# Online PLSA: Batch Updating Techniques Including Out-of-Vocabulary Words

Nikoletta K. Bassiou and Constantine L. Kotropoulos, *Senior Member, IEEE*

***Abstract*— A novel method is proposed for updating an already trained asymmetric and symmetric probabilistic latent semantic analysis (PLSA) model within the context of a varying document stream. The proposed method is coined online PLSA (oPLSA). The oPLSA employs a fixed-size moving window over a document stream to incorporate new documents and at the same time to discard old ones (i.e., documents that fall outside the scope of the window). In addition, the oPLSA assimilates new words that had not been previously seen (out-of-vocabulary words), and discards the words that exclusively appear in the documents to be thrown away. To handle the new words, Good-Turing estimates for the probabilities of unseen words are exploited. The experimental results demonstrate the superiority in terms of accuracy of the oPLSA over well known PLSA updating methods, such as the PLSA folding-in (PLSA fold.), the PLSA rerun from the breakpoint, the quasi-Bayes PLSA, and the Incremental PLSA. A comparison with respect to the CPU run time reveals that the oPLSA is the second fastest method after the PLSA fold. However, the better accuracy of the oPLSA than that of the PLSA fold. pays off the longer computation time. The oPLSA and the other PLSA updating methods together with online LDA are tested for document clustering and $F_1$ scores are also reported.**

***Index Terms*— Document clustering, document modeling, information retrieval, out-of-vocabulary (OOV) words, PLSA updating, probabilistic latent semantic analysis (PLSA), unsupervised learning.**

## I. Introduction

ACCESSING, processing, and retrieving text, audio and video data has been enhanced by various machine learning algorithms, which employ computationally efficient statistical methods to extract and process information from the data. Probabilistic latent semantic analysis (PLSA) [1] is such an unsupervised machine learning algorithm that has evolved from latent semantic analysis (LSA) [2]. It manipulates huge amounts of data within a solid probabilistic framework. Another example is the latent Dirichlet allocation (LDA) [3], which is the Bayesian extension of PLSA. Indeed, the PLSA was shown to be a special variant of the LDA with a uniform Dirichlet prior in a maximum *a posteriori* model [4]. PLSA, LSA, and LDA share more or less the same application domains, including data modeling [5]–[8], data classification [9]–[13], and retrieval of documents [1], [14], images, or videos [15], [16]. The majority of the aforementioned applications deal with large data volumes that cannot be stored in the memory to be processed at once. In addition, the data are accumulated or vary over time. As a result, several efforts have been made toward updating incrementally the LSA [17]–[19], the LDA [20]–[22], or the PLSA [23]–[25] with respect to new terms or documents. Recently, we have proposed the so-called recursive PLSA (RPLSA) for updating a trained PLSA model, when new documents are added in the document collection [26]. In this paper, we revisit the RPLSA by employing a fixed-size moving window to incorporate new documents, and discard old ones, similarly to the incremental PLSA (IPLSA) method [25]. In addition, out-of-vocabulary (OOV) words (i.e., words unseen so far) appearing in the new documents, are handled. The proposed method is coined online PLSA (oPLSA). The updating equations of the PLSA model are derived for both the asymmetric and the symmetric formulations, for every advance of the window to incorporate the newly arrived documents containing potentially OOV words, while discarding the old ones (i.e., documents that fall outside the scope of the moving window). To handle the OOV words, we resort to Good-Turing probability estimates [27] to initialize the corresponding conditional probabilities with nonzero values. Furthermore, in addition to the commonly used random initialization (referred to as Averaged Random), we test the initialization of PLSA parameters with two schemes, namely the Random Acol and the Random C, which have previously been applied to initialize the nonnegative matrix factorization (NMF) [28]. These initializations are tailored to the PLSA model by applying probability smoothing in order to cope with zero probability estimates. The oPLSA together with the initialization schemes is compared with well established PLSA updating methods, namely the PLSA folding-in (PLAS fold.) [29], the quasi-Bayes PLSA (QB PLSA) [24], the IPLSA [25], and the PLSA rerun from the breakpoint, in terms of accuracy and speed. The average absolute error between the probabilities derived by the aforementioned PLSA updating methods and those estimated by the PLSA applied from scratch manifests the superior modeling power of the oPLSA. The average log-likelihood of the updating methods under study as a function of iterations demonstrates the superior behavior of the oPLSA against the other updating methods under study. In addition, with respect to the average CPU run time the oPLSA is found to be less time consuming than all the other updating methods except the PLSA fold., which is the least time consuming method. However, the excessive computational time of the oPLSA is

Manuscript received December 28, 2012; revised December 19, 2013; accepted December 21, 2013.

The authors are with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki GR-54124, Greece (e-mail: nbassiou@aiia.csd.auth.gr; costas@aiia.csd.auth.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2299806

paid off by its higher accuracy than that of the PLSA fold. Finally, the updating methods under study and the online LDA algorithm are tested for document clustering with respect to the $F_1$ score. To sum up, the contributions of this paper against our previous work [26] lie in: 1) the oPLSA now has the ability to handle not only new documents, but also new terms within the new documents that are treated as unseen words thanks to Good-Turing probability estimates for the OOV words; 2) the oPLSA now has the ability to discard old documents that fall outside the domain of the moving window over the document stream. The vocabulary also varies by discarding terms that appear only in the documents thrown away; 3) the oPLSA works on a fixed-size moving window allowing batch document processing; and 4) efficient initialization schemes combined with smoothing are proposed for the parameter initialization.

The outline of this paper is as follows. In Section II, the PLSA is briefly presented. The traditional PLSA updating schemes are summarized in Section III, while the proposed updating algorithms are derived in Section IV. Experimental results are demonstrated in Section V, and the conclusion is drawn in Section VI.

## II. PLSA

PLSA performs a probabilistic mixture decomposition by means of the so-called aspect model [30], which associates an unobserved class variable to co-occurence. For text processing, we are interested in the occurrence of a word/term $w \in W = \{w_1, w_2, \ldots, w_M\}$ in a document $d \in D = \{d_1, d_2, \ldots, d_N\}$, while the unobserved class variable $z \in Z = \{z_1, z_2, \ldots, z_K\}$ usually represents the topic a document was generated from. It holds that $|Z| \ll \min(|D|, |W|)$, where $|\cdot|$ stands for the cardinality of the corresponding set. Furthermore, all the pairs $(d, w)$ are assumed to be independent and identically distributed, and conditionally independent given the respective latent class $z$. The data generation process can be better described by the following scheme [31]: 1) select a document $d$ with probability $P(d)$; 2) pick a latent topic $z$ for the document with probability $P(z|d)$; and 3) generate a term $w$ with probability $P(w|z)$. Accordingly, the joint distribution of a word $w$ in a document $d$ generated by a latent topic $z$ is given by $P(d, w, z) = P(d)P(z|d)P(w|z)$.

### A. Asymmetric Formulation

The joint distribution of $d$ and $w$ is obtained by summing over all possible realizations of $z$

$$P(d, w) = \sum_{z \in Z} P(d, w, z) = P(d) \underbrace{\sum_{z \in Z} P(z|d)P(w|z)}_{P(w|d)}. \quad (1)$$

As can be seen from (1), the document-specific term distribution $P(w|d)$ is obtained by a convex combination of the $|Z|$ aspects/factors $P(w|z)$. To determine $P(d)$, $P(z|d)$, and $P(w|z)$, the log-likelihood function

$$\mathcal{L} = \sum_{d \in D} \sum_{w \in W} n(d, w) \log P(d, w) \quad (2)$$

has to be maximized with respect to all the aforementioned probabilities. In (2), $n(d, w)$ denotes the term-document frequency. That is, the number of times $w$ occurs in $d$. The estimation of $P(d)$ can be carried out independently resulting in $P(d) = n(d) \big/ \sum_{d' \in D} n(d')$. The conditional probabilities $P(z|d)$ and $P(w|z)$ are estimated by means of the EM algorithm [31], [32], which alternates between the Expectation ($E$)-step

$$\hat{P}(z|d, w) = \frac{P(w|z)P(z|d)}{\sum_{z' \in Z} P(w|z')P(z'|d)} \quad (3)$$

and the Maximization ($M$)-step

$$P(w|z) = \frac{\sum_{d \in D} n(d, w)\hat{P}(z|d, w)}{\sum_{d \in D} \sum_{w' \in W} n(d, w')\hat{P}(z|d, w')} \quad (4)$$

$$P(z|d) = \frac{\sum_{w \in W} n(d, w)\hat{P}(z|d, w)}{n(d)}. \quad (5)$$

By alternating (3) with (4) and (5), a convergent procedure is obtained to a local maximum of the log-likelihood.

### B. Symmetric Formulation

An equivalent model can be obtained by applying the Bayes' rule to invert the conditional probability $P(z|d)$ [31], yielding $P(d, w) = \sum_{z \in Z} P(z)P(d|z)P(w|z)$. Let $R = \sum_{d \in D} \sum_{w \in W} n(d, w)$. Following similar lines to the asymmetric model, the following $E$-step and $M$-step result:
$E$-step

$$\hat{P}(z|d, w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z' \in Z} P(z')P(d|z')P(w|z')}. \quad (6)$$

$M$-step

$$P(w|z) = \frac{\sum_{d \in D} n(d, w)\hat{P}(z|d, w)}{\sum_{d \in D} \sum_{w' \in W} n(d, w')\hat{P}(z|d, w')} \quad (7)$$

$$P(d|z) = \frac{\sum_{w \in W} n(d, w)\hat{P}(z|d, w)}{\sum_{d' \in D} \sum_{w \in W} n(d', w)\hat{P}(z|d', w)} \quad (8)$$

$$P(z) = \frac{\sum_{d \in D} \sum_{w \in W} n(d, w)\hat{P}(z|d, w)}{R}. \quad (9)$$

The symmetric PLSA formulation can be rewritten in matrix notation as $\mathbf{P} = \mathbf{U}_K \mathbf{S}_K \mathbf{V}_K^T$, where $\mathbf{U}_K$ is the $|W| \times |Z|$ matrix with $jk$ element $P(w|z)$, $\mathbf{V}_K$ is the $|D| \times |Z|$ matrix with $ik$ element $P(d|z)$, $\mathbf{S}_K$ is the $|Z| \times |Z|$ diagonal matrix having as elements on its main diagonal $P(z)$, $z \in Z$, and $\mathbf{P}$ is the $|W| \times |D|$ matrix with elements the probabilities $P(w, d)$. Such a decomposition looks like the truncated singular value decomposition (SVD) employed within the LSA. Despite the just described resemblance, it should be stressed that the LSA and the PLSA solve different optimization problems. Indeed, the LSA minimizes the Frobenius norm between the original-term document matrix and its best $K$-rank approximation, while the PLSA maximizes the likelihood function of multinomial sampling. In other words, the PLSA minimizes the cross entropy (or Kullback–Leibler divergence) between the model and the empirical distribution.

