Online Topic Modeling for Short Texts

Suman Roy[1](✉), Vijay Varma Malladi[1], Ayan Sengupta[1], and Souparna Das[2]

[1] Optum Global Solutions India Pvt. Ltd. (UnitedHealth Group),
Bangalore 560 103, India
{suman.roy,malladi_varma,ayan_sengupta}@optum.com

[2] International Institute of Information Technology (IIIT-H), Hyderabad,
Hyderabad 500 032, India
souparna.das@students.iiit.ac.in

Abstract. Retrieval of knowledge from short texts has attracted a lot of attention these days as topic discovery from them can unearth hidden information. In many applications, such topics are needed to be learned on the fly for streaming short texts. In this work we propose an online topic discovery algorithm (OTDA) for short texts. It overcomes the inability of short texts to capture word co-occurrence information by adopting word-context semantic correlation through the skip-gram view of the corpus, following the approach of semantics-assisted NMF (SeaNMF) model due to Shi *et al.* This OTDA works with one data point or one chunk of data points at a time instead of keeping the entire data in the memory, and also admits the property of memorylessness. We consider a couple of public data sets and an internal data set to conduct experiments using one-pass and multi-pass iterations of the proposed algorithm. The results show encouraging performance of OTDA in terms of average Frobenius loss, Topic Coherence, Normalized Mutual Information (NMI), and emerging topic detection.

Keywords: Data mining · Online topic modeling · Short texts · Non-negative matrix factorization (NMF) · Average frobenius loss · Topic Coherence · NMI · Emerging topic

1 Introduction

Lot of applications involving short texts need to possess the ability to learn topics on the fly as new data points arrive in the context of an evolving system. For example, consider the case when an organization tries to address the issue of understanding customer feedback (which is typically short text) using topic modeling. With the constant churning of feedback from customers it is not very prudent to run the topic modeling algorithm on complete data on every update (whenever a new feedback is collected). Also as an organization introduces new

S. Das—This work was done when the author was an intern with Optum Global Solutions, Hyderabad during May-June'19.

E. Kafeza et al. (Eds.): ICSOC 2020, LNCS 12571, pp. 563–579, 2020.
https://doi.org/10.1007/978-3-030-65310-1_41

services and functionalities on the existing issues in their products/services, the nature of user supplied feedback texts changes over time. To capture the thematic content of evolving feedback materials an online topic modeling algorithm should be in place that can give more importance to the current feedback than the older feedback texts. Motivated by this, we propose an online topic discovery algorithm (OTDA) for streaming short texts which incorporates word-context semantic correlation learnt from the skip-gram view of the corpus much like the semantics-assisted NMF (SeaNMF) model [19], with the adaptivity of forgetting mechanism [5]. The SeaNMF model is solved using a block co-ordinate descent (BCD) algorithm. The well known methods for solving BCD need to hold the entire data matrix in the memory throughout the process of computation which can be prohibitive in case of large amount of data sets. Although various online NMF algorithms like the algorithm [6], have been proposed that can detect latent factors and track their evolution with new data arrival, none of them are suitable to be applied to short texts. To address this issue we incorporate a variant of the online NMF algorithm of [21] in OTDA, grounded in the framework of SeaNMF [19], to discover topics from very large scale/streaming short texts.

The OTDA algorithm works with one data point or one chunk of data points instead of storing the whole data in the memory. Further it updates the topic representation in an underlying space as well as context representation in terms of words on arrival of new data stream, by employing Projected Gradient Descent (PGD) algorithm in both the steps. Admission of context information improves the quality of incremental topic modeling as it can capture the semantics of the short text corpus based on word-document and word-context correlations, thus overcoming the problem of lacking word co-occurrence in short texts. To highlight the adaptivity of our learning algorithm we introduce a decay factor that exponentially reduces the contribution of history data, thereby imposing a forgetting (memorylessness) mechanism on the topic discovery process [5]. We design experiments to investigate the effect of the forgetting mechanism, and the results show that one needs to forget to adapt, that is, in absence of decay parameters the quality of generated topics suffers (NMI values go down) as new data points arrive, and richer topics are generated for streaming data when decay factors are present.

Contribution of This Work. This work has contributed to the body of online topic discovery in several ways. The topic learning incorporates word-context semantic correlation from SeaNMF model, however, it uses the framework of distributed clustering algorithm [22] without an increase in computational overhead and memory requirements. This algorithm has noticeable speed-up as the topic computation is done locally via a reduction in the memory footprint. Also like other online applications, we allow memorylessness with our method by introducing decay factors in the computation which causes the past history to be forgotten at exponential rate and attaches more importance to the current set of data. Extensive experimentation on real-life data sets produce interesting results on metrics, such as average Frobenius loss, Topic Coherence, NMI and emergent topic detection.

Organization: The paper is organized as follows. In the next section (Subsect. 1.1) we discuss the current literature related to this work. In Sect. 2 we review background material on NMF concepts and related matters. We propose our online topic discovery algorithm in Sect. 3. We discuss the data sets and the metrics used for experimentation in Sect. 4 and 5 respectively. The results of the experiments are furnished in Sect. 6. Finally we conclude in Sect. 7.

1.1 Related Work

Two groups of topic models are frequently employed to automatically extract topical contents from the documents, generative probabilistic models such as PLSA [10], LDA [3], and non-negative matrix factorization (NMF) [24]. They normally work well for lengthy documents. However these techniques do not produce meaningful results for short texts as term document matrix is very sparse which produces scarce word co-occurrence information and hence, generates poor quality topics [7,19]. There are lot of methods proposed in recent times to tackle this problem. These include aggregating short texts into pseudo-documents, and extracting cross document co-occurrence [16,27] using internal semantic relationship between words. While a pseudo-document generated in the first approach may contain many irrelevant short texts, noise and bias can creep in due to adoption of Wikipedia-centric notions of semantics in the second approach. To alleviate these problems, Shi *et al.* have proposed a novel semantics-assisted NMF (SeaNMF) model for short texts which incorporates word-context semantic correlations learned from the skip-gram view of the corpus [19]. Rest of the discussion on relevant prior art is divided into two parts, online topic discovery and online NMF.

