A novel updating scheme for probabilistic latent semantic indexing

Constantine Kotropoulos and Athanasios Papaioannou

Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki 54124 , Greece
{costas, apapaion}@aiia.csd.auth.gr

Abstract. Probabilistic Latent Semantic Indexing (PLSI) is a statistical technique for automatic document indexing. A novel method is proposed for updating PLSI when new documents arrive. The proposed method adds incrementally the words of any new document in the term-document matrix and derives the updating equations for the probability of terms given the class (i.e. latent) variables and the probability of documents given the latent variables. The performance of the proposed method is compared to that of the folding-in algorithm, which is an inexpensive, but potentially inaccurate updating method. It is demonstrated that the proposed updating algorithm outperforms the folding-in method with respect to the mean squared error between the aforementioned probabilities as they are estimated by the two updating methods and the original non-adaptive PLSI algorithm.

1 Introduction

Information Retrieval (IR) is the research topic that examines how people find information and how tools (such as search engines and catalogues) can be constructed to help people to retrieve information. IR has attracted the attention of researchers for more than 40 years. Nowadays, the World Wide Web is one example of information overload and its expansion has generated needs for more efficient access to global and corporate information repositories. Such repositories are usually text-based, but they increasingly include multimedia content. In this paper, we focus on text-based IR.

The paper builds on the *vector space* model [1], where the available textual data of the training corpus along with the query-documents are represented by numerical vectors. Each vector element corresponds to a different *term*, that is, a distinct word in the corpus [2]. It is generally agreed upon that the contextual similarity between documents exists also in their vectorial representation. Therefore, similarity can be assessed by a vector metric. There are two drawbacks in the original vector space model techniques such as word polysemy (i.e., when one word has many meanings e.g saturn) and synonymy (i.e., two or more words have the same meaning e.g. car and automobile). Polysemy tends to reduce precision, while synonymy tends to reduce recall.

Several vector space dimensionality reduction methods have been proposed in order to solve the two aforementioned problems. For example, *latent semantic indexing* (LSI) maps the documents and the terms onto the so-called *latent semantic space* [3].

LSI performs dimensionality reduction by using *singular value decomposition* (SVD). However, although LSI yields good results, many problems arise due to the lack of a statistical foundation. This happens because LSI assumes that words and documents form a joint Gaussian model. However, Gaussian models can generate negative values. Document vectors whose elements are simply the term counts cannot admit negative values. Contrary to the LSI, a method that has a firm statistical foundation is the *probabilistic latent semantic indexing* (PLSI) [4]. PLSI is based on a statistical model, the so called *aspect model* [5, 6]. It allows to deal with polysemous and synonymous words. It has been proved that it outperforms LSI in document and word clustering applications.

In this paper, a novel method is proposed for updating PLSI when new documents arrive. The proposed method adds incrementally the words of any new document in the term-document matrix and derives the updating equations for the probability of terms given the class (i.e. latent) variables and the probability of the documents given the latent variables. Such an updating scheme is very useful when we deal with applications that refresh their term-document matrix very often. A typical example is a web crawler [7]. The performance of the proposed method is compared to that of the folding-in algorithm, which is an inexpensive, but potentially inaccurate updating method. It is demonstrated that the proposed updating algorithm outperforms the folding-in method with respect to the mean squared error between the aforementioned probabilities as they are estimated by the two updating methods and the original non-adaptive PLSI algorithm.

The outline of the paper is as follows. Section 2 describes briefly LSI, while PLSI is presented in Section 3. The proposed updating algorithm is derived in Section 4. Experimental results are demonstrated in Section 5 and conclusions are drawn in Section 6.