## III. PLSA Updating

Updating refers to the process of adding new data in the initial data collection. Such a need emerges when documents and/or terms are added or deleted in document clustering or topic-detection. Updating is a necessity due to the huge data volumes continuously changing over time, and the memory and time limitations, preventing one to process the data as a whole at once. As a result, several methods have emerged for updating the LSA or PLSA models. These methods are frequently met in the literature with terms, such as online, incremental, or folding-in.

Let us begin with LSA updating techniques, which have motivated the development of PLSA updating techniques. For updating the LSA model, which is estimated by a truncated SVD, the most representative methods include recomputing the SVD, SVD folding-in [17], SVD-updating [2], [18], and SVD folding-up [19]. SVD-updating makes updating computationally efficient by leveraging suitable Cholesky factorizations, while SVD folding-up is a hybrid method alternating repeatedly between the SVD folding-in and the SVD-updating to prevent the loss of orthogonality.

Similarly, several methods for updating the PLSA model have been proposed. The PLSA fold. is the simplest method for updating the PLSA model, when new documents are added in the initial document collection [23], [33], [34]. It is based on an incremental variant of the EM algorithm discussed in [27] that recalculates only the probabilities of the topics given the new documents $P(z|d_{\text{new}})$ in the $M$-step leaving the probabilities of the words given the topics $P(w|z)$ unchanged. Usually, a very small number of iterations is needed for the EM to converge. Another incremental approach is proposed in [35] that is based on the generalized expectation maximization [27]. An incremental version of EM that updates the PLSA model parameters using only a subset of training data at each iteration is also proposed in [36], while in [37] the PLSA model parameters are updated by means of an online EM algorithm that works on the weighted mean values of the conditional probability $P(z|d, w)$.

The so-called IPLSA method is developed, whenever a batch of new incoming documents is added and a batch of old documents is discarded [25]. The PLSA fold. is used to fold-in new terms and documents in four steps. In the first step, the old documents and terms are discarded and the probabilities of the remaining terms and documents are estimated by renormalization. In the second step, the new documents are folded-in by means of the PLSA fold. for the asymmetric formulation. In the third step, the new terms are folded-in by exploiting the symmetric PLSA formulation. Finally, all the PLSA parameters are revised by applying the asymmetric PLSA algorithm in the fourth step.

Two new adaptation paradigms for PLSA are derived in [24], namely the QB PLSA for incremental learning and the MAP PLSA for corrective training. They are based on a Bayesian PLSA framework that uses a Dirichlet density kernel as prior. The QB PLSA estimates the model parameters by maximizing an approximate posterior distribution, or equivalently, a product of the likelihood function of currently observed documents and the prior density given the hyperparameters from previous documents. The MAP PLSA maximizes the posterior probability integrated by a prior density and a likelihood function. A maximum *a posteriori* estimator is also employed in Bayesian folding-in, which uses a Dirichlet density kernel as prior [38].

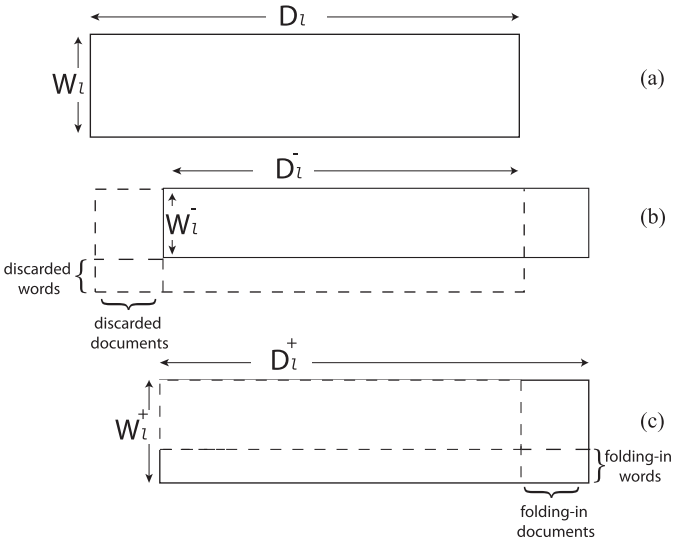



Fig. 1. Schematic representation of (a) the initial data volume, (b) the deletion of old documents and the words that appear exclusively in them, and (c) the insertion of new documents and their associated words in the word-document matrix as the window advances.

## IV. oPLSA

The oPLSA is a window-based method for updating the PLSA parameters for both the asymmetric and the symmetric formulations. Let us suppose that the PLSA model parameters have been extracted for the data volume observed in Fig. 1(a). Let $D_l$ and $W_l$ be the set of documents and the vocabulary associated with the PLSA model parameters. In each advance of the window, the documents that fall outside the window are discarded, as shown in Fig. 1(b). After removing the discarded documents, the vocabulary may change due to the deletion of words that appear exclusively in the discarded documents. Let $D_l^-$ and $W_l^-$ be the sets of already seen documents and words, respectively. Another consequence is the insertion of new documents into $D_l^-$ that are referred to as folding-in documents as well as the possible inclusion of new terms in the vocabulary $W_l^-$ that are referred to as folding-in words [Fig. 1(c)]. Clearly, the folding-in words can be treated as OOV words for the vocabulary $W_l^-$. Let $D_l^+$ and $W_l^+$ be the sets of documents and words that fall inside the new window position. In the RPLSA, we addressed the problem of adding new documents containing words from a fixed vocabulary without using a window. As a result, the RPLSA does not support model adaptation when documents and/or words are removed from the initial data volume nor does update the PLSA model parameters when new words are included in the vocabulary. The oPLSA method is formulated by means of a fundamental operation, namely the addition/deletion of a pivotal document to the document collection. This fundamental operation allows us to treat more complex operations, such as the addition/deletion of an actual document. For consistency

reasons with the addition of a single document, the removal of a single document is also formulated.

In the analysis following, the original PLSA algorithm is executed once for the initial word and document collections. Then, it is assumed that a window of size $S$ advances by a single document. The oPLSA algorithm performs three steps to update the existing PLSA model parameters between the $l$th and $(l+1)$th position of the window. The aim is to update the PLSA model parameters having been estimated by the EM algorithm at the $l$th position of the window to obtain the PLSA model parameters at the $(l+1)$th position of the window. In other words, to derive the equations for initializing the EM algorithm at the $(l+1)$th position of the window. For simplicity, we assume that the PLSA model parameters at the $l$th position of the window have been obtained after EM convergence.

### A. Asymmetric Formulation

*1) Discard Old Document and Their Exclusive Words:* For simplicity let us consider the case of a single document and a single exclusive word to be discarded at the $(l+1)$th window advance. When $d_{\text{out}}$ is discarded from the existing document collection $D_l$, a new document collection $D_l^- = D_l - \{d_{\text{out}}\}$ results. If the discarded document $d_{\text{out}}$ contains only a single exclusive word $w_{\text{out}}$, (i.e., a word that does not appear in any document in $D_l^-$), this word is also discarded from $W_l$ yielding the vocabulary of already seen word $W_l^- = W_l - \{w_{\text{out}}\}$. The corresponding PLSA model probabilities for $d_{\text{out}}$, $P_l(z|d_{\text{out}})$, and $w_{\text{out}}$, $P_l(w_{\text{out}|z})$, are eliminated and the PLSA model parameters for the remaining documents $d \in D_l^-$ and words $w \in W_l^-$ are renormalized as follows:

$$P^-(z|d)_l = \frac{P(z|d)_l}{\sum_{z' \in Z} P(z'|d)_l}, z \in Z, d \in D_l^- \tag{10}$$

$$P^-(w|z)_l = \frac{P(w|z)_l}{\sum_{w' \in W_l^-} P(w'|z)_l}, z \in Z, w \in W_l^-. \tag{11}$$

Clearly, (10) and (11) are still valid when more than one documents and words are to be discarded.

*2) Add a New Word and Document:*

a) Let a new document $d_{\text{in}}$ be inserted in the already seen document collection $D_l^-$, yielding $D_l^+ = D_l^- + \{d_{\text{in}}\}$. Suppose that the document is pivotal. That is, it contains only a single word $w_{\text{in}}$ that appears $\alpha$ times. This word, $w_{\text{in}}$, can be either a word from the vocabulary of already seen words $W_l^-$ or a new word denoted by $w_{\text{OOV}}$, being an OOV word at this point. In the latter case, the new word is inserted in $W_l^-$, expanding it into $W_l^+ = W_l^- + \{w_{\text{OOV}}\}$. Therefore, the entries of the augmented word-document matrix satisfy

$$n(d,w)_{l+1} = \begin{cases} \alpha, & \text{if } w = w_{\text{in}} \text{ and } d = d_{\text{in}} \\ n(d,w)_l, & \text{if } w \in W_l^- \text{and } d \in D_l^- \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $n(d,w)_l$ is the document-word matrix at the $l$th position of the window.

b) The addition of the new document requires the initialization of the latent-variable probability $P^+(z|d_{\text{in}})_l$. Clearly, for $d \in D_l^-$, $P^+(z|d)_l = P^-(z|d)_l$. Accordingly, one has to initialize $P^+(z|d_{\text{in}})_l$. Since the new document has a single word $w_{\text{in}}$, we assume that the assignment of the new document to the latent topics is driven by the single word it contains. Thus, $P^+(z|d_{\text{in}})_l = P^+(w_{\text{in}}|z)_l$, $\forall z \in Z$. The conditional probabilities at the right-hand side are estimated next.

c) For already seen words $w \in W_l^-$, $P^+(w|z)_l = P^-(w|z)_l$. For an OOV word $w_{\text{OOV}}$, the corresponding conditional probability $P^+(w_{\text{OOV}}|z)_l$ is initialized by the Good-Turing estimate of the probability of unseen words [27]. Let $n(w) = \sum_{d \in D_l^-} n(d,w)_l$, $w \in W_l^-$ be the number of appearances of the word $w \in W_l^-$ that has already been seen within the documents prior to the insertion of the pivotal document. Let $R_{l+1}$ denote the total number of words in all documents, including the pivotal one, that is

$$R_{l+1} = \sum_{d \in D_l^+} \sum_{w \in W_l^+} n(d,w)_{l+1} = R_l + a. \tag{13}$$

The Good-Turing estimate for an OOV word is given by [27]

$$p_{\text{GT}_l}(w_{\text{OOV}}) = \frac{n_1}{R_{l+1}} \tag{14}$$

where $n_1 = \sum_{w \in W_l^-: n(w)=1} 1$ is the total number of already seen words occurring exactly once (hapax legomena). The probability given by (14) is uniformly distributed among the topics

$$P^+(w_{\text{OOV}}|z)_l = \frac{1}{|Z|} p_{\text{GT}_l}(w_{\text{OOV}}). \tag{15}$$

The conditional probabilities of already seen words given the topics are renormalized using

$$P^+(w|z)_l = \left(1 - P^+(w_{\text{OOV}}|z)_l\right) \frac{P^-(w|z)_l}{\sum_{w \in W_l^-} P^-(w|z)_l} \tag{16}$$

where $w \in W_l^-$, so that $\sum_{w \in W_l^+} P^+(w|z)_l = 1$.