Online Topic Discovery. In one of the earlier work on online topic modeling based on LDA Blei *et al.* [2] develop a family of probabilistic time series models in order to analyze the time evolution of topics in documents. Another LDA-based model is proposed in [23] to model a topic as a continuous distribution over timestamps and the mixture distribution as a function of both word co-occurrences and the document's timestamp. AlSumait *et al.* [1] introduce a topic modeling framework based on the LDA model to make it work in an online fashion such that it incrementally builds an up-to-date model (mixture of topics per document and mixture of words per topic) as a set of documents appear. The authors [11] propose another online topic model for sequentially analyzing time evolution of topic along multi-scales in a large collection of documents. Some online topic models have been also proposed for short texts like tweet data, such as [18] wherein the authors model the generation process of tweets by estimating the ratio between topic words and general words for each user.

Online NMF. We do not come across any work which uses NMF to present online topic model, hence we discuss few pieces of works related to online NMF. Cao *et al.* have proposed an online NMF which finds two factor matrices to approximate the whole data matrix [6]. Although it performs well in practice it cannot be applied to large-scale or streaming data sets due to the memory limitations. Bucak and Gusel have proposed an incremental NMF [5] in which the term topic matrix at $(t+1)$th step is updated on the arrival of $(k+1)$th sample. It has been seen that this works well in practice but, it is time consuming as the updation of rules have slow convergence. Zhou *et al.* has proposed another variant of incremental NMF with volume constraint [26]. In [8] Guan and Tao propose an efficient online NMF algorithm that learns NMF in an incremental fashion using robust stochastic approximation. In [21] an online NMF algorithm has been proposed for efficient document clustering for very large and streaming data sets. The proposed algorithm in this paper is an improvement of this algorithm in the sense that we consider word context correlation in the model and incorporate decay factors that cause the past history to be forgotten at an exponential rate.

2 Basic NMF Model for Topic Discovery

In this section we discuss basic NMF method, its application to topic modeling and the recently proposed SeaNMF method [19] for short texts.

Notation: Let $\mathbb{R}$ denote the set of real numbers (or reals), $\mathbb{R}_+$ the set of non-negative real numbers and $\mathbb{N}$ the set of natural numbers. $\boldsymbol{x} \in \mathbb{R}^n$ denotes an n-dimensional vector of reals. $\mathbf{1}_K$ denotes a row vector of size 1 whose all elements are 1. Also $\|\boldsymbol{x}\|_1$ and $\|\boldsymbol{x}\|_2$ denote the ℓ_1 and ℓ_2 norms of vector $\boldsymbol{x}$ respectively. We use the notation $\mathbf{X} \in \mathbb{R}^{p\times q}$ to denote a matrix of real numbers having p and q number of rows and columns respectively (or having dimension $p \times q$). We denote the elements of a matrix $\mathbf{X} \in \mathbb{R}_+^{p\times q}$ as $[x_{ij}]_{\{1\le i\le p, 1\le j\le q\}}$. We use $\boldsymbol{X}_{i\cdot}$ and $\boldsymbol{X}_{\cdot j}$ to denote the ith row vector and the jth column vector of matrix $\mathbf{X}$. In some cases the column vector $\boldsymbol{X}_{\cdot j}$ will be also denoted as $\boldsymbol{x}_j$ as before. Further $\|\mathbf{X}\|_F^2$ denotes the sum of the squared elements in the matrix $\mathbf{X}$ (also called the Frobenius norm). The zero matrix $\mathbf{0}$ has all zero entries with its dimension to be read off from its context.

Basic NMF Model. The problem of Non-Negative Matrix Factorization (NMF) deals with factoring a given matrix into two non-negative matrices [13,24]. Given an input matrix $\mathbf{X} \in \mathbb{R}_+^{m\times n}$, an integer $K \ll \min(m,n)$, NMF tries to solve a lower-rank approximation, $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$. where $\mathbf{U} \in \mathbb{R}_+^{m\times K}$ and $\mathbf{V} \in \mathbb{R}_+^{n\times K}$ are factor matrices. This is done by considering the optimization problem that minimizes the following objective function/loss function (also called the error of approximation or the Frobenius loss):

$$\min \mathcal{L}(\mathbf{U},\mathbf{V}) \left(= \frac{1}{2}\left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_F^2\right), \text{ s.t.}, \mathbf{U} \ge 0, \mathbf{V} \ge 0 \tag{1}$$

Popular algorithms for solving the NMF problem with Frobenius loss as given by Eq. 1 are Multiplicative Update Rule (MUA) [14], Blockwise Co-ordinate Descent (BCD) [12], Projected Gradient Method (PGD) [15] to name a few. We shall mainly adopt PGD [15] which follows alternative minimization principle.

Topic Discovery Using NMF. In topic modeling, $\mathbf{X} \in \mathbb{R}_+^{m\times n}$ is called the term-document matrix where we assume a given corpus with n documents and

m terms. $\boldsymbol{X}_{\cdot l} \in \mathbb{R}_{+}^{l}$ represents the l-th column vector of $\mathbf{X}$, which corresponds to the bag-of-words representation of document l with respect to m terms, possibly using TF*IDF weight after some pre-processing, and column-wise ℓ_2 normalization. For solving the minimization problem in Eq. 1 one assumes a predetermined number of topics K.

Topic Modeling for Short Texts Using NMF. As short texts are sparse and consists of only a few terms many unrelated documents may lead to biased relationship between terms resulting in poor clustering (and topic extraction). Moreover, most of the algorithms for solving NMF fail to appropriately discover the relationship between terms and their contexts. To overcome this problem the authors in [19] propose a novel semantics-assisted NMF (SeaNMF) model to learn topics from short texts.