2 Latent Semantic Indexing

LSI has demonstrated an improved performance over the traditional vector space techniques and it has been successfully employed in many IR systems [3]. It is an optimal special case of multidimensional scaling [8] that aims at discovering something about the meaning behind the terms and about the topics in the documents, where the topic is an unobservable (i.e., a latent) variable. LSI models the semantics of the domain in order to yield additional relevant keywords and to reveal the "hidden" concepts of a given corpus while eliminating the high order noise. The attractive point of the method is that it captures the higher order "latent" structure of word usage across the documents rather than just the word surface level. This is done by modeling the association between the terms and the documents based on how terms co-occur across documents. The key idea of LSI is to map terms and documents to a vector space with reduced dimensionality, the latent semantic space. Let $\mathbf{X}$ be the $T \times N$ term-document co-occurrence matrix of rank $r <= \min(T, N)$. LSI is based on an application of SVD to $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^{\top} \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are both column-orthogonal matrices, $\mathbf{D}$ is an $r \times r$ diagonal matrix that contains the non-zero singular values of $\mathbf{X}$, and $\top$ is the transposition operator. An

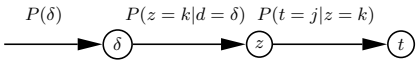


Fig. 1. The data generation process.

approximation of $\mathbf{X}$ is computed by preserving only the largest $K < r$ singular values of $\mathbf{D}$ in $\widetilde{\mathbf{D}}$ and setting the remaining singular values to zero:

$$\widetilde{\mathbf{X}} = \mathbf{U}\widetilde{\mathbf{D}}\mathbf{V}^{\top}. \tag{2}$$

Eq. (2) indicates that the new document-term matrix $\widetilde{\mathbf{X}}$ is no more sparse. So we hope to compute a meaningful association between document pairs and that terms with the same meaning will be mapped to the same subspace.

3 Probabilistic Latent Semantic Indexing

Recently, LSI has been criticized, because its probabilistic model does not match the observed data. Thus, a novel alternative is proposed the so called PLSI that is based on a multinomial model. It has been reported to yield better results for document and word clustering than the standard LSI [4]. PLSI is based on the so called aspect model [5]. In the sequel, the variables z, t, and d denote indices to topics, terms, and documents, respectively. The aspect model is a latent variable model for co-occurrence data which associates an unobserved class variable $z = 1, 2, \ldots, K$ with each observation. So, for any text document $d = 1, 2, \ldots, N$ we assume that the occurrence of a term $t = 1, 2, \ldots, T$ in the document is an observed variable and the topic z is an unobserved one. PLSI defines a generative model for term-document co-occurrences. The assumption is that each term t in a given document d is generated from a latent topic z, i.e. a term is conditionally independent from its original document given the latent topic it was generated from. The data generation process can be described as follows[9]:

1. Select a document $d = \delta$ with probability $P(d = \delta)$.
2. Pick a latent topic $z = k$ with probability $P(z = k|d = \delta)$.
3. Generate a term $t = j$ with probability $P(t = j|z = k)$.

Figure 1 depicts the data generation process. The generative process is described by the joint distribution of a term $t = j$, a latent topic $z = k$, and a document $d = \delta$:

$$P(d = \delta, z = k, t = j) = P(d = \delta)P(z = k|d = \delta)P(t = j|z = k) \tag{3}$$

and the joint distribution of the observed data is given by:

$$\begin{aligned} P(d = \delta, t = j) &= \sum_{k=1}^{K} P(d = \delta, z = k, t = j) \\ &= P(d = \delta) \sum_{k=1}^{K} P(z = k|d = \delta)P(t = j|z = k). \end{aligned} \tag{4}$$

From (4) one can notice that in contrast to document clustering models, document-specific term distributions $P(t|d)$ are obtained by a convex combination of the aspects or factors $P(t|z)$. Documents are not assigned to clusters. They are characterized by a specific mixture of factors with weights $P(z = k|d = \delta)$. So each word in a document is seen as a sample from a mixture model where mixture components are the multinomial $P(t = j|z = k)$ and the mixing proportions are $P(z = k|d = \delta)$. These mixing weights offer more modeling power and are conceptually very different from posterior probabilities in clustering models and (unsupervised) naive Bayes models.