*3) Fold in the New Word for Pivotal Documents:* The PLSA model probabilities at the window position $l+1$ are estimated by updating the PLSA model probabilities at the window position $l$. To achieve this, the computations between two successive EM iterations, as described in the Appendix, are taken into consideration. Thus, the conditional probability $P_1^+(w|z)_{l+1}$ of the word $w \in W_l^+$ given the latent topic $z \in Z$

is given by

$$P_1^+(w|z)_{l+1} = P_1(w|z)_{l+1} + \frac{n(d_{\text{in}},w)_{l+1}P^+(w|z)_l P^+(z|d_{\text{in}})_l}{\sum_{z'\in Z}P^+(w|z')_l P^+(z'|d_{\text{in}})_l}$$
$$= \begin{cases} P_1(w|z)_{l+1}, & \text{if } w \neq w_{\text{in}} \\ P_1(w|z)_{l+1} + \frac{\alpha P^+(w|z)_l P^+(z|d_{\text{in}})_l}{\sum_{z'\in Z}P^+(w|z')_l P^+(z'|d_{\text{in}})_l}, & \\ \qquad \text{if } w = w_{\text{in}} \quad \text{and} \quad w_{\text{in}} \in W_l^- & \\ \frac{\alpha P^+(w|z_k)_l P^+(z|d_{\text{in}})_l}{\sum_{z'\in Z}P^+(w|z')_l P^+(z'|d_{\text{in}})_l}, & \\ \qquad \text{if } w = w_{\text{in}} \quad \text{and} \quad w_{\text{in}} = w_{\text{OOV}} & \end{cases} \tag{17}$$

where $P_1(w|z)_{l+1}$ is defined in (38) in the Appendix.

Having derived (17), $P^+(w|z)_{l+1}$ is obtained by normalizing $P_1^+(w|z)_{l+1}$ as in (39)

$$P^+(w|z)_{l+1} = \frac{P_1^+(w|z)_{l+1}}{\sum_{w'\in W_l^+}P_1^+(w'|z)_{l+1}} = \begin{cases} \frac{P_1(w|z)_{l+1}}{A_{l+1}(w_{\text{in}})}, & \text{if } w \neq w_{\text{in}} \\ \frac{P_1^+(w|z)_{l+1}}{A_{l+1}(w_{\text{in}})}, & \text{if } w = w_{\text{in}}. \end{cases} \tag{18}$$

The denominator in (18) is given by

$$A_{l+1}(w_{\text{in}}) = \begin{cases} \sum_{w\in W_l^-}P_1(w|z)_{l+1} + P_1^+(w_{\text{in}}|z)_{l+1} - P_1(w_{\text{in}}|z)_{l+1}, & \\ \qquad \text{if } \mathrm{w_{in}} \in \mathrm{W_l^-} & \\ \sum_{w\in W_l^-}P_1(w|z)_{l+1} + P_1^+(w_{\text{in}}|z)_{l+1}, & \\ \qquad \text{if } \mathrm{w_{in}} = \mathrm{w_{OOV}}. & \end{cases} \tag{19}$$

Similarly, the conditional probability $P_2(z|d)_{l+1}$ of the latent topic $z \in Z$ given the document $d \in D_l^+$ is given by

$$P_2^+(z|d)_{l+1} = P^+(z|d)_l \sum_{w\in W_l^+}\frac{n(d,w)_{l+1}P^+(w|z)_l}{\sum_{z'\in Z}P^+(w|z')_l P^+(z'|d)_l} = \begin{cases} P_2(z|d)_{l+1}, & \text{if } d \in D_l^- \\ \frac{\alpha P^+(w_{\text{in}}|z)_l P^+(z|d)_l}{\sum_{z'\in Z}P^+(w_{\text{in}}|z')_l P^+(z'|d)_l}, & \text{if } d = d_{\text{in}} \end{cases} \tag{20}$$

where $P_2(z|d)_{l+1}$ is defined in (41) in the Appendix. When $d = d_{\text{in}}$, with the help of (17), (20) takes the form

$$P_2^+(z|d_{\text{in}})_{l+1} = \begin{cases} P_1^+(w_{\text{in}}|z)_{l+1} - P_1(w_{\text{in}}|z)_{l+1}, & \text{if } w_{\text{in}} \in W_l^- \\ P_1^+(w_{\text{in}}|z)_{l+1}, & \text{if } w_{\text{in}} = w_{\text{OOV}}. \end{cases} \tag{21}$$

Finally, $P^+(z|d)_{l+1}$ is updated taking into consideration (20) as follows:

$$P^+(z|d)_{l+1} = \frac{P_2^+(z|d)_{l+1}}{n(d)} = \begin{cases} \frac{P_2(z|d)_{l+1}}{n(d)} = P(z|d)_{l+1}, & \text{if } d \in D_l^- \\ \frac{P_2^+(z|d)_{l+1}}{\alpha}, & \text{if } d = d_{\text{in}}. \end{cases} \tag{22}$$

### B. Symmetric Formulation

*1) Discard Old Documents and Terms:* Following similar lines to Section IV-A, at the $(l+1)$th window advance, a document $d_{\text{out}}$ and a word $w_{\text{out}}$ are discarded from $D_l$ and $W_l$ yielding $D_l^- = D_l - \{d_{\text{out}}\}$ and $W_l^- = W_l - \{w_{\text{out}}\}$, respectively. The corresponding PLSA model probabilities for $w_{\text{out}}$, $P_l(w_{\text{out}|z})$, and $d_{\text{out}}$, $P_l(d_{\text{out}}|z)$, are eliminated. The PLSA model parameters for the remaining words $w \in W_l^-$ are renormalized as in (11) and for the documents $d \in D_l^-$ by:

$$P^-(d|z)_l = \frac{P(d|z)_l}{\sum_{z'\in Z}P(d|z')_l}, z \in Z, d \in D_l^-. \tag{23}$$

*2) Add a New Word and Document:*

a) The new pivotal document $d_{\text{in}}$, containing a single word $w_{\text{in}}$, is inserted in the existing document collection $D_l^-$, yielding $D_l^+ = D_l^- + \{d_{\text{in}}\}$. When $w_{\text{in}} \in W_l^-$, the word $w_{\text{in}}$ is an OOV word that is appended to the vocabulary of already seen words, yielding $W_l^+ = W_l^- + \{w_{\text{OOV}}\}$. The augmented document-word matrix is given by (12).

b) After the addition of $d_{\text{in}}$, $P^+(d, d \in D_l^-|z)_l = P^-(d, d \in D_l^-|z)_l$ is initialized similarly to the probability $P^+(z|d_{\text{in}})_l$ of the asymmetric formulation. That is, $P^+(d_{\text{in}}|z)_l = P^+(w_{\text{in}}|z)_l$, $\forall z \in Z$. For the rest, $P^+(d, d \in D_l^-|z)_l = P^-(d, d \in D_l^-|z)_l$.

c) When the appended document contains an OOV word, the probability of $P^+(w_{\text{OOV}}|z)_l$ is initialized as described in Section IV-A2c.

*3) Fold in the New Word for Pivotal Documents:* The PLSA model probabilities for the window position $l+1$ are estimated by updating the PLSA model probabilities at the window position $l$. To achieve this, the computations between two successive EM iterations, given in the Appendix, are taken into consideration. Thus, the conditional probability $P_1^+(w|z)_{l+1}$ of the word $w \in W_l^+$ given the latent topic $z \in Z$ is given by

$$P_1^+(w|z)_{l+1} = \begin{cases} P_1(w|z)_{l+1}, & \text{if } w \neq w_{\text{in}} \\ P_1(w|z)_{l+1} + \frac{\alpha P^+(z)P^+(w|z)_l P^+(d_{\text{in}}|z)_l}{\sum_{z'\in Z}P^+(z')P^+(w|z')_l P^+(d_{\text{in}}|z')_l}, & \\ \qquad \text{if } w = w_{\text{in}} \text{ and } w_{\text{in}} \in W_l^- & \\ \frac{\alpha P^+(w|z)_l P^+(z)P^+(d_{\text{in}}|z)_l}{\sum_{z'\in Z}P^+(z')P^+(w|z')_l P^+(d_{\text{in}}|z')_l}, & \\ \qquad \text{if } w = w_{\text{in}} \quad \text{and} \quad w_{\text{in}} = w_{\text{OOV}} & \end{cases} \tag{24}$$

$P^+(w|z)_{l+1}$ are obtained by renormalizing as described in (18) and (19).

