Hierarchical Online NMF for Detecting and Tracking Topic Hierarchies in a Text Stream

Ding Tu , Ling Chen , Mingqi Lv , Hongyu Shi , Gencai Chen

PII: S0031-3203(17)30455-7
DOI: 10.1016/j.patcog.2017.11.002
Reference: PR 6355

To appear in: *Pattern Recognition*

Received date: 23 September 2016
Revised date: 12 October 2017
Accepted date: 4 November 2017

Please cite this article as: Ding Tu , Ling Chen , Mingqi Lv , Hongyu Shi , Gencai Chen , Hierarchical Online NMF for Detecting and Tracking Topic Hierarchies in a Text Stream, *Pattern Recognition* (2017), doi: 10.1016/j.patcog.2017.11.002

Highlights

- Propose an efficient hierarchical NMF framework-HONMF.
- Propose a mechanism to adaptively determine the topic numbers.
- Propose a mechanism to evolve the topic hierarchy.
- Propose a mechanism to adaptively determine the topic numbers.

Hierarchical Online NMF for Detecting and Tracking Topic Hierarchies in a Text Stream

Ding Tu [a], Ling Chen [a, b, *], Mingqi Lv [c], Hongyu Shi [a], Gencai Chen [a]

[a] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

[b] Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China

[c] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

Abstract

Discovering and tracking topics in a text stream has attracted the interests of many researchers. A limitation of most existing methods is that they organize topics in flat structures. Topic hierarchy could reveal the potential relations between topics, which can help to find high quality topics when analyzing the text stream. In this paper, a hierarchical online non-negative matrix factorization method (HONMF) is proposed to generate topic hierarchies from text streams. The proposed method can dynamically adjust the topic hierarchy to adapt to the emerging, evolving, and fading processes of the topics. In the experiment, HONMF is evaluated under a variety of metrics. Compared with the baseline methods, our method can achieve better performance with competitive time efficiency.

Index Terms—Topic modeling; Hierarchical matrix factorization; Online learning

1 Introduction

Over the past decade, the rapid development of Internet applications brings large volume of textual data, e.g., news, customer reviews, and blogs. To analyze such a large amount of textual data, it is necessary to automatically obtain an appropriate representation to get deep insight of the data. In text mining research area, this task is termed as topic detecting and tracking (TDT). The typical TDT scenario is to find out latent topics while the system receives documents continuously [1], [2], [3], [4]. These latent topics could be detected as new topics (i.e., topic detecting) or evolved from existing topics (i.e., topic tracking).

In general, the TDT task faces three challenges: 1) How to find out the appropriate latent topics from the documents of the current time slot? 2) How to track the evolving processes of the existing

[*] Corresponding author. Tel.: + 86 13606527774.
E-mail address: lingchen@zju.edu.cn (L. Chen).

topics? 3) How to adapt to the life cycle of the latent topics (e.g., the emerging and fading processes of the topics)? In order to address these challenges, the TDT methods should work in a dynamic way, and several dynamic TDT methods have been proposed in the literatures from different perspectives, e.g., online probabilistic methods [1], [2], [31] and online matrix factorization methods [3], [4], [5]. The online probabilistic methods usually conduct online topic detecting by introducing the timestamp as a variable into the inference process. While in matrix factorization based topic detecting methods, the online process is accomplished by introducing constraints on topics from different time slots.

However, most existing TDT methods arrange topics in a flat structure. Treating the discovered topics equally ignores the potential relations between them, and thus it limits the representation ability of these methods. The hierarchical structure has been proven to be a better description than the flat structure, e.g., the topics are organized in a tree structure in the widely-used text corpora RCV1 [6] and 20NewsGroup [7]. Thus, it would be beneficial to detect and track topic hierarchies in a text stream, e.g., the hierarchical structure could be used to reduce the topic overlaps in the topic evolving process. Another challenge is that the relations between topics are not fixed in a text stream, as the relative closeness between the existing topics may change in a text stream.

Aiming at these problems, we propose a hierarchical online non-negative matrix factorization method (HONMF). Unlike the existing online topic detecting methods, the generated topics in HONMF are organized in a hierarchical structure. In addition, the method can track the evolving process of the topic hierarchy, which is accomplished by adaptively adding emerging topics and removing fading topics. Our contributions are as follows:

1) Propose an efficient hierarchical NMF framework to detect and track latent topics in a text stream.

2) Propose mechanisms to detect and track the evolving process of the topic hierarchy in a text stream, e.g., the emergence and elimination of the topics.

3) Evaluate the performance of the proposed method by comparing it with many baseline methods under several metrics, e.g., topic quality, topic smoothness, and time efficiency.

The rest of the paper is organized as follows: Section 2 introduces the related work of the paper. Section 3 describes the process of our approach, including the overall framework and the details. Section 4 gives the experiments and the results are discussed. Conclusions and future work are presented in Section 5.

2 Related Work

Topic detection has attracted the interests of many researchers. There have been a large amount of methods addressing this problem. These methods could be divided into two categories, i.e., probabilistic methods and non-probabilistic methods. Probabilistic methods [8], [9], [10], [11], [33] usually model topics as latent factors, and assume that the joint probability of the words and the documents could be described by the mixture of the conditional probabilities over the latent factors. On the contrary, non-probabilistic methods usually use NMF [12], [13], [32] and dictionary learning [14] to uncover the low-rank structures using matrix factorization.

All the above mentioned methods are static topic detection methods and cannot handle the topic evolving process in the temporal dimension. Thus, various extensions of these methods have been proposed to handle this issue. The extensions can be divided into two categories according to the base models they utilized:

The first category of methods uses probabilistic models, e.g., LDA [8] and HDP [9], as their base models. For example, TOT (topic over time) model [15] captures how a topic changes over time, where each topic is associated with a continuous distribution, and the distribution over a topic is influenced by the timestamp of documents. In [16], a continuous dynamic topic model is proposed by using Brownian motion to model topic evolution through time with arbitrary granularity. In [1], an online variation of Bayes algorithm is proposed for LDA, which is based on online stochastic optimization with a natural gradient step. In [2], an online variation of HDP with a new coordinate-ascent variational inference algorithm is presented. In [28], [29], [30], HDP is used to discover topics for each time slot, and the evolving of the topics are modeled by exploiting the similarity between the topics of adjacent time slots. Compared to LDA, HDP does not need to predefine topic number. However, these methods organize topics in flat structures.

The second category of methods extends matrix factorization based methods to find the topics in the text stream. In [17], an invertible matrix is used to represent the transition relations between the old topics and the new topics. Constraints are added on the matrix to find the optimal topics. Similarly, in [5], a transition matrix is used to represent the relations between the topics. However, there are two differences: 1) In [5], the l1 regularization of the transition matrix is integrated into the objective function, while in [17], it is only exploited to constrain the solution space; 2) In [5], the l1 regularization of the topic matrix is used to get sparse representations, while in [17], more strict orthogonality constraints are utilized to get coherent topics. Unlike these methods, in [4], a box constraint is used to eliminate the difference between the topics found in different time slots, and a time regularization factor is also added to penalize static topics. In [3], the task is solved from a different perspective, which finds the topics that fit the present data and the decomposition results of the past data. However, the topics are also organized in flat structures in these methods.

Another research area closely related to our work is the hierarchical topic modeling. Hierarchical latent representation learning and deep learning architectures have been exploited as a natural structure in several applications, especially for modeling textual data. According to the base models utilized, the hierarchical topic models can also be divided into two categories: hierarchical probabilistic topic models and hierarchical matrix factorization topic models.

Methods proposed in [18], [19], [20], [21] belong to the hierarchical probabilistic topic models. In [18], the original LDA is extended to HLDA by combining the nested Chinese restaurant process (nCRP). In HLDA, a document is generated by choosing a path from the root to a leaf and sampling topics along the path. Several extensions of HLDA have been proposed in the latter researches. In [19], HLDA is extended to a nonparametric topic-model tree to represent human choices by developing a new stick-breaking model. Recently, a hierarchical extension of HDP [20] is also presented to overcome the drawback of HLDA, which can select multiple paths for a document in the topic hierarchy.

Hierarchical matrix factorization based topic models are also proposed by many researchers [21], [22], [23]. In [21], a tree-structured sparse regularization is presented to learn dictionaries embedded in a topic hierarchy, which can be computed by a finite number of proximal operators. To overcome the drawbacks of convex NMF, in [22], a hierarchical convex NMF that can automatically adapt to the internal structures of a data set is proposed, which hence yields meaningful and interpretable clusters for non-convex data sets. In [23], an efficient hierarchical document clustering method is presented to reveal the hierarchical relations between the documents, which is based on a new algorithm for rank-2 NMF.

We summarize following findings from the existing work. First, compared to the online methods based on matrix factorization, the online methods based on probabilistic models usually have higher computational cost, which makes them not suitable for large amounts of documents arriving in real time [5]. Second, most online topic models discover latent topics with fixed topic numbers, which is not appropriate in many scenarios. Third, most online topic models generate topics with flat structures, which may increase the topic overlaps, as all topics evolve with an arriving document in this case. The hierarchical structure can be exploited to adaptively determine the topic numbers, as well as to reduce the topic overlaps in the topic evolving process, as the hierarchical structure can ensure that the topics evolve with the most related documents.