The SeaNMF approach is based on the idea that terms are dependent on contexts as they appear around them. Towards this the authors define term-context correlation matrix $\mathbf{R}$ [19] using Skip-gram view of the corpus in the presence of an M-dimensional context vector $\boldsymbol{c}$:

$$r_{ij} = \max\left[\log\left(\frac{\#(t_i, c_j)}{\#(t_i)\cdot p(c_j)}\right) - \log\kappa, 0\right], 1 \leq i \leq m, 1 \leq j \leq M \tag{2}$$

We use $\mathbb{V}$ to denote the the overall vocabulary of terms and contexts. The notation $\#(t_i, c_i)$ denotes the number of times t_i appears with context c_i in text corpora. Further $\#(t_i) = \sum_{c_j \in \mathbb{V}} \#(t_i, c_j)$ and $\#(c_j) = \sum_{t_i \in \mathbb{V}} \#(t_i, c_j)$ represent the number of times t_i and c_j occur in all possible term-context pairs respectively, and κ is the number of negative samples. Finally, $p(c_j)$ is a unigram distribution for sampling a context c_j defined as $p(c_j) = \frac{\#(c_j)}{\sum_{c_j \in \mathbb{V}} \#(c_j)}$. There are a few techniques to specify the sliding window for a context [19]. For example, each document can be selected as a window of context [19] for a term in short text corpus or it can be a long pseudo-text obtained by aggregating short texts belonging to a cluster. A fixed size window of neighboring words can act as a context for a word, and so on.

Finally, SeaNMF proceeds in two step. In the first step the term-context correlation matrix $\mathbf{R}$ is factored into two matrices, term-topic matrix $\mathbf{U} \in {\mathbb{R}_{+}}^{m\times K}$ and another newly introduced matrix context topic matrix $\mathbf{U}_c \in {\mathbb{R}_{+}}^{M\times K}$. In the second step the term document matrix $\mathbf{X} \in \mathbb{R}_{+}^{m\times n}$ is factored along with the term-topic matrix to obtain the document-topic matrix $\mathbf{V} \in \mathbb{R}_{+}^{n\times K}$ (some sparsity constraint may be imposed on $\mathbf{X}$ in the process). For details the reader is advised to consult [19].

The computational complexity of SeaNMF for short texts is same as the computational complexity of standard NMF [12] using MUA or BCD method and is equal to $O(nmK)$ for single iteration and $O(TnmK)$ for T iterations (assuming $K \ll \min(n, m)$). However, as $\mathbf{R}$ and $\mathbf{X}$ are sparse matrices the authors conclude that the complexity of the SeaNMF model using BCD is $O(zK)$ $(O(TzK))$ for single (T) iteration(s), where $z = \max(z_{\mathbf{R}}, z_{\mathbf{X}})$, and $z_{\mathbf{R}}$ and $z_{\mathbf{X}}$ are the nonzero elements in the matrices $\mathbf{R}$ and $\mathbf{X}$ respectively, $\max(z_{\mathbf{R}}, z_{\mathbf{X}}) \ll mn$ and

$K \ll \min(m,n)$, which is less expensive than the standard NMF. Further it is required to hold the matrices $\mathbf{X}$ and $\mathbf{R}$ at a storage cost of $O(mn)$ in the SeaNMF model.

3 Proposed Online Topic Modeling for Short Texts

We propose an online Topic Discovery (OTDA) algorithm that updates the matrices $\mathbf{U}, \mathbf{U}_c$ and $\mathbf{V}$ by adding the effects of subsequent samples in an incremental fashion.

3.1 An Incremental Form of NMF

Note the loss function in Eq. 1 can be decomposed as [12]:

$$\mathcal{L}(\mathbf{U},\mathbf{V}) = \left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_F^2 = \sum_{j=1}^{n} \left\|\mathbf{X}_{\cdot j} - \mathbf{U}\mathbf{V}_{\cdot j}{}^T\right\|_F^2 = \sum_{j=1}^{n} \|\boldsymbol{x}_j - \mathbf{U}\boldsymbol{v}_j\|_F^2 \tag{3}$$

Consider the problem of generating K topics from the data set. The term topic matrix will look like $\mathbf{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_K]$ which represents each topic as the weighted combination of terms. Further $\boldsymbol{v}_j = [g_{j1} \cdots g_{jK}]^T$ are the reconstruction weights of $\boldsymbol{x}_j$ from these representatives.

When $\mathbf{U}$ is fixed, the minimum value of $\mathcal{L}(\mathbf{U},\mathbf{V})$ is reached if and only if the cost function $\mathcal{L}(\mathbf{U},\boldsymbol{v}_j) = \|\boldsymbol{x}_j - \mathbf{U}\boldsymbol{v}_j\|_F^2$ is minimized for all $j, 1 \le j \le n$. Thus, one solves independent Non-negative Least Squares (NNLS) problems of the form,

$$\min_{\boldsymbol{v}_j \ge 0} \|\boldsymbol{x}_j - \mathbf{U}\boldsymbol{v}_j\|_F^2, j = 1, 2 \ldots n \tag{4}$$

and aggregate the solution as $\mathbf{V} = [\boldsymbol{v}_1 \cdots \boldsymbol{v}_n]$.

3.2 Computing Document Representations

In this step we let the topic representation $\mathbf{U}$ to be fixed. We solve the optimization problem in Eq. 5 to compute $\boldsymbol{v}^{(t)}$:

$$\min \frac{1}{2}\left(\left\|\boldsymbol{x}^{(t)} - \mathbf{U}\boldsymbol{v}^{(t)}\right\|_F^2 + \lambda \left\|\boldsymbol{v}^{(t)}\right\|_1^2\right) \text{ s.t.}, \boldsymbol{v}^{(t)} \ge 0, \mathbf{U} \text{ is given} \tag{5}$$

where $\lambda > 0$ is a constant. We also impose the sparsity on $\boldsymbol{v}^{(t)}$ by adding a suitable ℓ_1 norm on it. The NNLS problem given by Eq. 5 is the so-called Lasso problem [20] which can be solved using Projected Gradient (PGD) [15] with the gradient computed as: $\frac{\partial \mathcal{L}^{(t)}}{\partial \boldsymbol{v}^{(t)}} = -(\boldsymbol{x}^{(t)})^T\mathbf{U} + (\mathbf{U}\boldsymbol{v}^{(t)})^T\mathbf{U} + \lambda \mathbf{1}_K^T$.

3.3 Solving for Context Representation

In this step we try to compute the context representation of term in an incremental fashion. We assume that the M-dimensional context vector $\boldsymbol{c}^{(t)}$ is available at time instant t, this can be invariant with time or can be learned incrementally as new samples arrive, *e.g.*, it can be learned online as a cluster of data points for streaming data [25]. Thus at time point t we can compute the term context correlation matrix $\mathbf{R}^{(t)}$ with the aid of current context information $\boldsymbol{c}^{(t)}$ using Eq. 2.

