$ so that $\frac{f_1}{p_1} > \frac{f_2}{p_2} > \cdots > \frac{f_k}{p_k}$, find t such that

$$\frac{\frac{\beta}{\alpha} + \sum_{j=1}^{t} p_j}{\sum_{j=1}^{t} f_j} - \frac{p_t}{f_t} > 0 \tag{12}$$

and

$$\frac{\frac{\beta}{\alpha} + \sum_{j=1}^{t+1} p_j}{\sum_{j=1}^{t+1} f_j} - \frac{p_{t+1}}{f_{t+1}} \leqslant 0 \tag{13}$$

Then q_i's are given by

$$q_i = \begin{cases} \frac{f_i}{\lambda} - \frac{\alpha}{\beta} p_i & \text{if } 1 \leqslant i \leqslant t \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where λ is given by

$$\lambda = \frac{\sum_{i=1}^{t} f_i}{1 + \frac{\alpha}{\beta} \sum_{i=1}^{t} p_i} \tag{15}$$

The complexity of the algorithm is same as sorting, which is $O(k \log(k))$.

5. Fast $O(k)$ algorithm

The key part of the Algorithm 0 is to find the thresholding pair (f_t, p_t). Borrowing ideas from the $O(k)$ algorithm for finding median, we have the following average $O(k)$ algorithm for finding (f_t, p_t).

The general idea of Algorithm 1 is to divide and conquer. We first select a random term/element/word s (Line 4) as a pivot to partition the words into two sets L and G (Line 7–13). Then we decide whether t is in set L or G (Line 15–19). Then we recurse on the appropriate subset (L or G) until we find t (Line 20). On average, it takes $O(\log(k))$ iterations, and the cost of each iteration being roughly half of the previous one. The whole process is a geometric series converges as an average linear time algorithm.

Now we prove that the average computation cost is $O(k)$ using recursion. Let C_k be the average computation cost for n, then we have the following recursion:

$$C_k = k + \frac{1}{k}(C_1 + \cdots + C_{k-1}) \tag{16}$$

Algorithm 1. Find t

Input: A set of k distinct elements.
Output: The element $t \in [1, 2, \ldots, k]$ such that

$$\frac{\frac{\beta}{\alpha} + \sum_{j=1}^{t} p_j}{\sum_{j=1}^{t} f_j} - \frac{p_t}{f_t} > 0$$

and

$$\frac{\frac{\beta}{\alpha} + \sum_{j=1}^{t+1} p_j}{\sum_{j=1}^{t+1} f_j} - \frac{p_{t+1}}{f_{t+1}} \leqslant 0$$

1: $S_p \leftarrow 0, S_f \leftarrow 0$
2: $A \leftarrow \{1, 2, \ldots, k\}$
3: **repeat**
4: randomly select a pivot s from A
5: $L \leftarrow \emptyset, G \leftarrow \emptyset, S_f^* \leftarrow S_f, S_p^* \leftarrow S_p$
6: **for all** $i \in A$ **do**
7: **if** $\frac{f_i}{p_i} > \frac{f_s}{p_s}$ **then**
8: $G \leftarrow G \bigcup \{i\}, S_f^* \leftarrow S_f^* + f_i, S_p^* \leftarrow S_p^* + p_i,$
9: **else if** $\frac{f_i}{p_i} < \frac{f_s}{p_s}$ **then**
10: $L \leftarrow L \bigcup \{i\}$
11: **else**
12: $S_f^* \leftarrow S_f^* + f_i, S_p^* \leftarrow S_p^* + p_i$
13: **end if**
14: **end for**
15: **if** $\frac{\frac{\beta}{\alpha} + S_p^*}{S_f^*} - \frac{p_s}{f_s} > 0$ **then**
16: $A \leftarrow L, S_p \leftarrow S_p^*, S_f \leftarrow S_f^*, t \leftarrow s$
17: **else**
18: $A \leftarrow G$
19: **end if**
20: **until** $A = \emptyset$
Output: t = s

From Eq. (16), it can be derived that

$$C_k = 2k - \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{k}\right) \tag{17}$$

Hence the average computation cost is $O(k)$. k is the number of words with non-zero frequency in D. Because words not in data D do not influence the estimation of λ and are ranked lower than the pivot, we can ignore them in Algorithm 1. This property is nice, because k could be much smaller than the vocabulary size. For example, if D is a document generated by the mixture of document language model q and background English language model p, we only need to process words in the document while estimating the document language model p using Algorithm 1.