3 Hierarchical Online NMF for Topic Detecting

3.1 Topic Detecting using NMF

In vector space model, a corpus is represented by an $m \times n$ matrix $\boldsymbol{X}$, where m is the vocabulary size, and n is the number of documents. A common assumption of topic modeling is that a latent topic can be represented as a distribution over the words. Then, a topic is a vector $\boldsymbol{w}$ in $\mathbf{R}^m$, and an $m \times k$ topic-word matrix $\boldsymbol{W}$ can be obtained by vertically combining k topics. With $\boldsymbol{W}$, a document can be seen as a distribution over the k topics, which can be represented as a $k \times 1$ vector $\boldsymbol{h}$. Since using limited topics (usually $k << n$) to precisely fit all documents is impossible, it is common to use $\boldsymbol{WH}$ to approximate the document matrix. $\boldsymbol{H}$ is the topic-document matrix, where each column contains the topic distribution of a document. Good $\boldsymbol{W}$ and $\boldsymbol{H}$ could ensure that the difference between $\boldsymbol{WH}$ and the original document matrix $\boldsymbol{X}$ is small.

In the case of text streams, the documents arrive continuously. The document matrix $\boldsymbol{X}$ consists of document matrices from different time slots. In time slot t, suppose the current $m \times n$ document matrix is $\boldsymbol{X}^t$, then the NMF methods detect topics as follows: Given a topic number k, it tries to find an $m \times k$ topic-word matrix $\boldsymbol{W}^t$ and a $k \times n$ topic-document matrix $\boldsymbol{H}^t$, which satisfy the following objective function:

$$\underset{\boldsymbol{W}^t, \boldsymbol{H}^t}{\operatorname{argmin}} \left\| \boldsymbol{X}^t - \boldsymbol{W}^t \boldsymbol{H}^t \right\|_2^2 \quad \text{s.t. } \boldsymbol{W}^t, \boldsymbol{H}^t \geq 0, \tag{1}$$

3.2 The Online NMF

The proposed HONMF is based on ONMF [3], which should satisfy the following objective function:

$$\underset{\boldsymbol{W}^t\boldsymbol{H}^t}{\operatorname{argmin}} \left\| \boldsymbol{X}^t - \boldsymbol{W}^t\boldsymbol{H}^t \right\|_2^2 + \lambda \left\| \boldsymbol{H}^t \right\|_1 \quad \text{s.t. } \boldsymbol{W}^t, \boldsymbol{H}^t \geq 0. \tag{2}$$

By adding l1-norm regularization on $\boldsymbol{H}^t$, the objective function can increase the sparsity of the result, i.e., the topic distribution of a document concentrates on a small number of topics. This is beneficial to the hierarchical topic modeling, as the hierarchical partition methods usually assume that the data of two sub-nodes have non-overlap or little overlap.

A common feature of online NMF methods is that they involve the past data when solving $\boldsymbol{W}^t$. To join the past information in the present decomposition process, the objective function is modified as:

$$\underset{\boldsymbol{W}^t,\boldsymbol{H}^t}{\operatorname{argmin}} \sum_{j=1}^{t} \sigma(t, j) \left(\left\| \boldsymbol{X}^j - \boldsymbol{W}^t\boldsymbol{H}^j \right\|_2^2 + \lambda \left\| \boldsymbol{H}^j \right\|_1 \right) \quad \text{s.t. } \boldsymbol{W}^t, \boldsymbol{H}^t \geq 0, \tag{3}$$

where t indicates the current time slot. $\boldsymbol{H}^j$ with $j<t$ is the topic-document matrix in the past time slot j. This objective function guarantees the found topic matrix $\boldsymbol{W}^t$ fits both the current and past data, which can lead to smooth transition between topic-word matrices from adjacent time slots. $\sigma(t, j)$ is the time scaling function, which adjusts the weight of $\boldsymbol{H}^j$ according to the time interval between t and j. Usually, it is an exponential function and produces a smaller value with a larger interval, e.g., $\sigma(t, j) = \alpha^{t-j}$ $\alpha \leq 1$. Older data usually gets smaller weight.

Eq. 3 can be solved by using the optimizing rules in [3], which are presented in Table 1. After the optimizing process of $\boldsymbol{W}^t$, $\boldsymbol{H}^t$ can be got using LARS-Lasso [3]. The detailed convergence proof of the algorithm can be found in [3]. Two auxiliary matrices are used to store the intermediate results of ONMF. The auxiliary matrix $\boldsymbol{A}$ stores the information related to the document-specific topic distribution vectors in the past time slots, and the auxiliary matrix $\boldsymbol{B}$ stores the information related to the documents and the document-specific topic distribution vectors in the past time slots, which can be found in line 4 of Algorithm 1. β is used to replace the time scaling function, which has a value between 0 and 1. When a new document arrives, the weights of all past documents would be scaled by β. In ONMF, $\boldsymbol{W}^0$ is randomly initialized and $\boldsymbol{A}^0$, $\boldsymbol{B}^0$ are initialized with $\boldsymbol{A}^0=\varepsilon\boldsymbol{I}$, $\boldsymbol{B}^0=\varepsilon\boldsymbol{W}^0$, where ε is a very small value and $\boldsymbol{I}$ is a $k\times k$ identity matrix.

TABLE 1 OPTIMIZING RULES OF ONMF

Algorithm1 Optimizing rules for ONMF

Input: Document matrix $\boldsymbol{X}^t=[\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n]$, topic-word matrix of the last time slot $\boldsymbol{W}^{t-1}$, l1-norm regularization coefficient λ, past information of related node $\boldsymbol{A}$, $\boldsymbol{B}$, time scaling parameter β.

```
1.   $\boldsymbol{W} \leftarrow \boldsymbol{W}^{t-1}$ For $j$=1 to $n$
```

```
Computing $\boldsymbol{h}_j$ using LARS-Lasso with $\boldsymbol{h}_j = \arg\min_{\boldsymbol{h}} \|\boldsymbol{x}_j - \boldsymbol{W}\boldsymbol{h}\|_{fro}^2 + \lambda\|\boldsymbol{h}\|_1$   s.t. $\boldsymbol{h} \geq 0$
$\boldsymbol{A} \leftarrow \beta\boldsymbol{A} + \boldsymbol{h}_j\boldsymbol{h}_j^T$, $\boldsymbol{B} \leftarrow \beta\boldsymbol{B} + \boldsymbol{x}_j\boldsymbol{h}_j^T$
$\boldsymbol{A}=[\boldsymbol{a}_1, \boldsymbol{a}_2,\ldots, \boldsymbol{a}_k]$, $\boldsymbol{B}=[\boldsymbol{b}_1, \boldsymbol{b}_2,\ldots, \boldsymbol{b}_k]$, $\boldsymbol{W}^i=[\boldsymbol{w}_1, \boldsymbol{w}_2,\ldots, \boldsymbol{w}_k]$
While not converge
  For $c$=1 to $k$
    $\boldsymbol{w}_c \leftarrow (\boldsymbol{b}_c - \boldsymbol{W}\boldsymbol{a}_c) / \boldsymbol{A}[c,c] + \boldsymbol{w}_c$
    $\boldsymbol{w}_c \leftarrow \boldsymbol{w}_c / \max(\|\boldsymbol{w}_c\|_2, 1)$
  End For
End While
End For
```

3.3 Framework Overview

The key difference between ONMF and HONMF is that HONMF decomposes documents level by level in a topic hierarchy. The topics in the topic hierarchy are arranged with a decreasing topic area from up to bottom. The advantage of detecting topics in a hierarchical way is two-fold: First, it can reduce the overlaps between different topics and lead to more separable topics, which has been proven in [20]. Second, in an online topic model, the word distributions of the topics are evolving continuously, and thus the overlaps between the topics may become larger. By organizing topics in a hierarchy, the impact of documents from other topics is controlled, thus more consistent topics would be generated.

The overall process of using HONMF to detect topics in a text stream is shown in Fig. 1, where $\boldsymbol{W}_i^t$ stands for the topic matrix of the ith topic node in the hierarchy at time slot t. When the timestamp is not useful (e.g., discussing topics in the same time slot), it would be omitted for convenience in the rest of the paper.

At the initial stage of HONMF (i.e., t=1), a hierarchical ONMF process is performed and the initial topic hierarchy is generated as follows. First, an ONMF process is performed at the root node, where the topic number k_{sub} is determined by a fast density-based clustering process, and the result is $\boldsymbol{W}_{\text{root}}$ and $\boldsymbol{H}_{\text{root}}$. Second, every detected topic is regarded as a sub-topic node of the root node and is represented by a column vector $\boldsymbol{w}$ of $\boldsymbol{W}_{\text{root}}$. Then, the documents are assigned to the most related sub-topic nodes according to $\boldsymbol{H}_{\text{root}}$. Suppose the topic distribution vector of document $\boldsymbol{x}_i$ is $\boldsymbol{h}_i$, $\boldsymbol{x}_i$ would be assigned to the topic node with the largest proportion in $\boldsymbol{h}_i$. Third, ONMF processes are performed on the sub-topics, and the decomposition results are stored. Then, the separability of every leaf topic node $\boldsymbol{w}$ is calculated, which is defined as the diversity between the sub-topics of $\boldsymbol{w}$. Fourth, the leaf node $\boldsymbol{w}_i$ with the largest separability is chosen. The same document splitting processes are performed on $\boldsymbol{w}_i$ according to $\boldsymbol{W}_i$ and $\boldsymbol{H}_i$. Then, the sub-topics of $\boldsymbol{w}_i$ would become new leaf topic nodes. Fifth, the whole process iterates until some stopping criteria are met. In the construction process of the topic hierarchy, the past decomposition results of all topic nodes are stored (e.g., the auxiliary matrices $\boldsymbol{A}$ and $\boldsymbol{B}$). Then, it goes to the next time slot.