The conditional probability $P_2(d|z)_{l+1}$ of the document $d \in D_l^+$ given the latent topic $z \in Z$ is given by

$$P_2^+(d|z)_{l+1} = P^+(d|z)_l \sum_{w\in W_l^+}\frac{n(d,w)_{l+1}P^+(z)P^+(w|z)_l}{\sum_{z'\in Z}P^+(z')P^+(w|z')_l P^+(d|z')_l} = \begin{cases} P_2(d|z)_{l+1}, & \text{if } \quad \in D_l^- \\ \frac{\alpha P^+(z)P^+(w_{\text{in}}|z)_l P^+(d|z)_l}{\sum_{z'\in Z}P^+(z')P^+(w_{\text{in}}|z')_l P^+(d|z')_l}, & \text{if } \quad d = d_{\text{in}} \end{cases} \tag{25}$$

where $P_2(d|z)_{l+1}$ is defined in (43) in the Appendix. One may observe that (25) has the form of (20) written in terms of the conditional probabilities of documents given the topics. Accordingly for $d = d_{\text{in}}$, $P_2^+(d_{N+1}|z_k)_{l+1}$ is updated as in (21), that is

$$P_2^+(d_{\text{in}}|z)_{l+1} = \begin{cases} P_1^+(w_{\text{in}}|z)_{l+1} - P_1(w_{\text{in}}|z)_{l+1}, & \text{if} \quad w_{\text{in}} \in W_l^- \\ P_1^+(w_{\text{in}}|z)_{l+1}, & \text{if} \quad w_{\text{in}} = w_{\text{OOV}}. \end{cases} \tag{26}$$

Using (25) and (26), $P^+(d|z)_{l+1}$ is obtained by renormalizing $P_2^+(d|z)$ as in (45)

$$P(d|z)_{l+1}^+ = \frac{P_2^+(d|z)_{l+1}}{\sum_{d' \in D_l^+} P_2^+(d'|z)_{l+1}} = \begin{cases} \dfrac{P_2(d|z)_{l+1}}{B_{l+1}^+(w_{\text{in}})}, & \text{if} \quad d \in D_l^- \\ \dfrac{P_2^+(d|z)_{l+1}}{B_{l+1}^+(w_{\text{in}})}, & \text{if} \quad d = d_{\text{in}} \end{cases} \tag{27}$$

where the denominator in (27) is given by

$$B_{l+1}^+(w_{\text{in}}) = \sum_{d \in D_l^-} P_2^+(d|z)_{l+1} + P_2^+(d_{\text{in}}|z)_{l+1} = \begin{cases} \sum_{d \in D_l^-} P_2(d|z)_{l+1} + P_1^+(w_{\text{in}}|z)_{l+1} - \\ \qquad P_1(w_{\text{in}}|z)_{l+1}, \quad \text{if} \quad w_{\text{in}} \in W_l^- \\ \sum_{d \in D_l^-} P_2(d|z)_{l+1} + P_1^+(w_{\text{in}}|z)_{l+1}, \\ \qquad \text{if } w_{\text{in}} = w_{\text{OOV}}. \end{cases} \tag{28}$$

Finally, we proceed to updating $P(z)_{l+1}$ as follows:

$$P^+(z)_{l+1} = \frac{1}{R_{l+1}} \Bigg[ R_l P(z)_{l+1}^+ + \frac{n(d_{\text{in}}, w_{\text{in}})_{l+1} P^+(z)_l P^+(d_{\text{in}}|z)_l P^+(w_{\text{in}}|z)_l}{\sum_{z' \in Z} P^+(z')_l P^+(d_{\text{in}}|z')_l P^+(w_{\text{in}}|z')_l} \Bigg] = \begin{cases} \dfrac{1}{R_{l+1}} \big[ R_l P^+(z)_{l+1} + \\ P_1^+(w_{\text{in}}|z)_{l+1} - P_1(w_{\text{in}}|z) \big], \quad \text{if } w_{\text{in}} \in W_l^- \\ \dfrac{1}{R_{l+1}} \big[ R_l P^+(z)_{l+1} + P_1^+(w_{\text{in}}|z)_{l+1} \big], \\ \quad \text{if } w_{\text{in}} = w_{\text{OOV}} \end{cases} \tag{29}$$

where $R_{l+1}$ is given by (13).

### C. From Pivotal Documents to Actual Ones

When the new document to be added has more than one words, simply one has to repeat the updating procedure as many times as the number of the words in the document. Nothing prohibits repeating the updating equations of the PLSA parameters document-wise to assimilate the insertion of more than one documents during the transition phase from the window at the $l$th position to the $(l+1)$th position. After having absorbed all the documents in the transition phase, one may switch to the standard PLSA. The PLSA algorithm is then applied in order to refine the conditional probabilities estimated by the updating equations during the aforementioned transition phase.

### D. Initializations for PLSA

As it is widely known, the EM algorithm is an iterative method, which is guaranteed to converge to a fixed point that may be a local extremum or a saddle point of the likelihood function. This fact in combination with the initialization of the model parameters explains the sensitivity of the PLSA algorithm. Obviously, a good initialization of the PLSA model parameters is of great importance, since it may improve the convergence speed and the accuracy of the updates.

In most cases, a random initialization is used, which does not provide a satisfactory first estimate of the model parameters. To tackle this problem, frequently results from four/five random initializations are averaged. Such recipe was empirically found to improve the performance [1], [14]. In [28], several initializations are examined for the term-topic factor of the NMF algorithm, which is related to PLSA [39], [40]. Here, the least computationally expensive ones, namely the Random Acol and the Random C, are selected to initialize $P(w|z)$ in the PLSA algorithm.

In Random Acol, the probability of every word given a topic, $P(w|z)$, is initialized by averaging the joint probability $P(w, d)$ over $p$ randomly chosen documents. Random C initialization is similar to Random Acol. It initializes the probability of a word given a topic, $P(w|z)$, by averaging the joint probability $P(w, d)$ over $p$ documents randomly chosen among the $N$ documents whose associated vectors with elements $P(w, d)$ have the $N$ largest $\ell_2$ norms. The probability of a document given a topic $P(z|d)$ and that of a topic given a document $P(d|z)$ is initialized by averaging the probabilities of the words that are present in the document given the topics $P(w|z)$.

Obviously, the joint probability $P(w, d)$ has to be estimated from the training data set in both initialization schemes. By means of the Bayes rule, it holds that $P(w, d) = P(d)P(w|d)$, where

$$P(d) = \frac{n(d)}{\sum_{d' \in D} n(d')}, \quad n(d) = \sum_{w \in W} n(d, w)$$
$$\text{and} \quad P(w|d) = \frac{n(d, w)}{\sum_{w' \in W} n(w', d)}. \tag{30}$$

Equation (30) usually results in many zero probability values due to the sparseness of the document-word matrix. We choose to smooth the zero values in $P(w|d)$, since they produce zero values in $P(w, d)$, which may lead to zero initial estimates of $P(w|z)$ in both the Random Acol or Random C schemes. To achieve this, a small amount of the probability mass is removed from the seen events and it is redistributed among the unseen events. For this purpose, absolute discounting has been used [41]. Let us adapt the count-counts $n_r$, also used in Good-Turing probability estimates, as follows:

1) $n_0(d) = \sum_{w: n(w,d)=0} 1$, be the number of words that are never seen in the training document $d$;

2) $n_r = \sum_{w,d:n(w,d)=r} 1$, be the number of word-document pairs that appear $r$ times in the training document-word matrix.

Then, absolute discounting yields

$$P(w|d) = \begin{cases} \dfrac{n(w,d) - b}{\sum_{w' \in W} n(w',d)}, & \text{if} \quad n(w,d) > 0 \\ b \dfrac{|W| - n_0(d)}{\sum_{w' \in W} n(w',d)} P(w), & \text{if} \quad n(w,d) = 0 \end{cases} \tag{31}$$

where $P(w) = \sum_{d \in D} n(w,d) \big/ \sum_{w' \in W} \sum_{d \in D} n(w',d)$ and $b = n_1/(n_1 + 2n_2)$.

## V. Experimental Result

Let us refer by PLSA rerun from the breakpoint to the PLSA which employs as initial model parameters those estimated prior to the insertion of new documents. The oPLSA, the PLSA rerun from the breakpoint, and the state of the art updating methods PLSA fold., QB PLSA, and IPLSA, described in Sections III and IV, were implemented and their performance is compared with that of the standard PLSA algorithm, when applied from scratch. It is worth mentioning that the state of the art updating methods can handle OOV words provided that the probabilities of the OOV words given the latent topics are initialized as described in Section IV for the oPLSA.

Both the asymmetric formulation and the symmetric one have been implemented so that one can examine performance variations, if any. In addition, the initialization methods for PLSA model parameters, described in Section IV-D, have been tested to measure how the initialization affects the PLSA and its updating schemes.

The updating methods under study were assessed in terms of the accuracy by estimating the average absolute error between the model probabilities estimated by the PLSA updating methods and those estimated by the original PLSA executed from scratch. In addition, the average log-likelihood was computed to examine whether the PLSA updating methods converged close to the model parameters of the original PLSA algorithm executed from scratch. The average CPU run time per added document was measured as well. The performance of the PLSA, the oPLSA, the QB PLSA, the IPLSA, the PLSA rerun from the breakpoint, and the PLSA fold. was additionally compared with respect to the $F_1$ measure in a document clustering application. Finally, the performance of the oPLSA method was also compared with that of the online LDA, an algorithm based on online stochastic optimization, which analyzes massive document collections that arrive in a stream and are gradually discarded.

### A. Data Sets

Two data sets were built from two standard document corpora: the Reuters document corpus, Volume 1[1] and the TDT5 one.[2] The Reuters corpus contains around 203340 documents tagged with two topics. In the experiments, only the first topic was taken into consideration. The vocabulary of topics has size 79. Thus, the documents of the Reuters data set are distributed into 79 topics. In the TDT5 corpus, only English documents from the Agence France Presse were selected that are tagged as news and assigned with topic relevance annotations, resulting in a data set of 1039 documents distributed into 73 topics. The number of documents in each topic is shown in Fig. 2 for both data sets. The assignment of documents into topics is treated as ground truth. By keeping the number of topics fixed, we leave aside its influence in the PLSA updating procedures. All the documents were preprocessed to have their tags removed and their words stemmed. For stemming, the Porter stemmer was used [42]. A cut-off was also applied to discard terms appearing less than 10 times in each corpus, resulting into a vocabulary of 49451 and 2870 terms for the Reuters and the TDT5 corpus, respectively. The data sets differ by two orders of magnitude (i.e., thousands and hundreds of thousands) with respect to the number of documents, enabling us to conduct a crude assessment how the performance scales in terms of the just mentioned evaluation factors (i.e., accuracy, CPU time, and document clustering quality). The documents were further equally divided into training and test subsets. That is, from each topic half of the documents were randomly selected for training and the remaining half documents were retained to form the test subset. The documents are incrementally added to the training subset from the test subset in each advance of the window, as detained in Section V-B4 next. The procedure was repeated four times, resulting in four folds with different training and test subsets to assess performance reliably.