To further simplify the notation we suppress δ, k, and j hereafter. In order to determine $P(d)$, $P(z|d)$, and $P(t|z)$ we should maximize the log-likelihood function

$$\mathcal{L} = \sum_{d=1}^{N} \sum_{t=1}^{T} n(d,t)\ \log P(d,t) \tag{5}$$

where $n(d,t)$ denotes the term frequency, i.e the number of times t occurred in d. It is worth noting that an equivalent symmetric version of the model can be obtained by inverting the conditional probability $P(z|d)$ with the help of Bayes' rule, which results in

$$P(d,t) = \sum_{z=1}^{K} P(z)P(t|z)P(d|z). \tag{6}$$

Eq. (6) is just a re-parameterized version of the generative models described by (3) and (4).

The PLSI algorithm maximizes the log-likelihood of the model by using the Expectation Maximization (EM) algorithm[10]. EM alternates between two steps:

1. An expectation step (E-step) where posterior probabilities are computed for the latent variables z based on the current estimates of the parameters.
2. A maximization step (M-step), where parameters are updated for given posterior probabilities computed in the previous E-step.

For the aspect model in the symmetric parameterization Bayes' rule yields the E-step

$$P(z|d,t) = \frac{P(z)P(t|z)P(d|z)}{\sum_{z'=1}^{K} P(z')P(t|z')P(d|z')} \tag{7}$$

which is the probability that a term t in a particular document or context d is explained by the factor corresponding to z. By straightforward calculations, one arrives at the following M-step re-estimation equations [4]:

$$P(t|z) = \frac{\sum_{d=1}^{N} n(t,d)P(z|d,t)}{\sum_{d=1}^{N}\sum_{t'=1}^{T} n(t',d)P(z|d,t')} \tag{8}$$

$$P(d|z) = \frac{\sum_{t=1}^{T} n(t,d)P(z|d,t)}{\sum_{d'=1}^{N}\sum_{t=1}^{T} n(t,d')P(z|d',t)} \tag{9}$$

$$P(z) = \frac{1}{R}\sum_{d=1}^{N}\sum_{t=1}^{T} n(t,d)P(z|d,t) \tag{10}$$

where

$$R = \sum_{d=1}^{N} \sum_{t=1}^{T} n(t,d). \tag{11}$$

Alternating (7) with (8)-(10) defines a convergent procedure that approaches a local maxima of the log-likelihood.

In [4], a generalization of the EM algorithm for mixture models is proposed, the so called *tempered EM* (TEM). TEM is based on an entropic regularization and is closely related to the deterministic annealing. In short, a control parameter β (the inverse computational temperature) is introduced and the E-step is modified to

$$P_\beta(z|d,t) = \frac{P(z)[P(d|z)P(t|z)]^\beta}{\sum_{z'=1}^{K} P(z')[P(t|z')P(d|z')]^\beta}. \tag{12}$$

For $\beta = 1$, (12) is the standard E-step, while for $\beta < 1$ the likelihood part in Bayes' formula is discounted. It can be shown that TEM minimizes an objective function known as the free energy [11] and hence it defines a convergent algorithm. In the context of PLSI, the main advantage of TEM is that it avoids overfitting. In order to determine the optimal value of β the use of some held-out data is recommended [4]. The typical number of TEM iterations performed starting from randomized initial conditions is 40-60.

The PLSI model can be used to replace the original term-document representation by a representation in a low-dimensional "latent" space in order to perform term clustering or document retrieval. The components of the document in the low-dimensional space are $P(z = k|d)$, $k = 1, 2, \ldots, K$ and for each unseen document or query the aforementioned components are computed by maximizing the log-likelihood with $P(t|z = k)$ fixed [12]. It is obvious that PLSI is not a well-defined generative model of documents, since there is no direct way to assign a probability to an unseen document. However, a better performance for PLSI than LSI was reported on several corpora in [12]. In particular, PLSI is found to perform well even in the cases where LSI fails completely.