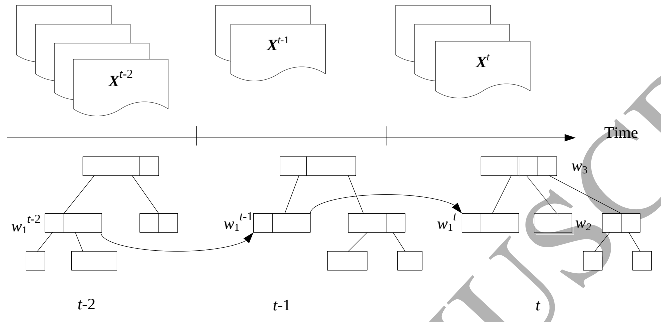

Fig.1. The working process of HONMF.

In the next time slot, the topic hierarchy is evolving with the newly arriving data in an online way. Similar hierarchical ONMF process is also conducted in an up to bottom way. The topics are either evolving from existing topics or added as emerging topics. Take the decomposition process of $\boldsymbol{w}_1$ as an example, we firstly detect novel documents, which are defined as documents that cannot be recovered by $\boldsymbol{W}_1^{t\text{-}1}$ with sparse constraints. Few novel documents indicate that the topic transition in adjacent time slots is smooth. Then, $\boldsymbol{w}_1^t$ is decomposed by directly joining its past information. For example, in Fig. 1, $\boldsymbol{w}_1^{t\text{-}2}$ and $\boldsymbol{w}_1^{t\text{-}1}$ are the same topic as $\boldsymbol{w}_1^t$ in the past time slots, and the decomposition of $\boldsymbol{w}_1^{t\text{-}1}$ is performed by considering the past information of $\boldsymbol{w}_1^{t\text{-}2}$ (similar relation exists between $\boldsymbol{w}_1^t$ and $\boldsymbol{w}_1^{t\text{-}1}$), as shown in Table 1. This process can be seen as that $\boldsymbol{w}_1^{t\text{-}2}$ is evolving to $\boldsymbol{w}_1^{t\text{-}1}$ and $\boldsymbol{w}_1^t$ in a text stream. Otherwise, if the number of novel documents is large, it is considered that there exist emerging topics. New topic nodes are added as sub-topic nodes of the current node (e.g., $\boldsymbol{w}_2$ is an emerging topic of $\boldsymbol{w}_3$). However, ONMF cannot work in this situation, so we propose a mechanism to modify the initial status of the current node to adapt to the emerging topics in ONMF (in Section 3.4.2). After the decomposition of the current node, the same node selection and decomposition processes are iterated. In the rest part of this paper, only the bottom nodes among the activated nodes are considered as leaf nodes.

Besides the emerging topics, existing topics may disappear in the evolving process. Nodes with few documents are regarded as inactivated topics. If these topics are inactivated for several time slots, the topic nodes would be removed from the topic hierarchy. The activation of these topics in early time slots can be viewed as the fading of existing topics. Removing inactivated topic nodes can speed up the whole decomposition process. A heartbeat mechanism is introduced to record the time period that the inactivated state of a topic lasts. If the time period is larger than a predefined threshold θ_a, they would be removed, and the corresponding topic matrices would be updated. Through this, our method can track the evolving of the topic hierarchy.

3.4 Method Details

In this section, we discuss the details of our method, including: 1) Exploit a fast density-based clustering process to adaptively determine the sub-topic number; 2) Find the emerging topics from novel documents; 3) Control the hierarchical ONMF process according to the separabilities and bandwidths of the topics.

3.4.1 Determining Topic Number

Existing hierarchical NMF methods usually produce binary trees or *k*-branch trees, i.e., every internal topic node has a fixed number of children. Setting k_{sub}=2 would lead to a deep topic hierarchy, which is not appropriate in many applications. Setting k_{sub}>2 may not suitable for all topic nodes. For example, it would split the high coherent topics and introduce abundant topics when the tree goes deeper. In HONMF, a clustering process analogous to mean shift clustering [24] is used to adaptively determine the number of sub-topic nodes in the initial stage.

Mean shift clustering is based on the mean shift process, which aims to find the modes of local density by moving a point along the mean shift vector. Given a data point $\boldsymbol{x}$ and a predefined bandwidth b, the mean shift vector $m(\boldsymbol{x})$ can be constructed as:

$$m(\boldsymbol{x}) = \sum_{i=1}^{n} \boldsymbol{x}_i g(\|\boldsymbol{x} - \boldsymbol{x}_i / b\|^2) / \sum_{i=1}^{n} g(\|\boldsymbol{x} - \boldsymbol{x}_i / b\|^2) - \boldsymbol{x}, \tag{4}$$

where $\boldsymbol{x}_i$ enumerates all the data points. The function $g(x)$ used in our method is a uniform kernel that satisfies:

$$g(x) = \begin{cases} 1 & x \le 1 \\ 0 & x > 1 \end{cases}. \tag{5}$$

In our method, a fast mean shift clustering (FMS) is used to determine the sub-topic number in the initial stage. To speed up the initial stage, the mean shift process is only performed for a set of seed documents in FMS. The first seed document is randomly selected. In the mean shift process of the seed document, all documents in the bandwidth are assigned to the same cluster as seed documents. Then, another seed document is selected from the documents that have not been assigned to any cluster. This process ends until all documents have been assigned, and the similar cluster centers are merged. Finally, the topic number k_{sub} is set as the number of cluster centers. The cosine distance is chosen to compute the difference between documents in FMS [25].

For the root node, the bandwidth of FMS is set as a large value (e.g., 0.95). The k_{sub} cluster centers of FMS are considered as the initial sub-topic vectors. If k_{sub}= 1, the bandwidth decreases with a fixed step φ, and the same FMS process is performed until k_{sub}> 1. Then, the ONMF processes are performed based on the initial k_{sub} topics. Since ONMF updates topic vectors with large steps at early iterations, the initial status should be modified to avoid that the final topic vectors deviate from the initial topic vectors too much, which can be achieved by increasing the weights of the initial topics. Therefore, the auxiliary matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are initialized as $\boldsymbol{A}=t_0\boldsymbol{I}$ and $\boldsymbol{B}=t_0\boldsymbol{D}_0$ to force the decomposition results to stay close to the initial topic vectors, where t_0 is a constant and its value is set according to the actual data set (e.g., in the experiment it is 0.1, which is close to the weight that 100 documents has been processed) and $\boldsymbol{D}_0$ is the initial topic-word matrix generated by FMS.

For every topic node, a corresponding bandwidth is recorded, which is the bandwidth when it first gets its sub-nodes. The initial bandwidth of a child node is set as b_p-φ, where b_p is the bandwidth of the parent node and φ is the bandwidth step size. With this setting, our method can guarantee that higher level topics are broader and lower level topics are narrower. If the topic node can only be decomposed into one cluster center at the current bandwidth, the bandwidth decreases with φ. In addition, a lower bound θ_{lb} is used for the bandwidth to prevent the method to find too narrow topics. The values of φ and θ_{lb} depend on the requirements and the characteristics of the text stream. Smaller step size leads to a deeper topic hierarchy and vice versa. Smaller θ_{lb} leads to narrower leaf topic nodes and vice versa.

Note that FMS sometimes produces very small clusters, which may be outliers or small topics that are not suitable for the current bandwidth. To prevent this, the document proportion of a topic should be larger than a topic size threshold θ_t (e.g., 5% of all documents in the parent topic node) and larger than 5 documents. The small clusters are merged with the most similar clusters. If it is a small topic, it may be split from another topic at an appropriate bandwidth.

3.4.2 Finding Emerging Topics

ONMF assumes that the word distribution of each topic is evolved between adjacent time slots with a fixed topic number. However, this assumption is not appropriate in many scenarios, as there exist topics that never appear in the past time slots, i.e., the emerging topics.

For a topic node, the documents that cannot be described by existing topics are called novel documents. The novel documents are defined as documents that satisfy the following conditions:

$$\boldsymbol{h}_i = \arg\min_{\boldsymbol{h}} \left\| \boldsymbol{x}_i \text{ - } \boldsymbol{W}^{t-1}\boldsymbol{h} \right\|_2^2 + \lambda \left\| \boldsymbol{h} \right\|_1 \quad \text{s.t. } \boldsymbol{h} \geq 0, \tag{6}$$

$$(\boldsymbol{x}_\mathrm{r})^\mathrm{T} \boldsymbol{x}_i / (\left\| \boldsymbol{x}_i \right\|_2 \left\| \boldsymbol{x}_i^r \right\|_2) < \theta \; \boldsymbol{x}_\mathrm{r} = \boldsymbol{W}^{t-1}\boldsymbol{h}_i, \tag{7}$$

where $\boldsymbol{x}_r$ is the recovered $\boldsymbol{x}_i$, which is got by multiplying $\boldsymbol{W}^{t\text{-}1}$ with $\boldsymbol{h}_i$. If the cosine similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_r$ is smaller than a novel document threshold θ_n, $\boldsymbol{x}_i$ is regarded as a novel document of the current topic node. The novel document threshold θ_n should be highly related to the granularity of the topic node. Therefore, we set θ_n= 1-b, where b is the bandwidth of the current node. As a result, θ_n of a higher level topic is smaller and vice versa.