The worst-case running time is $O(k^2)$, because the system could be unlucky and pick up a pivot resulting a very unbalanced partition around the largest remaining item. The algorithm can also be modified to a worst-case $O(k)$ algorithm. This can be achieved by using s corresponding to the median of $\frac{f_i}{p_i}$ at Line 4 instead of randomly selecting s.

If implemented appropriately, the division operations can be mostly avoided in this algorithm. For example, Line 7 can be implemented as if $f_i p_s > f_s p_i$. As we know, division is a much more time consuming floating point operation than multiplication. Our simulations show that the EM algorithm converges to the results of our exact algorithm, while our algorithm takes about the same time as 1–3 EM iterations

to finish. Needless to say that the EM algorithm needs many iterations to converge (Zhang, Xu, Jamie, 2002). The source code of our algorithm written in *C* is provided online at www.soe.ucsc.edu/~yiz/shared/fastem.cpp.

6. Experimental results

We compared our algorithm with the EM algorithm in the task of finding language model for a topic as describe in Section 1 and Zhai and Lafferty (2001). In our experiments, we used 50 TREC topics (Topic 101–Topic 150) and 197,084 AP News and Wall Street Journal news published in 1988–1990 from TREC CDs 1, 2, and 3. The corpus language model p is calculated using all documents as described in Section 1. α is set to 0.9 arbitrarily.

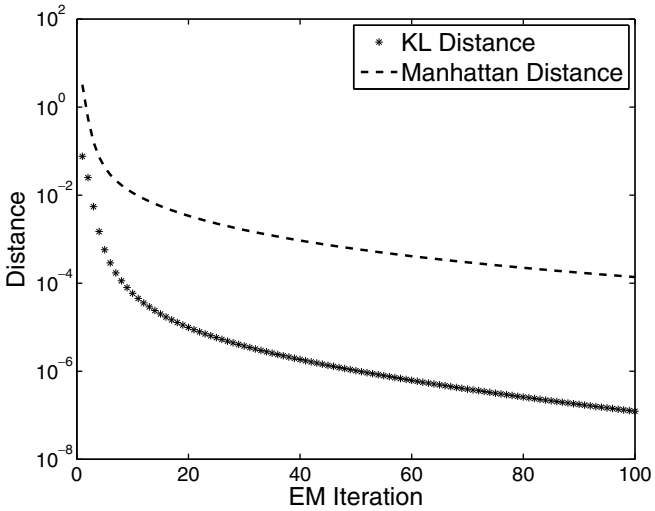

Fig. 2. The multinomial distribution found by the EM algorithm is converging to the distribution calculated directly by our fast algorithm.

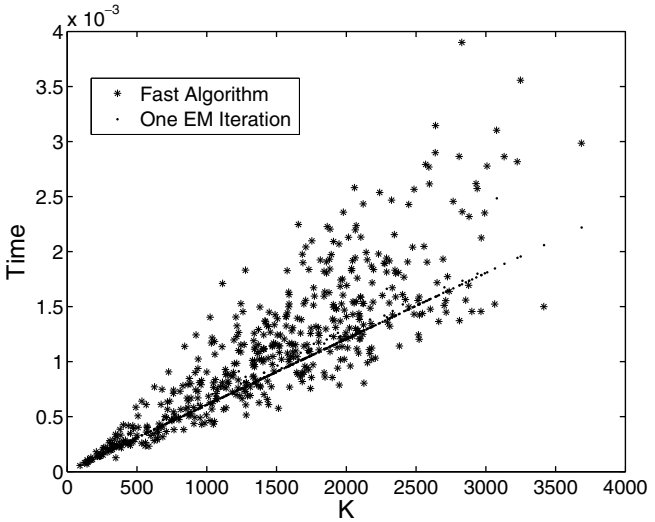

Fig. 3. The behavior of the algorithms for different k. The horizontal axis corresponds to the number of unique words (k). The vertical axis corresponds to the wall clock time for running the algorithm on a PC (one Intel Xeon 2.80GHZ CPU, 1.00 GB of RAM).