Now we solve for the underlying representation of context in the form of context-topic matrix $\mathbf{U}_c^{(t)}$ by minimizing the following cost function in Eq. 6 keeping $\mathbf{U}$ as constant. Also below, we impose the condition that the computed $\mathbf{U}_c^{(t)}$ will be dense by using a ℓ_2-regularization term for it, where $\beta > 0$ is a constant. Again this NNLS can be solved using a standard optimization algorithm.

$$\frac{1}{2}\left\|\mathbf{R}^{(t)} - \mathbf{U}(\mathbf{U}_c^{(t)})^T\right\|_F^2 + \beta\left\|\mathbf{U}_c^{(t)}\right\|_F^2 \text{ s.t., } \mathbf{U}_c^{(t)} \geq 0, \ \mathbf{U} \text{ is given} \tag{6}$$

3.4 Updating Topic Representations

The topic represented in the form of term-topic matrix $\mathbf{U}$ is updated in this step. At time instant t, as $\boldsymbol{x}^{(t)}$ arrives, OTDA first solves for $\boldsymbol{v}^{(t)}$ and $\mathbf{U}_c^{(t)}$ using $\mathbf{U}^{(t-1)}$, and then updates $\mathbf{U}$ by minimizing the following loss function:

$$\mathcal{L}^{(t)}(\mathbf{U}^{(t)}) = \left[\frac{\gamma_0}{2}\sum_{s=1}^{t}\mu\left\|\mathbf{R}^{(s)} - \mathbf{U}^{(t)}\mathbf{U}_c^{(s)^T}\right\|_F^2 + \sum_{s=1}^{t}\frac{\gamma_s}{2}\left\|\boldsymbol{x}^{(s)} - \mathbf{U}^{(t)}\boldsymbol{v}^{(s)}\right\|_F^2\right] \tag{7}$$

under the constraints $\mathbf{U}^{(t)} \geq 0$. Further $\boldsymbol{v}^{(s)}$ is obtained as a solution of the minimization problem given in Eq. 5, and $\mathbf{U}_c^{(s)}$ is found by solving Eq. 6.

We introduce decay factors [5] to ensure that the effects of new samples on the representation is higher, while that of old ones wane (memorylessness). That is, γ_0, γ_s $(s = 1, 2 \ldots, t)$ are the decay factors which cause the past history to be forgotten at an exponential rate. We define,

$$\begin{aligned}\gamma_j &= \ {\gamma_0}^{(t-2r)}, \quad j \leq 2r \\ &= {\gamma_0}^{(t-j)}\gamma_f, \ 2r < j \leq t\end{aligned}$$

We assume $\gamma_0 < 1(\gamma_0 \approx 0.5), \gamma_f < 1(\gamma_f \approx 0.9)$ and $r = 1$.
The gradient of $\mathcal{L}^{(t)}$ *wrt* $\mathbf{U}^{(t)}$ is given by

$$\begin{aligned}\nabla_{\mathbf{U}^{(t)}}\left(\mathcal{L}^{(t)}(\mathbf{U}^{(t)})\right) = -\gamma_0\sum_{s=1}^{t}\mu[\boldsymbol{R}^{(s)}\mathbf{U}_c^{(s)} - \mathbf{U}^{(t)}\mathbf{U}_c^{(s)^T}\mathbf{U}_c^{(s)}] \\ -\sum_{s=1}^{t}[\gamma_s(\boldsymbol{x}^{(s)}\boldsymbol{v}^{(s)^T} - \mathbf{U}^{(t)}\boldsymbol{v}^{(s)}\boldsymbol{v}^{(s)^T})]\end{aligned} \tag{8}$$

One can update $\mathbf{U}^{(t)}$ using PGD assuming an initial value of $\mathbf{U}_0^{(t)}$. However, when we implement the first-order PGD we do not get quality results as expected, because there are some known drawbacks for the first-order PGD, for instance, large step size in the update leads to slow convergence etc. Hence we use second order PGD for which we compute the Hessian matrix of $\mathcal{L}^{(t)}$ *wrt* $\mathbf{U}^{(t)}$,

$$\mathcal{H}_{\mathbf{U}^{(t)}}\left(\mathcal{L}^{(t)}(\mathbf{U}^{(t)})\right) = 2\sum_{s=1}^{t}\left[\mu \cdot \gamma_0 \cdot {\mathbf{U}_c^{(s)}}^T \mathbf{U}_c^{(s)} + \gamma_s \boldsymbol{v}^{(s)} {\boldsymbol{v}^{(s)}}^T\right] \tag{9}$$

Finally we adopt the following update rule for the second order PGD that can guarantee faster convergence without using any parameter:

$$\mathbf{U}_{k+1}^{(t)} = \mathcal{P}\left[\mathbf{U}_k^{(t)} - \nabla_{\mathbf{U}^{(t)}}\left(\mathcal{L}^{(t)}(\mathbf{U}_k^{(t)})\right)\mathcal{H}_{\mathbf{U}^{(t)}}^{-1}\left(\mathcal{L}^{(t)}(\mathbf{U}_k^{(t)})\right)\right] \tag{10}$$

where $\mathcal{H}^{-1}$ is the inverse of the Hessian matrix $\mathcal{H}$. As the computation of $\mathcal{H}^{-1}$ matrix is time consuming we adopt Conjugate Gradient to calculate it. The second-order PGD has been shown in Algorithm 1. For notational convenience we introduce the following first-order and second-order terms respectively.

$$\mathbf{W}^{(t)} = \sum_{s=1}^{t}\left[\gamma_0 \cdot \mu \cdot \boldsymbol{R}^{(s)}\mathbf{U}_c^{(s)} + \gamma_s \cdot \boldsymbol{x}^{(s)} {\boldsymbol{v}^{(s)}}^T\right] \tag{11}$$

$$\mathbf{H}^{(t)} = \sum_{s=1}^{t}\left[\gamma_0 \cdot \mu \cdot {\mathbf{U}_c^{(s)}}^T \mathbf{U}_c^{(s)} + \gamma_s \cdot \boldsymbol{v}^{(s)} {\boldsymbol{v}^{(s)}}^T)\right] \tag{12}$$