The key characteristic of emerging topics is that there are a large amount of novel documents. Only a small number of novel documents is regarded as outliers and cannot be viewed as a clue of emerging topics. We assume that the number of novel documents should occupy a certain fraction of documents in the current topic node, i.e., the ratio between novel documents and the documents in the current topic node should be larger than a predefined emerging topic threshold θ_e (e.g., 20%). Note that an emerging topic is related to a certain topic bandwidth. If the bandwidth of the current topic node is large, the fraction of novel documents is usually small. The reason is two-fold: 1) Since a topic node with larger bandwidth usually has a smaller θ_n, the probability of a document to be judged as novel is lower; 2) The topic nodes with larger bandwidth usually are those near to the root node, the number of documents assigned to which is large.

In [26], a method is proposed to get emerging topics from novel documents. The limitation of the method is that the number of emerging topics should be predefined. Therefore, we also use FMS to determine the number of emerging topics: Suppose the document set of the current node is $\boldsymbol{X}$ and the document set of the found novel documents is $\boldsymbol{X}_{novel}$. The FMS process is conducted on $\boldsymbol{X}_{novel}$, and the bandwidth is set as the bandwidth of the current node. Then, the emerging topics are detected as follows. All candidate emerging topics (represented by document clusters) are ranked by their average similarities with the existing topics, and the one with the lowest similarity is selected as an emerging topic every time. Note that too small document clusters are not considered as candidate emerging topics (the same topic size threshold θ_t is used to measure whether a cluster is small). The selection process ends until the existing topics and the emerging topics contain more than 90% documents of $\boldsymbol{X}$. After that, the initial topic vectors of the emerging topics are computed as the mean vectors of documents belonging to them. To combine the existing topics $\boldsymbol{W}$ and the emerging topics $\boldsymbol{W}_e$, the initial state of the decomposition process is modified as:

$$\hat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{\Sigma}_1 \\ \boldsymbol{\Sigma}_2 & t_0\boldsymbol{I} \end{bmatrix}, \qquad \hat{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B} & t_0\boldsymbol{W}_{\mathrm{e}} \end{bmatrix}, \qquad \hat{W} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{W}_{\mathrm{e}} \end{bmatrix}, \tag{8}$$

where k_e is the number of the selected emerging topics, $\boldsymbol{I}$ is a $k_e \times k_e$ identity matrix, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are matrices with all elements equal to ε, t_0 is a constant to adjust the weights of the emerging topics to prevent that the decomposition result deviates too much from the initial topics.

3.4.3 Split Node Selection and Stop Criterion

The main idea of HONMF is to reduce the topic overlaps and control the topic consistency in the topic evolving process. In each time slot, a topic hierarchy is generated based on a tree structure and a top-down level-by-level factorization strategy. In each level, the topic granularity is controlled by the topic bandwidth, and the documents are assigned to the most relevant topic node.

In every iteration of the HONMF process, it chooses the leaf node with the largest separability and topic bandwidth to be split. The inverse of the average pairwise cosine similarity between the sub-topic nodes is exploited as node separability. Suppose the topic matrix of the activated topics in the current node is $\boldsymbol{W}_a$, the separability is defined as:

$$Separ(\boldsymbol{W}_{\mathrm{a}}) = (n(n-1)) \ / (\sum_{i,j}((\boldsymbol{w}_j{}^{\mathrm{T}}\boldsymbol{w}_i) / (\|\boldsymbol{w}_i\|_2 \|\boldsymbol{w}_j\|_2))) \ \ \boldsymbol{w}_i, \boldsymbol{w}_j \in \boldsymbol{W}_{\mathrm{a}} \ , \tag{9}$$

where n is the number of the activated sub-topic nodes. If a topic node is assigned with few documents in the current time slot (the assigned documents are less 1% of the parent topic node or less than 5), it is judged as an inactivated topic. A large separability indicates that the sub-topics differ much. The separability of the current topic node would be -1 in the following two conditions: 1) The document number of the current topic node is smaller than a constant number; 2) It cannot be split to at least two sub-topics with the predefined smallest topic bandwidth.

The leaf nodes with the largest bandwidth are selected as the candidates to be split. The reason is that the result topic hierarchy should be organized from top to bottom with decreasing topic granularity and balanced height. If only considering the separability, the topic granularity of the leaf nodes would differ much.

To get the bandwidth and separability, every leaf node would be performed a trial ONMF process at first. If a leaf node already has sub-topics in the topic hierarchy, its bandwidth is the original bandwidth. If not, the topic bandwidth is the maximum bandwidth, with which the topic can be split into more than one sub-topic in the trial ONMF process. Otherwise, the bandwidth would be set as θ_{lb}. The candidate with the largest separability is chosen as the next topic node to be split.

If the separabilities of all candidates are -1, the leaf nodes with the second largest bandwidth are selected as candidates. The HONMF process ends until no leaf node can be further split or the number of leaf nodes reaches a predefined number k. The pseudocode of HONMF in the evolving stage is presented in Table 2.

TABLE 2 HIERARCHICAL ONLINE NMF

Algorithm2Hierarchical online NMF

```
Input: document matrix $\boldsymbol{X}^t$=[$\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$,…,$\boldsymbol{x}_n$], topic hierarchy of the last time slot $\boldsymbol{T}^{t-1}$, l1-norm regularization coefficient $\lambda$.
Initialize the set of leaf nodes $S_l = \varnothing$, $\boldsymbol{T}^t \leftarrow \boldsymbol{T}^{t-1}$
For the root node $\boldsymbol{w}_{root}$, find $\boldsymbol{H} = \arg\min_{\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{WH}\|_2^2 + \lambda\|\boldsymbol{H}\|_1 \quad s.t.\boldsymbol{H} \geq 0$
Find novel documents that satisfy $(\boldsymbol{x}_r)^T \boldsymbol{x}_i / (\|\boldsymbol{x}_i\|_2 \|\boldsymbol{x}_i^r\|_2) < \theta$ $\boldsymbol{x}_r = \boldsymbol{W}^{t-1}\boldsymbol{h}_i$
Use Eq. 4 to detect emerging topics. If emerging topics $\boldsymbol{W}_e$ are found, update $\boldsymbol{W}$, $\boldsymbol{A}$, and $\boldsymbol{B}$ using Eq. 8. Add emerging topic nodes in $\boldsymbol{T}^t$.
[$\boldsymbol{W}$, $\boldsymbol{H}$]=ONMF($\boldsymbol{X}^t$, $\lambda$, $\boldsymbol{A}$, $\boldsymbol{B}$);
Split documents to the related topics according to $\boldsymbol{H}$, find the activated topics $\boldsymbol{W}_a$.
Compute the separability of $\boldsymbol{w}_{root}$.
    $Separ(\boldsymbol{W}_a) = (n(n-1)) / (\sum_{i,j}((\boldsymbol{w}_j^T \boldsymbol{w}_i) / (\|\boldsymbol{w}_i\|_2 \|\boldsymbol{w}_j\|_2))) \quad \boldsymbol{w}_i, \boldsymbol{w}_j \in \boldsymbol{W}_a$ ,
$S_l$.add($\boldsymbol{w}_{root}$)
While num($S_l$)<$k$ and min($Separ$($\boldsymbol{w}_i$.$\boldsymbol{W}$))>0 ($\boldsymbol{w}_i \in S_l$)
    Find the next topic node $\boldsymbol{w}$ to be split.
        $maxband = \max(\boldsymbol{w}_i.b)$ s.t. $Separ(\boldsymbol{w}_i.\boldsymbol{W}) > 0$, $\boldsymbol{w}_i \in S_l$
```

```
         $\boldsymbol{w} = \arg\max_{\boldsymbol{w}_i}(Separ(\boldsymbol{w}_i.\boldsymbol{W}))$ s.t. $\boldsymbol{w}_i.b = maxband$, $\boldsymbol{w}_i \in S_l$
11.      $S_l$.remove($\boldsymbol{w}$)
12.      For each activated sub-topic of $\boldsymbol{w}$, do steps 2-8.
13.  **End While**
14.  Update the status of all inactivated nodes in $\boldsymbol{T}$
15.  Remove fading topics
```

3.4.4 Topic Tracking

The existing topics could be tracked by monitoring their evolving process. The evolving of an existing topic could be divided into the following three categories:

Stability of a topic: If the topic distribution of the text stream is relatively stable, the topic hierarchy would also be stable, and the positions of the topic nodes would maintain unchanged.

Upward movement of a topic: As shown in Fig. 2, when documents belonging to $\boldsymbol{w}_l$ in the past time slot are assigned to $\boldsymbol{w}_e$ in the current time slot and $\boldsymbol{w}_\mathrm{l}$ is assigned few documents in the current time slot, it could be regarded as $\boldsymbol{w}_l$ moves upward to the position of $\boldsymbol{w}_e$. This situation could be also viewed as that an emerging topic $\boldsymbol{w}_e$ is almost the same to a fading topic $\boldsymbol{w}_l$ in the lower level.

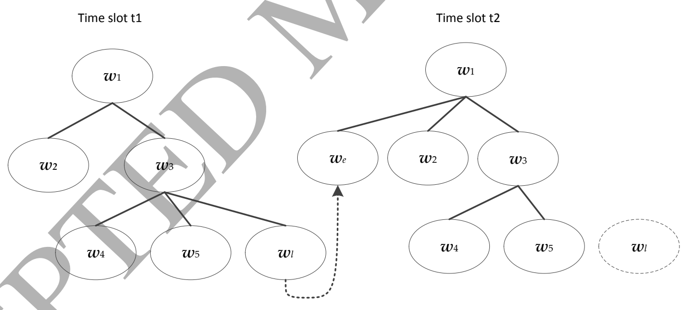

Fig. 2. Upward movement of topic.