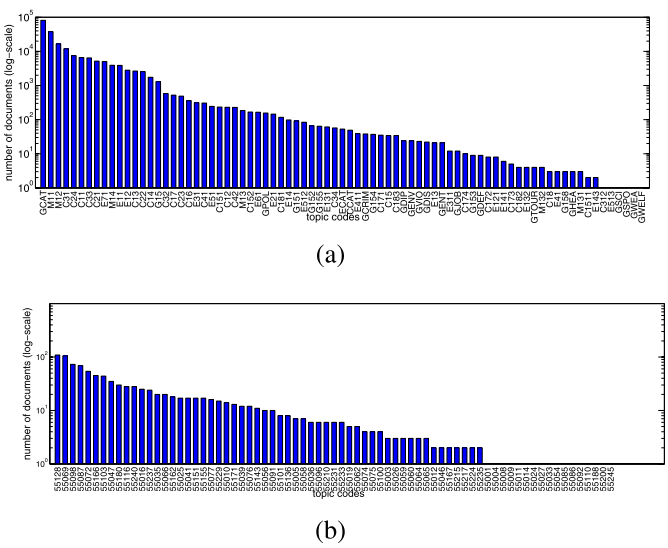



Fig. 2. Distribution of documents in topics for (a) Reuters corpus and (b) TDT5 corpus. For clarity, the $y$-axis is in logarithmic scale.

### B. Experiment Setup

The algorithms require setting the following parameters.

1) Number of latent topics $K$. It corresponds to the 79 and 73 document categories for the Reuters and TDT5 corpus, respectively. To set $K$, the weighted gap (WGap) statistic was used [43], [44] that is more robust than the gap statistic originally proposed in [45]. Let the

[1] http://about.reuters.com/researchandstandards/corpus/index.asp

[2] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T18

total number of clusters vary from $k = 1$ to $k_{\max}$. The documents $d_1, d_2, \ldots, d_N$ are partitioned into $k$ clusters $C_1, C_2, \ldots, C_k$ by the $K$-means clustering method. The weighted within-dispersion measure $\overline{W}_k$, is computed by

$$\overline{W}_k = \sum_{i=1}^{s} \frac{1}{2n_s} \overline{D}_s = \sum_{i=1}^{s} \frac{1}{2n_s(n_s-1)} D_s, \quad k = 1, 2, \ldots, k_{\max} \tag{32}$$

where $n_s = |C_s|$ is the number of documents in cluster $C_s$ and $D_s = \sum_{t=1}^{n_s} \sum_{t'=1}^{n_s} r_{tt'}$ is the sum of pairwise distances between all points in cluster $C_s$. Clearly, $r_{tt'}$ denotes the distance between the documents $d_t$ and $d_{t'}$. Each document $d_t$ is represented by a vector containing the joint probability values $P(w, d_t)$ of every word $w$ in the vocabulary and document $d_t$. A typical choice of distance is the squared Euclidean one. The weighted gap statistic is then defined by

$$\overline{\text{Gap}}(k) = \mathbb{E}[\log(\overline{W}_k)] - \log(\overline{W}_k), \quad k = 1, 2, \ldots, k_{\max} \tag{33}$$

where $\mathbb{E}[\log(\overline{W}_k)]$ is the expected $\log(\overline{W}_k)$, when a reference uniform distribution is assumed. The WGap method identifies the optimal number $K^*$ of clusters according to [44]

$$K^* = \arg\max_k = 2, \ldots, k_{\max} - 1 \left( \underbrace{2\overline{\text{Gap}}(k) - \overline{\text{Gap}}(k-1) - \overline{\text{Gap}}(k+1)}_{\text{DD}\overline{\text{Gap}}(k)} \right). \tag{34}$$

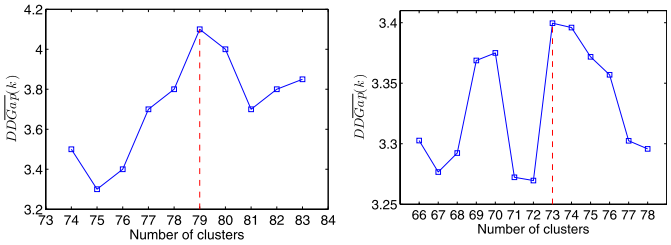
The plot of $\text{DD}\overline{\text{Gap}}(k)$ versus the number of clusters $k$ shown in Fig. 3 supports the choices $K^* = 79$ and $K^* = 73$, for the two corpora, respectively.

2) The criterion value $\epsilon$ used to check the convergence of the EM algorithm. The convergence criterion, that was used, compares the relative log-likelihood change between two successive $EM$-steps, $l-1$ and $l$, to $\epsilon$, that is

$$\left| \frac{[\mathcal{L}]^l - [\mathcal{L}]^{l-1}}{[\mathcal{L}]^{l-1}} \right| \leq \epsilon. \tag{35}$$

Experiments were run for different values of $\epsilon$. A typical value is $10^{-5}$.

3) The initialization method for the conditional probability of a word given a topic $P(w_j|z_k)$ according to Section IV-D. Three different choices were considered:
   a) averaged Random Initialization using the rand function that returns uniformly distributed numbers in the interval $(0, 1)$. Averaged results for five random initializations were used;
   b) random Acol with $p = 20$ and $p = 10$ for the Reuters and the TDT5 corpora, respectively;
   c) random C with $p = 20$ and $p = 10$ for the Reuters and the TDT5 corpora, respectively.




(a) (b)

Fig. 3. $\text{DD}\overline{\text{Gap}}(k)$ versus the number of clusters $k$, for (a) Reuters corpus and (b) TDT5 corpus.

4) The window size and the moving step of the window. The window size was set equal to the number of the training documents for each data set. That is, the window is equal to 520 documents for the TDT corpus and 101670 documents for the Reuters corpus. The moving step of the window was set equal to 2% of the window size for the TDT corpus, resulting in a moving step of 10 documents. For the Reuters corpus, the moving step of the window was set equal to 10%, resulting in a moving step of 10167 documents.

The experimental procedure consists of the following steps. First, the PLSA model parameters were initialized according to the methods described in Section IV-D and the PLSA algorithm was executed for each training data set. Next, in each advance of the window, old documents (as many as the moving step of the window), and their exclusive terms were discarded from the training subset and new documents (as many as the discarded documents) and their exclusive words were appended to the training subset. Then, the original PLSA algorithm was executed from scratch for the modified set, while the PLSA fold., the PLSA rerun from the breakpoint, the QB PLSA, the IPLSA, and the oPLSA update the model parameters estimated prior to the deletion and addition of the documents. Finally, the original PLSA was applied having been initialized by the model parameters that were updated by the aforementioned online PLSA algorithms until convergence.

### C. Evaluation

*1) Accuracy of Model Parameters:* For both the asymmetric and the symmetric formulations, the oPLSA, the QB PLSA, the IPLSA, the PLSA fold., and the PLSA rerun from the breakpoint were compared with the original PLSA algorithm computed from scratch, when unseen documents from the test subset are added and old documents are removed as the window advances. This was done, by averaging over the $K$ latent variables the absolute difference between the probabilities $P(w|z)$ and $P(z|d)$ derived by the original PLSA and the same probabilities estimated by the oPLSA, the QB PLSA, the IPLSA, the PLSA fold., and the PLSA rerun from the breakpoint, after the window advance. The Bayes' chain rule was used to estimate the probabilities $P(z|d)$ for the symmetric formulation. The results obtained were further averaged across all window advances. The above procedure was repeated for the three initialization methods. The mean

TABLE I

MEAN AND STANDARD DEVIATION (STD) ACROSS THE FOUR DIFFERENT FOLDS ON THE REUTERS CORPUS OF THE AVERAGE ABSOLUTE ERROR BETWEEN THE PROBABILITY $P(z/d)$ ESTIMATED BY THE OPLSA, THE QB PLSA, THE IPLSA, THE PLSA FROM BREAKPOINT (PLSA BRK.), OR THE PLSA FOLD AND THAT ESTIMATED BY THE ORIGINAL PLSA EXECUTED FROM SCRATCH FOR THE THREE INITIALIZATION METHODS (RANDOM C, RANDOM ACOL, AND AVERAGED RANDOM): (A) ASYMMETRIC. (B) SYMMETRIC FORMULATION

| *Method* | *Random C* | | *Random Acol* | | *Averaged Random* | |
|---|---|---|---|---|---|---|
| | *mean* | *std* | *mean* | *std* | *mean* | *std* |
| *oPLSA* | **0.2126** | 0.0014 | **0.2311** | 0.0025 | **0.2714** | 0.0050 |
| *QB PLSA* | 0.2425 | 0.0010 | 0.2627 | 0.0033 | 0.2920 | 0.0072 |
| *IPLSA* | 0.2375 | 0.0019 | 0.2560 | 0.0028 | 0.2909 | 0.0054 |
| *PLSA brk.* | 0.3024 | 0.0073 | 0.3159 | 0.0040 | 0.3436 | 0.0055 |
| *PLSA fold.* | 0.3637 | 0.0059 | 0.3795 | 0.0034 | 0.4014 | 0.0060 |

$(A)$

| *Method* | *Random C* | | *Random Acol* | | *Averaged Random* | |
|---|---|---|---|---|---|---|
| | *mean* | *std* | *mean* | *std* | *mean* | *std* |
| *oPLSA* | **0.2400** | 0.0034 | **0.2575** | 0.0025 | **0.2977** | 0.0061 |
| *QB PLSA* | 0.2871 | 0.0030 | 0.3013 | 0.0028 | 0.3250 | 0.0062 |
| *IPLSA* | 0.2779 | 0.0021 | 0.2959 | 0.0039 | 0.3161 | 0.0054 |
| *PLSA brk.* | 0.3417 | 0.0024 | 0.3562 | 0.0039 | 0.3809 | 0.0072 |
| *PLSA fold.* | 0.3855 | 0.0035 | 0.4008 | 0.0039 | 0.4246 | 0.0051 |