4 Updating Scheme for Probabilistic Latent Semantic Indexing

One open problem for PLSI is its updating scheme. In the literature, the only available solution is the well-known method of *folding-in* of a new document, where we project the new document vector to the latent space [13]. However, this method is suitable for document queries and not when new documents are added in the term-document matrix and PLSI model has to be retrained. This happens because the folding-in method calculates only the mixing proportion $P(z|d)$ while the factors $P(t|z)$ are kept fixed.

A novel method is proposed in this paper for updating all the PLSI model parameters. To distinguish between $P(t|z)$ and $P(d|z)$ we introduce the notation $P_1(t|z) = P(t|z)$ and $P_2(d|z) = P(d|z)$. Let us focus on the computations that take place when we proceed from iteration l to iteration $l + 1$ of the EM algorithm. The E-step for itera-

tion $l+1$ is given by

$$P(z|d,t)_{l+1} = \frac{P(z)_l P_1(t|z)_l P_2(d|z)_l}{\sum_{z'=1}^{K} P(z)_l P_1(t|z')_l P_2(d|z')_l}. \tag{13}$$

The M-step for updating $P_1(t|z)$ at iteration $l+1$ is rewritten as

$$P_1'(t|z)_{l+1} = \sum_{d=1}^{N} n(t,d) P(z|d,t)_{l+1} \tag{14}$$

$$P_1(t|z)_{l+1} = \frac{P_1'(t|z)_{l+1}}{\sum_{t'=1}^{T} P_1'(t'|z)_{l+1}}. \tag{15}$$

By substituting (13) into (14) we obtain:

$$P_1'(t|z)_{l+1} = P_1(t|z)_l \sum_{d=1}^{N} \left[\frac{n(t,d)\, P_2(d|z)_l}{\sum_{z'=1}^{K} P(z')_l\, P_1(t|z')_l\, P_2(d|z')_l} \right] P(z)_l \tag{16}$$

Similarly, the M-step for updating $P_2(d|z)$ at iteration $l+1$ is rewritten as

$$P_2'(d|z)_{l+1} = P_2(d|z)_l \sum_{t=1}^{T} \left[\frac{n(t,d)\, P_1(t|z)_l}{\sum_{z'=1}^{K} P(z')_l\, P_1(t|z')_l\, P_2(d|z')_l} \right] P(z)_l \tag{17}$$

$$P_2(d|z)_{l+1} = \frac{P_2'(d|z)_{l+1}}{\sum_{d'=1}^{N} P_2'(d'|z)_{l+1}}. \tag{18}$$

Let us assume that a new document indexed by $d = N+1$ is added at the end of the lth iteration that contains only one word that appears a times. We also assume that the addition of the new document alters neither the number of topics nor the vocabulary of terms. Without any loss of generality, let us assume that the single word is the first word in the vocabulary, i.e. $t = 1$. Therefore, $n(1, N+1) = a$ and $n(t, N+1) = 0$, $t = 2, \ldots, T$. Let $\mathbf{P}_2$ be the $N \times T$ matrix with elements $P_2(d|z)$, $d = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$. To initialize the recursion for the $(l+1)$th iteration, we simply append a new row to $\mathbf{P}_2$ with elements $P_2(N+1|z)_l$ that are numbers uniformly distributed in the interval $[0, 1]$ and we normalize so that each column in $\mathbf{P}_2$ has a unit sum. Under the just described conditions, it can be proven that (16) takes the form

$$P_1''(t|z)_{l+1} = P_1'(t|z)_{l+1} + P_1(t|z)_l \frac{n(t, N+1)\, P_2(N+1|z)_l}{\sum_{z'=1}^{K} P(z')\, P_1(t|z')_l\, P_2(N+1|z')_l} P(z)_l \tag{19}$$

where $P_1'(t|z)_{l+1}$ is simply the value predicted by (16) before the addition of the new document. Eq. (19) is further simplified to

$$P_1''(1|z) = \begin{cases} P_1'(1|z)_{l+1} + P_1(1|z)_l \cdot \\ \cdot \frac{a\, P_2(N+1|z)_l}{\sum_{z'=1}^{K} P(z')_l\, P_1(1|z')_l\, P_2(N+1|z')_l} P(z)_l & t = 1 \\ P'(t|z)_{l+1} & \text{if } t \neq 1. \end{cases} \tag{20}$$