Downward movement of a topic: The downward movement of a topic is usually caused by the expansion of the existing topic. Along with the topic evolving, the bandwidth of an existing narrow topic may become larger as more documents being assigned to it. Then, the existing narrow topic would be split as distinct sub-topics. For example, as shown in Fig. 3, suppose the existing narrow topic $\boldsymbol{w}_n$ is a leaf topic node and the broader topic evolving from $\boldsymbol{w}_n$ is $\boldsymbol{w}_b$. Since $\boldsymbol{w}_b$ is a broader topic, it would be split into several sub-topics. With an appropriate topic bandwidth, a sub-topic $\boldsymbol{w}_s$ would be almost the same as $\boldsymbol{w}_n$, and it can be regarded as $\boldsymbol{w}_n$ moves downward to the position of $\boldsymbol{w}_s$.

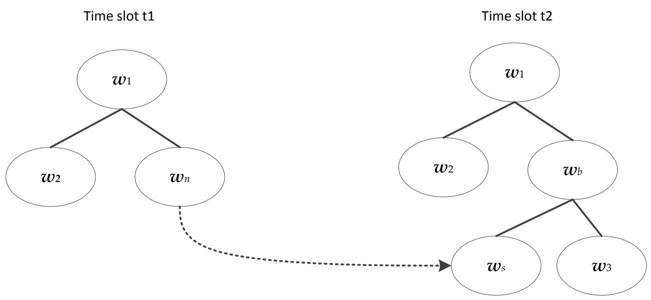

Fig.3. Downward movement of topic.

3.4.5 Time Complexity

The computation complexity of our method mainly consists of two parts: 1) The hierarchical ONMF process; 2) The FMS process. Suppose the dimension number (vocabulary size) is m, the document number is n, and the leaf topic number is k.

For the hierarchical ONMF process in the current time slot, the computation time is mainly consumed by the update steps in Table 1. The time complexity of LARS-Lasso in Table1 is O(m), and the time complexity of the rest part is O(mk^2). Then, the time complexity of processing a document in ONMF is O(mk^2). Suppose in HONMF, every internal topic has l sub-topic nodes in average and the final number of the leaf topics is k, the height of the final topic hierarchy would be about $\log_l k$. Then, every document would be processed $\log_l k$ times in the hierarchical ONMF process in average, and the total time consumed in this part is O($nml^2\log_l k$). Since the total time complexity of ONMF is O(nmk^2), the ratio between the two is $(l^2\log_l k)/k^2$. Suppose $\log_l k=a$, then $(l^2\log_l k)/k^2=a/l^{2a-2}$. Since l (the average sub-topic number of a topic) is larger than 2 and a>1, it is easy to see a/l^{2a-2}<1 if a>1, i.e., the time complexity of hierarchical ONMF process in HONMF is smaller than ONMF.

In the FMS process, the time complexity is O(mn) for a document. For a topic node, suppose the number of documents assigned to it is n_t, and the corresponding sub-topic number is l. Then, the FMS process would take nearly l turns, i.e., the time complexity is O(lmn_t). Since the document number of all topic nodes is $\log_l kn$ (every document has been processed $\log_l k$ times in average), the time complexity of FMS at the first time slot is O($nml\log_l k$), which is a bit smaller than that of the hierarchical ONMF process, i.e., O($nml^2\log_l k$). For the latter time slots (evolving stage), FMS is triggered only when a leaf node is further split or emerging topics are detected. Since the number of the documents involved in these cases is smaller than n, if the data distribution in the text stream is not skewed, the time consumed in this part should only occupy a small fraction of the whole computation time.

According to the analysis above, it can be seen that the time complexity of our method is O($nm(l^2+l)\log_l k$) in the first time slot and O($lmn_m+nml^2\log_l k$) in the latter time slots, where n_m is the number of the documents involved in the FMS process.

4 Experiments

In this section, the experimental settings are presented, and the effectiveness of the proposed method is evaluated under several metrics.

4.1 Dataset

Two datasets are used in our experiments. The first dataset is a subset of the Nist Topic Detection and Tracking corpus (TDT2) [27], which consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI), and 2 television programs (CNN, ABC). In this subset, those documents appearing in two or more categories are removed, and only the largest 30 categories are kept. Finally, 10212 documents and about 26000 distinct words are left in total. The second dataset is the 20Newsgroup dataset, which has 20 classes and about 20000 documents. A resampling procedure has been conducted to the original dataset. After that, 16 classes are selected in our experiment by reducing some broad topics (e.g., talk.politics.misc), and 3000 documents are left.

For every different setting of k, there are 50 randomly sampled document sets of TDT2. To meet the need of our experiments, all samples are reordered by timestamp (the original samples in the subset are sorted by class label), and the dimensions are reduced in the preprocessing step to improve the performance and reduce the execution time. For every sampled document set, the vocabulary size is reduced to 40% of the original size. The vocabulary size of a document set is 10000-20000 after excluding terms with 0 frequencies, and the top 40% of the terms are retained after sorting them by TF-IDFs. All documents are normalized by their lengths to eliminate the influence of document size.

4.2 Evaluation Metrics

In the experiments, four evaluation metrics are utilized to measure the performance:

Normalized Mutual Information (NMI): NMI is the most widely used metric to measure clustering quality. It is particularly useful when the number of generated clusters is different from that of ground truth classes, and it can be used to determine the optimal number of clusters. A larger NMI value between two partitions of a dataset indicates that the two partitions have larger similarity. The definition of NMI between two partitions *X*, *Y* is as follows:

$$NMI(X,Y) = 2I(X,Y)/(H(X)+H(Y)), \tag{10}$$

where *I*(*X*, *Y*) is the mutual information of *X* and *Y*, and *H*(*X*) is the information entropy of *X*.

Micro averaged F1 (MicroF1): It is a widely used evaluation metric in TDT literatures (e.g., [23]). Compared with NMI, MicroF1 can measure the performance of clustering methods more accurately when the class number is large. Assume that the generated result contains k_r clusters and the number of the labels is k_l. Then, for every ground truth cluster C_i, we can find a result cluster C_r, with which the overlap of the data points is maximal. MicroF1 is computed as follows:

$$P = \sum_{i=1}^{k_l} |C_i \cap C_r| / \sum_{i=1}^{k_l} |C_r|, (11)\ R = \sum_{i=1}^{k_l} |C_i \cap C_r| / \sum_{i=1}^{k_l} |C_i|, \tag{11}$$

$$MicroF1 = 2PR/(P+R). \tag{12}$$

Mean average precision (MAP): The MicroF1 metric evaluates clustering performance from a single point view. To evaluate the performance from a global view, MAP is a common choice, which is computed as follows:

$$MAP = \int_0^1 P(R)dR, \tag{13}$$

where *R*, *P* are recall and precision, respectively. The calculation of *R* and *P* is the same as MicroF1.

Normalized Discounted Cumulative Gain (NDCG): NDCG is used to evaluate the quality of the discovered topics. For every ground truth topic, the mean vector of all documents is generated. Then, the NDCG score is calculated between the mean vector and the topic whose overlap with this ground truth topic is maximal. Eq. 14 is the definition of NDCG:

$$DCG(n) = \sum_{j=1}^{m} (2^{r(j)}-1)/\log(1+j), \qquad NDCG = DCG/IDCG, \tag{14}$$

where r(*j*) is the relativeness between the detected and the ground truth topic distributions of the *j*th word. IDCG is the possible maximum value of DCG.

4.3 Compared Methods

In order to evaluate our method, seven baseline methods are compared in the experiments:

fix-NMF-l1: fix-NMF-l1 only performs NMF on the data of the first time slot and does not change the topic distribution in the latter time slots. The NMF algorithm is implemented with multiplicative-updates rules and l1-norm regularization [5].

t-NMF-l1: t-NMF-l1 uses the same NMF implementation as the fix-NMF-l1. The difference between t-NMF-l1 and fix-NMF-l1 is that in every time slot, t-NMF-l1 generates a new topic matrix and initializes with $\mathbf{W}^{t-1}$.

Hie-NMF-l1: Hie-Rank2 can cluster documents into hierarchies by repeatedly performing rank2-NMF [23]. The rank2-NMF is a fast active-set-type algorithm for NMF with k=2. To improve the performance, the original rank2-NMF is replaced with NMF-l1. Hie-NMF-l1 is performed in each time slot.

JPP: JPP is a time-based collective factorization algorithm for topic discovery [5]. It considers the difference between the previous decomposition and the current decomposition in the objective function to achieve online decomposition.

ONMF: ONMF [3] is an extension of an online dictionary learning method by adding positive constraints on the dictionary and the coefficients.

OLDA: Online LDA [1] is an online probabilistic topic model. It supports batch mode. In every time slot, it updates the parameters of the topic-word distributions and estimates the document-topic distributions for evaluation [1].

HLDA: HLDA [18] is a hierarchical probabilistic topic model. In HLDA, a document is generated by choosing a path from the root to a leaf and sampling topics along the path. In the experiment, every path is regarded as a topic in HLDA. Therefore, the documents are assigned to the corresponding leaf topics. HLDA is not an online method and is used as a static hierarchical probabilistic baseline. The result of HLDA is generated by using a sampled document set in a time.

HONMF: The proposed hierarchical online NMF method for detecting topic hierarchies in a text stream. It can track the evolving process of the topics and the topic hierarchy. When evaluate HONMF, only the activated leaf nodes are used.