$(B)$

TABLE II

MEAN AND STANDARD DEVIATION (STD) ACROSS THE FOUR DIFFERENT FOLDS ON THE TDT5 CORPUS OF THE AVERAGE ABSOLUTE ERROR BETWEEN THE PROBABILITY $P(z/d)$ ESTIMATED BY THE OPLSA, THE QB PLSA, THE IPLSA, THE PLSA FROM BREAKPOINT (PLSA BRK.), OR THE PLSA FOLD AND THAT ESTIMATED BY THE ORIGINAL PLSA EXECUTED FROM SCRATCH FOR THE THREE INITIALIZATION METHODS (RANDOM C, RANDOM ACOL, AND AVERAGED RANDOM): (A) ASYMMETRIC AND (B) SYMMETRIC FORMULATION

| *Method* | *Random C* | | *Random Acol* | | *Averaged Random* | |
|---|---|---|---|---|---|---|
| | *mean* | *std* | *mean* | *std* | *mean* | *std* |
| *oPLSA* | **0.1200** | 0.0035 | **0.1429** | 0.0017 | **0.1749** | 0.0064 |
| *QB PLSA* | 0.1660 | 0.0018 | 0.1886 | 0.0012 | 0.2171 | 0.0073 |
| *IPLSA* | 0.1500 | 0.0003 | 0.1786 | 0.0016 | 0.2046 | 0.0052 |
| *PLSA brk.* | 0.2023 | 0.0037 | 0.2243 | 0.0020 | 0.2687 | 0.0059 |
| *PLSA fold.* | 0.2410 | 0.0042 | 0.2672 | 0.0019 | 0.3036 | 0.0141 |

$(A)$

| *Method* | *Random C* | | *Random Acol* | | *Averaged Random* | |
|---|---|---|---|---|---|---|
| | *mean* | *std* | *mean* | *std* | *mean* | *std* |
| *oPLSA* | **0.1356** | 0.0043 | **0.1595** | 0.0026 | **0.1920** | 0.0074 |
| *QB PLSA* | 0.1810 | 0.0033 | 0.2036 | 0.0014 | 0.2307 | 0.0085 |
| *IPLSA* | 0.1750 | 0.0005 | 0.1989 | 0.0020 | 0.2216 | 0.0059 |
| *PLSA brk.* | 0.2230 | 0.0042 | 0.2445 | 0.0022 | 0.2848 | 0.0079 |
| *PLSA fold.* | 0.2490 | 0.0044 | 0.2730 | 0.0021 | 0.3097 | 0.0147 |

$(B)$

and the standard deviation across the four different folds of the average absolute error for the conditional probability of the latent topics given the documents, $P(z|d)$, are shown in Tables I and II, for the Reuters and TDT5 corpora, respectively. To keep the message conveyed to the reader clear, the errors for the conditional probability of the words given the latent topics, $P(w|z)$, are not included, because they are extremely low and the differences between the updating methods are negligible in both formulations.

As can be seen from Tables I and II, the proposed oPLSA yields on average model parameters closer to those estimated by the original PLSA applied from scratch to the modified document-word matrix due to deletions and additions of documents and words than the other online methods under study. For the asymmetric formulation, there is a relative decrease ranging between 6% and 41% in the Reuters corpus and between 14% and 50% in the TDT5 corpus in the average absolute error of $P(z|d)$ estimated by the oPLSA and that estimated by the original PLSA executed from scratch compared with the same error committed, when the other updating methods under study replace the oPLSA. For the symmetric formulation, there is a relative decrease in the same average absolute error ranging between 6% and 37% in the Reuters corpus and between 13% and 45% in the TDT5 corpus. The IPLSA and the QB PLSA exhibit the second and third best performance, respectively. It is worth mentioning that in all cases, the standard deviation of the average absolute error across the different data sets is much smaller than the performance gains, supporting the statistical significance of the improvements.

Studying Tables I and II with respect to the initialization method used in the PLSA, the effect of the initialization method to each updating method is deduced. The averaged random initialization yields the worst performance for all the

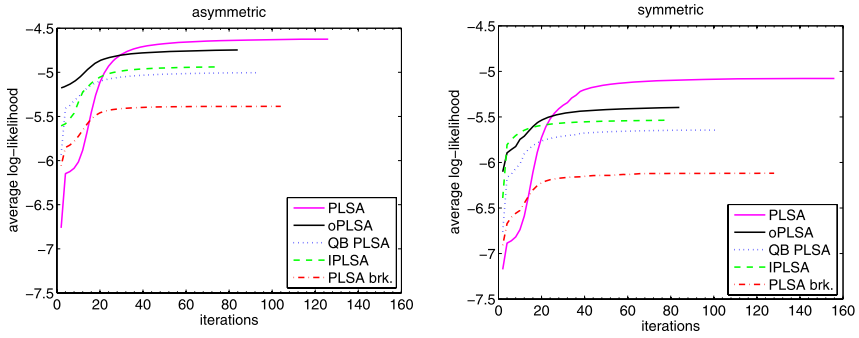



Fig. 4. Average log-likelihood of the asymmetric and the symmetric PLSA executed from scratch, the oPLSA, the QB PLSA, the IPLSA, and the PLSA brk., when Random C initialization was applied in the Reuters corpus.

TABLE III

AVERAGE CPU TIME (IN MS) PER DOCUMENT FOR UPDATING PLSA IN ASYMMETRIC OR SYMMETRIC FORMULATIONS UNDER STUDY AND THE PLSA EXECUTED FROM SCRATCH. TIMINGS ARE REPORTED FOR THE THREE INITIALIZATION METHODS (RANDOM C, RANDOM ACOL, AND AVERAGED RANDOM) AND THE (A) REUTERS AND (B) TDT5 CORPORA

| *Method* | *asymmetric* | | | *symmetric* | | |
|---|---|---|---|---|---|---|
| | *Random C* | *Random Acol* | *Averaged Random* | *Random C* | *Random Acol* | *Averaged Random* |
| *PLSA scratch* | 145.1 | 155.9 | 162.0 | 161.1 | 171.9 | 187.8 |
| *oPLSA* | 68.2 | 75.6 | 84.8 | 83.9 | 89.6 | 97.8 |
| *QB PLSA* | 79.1 | 86.5 | 97.5 | 96.0 | 107.1 | 109.6 |
| *IPLSA* | 86.9 | 92.5 | 102.0 | 109.6 | 114.8 | 124.3 |
| *PLSA brk.* | 96.3 | 104.2 | 111.7 | 113.4 | 129.6 | 135.2 |
| *PLSA fold.* | 28.6 | 31.9 | 36.1 | 33.5 | 36.1 | 39.4 |

(*A*)

| *Method* | *asymmetric* | | | *symmetric* | | |
|---|---|---|---|---|---|---|
| | *Random C* | *Random Acol* | *Averaged Random* | *Random C* | *Random Acol* | *Averaged Random* |
| *PLSA scratch* | 20.5 | 21.7 | 23.7 | 24.1 | 24.9 | 28.3 |
| *oPLSA* | 4.4 | 4.8 | 5.7 | 4.9 | 5.4 | 6.0 |
| *QB PLSA* | 7.2 | 7.6 | 7.9 | 8.1 | 8.9 | 9.3 |
| *IPLSA* | 7.7 | 8.1 | 8.6 | 8.9 | 9.4 | 9.7 |
| *PLSA brk.* | 8.4 | 10.7 | 11.1 | 9.2 | 11.2 | 11.4 |
| *PLSA fold.* | 2.8 | 2.7 | 3.3 | 2.8 | 2.7 | 3.3 |

(*B*)

updating methods in both the asymmetric and the symmetric formulations, thus confirming the need for a more reliable initialization. The Random C initialization yields better results than the Random Acol initialization for either the asymmetric or the symmetric formulations in both corpora. More precisely, when the Random C initialization is chosen in the Reuters corpus, there is a relative decrease in the average absolute error of the asymmetric conditional probability of the latent topics given the documents ranging between 4% and 8% than that obtained with the Random Acol. The corresponding relative decrease in the TDT5 corpus ranges between 9% and 16%. For the average absolute error of the symmetric conditional probability of the latent topics given the documents, the corresponding relative decrease when the Random C initialization is chosen instead of the Random Acol ranges between 4% and 7% in the Reuters corpus and between 9% and 15% in the TDT5 corpus. Thus, the Random C initialization is found more suitable than the Random Acol and the Averaged Random.

In Fig. 4, the average log-likelihood is plotted at each iteration of the EM algorithm for the various algorithms studied in the asymmetric and the symmetric formulation, when Random C initialization was employed. The average log-likelihood for the PLSA fold. is not depicted, since the PLSA fold. converges after a just a few iterations. It is seen that the proposed oPLSA achieves an average log-likelihood close to that of the PLSA and higher than that of all other updating methods under study for both the asymmetric and the symmetric formulation in the Reuters corpus. In addition, the oPLSA achieves a lower average absolute error than the PLSA fold. as well as any other updating method studied here. The oPLSA and the IPLSA need considerably less iterations to converge than the original PLSA executed from scratch, the PLSA rerun from the breakpoint, and the QB PLSA do. Comparing also the average log-likelihood achieved in the asymmetric and the symmetric formulation, it can be seen that the average log-likelihood in the asymmetric formulation is higher than that in the symmetric one, thus explaining the better performance of the former. This can be attributed to the estimation of one additional model parameter [i.e., $P(z)$] in the symmetric formulation.

*2) CPU Run Time:* In Table III, the average CPU run time (in ms) per document across the four folds is summarized when the oPLSA, the QB PLSA, the IPLSA, the PLSA rerun from the breakpoint, the PLSA fold., or the PLSA executed from scratch are employed for the three initialization methods under study in both formulations. The experiments on the TDT corpus were performed on a Intel Core 2 Duo 64-bit processor running at 2.4 GHz with 4 GB RAM, while the experiments on the Reuters corpus were performed on i3 64-bit processor running at 2.3 GHz with 16 GB RAM. MATLAB R2010a for the Mac OS X 10.6 was used. By examining Table III, it can be seen that the original PLSA executed from scratch is the most time consuming method, while the PLSA fold. is the least time-consuming one. The oPLSA is proved to be faster than all the updating methods under study except the PLSA fold. (i.e., the QB PLSA, the IPLSA, and the PLSA rerun from the breakpoint). More precisely, in the Reuters corpus, the oPLSA is relatively faster than the QB PLSA approximately by 9% to 12% for the asymmetric formulation, and by 7% to 15% for the symmetric formulation. Similarly, in the TDT5 corpus, the oPLSA performs faster than the QB PLSA by 28% to 38% for the asymmetric formulation, and by approximately 35% to 39% for the symmetric formulation. The oPLSA is also relatively faster than the IPLSA by 12% to 19% for the asymmetric formulation, and by 16% to 22% for the symmetric one in the Reuters corpus. The oPLSA is relatively less time consuming than the IPLSA by 34% to 42% for the asymmetric formulation, and by 38% to 45% for the symmetric one in TDT5 corpus.