Let

$$A'_{l+1} = \sum_{t=1}^{T} P'_1(t|z)_{l+1} \tag{21}$$

$$A''_{l+1} = \sum_{t=1}^{T} P''_1(t|z)_{l+1} = A'_{l+1} + P''_1(1|z)_{l+1} - P'_1(1|z)_{l+1}. \tag{22}$$

Eq. (15) is simply rewritten as

$$P_1(t|z)_{l+1} = \frac{P''_1(t|z)_{l+1}}{\sum_{t'=1}^{T} P''_1(t'|z)_{l+1}} = \begin{cases} \frac{P''_1(1|z)_{l+1}}{A''_{l+1}} & \text{if } t = 1 \\ \frac{A'_{l+1}}{A''_{l+1}} P_1(t|z)_{l+1} & \text{otherwise.} \end{cases} \tag{23}$$

Similarly, it can be shown that (17) results in

$$P''_2(d|z) = \begin{cases} P''_1(1|z) - P'_1(1|z) & \text{if } d = N+1 \\ P'_2(d|z)_{l+1} & \text{otherwise.} \end{cases} \tag{24}$$

Let

$$B'_{l+1} = \sum_{d=1}^{N} P'_2(d|z)_{l+1} \tag{25}$$

$$\begin{aligned} B''_{l+1} &= \sum_{d=1}^{N+1} P''_2(d|z)_{l+1} = B'_{l+1} + P''(N+1|z)_{l+1} \\ &= B'_{l+1} + P''_1(1|z)_{l+1} - P'_1(1|z)_{l+1}. \end{aligned} \tag{26}$$

Then (18) takes the form

$$P_2(d|z)_{l+1} = \frac{P''_2(t|z)_{l+1}}{\sum_{d'=1}^{N+1} P''_2(d'|z)_{l+1}} = \begin{cases} \frac{P''_2(N+1|z)_{l+1}}{B''_{l+1}} & \text{if } d = N+1 \\ \frac{B'_{l+1}}{B''_{l+1}} P_2(d|z)_{l+1} & \text{otherwise.} \end{cases} \tag{27}$$

Finally, we proceed to updating $P(z)_{l+1}$. Let

$$R_l = \sum_{d=1}^{N} \sum_{t=1}^{T} n(t,d) \tag{28}$$

$$P'(z)_{l+1} = \frac{1}{R_l} \sum_{d=1}^{N} \sum_{t=1}^{T} n(t,d) P(z|d,t)_{l+1} \tag{29}$$

be the values admitted by R and $P(z)$, defined by (11) and (10), before appending the $(N+1)$th document. It is straightforward to show that

$$R_{l+1} = R_l + a \tag{30}$$

$$P(z)_{n+1} = \frac{1}{R_{l+1}} \left[R_l \, P'(z)_{l+1} + P''_1(1|z)_{l+1} - P'_1(1|z)_{l+1} \right]. \tag{31}$$

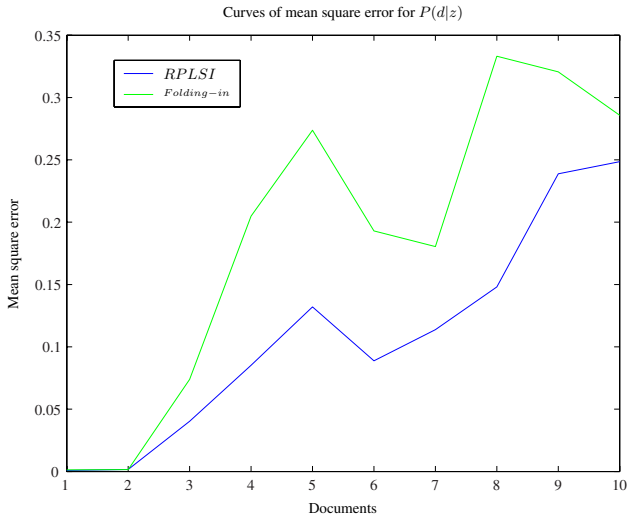
The method can be generalized for a document with more than one terms, if we assume that every time we deal with an elementary document having just one word and we incrementally append as many incremental documents as the terms found in the document. Additional recursions can be applied in order to process more than one documents. The proposed method will be referred to as *recursive probabilistic latent semantic indexing* (RPLSI).