Algorithm 1: 2nd order PGD for updating $\mathbf{U}^{(t)}$

```
Input        : Number of topics $K$, Initial term-topic matrix $\mathbf{U}_0^{(t)}$, and
               document-topic matrix $\mathbf{V}^{(t)}$, and other terms $\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)}$
/* Using Conjugate Gradient Descent (CGD); $k$ is the index of
   iterations and $\Gamma$ is no. of iterations                        */
for $k = 1, \ldots, \Gamma$ do
    Compute the gradient $\Delta_k = \mathbf{W}^{(t)} - \mathbf{U}_{k-1}^{(t)}\mathbf{H}^{(t)}$          † ;
    Solve $\mathbf{Q}$ such that $\mathbf{Q}\mathbf{H}^{(t)} = \Delta_k$ ;
    $\mathbf{U}_k^{(t)} = \max\left(\mathbf{0}, \mathbf{Q} + \mathbf{U}_{k-1}^{(t)}\right)$
end
```

3.5 Online Topic Discovery

Using second-order PGD we can design an online algorithm for topic discovery for short texts. This algorithm procedure can be performed using one pass and multiple passes. The complete one-pass algorithm is mentioned in Algorithm 2.

This algorithm follows mini-batch implementation [4] which is at the confluence of Stochastic Gradient Descent and the traditional batch descent algorithms. As this algorithm imports p data points at each step, the OTDA algorithm can be expected to converge faster. Consequently the update rules for $\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)}$ are given by,

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} + \sum_{i=1}^{p} \left[\gamma_0 \cdot \mu \cdot \boldsymbol{R}^{(t,i)} \mathbf{U}_c^{(t,i)} + \gamma_t \cdot \boldsymbol{x}^{(t,i)} {\boldsymbol{v}^{(t,i)}}^T\right] \tag{13}$$

$$\mathbf{H}^{(t)} = \mathbf{H}^{(t-1)} + \sum_{i=1}^{p} \left[\gamma_0 \cdot \mu \cdot {\mathbf{U}_c^{(t,i)}}^T \mathbf{U}_c^{(t,i)} + \gamma_t \cdot \boldsymbol{v}^{(t,i)} {\boldsymbol{v}^{(t,i)}}^T)\right] \tag{14}$$

Notice that we do not recompute $\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)}$ afresh each time. Rather we update $\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)}$ by using Eqs. 13 and 14 respectively. Although only a single pass over the data seems to be feasible in data stream applications, multiple passes can be run in many applications. In the multi-pass OTDA, the document topic assignment matrix $\mathbf{V}$ can be updated using term-topic matrix $\mathbf{U}$. Moreover, the first and second order information $\mathbf{W}$ and $\mathbf{H}$ in the previous pass can be updated and utilized. When multiple passes are feasible one can expect to obtain more accurate results.

Algorithm 2: One-pass OTDA in the mini-batch model (n is the total no of data points

```
Input          : Term-document matrix $\mathbf{X}$, Initial term-topic matrix $\mathbf{U}^{(0)}$, No
                 of data points at each step = $p$, No of steps $S = \left\lceil \frac{n}{p} \right\rceil$, Initial
                 Emerging topic set Etopics(1) = $\emptyset$, Confidence level CL
Initialization: $\mathbf{W}^{(0)} = \mathbf{0}, \mathbf{H}^{(0)} = \mathbf{0}$
for $t = 1, \ldots, S$ do
    Draw $\boldsymbol{X}^{(t)}$ ($p$ data points) from from $\mathbf{X}$;
    Compute $\boldsymbol{v}^{(t)}$ by solving the optimization problem given in Eqn. 5;
    Update $\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)}$ using Eqns 13 and 14 respectively;
    Update $\mathbf{U}^{(t)}$ by Algorithm 1;
    if  $t > 1$ then
        Etopics($t$) = Edetect($CL$) using the algorithm in [1]
    end section 3.2
    ;
end
```

3.6 Computational Savings

As the OTDA proceeds by solving Eqs. 5, 6 and 7 it incurs computational cost of $O(mnK), O(mmK)$ and $O(nmK)$ at each of these steps respectively. However, since $\mathbf{X}$ and $\mathbf{R}$ are sparse matrices, we only need to multiply the nonzero elements with factor matrices. Hence the cost for these operations will be

$O(z_{\mathbf{X}}K), O(z_{\mathbf{R}}K)$ and $O(zK)$, where $z = \max(z_{\mathbf{R}}, z_{\mathbf{X}})$ for single iteration[1]. The proposed OTDA will therefore will have a cost of $O(zK)$ for single iteration. The Frobenius loss of OTDA is frequently very close to the Frobenius loss of the SeaNMF algorithm after $T \leq 2$ iterations as witnessed by our experimentation, which will save computational cost appreciably ($\approx O(zK)$ cost only). Also our one-pass OTDA needs to only load the data matrix once which involves low IO cost. Our experiment results shows that we often do not need many passes to obtain very accurate results.

3.7 Topic Detection and Tracking

Our dynamic topic model enables capturing the topics and their evolution over time. The vector $\mathbf{U}_{\cdot k}^{(t)}$ portrays the evolution of topic k at time t. As each topic is represented in the form of a column vector, represented as a weighted combination of terms the dissimilarity between the representation of a topic k at time point $t+1$ and t, is defined as $\mathrm{Dist}(k,t) = \left\|\mathbf{U}_{\cdot k}^{(t+1)} - \mathbf{U}_{\cdot k}^{(t)}\right\|_2$. We consider a topic to be *emerging* if it is different from its peers in the same stream, or from all the topics seen so far. The identification of emerging topics can be modeled by considering the K topic distances computed at time t using a confidence level CL. Then we use the algorithm in [1] (Sect. 3.2) to compute nominated emerging topics in which the function Edetect(CL) returns the emerging topics Etopics(t) generated in the time slice t and $(t+1)$.