Comparing the CPU run time with respect to the initialization method, it can be deduced that, when the Random C initialization is exploited in all updating methods, updating is faster by 11% to 21% than that when Averaged Random is used in the Reuters corpus. Updating with Random C initialization is faster by 4% to 12% than that when Random Acol is chosen. In the TDT5 data set, all updating methods perform faster by 8% to 24%, when the Random C initialization was employed instead of the Averaged Random initialization. Faster updating by 3% to 21% was achieved when Random C initialization is chosen instead of Random Acol.

Finally, all updating methods perform faster when the asymmetric formulation is used than when the symmetric one is employed in both data sets. In addition, despite the two orders of magnitude difference, PLSA fold. remains at least twice faster than the oPLSA, suggesting that the relative performance of the proposed algorithm does not depend on the size of the data set.

*3) Document Clustering:* The performance of the PLSA updating methods under study was tested in document clustering. After having estimated the model parameters, for each document a feature vector is defined that has the conditional probabilities of the latent topics given the documents as elements. For each class, a prototype vector is created by averaging the feature vectors of all the documents already assigned to this class. The assignment of a document to a class was determined by means of the cosine similarity between the feature vector associated to the document and the prototype vector of each class.

TABLE IV
DOCUMENT CLUSTER-TOPIC CONTINGENCY TABLE

| | *In topic* | *Not in topic* |
|---|---|---|
| *In cluster* | a | b |
| *Not in cluster* | c | d |

Following the performance measurements described in [25], the $F_1$ measure was estimated. Let the $2 \times 2$ contingency table for a document cluster-topic pair be as in Table IV, where $a, b, c$, and $d$ denote the number of document pairs in the four cases. $F_1$ is defined as the harmonic mean of precision and recall, i.e., $F_1 = 2 \cdot \text{Precision} \cdot \text{Recall}/(\text{Precision}+\text{Recall})$. Clearly, the precision and the recall are given by

$$\begin{aligned} \text{Precision} &= \frac{a}{a+b}, \quad \text{if} \quad a+b>0 \\ \text{Recall} \quad &= \frac{a}{a+c}, \quad \text{if} \quad a+c>0. \end{aligned} \tag{36}$$

To obtain a global $F_1$ measurement, microaveraging was used. That is, the contingency tables of the topics were merged by summing the corresponding cells and then the merged table was used to derive the global $F_1$ measurement.

Fig. 5 illustrates the average $F_1$ measure across the four folds for the original PLSA executed from scratch, the oPLSA, the QB PLSA, the IPLSA, the PLSA rerun from the breakpoint, or the PLSA fold., when any of the three initializations is used in both formulations. Examining the bar plots, it becomes clear that the oPLSA produces more descriptive document clusters than the other updating methods in all cases, since the $F_1$ values for the oPLSA are the highest among all the PLSA updating methods. More precisely, the $F_1$ values for the oPLSA are higher than the IPLSA $F_1$ values by 2% to 6% on average in the Reuters corpus. Similarly, the relative $F_1$ improvement between the oPLSA and the IPLSA is 1% to 3% in the TDT5 corpus. On average, the $F_1$ values for the oPLSA are by 2% to 7% and by 5% to 8% higher than the QB PLSA $F_1$ values in the Reuters and the TDT5 corpora, respectively. Similarly, the oPLSA outperforms the PLSA rerun from breakpoint by 8% to 13% on the Reuters corpus and by 10% to 16% on the TDT5 corpus, while the relative improvement of oPLSA over the PLSA fold. ranges between 5% and 8% on the Reuters corpus and by 10% to 13% on the TDT5 corpus. It can also be seen that the $F_1$ values for the oPLSA are closer to those for the original PLSA executed from scratch and the $F_1$ values for any other updating method, revealing thus the ability of the proposed updating method to produce similar clusters to the ones generated by the original PLSA executed from scratch.

Running online LDA[3] with parameters $k = 0.7$ and $\tau_0 = 0.7$ in the same window setting, but without updating the OOV words, resulted in document clusterings with $F_1$ values equal to 0.67 for the Reuters corpus and 0.73 for the TDT corpus. Thus, the asymmetric oPLSA outperforms online LDA in terms of $F_1$ by 5% and 14% on the Reuters and the TDT

[3]http://www.cs.princeton.edu/~blei/downloads/onlineldavb.tar

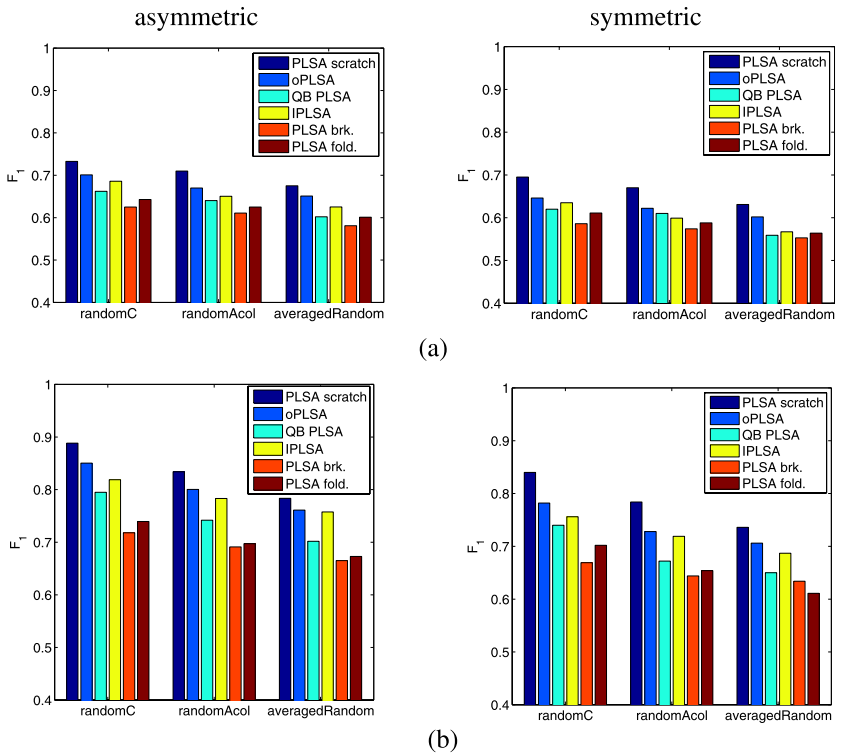



Fig. 5. Average $F_1$ measure for the document clustering derived by the PLSA and the PLSA updating methods when the three initializations are employed for both formulations: (a) Reuters and (b) TDT5 corpora.

corpus, respectively. These preliminary results by no means indicate inferiority of the online LDA, because no special effort was paid to properly tune the online LDA parameters.

## VI. Conclusion

A novel PLSA updating method, the so called oPLSA, has been proposed to handle document insertions that may potentially contain OOV words. The method works on a moving window framework allowing for the deletion of documents that fall outside the scope of the window as well as the removal of the words that exclusively appear in the documents to be discarded. Thus, the oPLSA has a fixed memory with respect to the document-word matrix, requiring less storage space. The oPLSA together with the other updating methods under study (i.e., the PLSA fold., the PLSA rerun from the breakpoint, the QB PLSA, and the IPLSA), have been enhanced with efficient parameter initializations thanks to the Random C and the Random Acol schemes. The promising experimental results demonstrated here allow us to claim that: 1) the oPLSA method outperforms the PLSA fold., the PLSA rerun from the breakpoint, the QB PLSA, and the IPLSA with respect to parameter estimation accuracy as measured by the average absolute error between the probabilities estimated by the updating methods under comparison and the probabilities estimated by applying the original PLSA algorithm to the modified document collection from scratch; 2) the oPLSA achieves a higher average log-likelihood value upon EM convergence compared with that of all the updating methods under study. This observation further supports the higher parameter estimation accuracy of the proposed method; 3) the oPLSA is the second less time consuming method after the PLSA fold. The additional time requirements for the oPLSA, are compensated by the better performance as validated by the other figures of merit (i.e., accuracy, $F_1$ measure in document clustering). Studying the performance of the updating methods in document clustering, the oPLSA algorithm has been shown to be more effective with respect to the $F_1$ measure than the other updating methods under study; and 4) the oPLSA is not found to be inferior to online LDA.

In the future, we plan to enhance the oPLSA algorithm so that it handles documents belonging to topics that are not seen in the initial document collection, making thus the oPLSA algorithm even more adaptive to an online environment. Clearly LDA concepts may help addressing a number of topics not necessarily fixed *a priori*.

## Appendix

## EM Computations Between Two Successive Iterations Without Any Window Advance

### A. Asymmetric Formulation

The computations are presented, taking place by proceeding from iteration $l$ to iteration $l+1$ of the EM algorithm, when documents or words are neither removed nor added). For $w \in W$, $d \in D$ and $z \in Z$, the $E$-step at iteration $l+1$ is given by

$$\hat{P}(z|d,w)_{l+1} = \frac{P(w|z)_l P(z|d)_l}{\sum_{z' \in Z} P(w|z')_l P(z'|d)_l}. \quad (37)$$

Let

$$P_1(w|z)_{l+1} = P(w|z)_l \sum_{d\in D} \frac{n(d,w)P(z|d)_l}{\sum_{z'\in Z} P(w|z')_l P(z'|d)_l}. \quad (38)$$

After the substitution of (37) into (4) and (5), the $M$-step equations are rewritten as

$$P(w|z)_{l+1} = \frac{P_1(w|z)_{l+1}}{\sum_{w'\in W} P_1(w'|z)_{l+1}} \quad (39)$$

$$P(z|d)_{l+1} = \frac{P_2(z|d)_{l+1}}{n(d)} \quad (40)$$

where

$$P_2(z|d)_{l+1} = P(z|d)_l \sum_{w\in W} \frac{n(d,w)P(w|z)_l}{\sum_{z'\in Z} P(w|z')_l P(z'|d)_l}. \quad (41)$$

### B. Symmetric Formulation

Following similar lines to Section VI-A, the computations are presented, taking place when proceeding from iteration $l$ to iteration $l+1$ of the EM algorithm, when no documents or words are neither removed nor are added. The $E$-step for iteration $l+1$ is given by

$$\hat{P}(z|d,w)_{l+1} = \frac{P(z)_l P(d|z)_l P(w|z)_l}{\sum_{z'\in Z} P(z')_l P(d|z')_l P(w|z')_l}. \quad (42)$$

Let

$$P_2(d|z)_{l+1} = P(d|z)_l \left[\sum_{w\in W} \frac{n(d,w)P(w|z)_l}{\sum_{z'\in Z} P(z')_l P(w|z')_l P(d|z')_l}\right] P(z)_l. \quad (43)$$

After the substitution of (42) into (7)–(9), $P(w|z)_{l+1}$ is given by (40) and the remaining $M$-step equations take the form

$$P(d|z)_{l+1} = \frac{P_2(d|z)_{l+1}}{\sum_{d'\in D} P_2(d'|z)_{l+1}} \quad (44)$$

$$P(z)_{l+1} = \frac{1}{R_l} \sum_{d\in D}\sum_{w\in W} \frac{n(d,w)P(z)_l P(d|z)_l P(w|z)_l}{\sum_{z'\in Z} P(z')_l P(d|z')_l P(w|z')_l}. \quad (45)$$

## REFERENCES


[1] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. Uncertainty Artif. Intell.*, 1999, pp. 286–296.