5 Experimental results

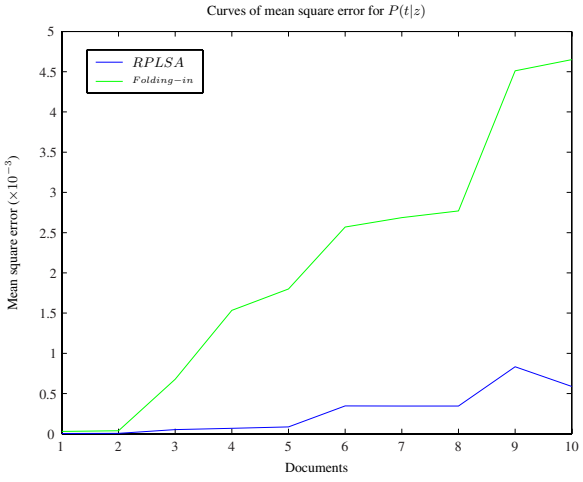
To demonstrate the performance of the proposed updating algorithm for PLSI we have employed a subset of 348 documents from the 20-Newsgroups corpus [14]. The documents used belong to 4 classes. For each document we have kept only 100 terms, those having the highest information gain. After having estimated the parameters of PLSI for the corpus of 348 documents, we start appending a number of documents incrementally. We have compared the accuracy of the proposed updating method with that of the folding-in method. PLSI computes the probabilities $P(t|z)$ and $P(d|z)$, $z = 1, 2, \ldots, K$ for $K = 4$ by resetting the calculation each time a new document is appended. RPLSI and folding-in update the probability values each time a new document is appended. Subsequently, the mean squared error (MSE) has been measured between the exact probability values determined by PLSI and the values estimated by RPLSI by averaging over the latent variable z that refers to classes. The mean squared error between the exact probability values determined by PLSI and the values estimated by folding-in has also been measured. The computations have been performed for 10 and 20 documents. Figures 2 and 3 demonstrate that the proposed method outperforms the folding-in method with respect to the MSE for both $P(t|z)$ and $P(d|z)$. It can be seen that the proposed RPLSI yields almost always a smaller MSE than the folding-in method when a new document is appended. The estimation of $P(t|z)$ is more accurate than $P(d|z)$.

6 Conclusions

A new method for updating the parameters of the PLSI has been proposed that does not require to train the model from the beginning. The proposed method updates not only the probabilities of the new document as folding-in method does, but also all the probabilities of the terms and documents. We have reported first promising experimental results that indicate a better performance than folding-in. In the future, experiments will be conducted with corpora having more documents and more classes. The proposed technique was derived with respect to certain assumptions. Relaxing the constraints is another point of further research. We do not claim that the proposed method yields a new language model. Therefore, it is pointless either to measure perplexity or to compare the model with the latent Dirichlet allocation method.