4 Data Sets for Experimentation

We have considered four sets of short text data for experimental purposes, three of which are public datasets and the fourth is an internal data set. Public data sets are Yahoo manner, SearchSnippets and StackOverflow. Yahoo manner data set (Yahoo) is a subset of the Yahoo Answers Manner Questions, version 2.04[2]. The data set SearchSnippets (Snippets) is selected after searching through the transactions on the web using predefined phrases of 8 different domains. StackOverflow (Stack) is the challenge data set published online[3]. The fourth data set (Optum) contains feedback texts that are provided by customers (from an offshore center of Optum) in certain healthcare domains. Three public data sets are labeled with categories, for which we generate the same number of topics. For Optum feedback texts we assume 9 topics by using the standard criterion of selecting optimal number of clusters.

[1] We assume a low average number of PGD iterations for updating $\mathbf{U}$ or $\mathbf{V}$ in one round, and also a low average number of trials needed for implementing the Armijo rule [15,21].

[2] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l.

[3] Kaggle.com.

Table 1. Statistics of data sets considered

Data set	# docs	# terms	density(X)	density(R)	doc-length	#cats	#topics generated
Yahoo	24555	14370	0.0482	0.1598	11.1	8	8
Snippets	10060	23031	0.0561	0.513	17.87	8	8
Stack	10000	8162	0.0858	0.354	8.22	8	8
ptum	9999	4372	0.3736	1.896	28.41	NA	9

Some basic statistics of these data sets are shown in Table 1. '#docs' represents the number of documents in each data set, and '#terms' the number of terms in the vocabulary. The quantity 'density' is defined as $\frac{\#\text{non-zero}}{\#\text{docs} \cdot \#\text{terms}}$, where #non-zero is the number of non-zero elements in the matrix. The entities 'density($\mathbf{X}$)' and 'density($\mathbf{R}$)' represent the density of term-document matrix $\mathbf{X}$ and term-context correlation matrix $\mathbf{R}$, respectively. 'doc-length' represents the average length of the documents. '#cats' denotes the number of distinct categories.

5 Evaluation Metrics

We present an evaluation of our approach by comparing the performance of our online topic discovery algorithm with other relevant algorithms on three characteristics, average Frobenius loss [14,21], Topic Coherence (Coherence) [17] and Normalized Mutual Information (NMI) [7]. As a topic can be related to a cluster we use a cluster-related metric Normalized Mutual Information (NMI) to measure the efficacy of our method, especially for labeled data. Due to which, it is not possible to compute NMI values for Optum dataset.

For comparison with our OTDA on average Frobenius loss, we use the work on clustering using online NMF due to Wang *et al.* [21] (ClusterONMF). There is an old work of online NMF for latent factor tracking due to Cao *et al.* [6] (LatentONMF), however it is shown that ClusterNMF performs better than LatentONMF in terms of average Frobenius loss [21], and hence we do not consider LatentONMF in our experimentation. When we compare our OTDA using Coherence and NMI we use three baseline methods other than ClusterONMF, - adaptive Online-LDA (A-OLDA) [1], Online Learning for LDA (L-OLDA) [9] and Dynamic Topic Model (DTM) [2].

6 Experimental Results

We present experimental results on the data sets discussed before. For the benefit of reproducible research we upload all our codes and the baseline methods on https://github.com/varma-ds/OTDA. We have tweaked parameters appearing in loss functions in Sect. 3, but they do not have much effect on the results. So, we use default hyperparameter settings for each of the baselines. We use Scikit-learn's online LDA implementation[4] for L-OLDA and Gensim's LdaSeqModel[5]

[4] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html.

[5] https://radimrehurek.com/gensim/models/ldaseqmodel.html.

implementation for DTM. For L-OLDA and A-OLDA we use document topic prior value as $1/K$.

6.1 OTDA with Conjugate Gradient

We mainly focus on OTDA with second order methods using conjugate gradient method. The performance of OTDA with first order PGD is not satisfactory, and hence is not presented (for space constraints).

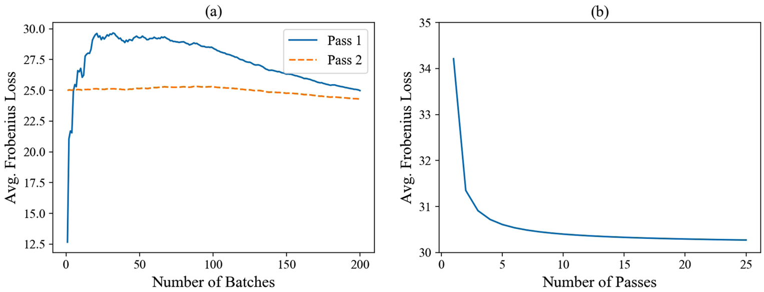

Fig. 1. (a) Average Frobenius loss for one-pass and two-pass with increasing number of batches (b) Average Frobenius loss with increasing number of passes on Yahoo dataset

For the below experiments, we assume that data is divided into different batches of some fixed size. For each batch, both term-document matrix and word context correlation matrix are generated using fixed vocabulary. For computing word-context correlation matrix each short text is considered as a context. Using this information word context correlation matrix is updated for each batch. Other context information like fixed size window of words, streaming text clusters [25] etc. can be also considered.

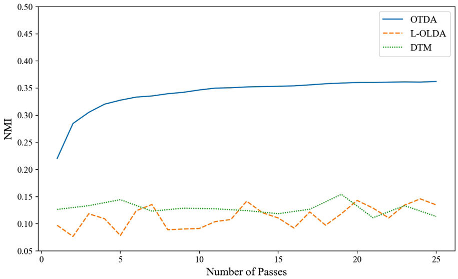

Fig. 2. NMI measured at the end of each pass on Yahoo data

We present the results for the average Frobenius loss for one-pass and two-pass with increasing number of batches. For the second pass we compute the average Frobenius loss using all the n data points. For the first pass, average Frobenius loss is calculated only for the data points seen so far. From Fig. 1(a), we can see that with only one pass of the algorithm, the average Frobenius loss increases at first and then starts decreasing as the number of batches increases. If two passes are allowed the average Frobenius loss remains almost constant, but it is smaller than the values in the first pass as it learns the topics from the initial batch only. All the data sets show almost similar pattern. Results in Fig. 1(b) indicate that the average Frobenius loss continues to decrease as we increase number of passes with diminishing returns for almost all data sets. Here we reproduce the results for only one initialization and omit results for other initializations for space constraints. We show results for Yahoo data set as other datasets exhibit similar patterns only.