[2] M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using linear algebra for intelligent information retrieval," Dept. Comput. Sci., Univ. Tennessee, Knoxville, TN, USA, Tech. Rep. UT-CS-94-270, 1994.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. 5, pp. 993–1022, 2003.

[4] M. Girolami and A. Kaban, "On an equivalence between PLSI and LDA," in *Proc. 26th ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Toronto, ON, Canada, 2003, pp. 433–434.

[5] J. Blitzer, A. Globerson, and F. Pereira, "Distributed latent variable models of lexical co-occurrences," in *Proc. 10th Int. Workshop Artif. Intell. Statist.*, 2005, pp. 1–8.

[6] S. Wang, D. Schuurmans, F. Peng, and Y. Zhao, "Combining statistical language models via the latent maximum entropy principle," *Mach. Learn.*, vol. 60, nos. 1–3, pp. 229–250, 2005.

[7] P. Smaragdis, B. Raj, and M. Shashanka, "Missing data imputation for time-frequency representations of audio signals," *J. Signal Process. Syst.*, vol. 65, no. 3, pp. 361—370, 2011.

[8] P. Li, J. Cheng, Z. Li, and H. Lu, "Correlated PLSA for image clustering," in *Advances in Multimedia Modeling*, K.-T. Lee, W.-H. Tsai, H.-Y. Liao, T. Chen, J.-W. Hsieh, and C.-C. Tseng, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 307–316.

[9] L. Cai and T. Hofmann, "Text categorization by boosting automatically extracted concepts," in *Proc. 26th ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Toronto, ON, Canada, 2003, pp. 182–189.

[10] T. Hofmann, "Learning the similarity of documents: An information-geometric approach to document retrieval and categorization," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA, USA: MIT Press, 2000, pp. 914–920.

[11] A. Vinokourov and M. Girolami, "A probabilistic framework for the hierarchic organisation and classification of document collections," *J. Intell. Inf. Syst.*, vol. 18, nos. 2–3, pp. 153–172, 2002.

[12] E. Gaussier, C. Goutte, K. Popat, and F. Chen, "A hierarchical model for clustering and categorising documents," in *Proc. 24th BCS-IRSG ECIR Res.*, Glasgow, U.K., Mar. 2002, pp. 292–247.

[13] Z. Zeng, S. Zhang, H. Li, W. Liang, and H. Zheng, "A novel approach to musical genre classification using probabilistic latent semantic analysis model," in *Proc. IEEE ICME*, Jul. 2009, pp. 486–489.

[14] T. Brants, F. Chen, and I. Tsochantaridis, "Topic-based document segmentation with probabilistic latent semantic analysis," in *Proc. 11th Int. Conf. Inf. Knowl. Manag.*, Washington, DC, USA, Nov. 2002, pp. 211–218.

[15] R. Lienhart and M. Slaney, "PLSA on large scale image databases," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 18. Honolulu, HI, USA, Apr. 2007, pp. 1217–1220.

[16] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via PLSA," in *Proc. 9th Eur. Conf. Comput. Vis.*, vol. 4. May 2006, pp. 517–530.

[17] J. R. Bellegarda, "Fast update of latent semantic spaces using a linear transform framework," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1. 2002, pp. 769–772.

[18] G. W. O'Brien, "Information management tools for updating an SVD-encoded indexing scheme," Comput. Sci. Dept., Univ. Tennessee, Knoxville, TN, USA, Tech. Rep. UT-CS-94-258, 1994.

[19] J. E. B. Tougas, "Folding-up: A hybrid method for updating the partial singular value decomposition in latent semantic indexing," M.S. thesis, Dept. Comput. Sci., Dalhousie Univ., Halifax, NS, Canada, Dec. 2005.

[20] M. Hoffman, D. Blei, and F. Bach, "Online learning for latent Dirichlet allocation," in *Proc. Adv. NIPS*, 2010, pp. 856–864.

[21] K. Canini, L. Shi, and T. Griffiths, "Online inference of topics with latent Dirichlet allocation," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, vol. 5. 2009, pp. 65–72.

[22] A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in *Proc. 7th SIAM Int. Conf. Data Mining*, 2007, pp. 437–442.

[23] T. H. Brants, I. Tsochantaridis, T. Hofmann, and F. R. Chen, "Methods, apparatus, and program products for performing incremental probabilistic latent semantic analysis," U.S. Patent 20 060 112 128, May 25, 2006.

[24] J. T. Chien and M. S. Wu, "Adaptive Bayesian latent semantic analysis," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 1, pp. 198–207, Jan. 2008.

[25] T. C. Chou and M. C. Chen, "Using incremental PLSI for threshold-resilient online event analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 3, pp. 289–299, Mar. 2008.

[26] N. Bassiou and C. Kotropoulos, "RPLSA: A novel updating scheme for probabilistic latent semantic analysis," *Comput. Speech Lang.*, vol. 25, no. 4, pp. 741–760, Oct. 2011.

[27] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed. Norwell, MA, USA: Kluwer, 1998, pp. 355–368.

[28] A. Langville, C. Meyer, and R. Albright, "Initializations for the nonnegative matrix factorization," in *Proc. 12th. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 23–26.

[29] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1999, pp. 50–57.

[30] T. Hofmann and J. Puzicha, "Unsupervised learning from dyadic data," Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep. TR-98-042, 1998.

[31] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, nos. 1–2, pp. 177–196, Jan. 2001.

[32] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. R. Statist. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[33] D. Gildea and T. Hofmann, "Topic-based language models using EM," in *Proc. 6th Eur. Conf. Speech Commun. Technol.*, 1999, pp. 2167–2170.

[34] T. Brants, "Test data likelihood for PLSA models," *Inf. Retr.*, vol. 8, no. 2, pp. 181–196, 2005.

[35] H. Wu, D. Zhang, Y. Wang, and X. Cheng, "Incremental probabilistic latent semantic analysis for automatic question recommendation," in *Proc. ACM Conf. Recommender Syst.*, Lausanne, Switzerland, Oct. 2008, pp. 99–106.

[36] J. Xu, G. Ye, Y. Wang, G. Herman, B. Zhang, and J. Yang, "Incremental EM for probabilistic latent semantic analysis on human action recognition," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Sep. 2009, pp. 55–60.

[37] J. Xu, G. Ye, Y. Wang, W. Wang, and J. Yang, "Online learning for PLSA-based visual recognition," in *Proc. 10th Asian Conf. Comput. Vis.*, 2011, pp. 95–108.

[38] A. Hinneburg, H.-H. Gabriel, and A. Gohr, "Bayesian folding-in with Dirichlet kernels for PLSI," in *Proc. 7th IEEE Int. Conf. Data Mining*, Omaha, NE, USA, Oct. 2007, pp. 499–504.

[39] E. Gaussier and C. Goutte, "Relation between PLSI and NMF and implications," in *Proc. 28th ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2005, pp. 601–602.

[40] H. Q. Ding, T. Li, and W. Peng, "NMF and PLSI: Equivalence and a hybrid algorithm," in *Proc. 29th ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2006, pp. 641–642.

[41] H. Ney, S. Martin, and F. Wessel, "Statistical language modeling using leaving-one-out," in *Corpus-Based Methods in Language and Speech Processing*, S. Young and G. Bloothooft, Eds. Dordrecht, The Netherlands: Kluwer, 1997, pp. 174–207.

[42] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, Jul. 1980.

[43] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2006–2021, Nov. 2010.

[44] M. Yan and K. Ye, "Determining the number of clusters using the weighted gap statistic," *Biometrics*, vol. 63, no. 4, pp. 1031–1037, Apr. 2007.

[45] G. W. R. Tibshirani and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. R. Statist. Soc. (Ser. B)*, vol. 63, no. 2, pp. 411–423, 2001.

**Nikoletta K. Bassiou** received the B.Sc. and Ph.D. degree in informatics from the Aristotle University of Thessaloniki, Thessaloniki, Greece.

She is currently a Post-Doctoral Researcher with the Department of Informatics, Artificial Intelligence and Information Analysis Laboratory, Aristotle University of Thessaloniki. Her current research interests include audio, speech and language processing, signal processing, pattern recognition, and information retrieval.

**Constantine L. Kotropoulos** (S'88–M'94–SM'06) was born in Kavala, Greece, in 1965. He received the Diploma (Hons.) degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1988 and 1993, respectively.

He is currently an Associate Professor with the Department of Informatics, Aristotle University of Thessaloniki. From 1989 to 1993, he was a Research and Teaching Assistant with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki. In 1995, he joined the Department of Informatics, Aristotle University of Thessaloniki, as a Senior Researcher, and served as Lecturer from 1997 to 2001 and an Assistant Professor from 2002 to 2007. He was a Visiting Research Scholar with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, USA, from 2008 to 2009. He was with the Signal Processing Laboratory, Tampere University of Technology, Tampere, Finland, in 1993. He has co-authored 48 journal papers, 165 conference papers, and contributed seven chapters to edited books. He is the co-editor of the book *Nonlinear Model-Based Image/Video Processing and Analysis* (J. Wiley and Sons, 2001). His current research interests include audio, speech, and language processing, signal processing, pattern recognition, multimedia information retrieval, biometric authentication techniques, and human-centered multimodal computer interaction.

Prof. Kotropoulos was a Scholar of the State Scholarship Foundation of Greece and the Bodossaki Foundation. He is a member of EURASIP, IAPR, and the Technical Chamber of Greece. He is an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS, a member of the editorial board of *Advances in Multimedia, ISRN Artificial Intelligence, Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization Journal*, and serves as a EURASIP Local Liaison Officer for Greece.