We carried out two experiments. In the first experiment, all relevant documents are used as observed training data sequence to learn a topic/relevance language model for each topic. The average of k (the number of unique words) per topic is 7558. Fig. 2 shows the Manhattan distance and Kullback–Leibler distance between the multinomial distribution found by the EM algorithm at each iteration and the distribution calculated by our fast algorithm. We can see that the EM result is actually converging to the result calculated directly by our fast algorithm. As we know, EM is guaranteed to converge to the unique optimal value for mixture of multinomial because the search space is convex. Thus empirically, our result is truly optimal. The new algorithm takes about 5.5 ms wall clock time to learn a topic model. Each EM iteration takes about 4.6 ms, while the exact time for EM algorithm depends on the stopping criteria.

In the second experiment, we vary the number of relevant documents for each topic from 1 to 20. The goal is to tell the behavior of the algorithms for different sizes of k. Fig. 3 shows the results. Each point in the figure tells the wall clock time for learning a topic model using the new algorithm or the wall clock time for finishing one EM iteration. Empirically, the time complexity of the new algorithm is $O(k)$. Although the new algorithm has a worst-case running time of $O(k^2)$ in theory, the worst case scenario or something similar is not observed in our experiment.

7. Conclusion

We provide an exact solution and a fast algorithm to solve the problem of finding the maximum likelihood estimation of the word mixtures, given fixed mixture weights and the density of another multinomial. Experimental result shows the result of the EM algorithm converges to the result calculated exactly with our algorithm, while our algorithm is guaranteed to find the exact unique optimal result at a very fast speed. The result of the paper is beneficial for several information retrieval language modeling applications, such as relevance feedback and novelty detection. It can also benefit other language modeling applications, such as speech recognition and part-of-speech tagging, etc.

Researchers have suggested that the mixture language modeling approach can serve as a feature selection technique. We prove that the probabilities of some words in language model q are exactly zero (Eq. (14)). The linear interpolation weight α serves as a tuning parameter that implicitly controls the number of selected features. The fast algorithm implements the exact MLE estimation for mixture language models as a feature selection technique explicitly.

Acknowledgements

We thank Jamie Callan, Chengxiang Zhai, Thomas Minka and anonymous reviewers for their feedback. Part of the work was supported by National Science Foundation IIS-0713111, AFRL/AFOSR and the industrial sponsors of the Information Retrieval and Knowledge Management Lab at University of California, Santa Cruz.

References

Allan, J., Wade, C., & Bolivar, A. (2003). Retrieval and novelty detection at the sentence level. In *SIGIR'03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 314–321). New York, NY, USA: ACM Press.

Hiemstra, D., Robertson, S., & Zaragoza, H. (2004). Parsimonious language models for information retrieval. In *SIGIR'04* (pp. 178–185). New York, NY, USA: ACM Press.

Kraaij, W., Pohlmann, R., & Hiemstra, D. (1999). Twenty-One at TREC-8: Using language technology for information retrieval. In *TREC 8: Proceedings of the 1999 text retrieval conference*. National Institute of Standards and Technology [special publication].

Lavrenko, V., & Croft, W. B. (2001). Relevance-based language models. In *Research and development in information retrieval* (pp. 120–127).

Miller, D. R., Leek, T., & Schwartz, R. M. (1999). A hidden Markov model information retrieval system. In *Proceedings of SIGIR-99, 22nd ACM international conference on research and development in information retrieval* (pp. 214–221).

Zhai, C., & Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th international conference on information and knowledge management*.

Zhang, Y., Callan, J., & Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. In *SIGIR: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval*.

Zhang, Y., Xu, W., & Jamie, C. (2002). Exact maximum likelihood estimation for word mixtures. In *Text learning workshop in international conference on machine learning* Sydney, Australia.

Zhou, X., Hu, X., Zhang, X., Lin, X., & Song, I.-Y. (2006). Context-sensitive semantic smoothing for the language modeling approach to genomic IR. In *SIGIR: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*.