The window size is 500 documents, which almost equals to the average document amount in one month in a sampled document set. For all methods using l1-norm regularizations, λ is 0.001, as they achieve the best results. The max iteration number is 1000 for all methods. The minimum number of documents in a cluster is 10 for HONMF. The time scaling parameter β in ONMF and HONMF is 0.9995, i.e., the impact of the (n-500)*th* document is down with a ratio around 0.8. The lower bound θ_{lb} of the topic bandwidth is 0.7. The bandwidth step size φ is 0.05. The rest parameters of JPP, HLDA, OLDA, and Hie-Rank2 are set and optimized under the directions of the original works. The parameters are kept as default value if there is no clear specification in the following experiments.

4.4 Topic Quality

The evaluation of topic quality is conducted on the sampled document sets with different topic numbers (*k*=5, 10, 15). Table 3 gives the detailed information about the data.

TABLE 3 STATISTICAL INFORMATION OF THE SAMPLED DATA

k	Document set number	Mean document number	Mean vocabulary size
5	50	1660	5110
10	50	2987	7537
15	50	4930	9863

The results of all the eight methods are reported in Fig. 4. fix-NMF-l1 performs the worst among the eight methods. The reason is that the word-topic distribution and the topic-document distribution are changing among the time slots, which reduces the effectiveness of using a fixed model to predict the topics hidden behind a document. t-NMF-l1 can be viewed as an online NMF model to some extent, as the topic-word matrices between adjacent time slots have some "coherence", which also can be seen in Section 4.5. However, the coherence between two topic-word matrices in adjacent time slots is not strictly defined in the objective function like other online methods, so it performs the worst among the five online methods.

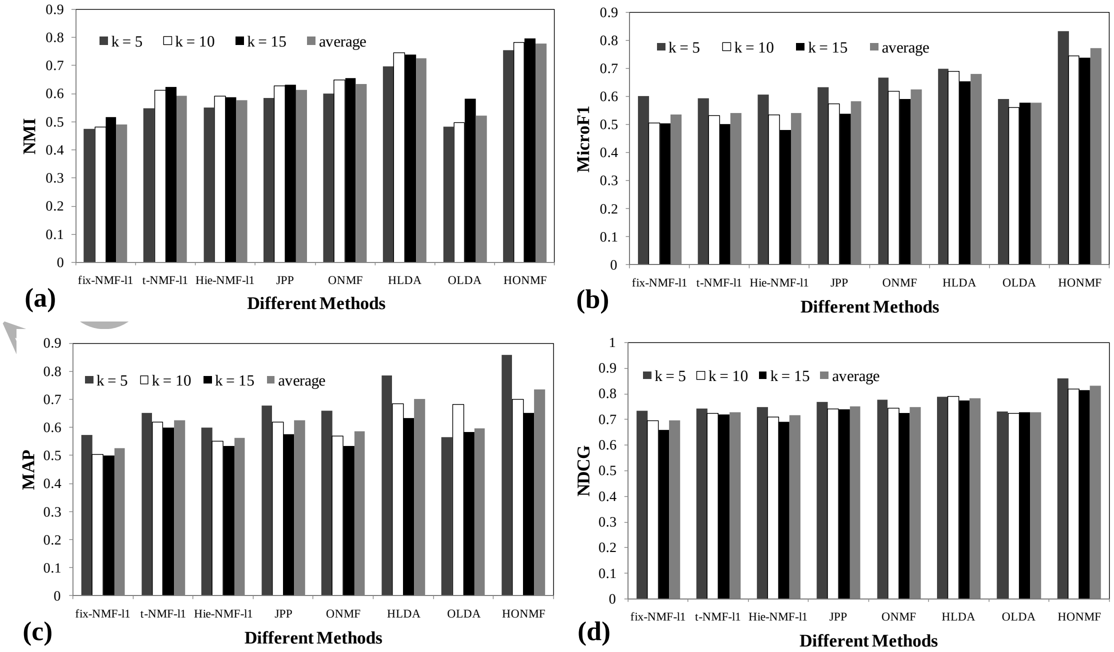

Fig.4. The comparison of different methods with respect to topic quality.

Among the remaining online methods, JPP and ONMF have similar results. The difference is that JPP has lower MicroF1 values, and ONMF has lower MAP values. The results of OLDA are not good in this experiment. We have checked the results of OLDA and found that the topic distributions of the documents are not as concentrated as other online methods. The reason might be that other online methods add the l1-norm regularization, while OLDA does not. Compared with these three methods, HONMF performs well under all the metrics. This can be interpreted as that HONMF produces much more stable results, i.e., the precision and recall values do not have large variations, as the main parts of the documents belonging to each topic are well apart.

Among the hierarchical NMF methods, HONMF outperforms Hie-NMF-l1. The reason might be that HONMF can utilize the past information, while Hie-NMF-l1 only uses information in the current time slot. For HONMF, if the l1-norm regularization is not used, there is a 0.1 reduction under several metrics. The reason is that the partition error would be enlarged during the hierarchical ONMF process, so the decomposition at each topic node should be as accurate as possible. As mentioned above, the l1-norm regularization can find more sparse representations, and the documents belonging to different topics can be better separated.

Among the probabilistic methods, HLDA performs better than OLDA. The reason can be two-fold: 1) HLDA uses the whole document set to train a model iteratively while OLDA does not; 2) HLDA generates more topics than OLDA, and the two methods have better performance with more topics [18]. The reason HLDA generates more topics is that HLDA does not provide a parameter to control topic number directly, and it tends to produce many small topics.

4.5 Topic Smoothness

In this experiment, we evaluate the smoothness between the two topic-word matrices generated in adjacent time slots. The smoothness between $\boldsymbol{W}^{t\text{-}1}$ and $\boldsymbol{W}^{t}$ can be defined as follows:

$$\boldsymbol{H}_{t-1}^{t} = \arg\min_{\boldsymbol{H}} \left\| \boldsymbol{X}^{t} - \boldsymbol{W}^{t-1}\boldsymbol{H} \right\|_2^2 + \lambda \left\| \boldsymbol{H} \right\|_1 , \tag{15}$$

$$\boldsymbol{H}_{t}^{t-1} = \arg\min_{\boldsymbol{H}} \left\| \boldsymbol{X}^{t-1} - \boldsymbol{W}^{t}\boldsymbol{H} \right\|_2^2 + \lambda \left\| \boldsymbol{H} \right\|_1 , \tag{16}$$

$$Smoothness = (NMI(C(\boldsymbol{H}_{t}^{t-1}), C(\boldsymbol{H}^{t-1})) + NMI(C(\boldsymbol{H}^{t}), C(\boldsymbol{H}_{t-1}^{t}))) / 2, \tag{17}$$

where $C(\boldsymbol{H})$ assigns documents to different topics according to the topic weights in $\boldsymbol{H}$. The hierarchical NMF methods (Hie-NMF-l1 and HONMF) only use leaf nodes to form a new topic matrix ***W***. Every document is assigned to the topic that has the highest correlation. Table 4 shows the topic smoothness of all methods. Since HLDA is a static method, it is not involved in this experiment.

TABLE 4 THE TOPIC SMOOTHNESS OF ALL METHODS

Metric	Method	5	10	15	average
Smoothness	**fix-NMF-l1**	1	1	1	1
	t-NMF-l1	0.554	0.484	0.541	0.526
	Hie-NMF-l1	0.392	0.430	0.462	0.428
	JPP	0.584	0.627	0.632	0.614
	ONMF	**0.707**	0.664	0.623	0.665
	OLDA	0.513	0.610	0.724	0.617
	HONMF	0.699	**0.708**	**0.746**	**0.718**

The smoothness of fix-NMF-l1 is 1, as the topic-word matrix is generated in the first time slot and does not change in the latter. HONMF gets the highest topic smoothness among all the left methods. The reason might be that HONMF generates more representative and strictly apart topics, which can separate the major proportions of different classes. However, other online methods may generate overlapped topics in the evolving process. The reason Hie-NMF-l1 performs poor might be that Hie-NMF-l1 is an offline method and it is randomly initialized in each time slot. In addition, it can be observed that the smoothness of OLDA increases with the increasing of topic number *k*. If there exist many emerging topics in the text stream and the quality of the detected topics is high, the smoothness of the topics would be reduced.

4.6 The Topic Hierarchy Evolving

In this part, an example of the topic hierarchy is presented. The experiment is conducted on the 20Newsgroup dataset, and the window size is 1000 documents. The topic number is 20. At the first time slot, there are documents from 8 topics. At the second time slot, documents from 4 new topics are added into the text stream. At the last time slot, 3 new topics are added. The documents in a window are randomly ordered. The standard topic hierarchy of this dataset is presented in Fig. 5, which is provided by the data contributor. For every ground truth leaf topic, the MicroF1 scores of all topics in the generated topic hierarchy are computed, and the one with the highest score is labeled as the same topic. The results of the three time slots are presented in Fig. 6. From up to down, the figures are the result topic hierarchies in time slots 1-3, respectively.

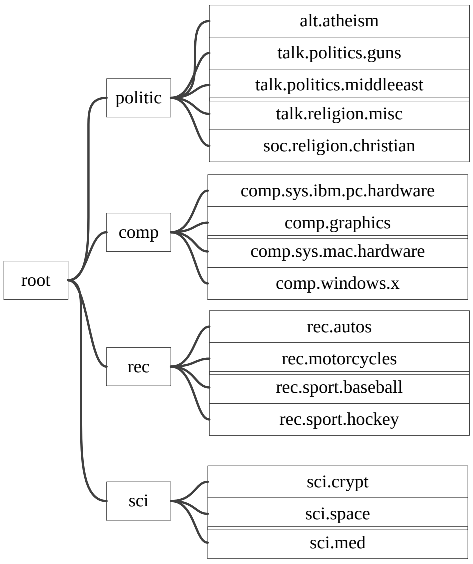

Fig. 5. The standard topic hierarchy of 20NewsGroup.