We further compute the NMI values for the labeled data sets using OTDA and plot the results in Fig. 2 for Yahoo dataset. It shows that NMI continues to go up as we increase number of passes to an extent and then stabilizes.

6.2 Comparison with Online Methods

We now compare the performance of OTDA with ClusterONMF [6], A-OLDA [1], L-OLDA [9] and DTM [2] using the metrics topic coherence and NMI. Additionally, we compare Frobenius loss for both OTDA and ClusterNMF during learning. We publish the best score achieved by each of the models for all the datasets in Table 2.

Average Frobenius Loss. We compare our OTDA with ClusterONMF in terms of the average Frobenius loss (using Eq. 4.22 in [6]). We report the result for only one initialization and different batch sizes. Further we produce the results for only one-pass of the algorithm for obvious reasons. We compute the Frobenius loss given using Eq. 4.20 in [6] (at the final iteration) for each batch due to the method of Lee and Seung, which is shown as a dashed line (labeled by L-S) in Fig. 3, wherein which we report the average Frobenius loss for 3 different method on Yahoo dataset only. Similar behavior is observed in other 3 datasets as well, the description of which is omitted in this paper due to space constraint. In all the cases OTDA produces higher loss than ClusterONMF. It is expected as we minimize the Frobenius loss along with another term involving context information, that acts like a regularization term (see Eq. 7).

Table 2. Performance of OTDA against baselines. (Best scores across different batch sizes and number of passes are chosen for each model)

Data	Model	Loss at learning	Topic quality	
		Avg. Frobenius loss	Coherence	NMI
Yahoo	OTDA	0.748	**0.485**	**0.390**
	ClusterNMF	0.712	0.449	0.350
	L-OLDA	–	0.302	0.112
	A-OLDA	–	0.269	0.054
	DTM	–	0.340	0.123
Snippets	OTDA	10.114	**0.656**	0.280
	ClusterNMF	9.746	0.411	0.190
	L-OLDA	–	0.491	0.176
	A-OLDA	–	0.271	0.030
	DTM	–	0.560	**0.285**
Stack	OTDA	1.213	**0.327**	**0.186**
	ClusterNMF	1.112	0.084	0.185
	L-OLDA	–	0.295	0.077
	A-OLDA	–	0.190	0.035
	DTM	–	0.322	0.113
Optum	OTDA	1.762	**0.468**	–
	CLusterNMF	1.569	0.430	–
	L-OLDA	–	0.183	–
	A-OLDA	–	0.186	–
	DTM	–	0.191	–

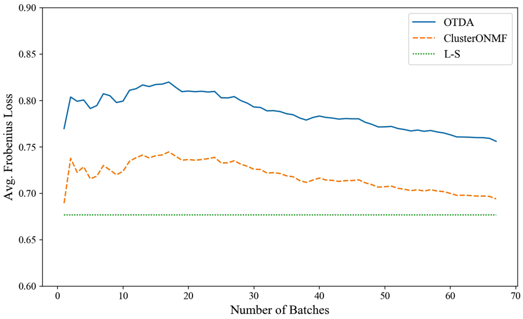

Fig. 3. Comparison of Average Frobenius loss on Yahoo

Topic Coherence. We compute topic coherence for all the data sets as shown in Table 2. For all of them OTDA performs better than all other baselines. While for Snippets, Stack and Optum datasets appreciable improvement of Coherence is observed for OTDA, Snippets data set shows marginal gain with both OTDA and DTM. Further, in Fig. 4(a) we observe that with increase in batch size, topic coherence values reduce as the models tend to assign more diverse and non-coherent words associated with topics.

NMI. Quantitative evaluation using NMI metric is conducted on the three data sets with label information, *e.g.*, Yahoo, Snippets and Stack have the same number of clusters being equal to 8. Table 2 depicts the comparison of clustering for each method on three labeled data sets. Overall, OTDA always outperforms ClusterONMF in terms of NMI values. For Yahoo and Snippets data, OTDA shows an improvement of 5-8% in NMI values in comparison to ClusterONMF. On the other hand, DTM performs slightly better than OTDA on Snippets dataset. Figure 4(b) shows the comparison of different models on Yahoo dataset. We observe a competitive performance between OTDA and ClusterNMF with increasing batch size.

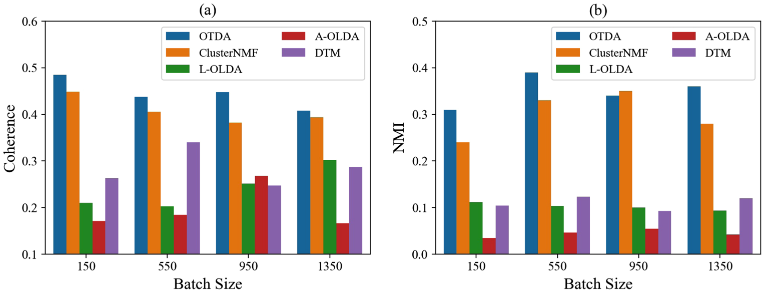

Fig. 4. Comparison of different models *w.r.t.* Topic Coherence and NMI on Yahoo data

6.3 Effect of Decay on Streaming Data

We now examine the effect of decay factors on the streaming data. For that we curate Yahoo data set as follows. We divide the data set into 4 groups and 2 types, characterized by the categories, that is, each type will contain exactly 4 distinct categories of data. Details of this curated data is shown in Table 3.

We assume each group corresponds to one batch and data arrives in batches. Using this curated data we have experimented with and without decay factors in OTDA formulation. The results are presented in Fig. 5. It shows that in absence of decay factors when a new type of data arrives, NMI reduces. But, with the introduction of decay factors in OTDA, the algorithm is able to forget the past topic distributions and learn the new topic distributions.