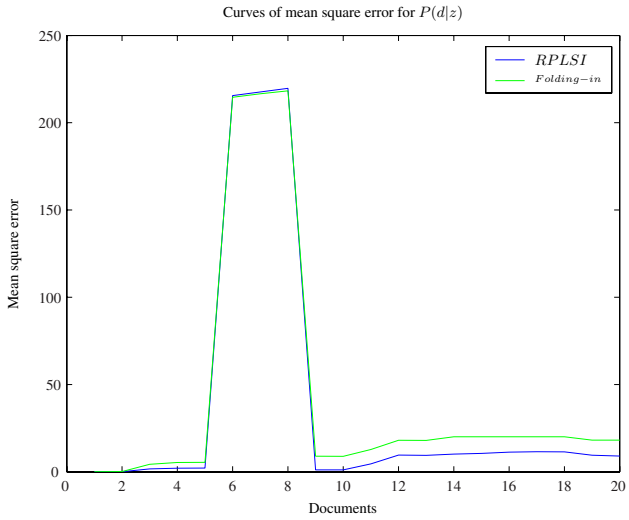


(a)

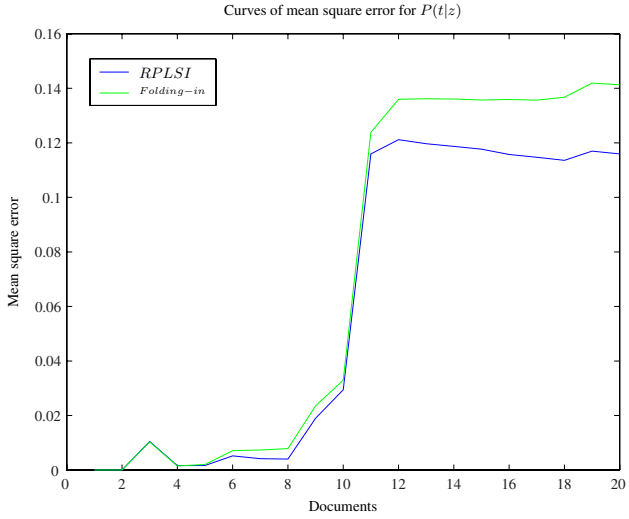


(b)

Fig. 2. (a) Mean squared error for $P(d|z)$ for 10 documents. (b) Mean squared error for $P(t|z)$ for 10 documents.



(a)



(b)

Fig. 3. (a) Mean squared error for $P(d|z)$ for 20 documents.(b) Mean squared error for $P(t|z)$ for 20 documents.

Acknowledgments

This work has been supported by the FP6 European Union Network of Excellence MUSCLE "Multimedia Understanding through Semantics, Computation and Learning" (FP6-507752). The authors would like to thank the anonymous reviewers for their constructive criticism and their careful proofreading.

References

1. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18** (1975) 613–620
2. Yates, R.B., Neto, B.R.: Modern Information Retrieval. ACM Press (1999)
3. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal American Society of Information Science **41** (1990) 391–407
4. Hofmann, T.: Probabilistic latent semantic analysis. In: Proc. Uncertainty in Artificial Intelligence, UAI'99, Stockholm (1999)
5. Hofmann, T., Puzicha, J.: Unsupervised learning from dyadic data. Technical Report TR-98-042, International Computer Science Institute, Berkeley, CA (1998)
6. Saul, L., Pereira, F.: Aggregate and mixed-order Markov models for statistical language processing. In Cardie, C., Weischedel, R., eds.: Proc. 2nd Conf. Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Somerset, New Jersey (1997) 81–89
7. Almpanidis, G., Kotropoulos, C.: Combining text and link analysis for focused crawling. In: Proc. Int. Conf. Advances Pattern Recognition. Volume LNCS 3686. (2005) 278–287
8. Bartell, B.T., Cottrell, G.W., Belew, R.K.: Latent semantic indexing is an optimal special case of multidimensional scaling. In: Proc. Research and Development in Information Retrieval. (1992) 161–167
9. Hofmann, T.: Probabilistic latent semantic indexing. In: Proc. Research and Development in Information Retrieval. (1999) 50–57
10. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm (with discussion). Journal Royal Statistical Society, Series B **39** (1977) 1–38
11. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. Learning in Graphical Models (1999) 355–368
12. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning **42** (2001) 177–196
13. Berry, M.W., Browne, M.: Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM (1999)
14. Lang, K.: Newsweeder: Learning to filter netnews. In: Proc. 12th Int. Conf. Machine Learning. (1995) 331–339