In Fig. 6, it can be seen that the generated topic hierarchy has high consistence with the golden standard in Fig. 5. In the first time slot, the "rec.motocycles" is separated from the two sport topics, as the otherness between them is significant at the initial topic bandwidth. In the second time slot, several downward movements of the topics can be observed, e.g., "alt.atheism" and "comp.sys.mac.hardware". In the third time slot, three new topics under the "sci" topic are correctly detected. The upward movement of the topic node cannot be seen in this experiment. However, we can imagine that if only a small number of documents of "talk.religion.misc" appear in the second time slot, these documents would be assigned to "alt.atheism" and further be assigned to a sub-topic of "alt.atheism". After several time slots, "alt.atheism" and "talk.religion.misc" have been found under the same parent topic (as shown in Fig. 6), and this can be regarded as that the sub-topic corresponding to "alt.atheism" performs an upward movement.

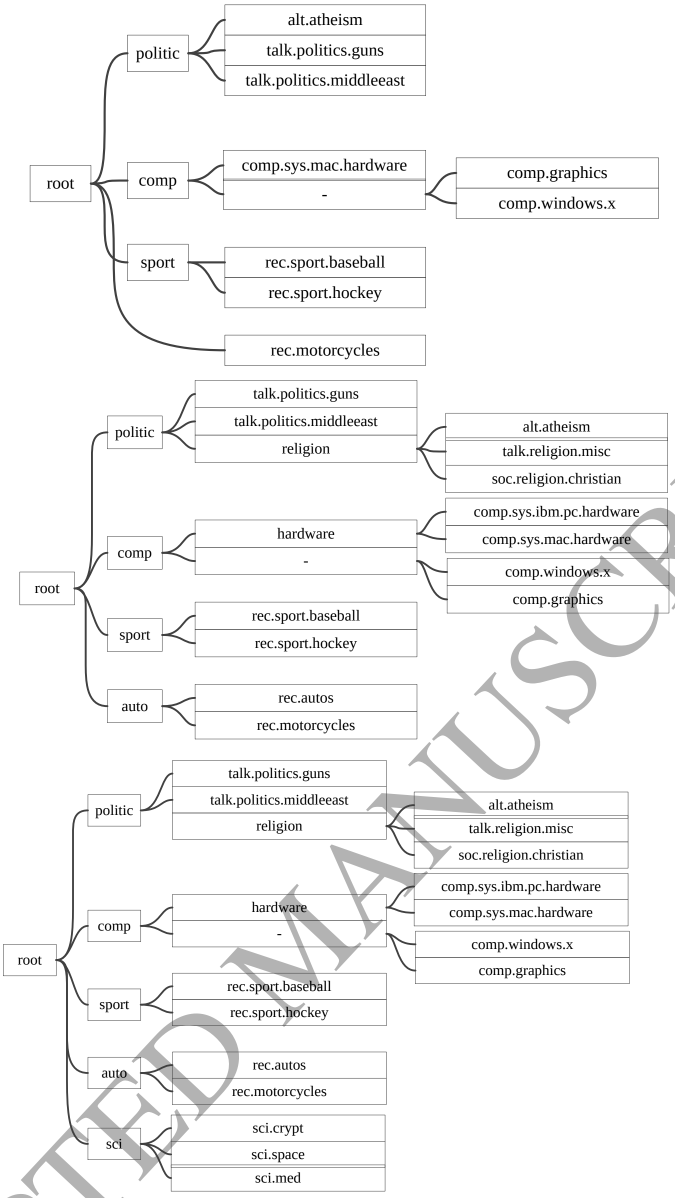

Fig. 6. The generated topic hierarchies.

4.7 Emerging Topic Detection

In this experiment, the performance of emerging topic detection is evaluated. The whole data of TDT2 is used, which is about 10000 documents with 30 classes. All documents are ordered by their timestamps. The window size is 1000 documents. The three baseline methods are ONMF, JPP, and OLDA. In every time slot, only classes that have more than 10 documents and never appear in the past time slots are regarded as the ground truth emerging topics. The ground truth emerging topics in the initial stage are not considered. Emerging documents are documents belonging to the emerging topics in the current time slot. MicroF1 is used as evaluation metric. Assume that the generated result contains k_r topics, and the number of ground truth emerging topics is k_l. For every ground truth emerging topic C_i, we can find a result topic C_r, with which the data overlap is maximal. Then, the MicroF1 of the ground truth emerging topics can be computed using Eq. 11 and 12. The data statistic of the text stream is represented in Table 5.

Table 6 is the MicroF1 scores of all the compared methods. It can be seen that our method outperforms all baseline methods. OLDA seems to be not sensitive to the ground truth emerging topics. JPP has better ability at finding the ground truth emerging topics than ONMF. The result of our method is also better than the result in [26] (0.54), despite the influence of document partition strategy. The reason HONMF performs well might be that the topics found in the last time slot have high accuracy, and the accuracy of novel document detection is relatively high.

TABLE 5 THE DATA STATISTIC INFORMATION

Time	Number of appeared topics	Number of emerging topics	Number of emerging documents
1	-	-	-
2	11	2	340
3	13	3	112
4	16	-	-
5	16	1	58
6	17	3	130
7	20	6	258
8	26	1	193
9	27	3	168
10	30	-	-

TABLE 6 MICROF1 OF ALL COMPARED METHODS

Time	HONMF	JPP	ONMF	OLDA
1	-	-	-	-
2	**0.966**	0.277	0.537	0.373
3	**0.930**	0.826	0.802	0.460
4	-	-	-	-
5	**0.928**	0.612	0.667	0.356
6	0.517	**0.570**	0.476	0.360
7	**0.650**	**0.796**	0.500	0.415
8	**0.682**	0.614	0.570	0.396
9	**0.912**	0.767	0.664	0.803
10	-	-	-	-
Avg	**0.798**	0.637	0.602	0.451

4.8 Time Efficiency

The execution time of different methods is evaluated on the sampled document sets of TDT2. The execution time of our method is recorded when the number of leaf nodes reaches k. The default experiment setting is used, i.e., document number n=2000, topic number k=10, and vocabulary size m=4000. The execution time is obtained as the mean value of 10 runs.

All methods decompose the data matrix in a whole, i.e., the window size is the same as n. We compare them under different settings of k, m, and n. The results are presented in Fig. 7. For all methods, only HONMF has different time complexities in the initial stage and the evolving stage. Therefore, HONMF-500 is compared in this experiment, which is HONMF with window size=500. The runtime of HLDA is between 1-4 hours in this experiment, so it is not involved in Fig. 7. The horizontal axes of Fig. 7(a) and Fig. 7(b) are the logarithmic values of the document number and the vocabulary size, respectively.

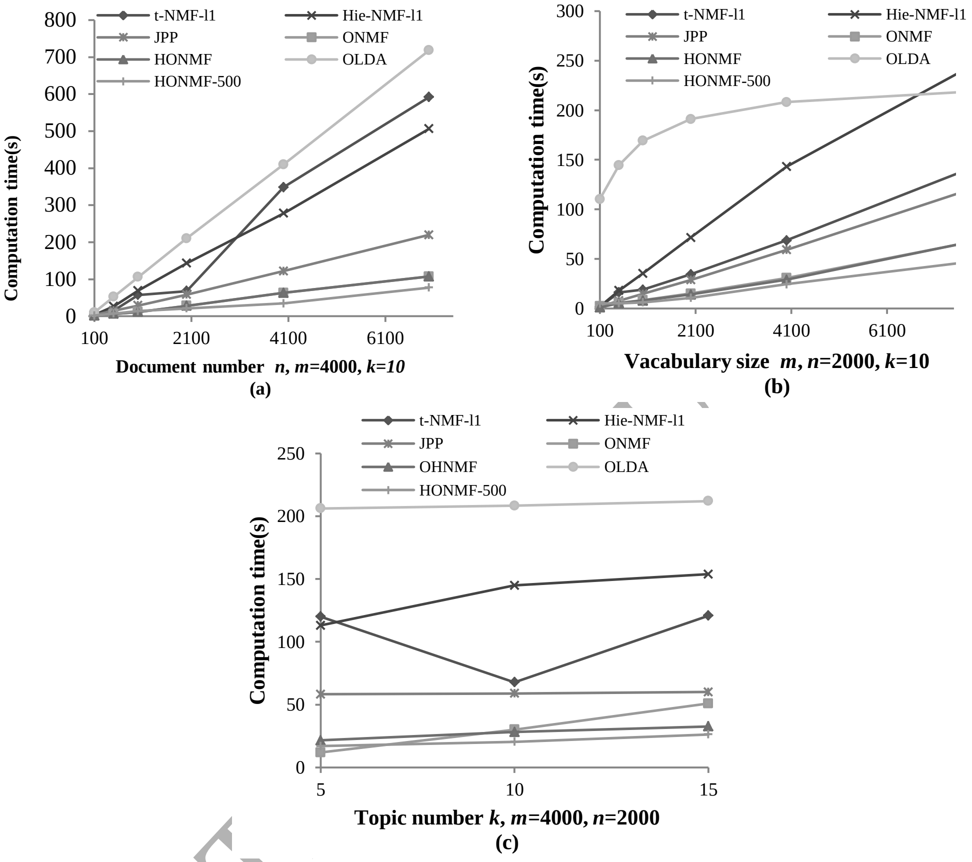

Fig. 7. The computation time of different settings.