Table 3. Curated Yahoo data

Group	Data indices	Type	Categories
1	0–3999	1	Family, Maths, Cleaning, Dogs
2	4000–8999	2	Cooking, Finance, Repairs, Diet
3	9000–13999	1	Family, Maths, Cleaning, Dogs
4	14000–18999	2	Cooking, Finance, Repairs, Diet

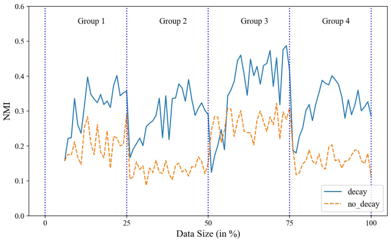

Fig. 5. Decay effect on NMI for Yahoo data

6.4 Emerging Topic Detection

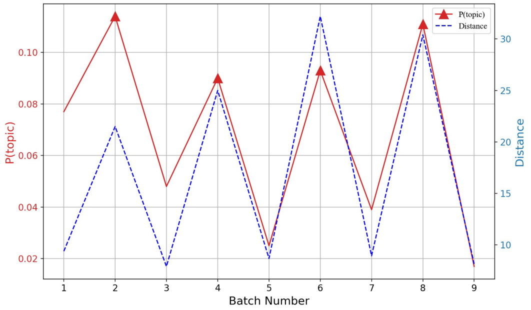

Fig. 6. Probability distribution and Distance of the topic **Maths** across different batch numbers (Trending regions are highlighted)

To test the ability of OTDA to detect novel topics as they evolve, we create synthesized data by mixing Yahoo and Stack Overflow data sets from which we take 10 categories (all the categories from Yahoo and only 2 categories from Stack overflow) in the following manner: (1) we add 9 categories in equal proportions (*i.e.p.*) excluding the topic **Maths**; (2) we add all the 10 categories *i.e.p.* including **Maths**; (3) we repeat step 1 and 2 four times; and (4) in the 9th time instant we have added 9 categories *i.e.p.* excluding **Maths**. With this synthesized data, we are able to detect the topic **Maths** as an emerging one at 2nd, 4th, 6th and 8th time instances at 90% confidence level (Fig. 6). The detected Topic probability distribution of **Maths** is also presented in Fig. 6.

7 Conclusion

We have proposed an efficient online NMF algorithm for discovering topics from short texts which processes incoming data incrementally. There are several reasons for choosing NMF over LDA to design this online algorithm.

While our method advocates optimizing loss function directly, other variations of (LDA-based) online topic discovery algorithms using variational inference techniques produce approximations of the actual results. Further, all Markov chain Monte Carlo-based topic extractions (*e.g.*, LDA) are asymptotically exact although computationally expensive. This makes our model a perfect fit for accurate as well as fast, scalable alternative to other topic models.

References

1. AlSumait, L., Barbará, D., Domeniconi, C.: On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In: Proceedings of ICDM 2008, pp. 3–12 (2008)
2. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proceedings of ICML 2006, pp. 113–120 (2006)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Bottou, L.: Stochastic learning. In: Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, Revised Lectures, pp. 146–168 (2003)
5. Bucak, S.S., Gunsel, B.: Incremental subspace learning via non-negative matrix factorization. Pattern Recogn. **42**(5), 788–797 (2009)
6. Cao, B., Shen, D., Sun, J.T., Wang, X., Yang, Q., Chen, Z.: Detect and track latent factors with online nonnegative matrix factorization. In: Proceedings of IJCAI 2007, pp. 2689–2694 (2007)
7. Cheng, X., Guo, J., Liu, S., Wang, Y., Yan, X.: Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: Proceedings of the 13th SIAM International Conference on Data Mining 2013, pp. 749–757 (2013)
8. Guan, N., Tao, D., Luo, Z., Yuan, B.: Online nonnegative matrix factorization with robust stochastic approximation. IEEE Trans. Neural Netw. Learn. Syst. **23**(7), 1087–1099 (2012)
9. Hoffman, M.D., Blei, D.M., Bach, F.R.: Online learning for latent Dirichlet allocation. In: Advances in Neural Information Processing Systems, vol. 23, pp. 856–864 (2010)
10. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR 1999, pp. 50–57. ACM (1999)
11. Iwata, T., Yamada, T., Sakurai, Y., Ueda, N.: Online multiscale dynamic topic models. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 663–672 (2010)
12. Kim, J., He, Y., Park, H.: Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. J. Global Optim. **58**(2), 285–319 (2013). https://doi.org/10.1007/s10898-013-0035-4
13. Kuang, D., Choo, J., Park, H.: Nonnegative matrix factorization for interactive topic modeling and document clustering. In: Celebi, M.E. (ed.) Partitional Clustering Algorithms, pp. 215–243. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-09259-1_7
14. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems, vol. 13, pp. 556–562. MIT Press (2001)
15. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. **19**(10), 2756–2779 (2007)

16. Quan, X., Kit, C., Ge, Y., Pan, S.J.: Short and sparse text topic modeling via self-aggregation. In: Proceedings of IJCAI 2015, pp. 2270–2276. AAAI Press (2015)
17. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: Proceedings of WSDM 2015, pp. 399–408. ACM (2015)
18. Sasaki, K., Yoshikawa, T., Furuhashi, T.: Online topic model for twitter considering dynamics of user interests and topic trends. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of EMNLP 2014, pp. 1977–1985. ACL (2014)
19. Shi, T., Kang, K., Choo, J., Reddy, C.K.: Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In: Proceedings of WWW 2018, pp. 1105–1114 (2018)
20. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. B **58**, 267–288 (1996)
21. Wang, F., Tan, C., König, A.C., Li, P.: Efficient document clustering via online nonnegative matrix factorizations. In: Eleventh SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics (2011)
22. Wang, F., Tan, C., Li, P., König, A.C.: Efficient document clustering via online nonnegative matrix factorizations. In: Proceedings of the 11th SIAM International Conference on Data Mining (SDM), pp. 908–919 (2011)
23. Wang, X., McCallum, A.: Topics over time: a non-Markov continuous-time model of topical trends. In: Proceedings of the 12th KDD, pp. 424–433. ACM (2006)
24. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2003. ACM (2003)
25. Zhong, S.: Efficient streaming text clustering. Neural Netw. **18**(5–6), 790–798 (2005)
26. Zhou, G., Yang, Z., Xie, S., Yang, J.: Online blind source separation using incremental nonnegative matrix factorization with volume constraint. IEEE Trans. Neural Networks **22**(4), 550–560 (2011)
27. Zuo, Y., Wu, J., Zhang, H., Lin, H., Wang, F., Xu, K., Xiong, H.: Topic modeling of short texts: a pseudo-document view. In: KDD 2016, pp. 2105–2114. ACM (2016)