Fig. 7(a) is the execution time under different values of *n*. It can be seen that HONMF-500 consumes the least time among all methods, and the time complexities of OLDA, HONMF, and ONMF are nearly linear to *n*. OLDA costs the most time. Despite the FMS process to determine the number of sub-topics in the initial stage, our method costs less time than ONMF in the latter time slots. This can be concluded by comparing HONMF-500 and ONMF, as ONMF is a fully online method and the main difference between HONMF-500 and HONMF is that HONMF cost more time than HONMF-500 in the initial stage. Fig. 7(b) is the execution time under different values of *m*. It can be seen that HONMF and ONMF cost the least time among all methods, and the increasing trends of all methods are nearly linear to *m* except OLDA. Fig. 7(c) is the execution time under different values of *k*. It can be seen that the compared methods have different performances under this setting. The execution time of JPP is stable, and the reason might be that the execution time of JPP is heavily affected by the iteration number (similar reason can be used to explain the results of t-NMF-l1 and OLDA). ONMF has larger time complexity than HONMF under this setting, which is consistent with our analysis in Section 3.6. HONMF-500 costs the least time among all methods. From this experiment, it can be concluded that compared to the baseline methods, HONMF has the lowest time complexity.

5 Conclusions and Future Work

Detecting and tracking latent topics in a text stream is important and various methods have been proposed to address this problem. However, using the existing online topic models, the discovered topics may be not consistent when evolving in the text stream, as the overlap between them may enlarge. In this paper, we propose a hierarchical online NMF to detect and track topic hierarchies in a text stream, which can exploit the relations between the topics to get coherent topics. In addition, our method can adaptively determine the topic numbers and capture the evolving patterns of the topic hierarchy in the evolving process. Compared to the baseline methods, our methods can find high quality topics in less time.

There are some works to be done to refine our work: 1) The node selection metric used in our method cannot get good result under some conditions, and an alternative method could be used to fix this disadvantage; 2) In our method, a document is assigned along one path, and it can be extended by exploiting multiple paths, like nested HDP [20].

Acknowledgment

This work was funded by China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2014-1-5), the Natural Science Foundation of China (No. 61332017), the Science and Technology Department of Zhejiang Province (No. 2015C33002), the Zhejiang Provincial Natural Science Foundation of China (Nos. LY18F020033, LY15F020025), and the Joint Funds of the National Natural Science Foundation of China (No. U1509214).

References

[1] M. D. Hoffman, D. M. Blei, and F. R. Bach, "Online Learning for Latent Dirichlet Allocation," in *Proceedings of the 24th Annual Conference*

on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2010, pp. 856-864.

[2] C. Wang, J. W. Paisley, and D. M. Blei, "Online Variational Inference for the Hierarchical Dirichlet Process," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, 2011, pp. 752-760.

[3] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19-60, 2010.

[4] A. Saha and V. Sindhwani, "Learning Evolving and Emerging Topics in Social Media: A Dynamic NMF Approach with Temporal Regularization," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, Seattle, Washington, USA, 2012, pp. 693-702.

[5] C. K. Vaca, A. Mantrach, A. Jaimes, and M. Saerens, "A Time-based Collective Factorization for Topic Discovery and Monitoring in News," in *Proceedings of the 23rd International Conference on World Wide Web*, Seoul, Korea, 2014, pp. 527-538.

[6] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *J. Mach. Learn. Res.*, vol. 5, pp. 361-397, 2004.

[7] K. Lang, "NewsWeeder: Learning to Filter Netnews," in *Proceedings of the 25th International Conference on Machine Learning*, Tahoe City, California, USA, 1995, pp. 331-339.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993-1022, 2003.

[9] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes," *J. Am. Stat. Assoc.*, vol. 101, pp. 1566-1581, 2006.

[10] H. Wu, J. Bu, C. Chen, J. Zhu, L. Zhang, H. Liu, C. Wang, and D. Cai, "Locally Discriminative Topic Modeling," *Pattern Recogn.*, vol. 45, pp. 617-625, 2012.

[11] H. Zhang, T. W. S. Chow, and M. K. M. Rahman, "A New Dual Wing Harmonium Model for Document Retrieval, "*Pattern Recogn.*, vol. 42, pp. 2950-2960, 2009.

[12] D. D. Lee and H. S. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature*, vol. 401, pp. 788-791, 1999.

[13] B. Jiang, H. Zhao, J. Tang, and B. Luo, "A Sparse Nonnegative Matrix Factorization Technique for Graph Matching Problems," *Pattern Recogn.*, vol. 47, pp. 736-747, 2014.

[14] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary Learning Algorithms for Sparse Representation," *Neural Computation*, vol. 15, pp. 349-396, 2003.

[15] X. Wang and A. McCallum, "Topics Over Time: A non-Markov Continuous-Time Model of Topical Trends," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, 2006, pp. 424-433.

[16] C. Wang, D. M. Blei, and D. Heckerman, "Continuous Time Dynamic Topic Models," in *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence*, Helsinki, Finland, 2008, pp. 579-586.

[17] B. Cao, D. Shen, J.-T. Sun, X. Wang, Q. Yang, and Z. Chen, "Detect and Track Latent Factors with Online Nonnegative Matrix Factorization," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007, pp. 2689-2694.

[18] D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The Nested Chinese Restaurant Process and Bayesian Nonparametric Inference of Topic Hierarchies," *J. ACM*, vol. 55, pp.1-30, 2010.

[19] X. Zhang, D. B. Dunson, and L. Carin, "Hierarchical Topic Modeling for Analysis of Time-Evolving Personal Choices," in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems*, Granada, Spain, 2011, pp. 1395-1403.

[20] J. W. Paisley, C. Wang, D. M. Blei, and M. I. Jordan, "Nested Hierarchical Dirichlet Processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, pp. 256-270, 2015.

[21] R. Jenatton, J. Mairal, G. Obozinski, and F. R. Bach, "Proximal Methods for Hierarchical Sparse Coding," *J. Mach. Learn. Res.*, vol. 12, pp. 2297-2334, 2011.

[22] K. Kersting, M. Wahabzada, C. Thurau, and C. Bauckhage, "Hierarchical Convex NMF for Clustering Massive Data," in *Proceedings of the 2nd Asian Conference on Machine Learning*, Tokyo, Japan, 2010, pp. 253-268.

[23] D. Kuang and H. Park, "Fast Rank-2 Nonnegative Matrix Factorization for Hierarchical Document Clustering," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, Illinois, USA, 2013, pp. 739-747.

[24] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.

24, pp. 603-619, 2002.

[25] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A Study of the Cosine Distance-Based Mean Shift for Telephone Speech Diarization," *IEEE Trans. Audio, Speech, Language, Process.*, vol. 22, pp.217-227, 2014.

[26] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani, "Emerging Topic Detection using Dictionary Learning," in *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, Glasgow, United Kingdom, 2011, pp. 745-754.

[27] D. Cai, X. He, and J. Han, "Locally Consistent Concept Factorization for Document Clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, pp. 902-913, 2011.

[28] A. Beykikhoshk, O. Arandjelović, D. Phung, and S. Venkatesh, "Discovering Topic Structures of a Temporally Evolving Document Corpus," *Knowledge & Information Systems*, vol. 16, pp. 1-34, 2015.

[29] V. Andrei and O. Arandjelović, "Temporal Quasi-semantic Visualization and Exploration of Large Scientific Publication Corpora," in *Proceedings of the International Conference on Artificial Intelligence Workshop on Big Scholarly Data*, 2016, pp. 9-15.

[30] V. Andrei and O. Arandjelović, "Complex Temporal Topic Evolution Modelling Using the Kullback-Leibler Divergence and the Bhattacharyya Distance," *Journal on Bioinformatics and Systems Biology*, vol. 16, 2016.

[31] I. Pruteanu-Malinici, L. Ren, J. Paisley, E. Wang, and L. Carin, "Hierarchical Bayesian Modeling of Topics in Time-stamped Documents," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 996-1011, 2010.

[32] S.S. Bucak and B. Gunsel, "Incremental Subspace Learning via Non-negative Matrix Factorization," *Pattern Recogn.*, vol. 42, pp. 788-797, 2009.

[33] S.H. Na and J.H. Lee, "Memory-restricted Latent Semantic Analysis to Accumulate Term-document Co-occurrence Event," *Pattern Recogn.*, vol. 33, pp. 1623-1631, 2012.

Ding Tu received his B.Sc. degree in Software Engineering from the Zhejiang University, China, in 2009. He is currently a Ph.D. candidate in the College of Computer Science and Technology at Zhejiang University, China. His research interests include unstructured data management and nature language processing.

Ling Chen received his B.S. and Ph.D. degrees in computer science from Zhejiang University, China, in June 1999 and March 2004, respectively. He is currently an associate professor of computer science and technology. His research interests include distributed systems, HCI, databases, AI, and pattern recognition.

Mingqi Lv received his Ph.D. degree in computer science and technology from Zhejiang University, China. He is currently an assistant professor of computer science and technology. His research interests include ubiquitous computing and data mining.

Hongyu Shi received his B.Sc. degree in Computer Science and Technology from the East China University of Science and Technology, China, in 2016. He is currently a Ph.D. candidate in the College of Computer Science and Technology at Zhejiang University, China. His research interests include ubiquitous computing and data mining.

Gencai Chen graduated from Hangzhou University, China, in 1973. After that he became faculty member of the Physics Department, Hangzhou University. He studied in the Computer Department of State University of New York at Buffalo from 1987 to 1988. His research interests include database applications, affective computing, computer supported cooperative work, and data-mining.