Information Sciences 486 (2019) 271–286

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Discovering author interest evolution in order-sensitive and Semantic-aware topic modeling

Min Yang[a], Qiang Qu[a], Xiaojun Chen[b,*], Wenting Tu[c], Ying Shen[d], Jia Zhu[e]

[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[b] College of Computer Science and Software, Shenzhen University, China
[c] Department of Computer Science, Shanghai University of Finance and Economics, China
[d] School of Electronics and Computer Engineering, Peking University Shenzhen Graduate School, China
[e] School of Computing Science, South China Normal University, China

ARTICLE INFO

Article history:
Received 9 September 2016
Revised 12 February 2019
Accepted 16 February 2019
Available online 22 February 2019

Keywords:
Topic model
Dynamic author topic model
Ordering-sensitive
Semantic-aware

ABSTRACT

Modeling the interests of authors over time from documents has important applications in broad applications such as recommendation systems, authorship identification and opinion extraction. In this paper, we propose an Ordering-sensitive and Semantic-aware Dynamic Author Topic Model (OSDATM), which monitors the evolution of author interest in time-stamped documents. The model further uses the discovered author interest information to discover better topics. Unlike traditional topic models, OSDATM is sensitive to the ordering of words, thus it extracts more information from the semantic meaning of the context. The experimental results show that OSDATM learns better topics than state-of-the-art topic models. In addition, the dynamic interests of authors that the OSDATM model discovers are interpretable and consistent with the truth.

1. Introduction

Topic modeling has been applied to plenty of applications, including collaborative filtering [17], information retrieval [38], authorship identification [26] and opinion extraction [16], etc. Existing topic models [3,8,18] assume that each document is a mixture of a certain number of latent topics, where each topic, in turn, defines the topic probabilities over words. These models, such as latent Dirichlet allocation (LDA) [5], are generative probabilistic models for text modeling. Several non-parametric extensions of LDA have been successfully applied to characterize the contents of documents [28,31]. However, the inference of those non-parametric models is computationally hard, such that inaccurate or slow approximations are resorted to calculate the posterior distributions over the topics.

A major limitation of the aforementioned topic modeling approaches and many of their extensions is the “bag-of-words” assumption, which assumes that each document is characterized by the “bag-of-words” features. This assumption is favorable in the computational point of view, but ignores the word order and cannot capture the semantic regularities of documents. For instance, the sentences “the department chair couches offers” and “the chair department offers couches” have the same unigram features, while representing different meanings and topics. When deciding the word “chair” in the

* Corresponding author.
E-mail addresses: min.yang@siat.ac.cn (M. Yang), qiang@siat.ac.cn (Q. Qu), xjchen@szu.edu.cn (X. Chen), tu.wenting@mail.shufe.edu.cn (W. Tu), shenying@pkusz.edu.cn (Y. Shen), jzhu@m.scnu.edu.cn (J. Zhu).

https://doi.org/10.1016/j.ins.2019.02.040
0020-0255/

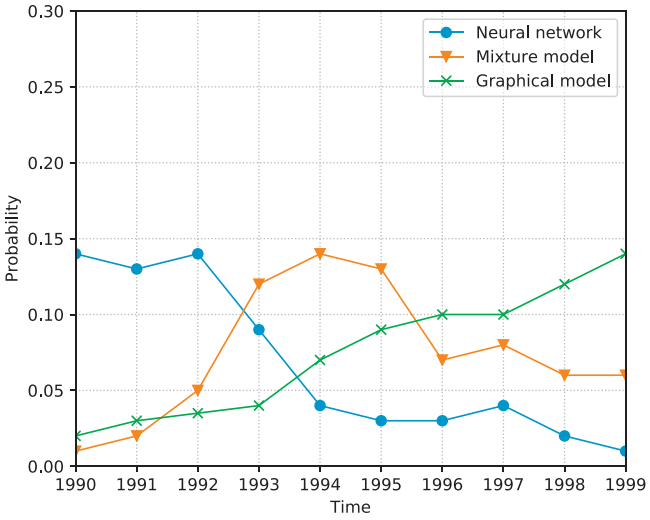


Fig. 1. Change curve of interested topics learned from NIPS dataset.

first sentence is generated by which topic, knowing that it is immediately preceded by the word "department" helps us to find that it is related to the university administration topic [33].

On the other hand, personalization techniques have shown their increasing importance of assisting users in seeking the information of their own interests. By modeling the interests of the author over time, we can answer some important questions: which topics the author likes in history and at present, which authors are similar to each other regarding their interests, and which authors' interests are getting closer over time so that they are likely to collaborate in the future [26]. Author interests play a significant role in plenty of NLP tasks, such as recommender systems, authorship identification and opinion extraction. Motivated by the demand for exploiting the preferences and behavioral patterns of individuals, a large number of studies have been proposed to discover the interests of authors. One common technique is to build generative models that characterize the interests of authors as latent variables. However, these methods discover static interests, while there is strong evidence that the author's interest constantly changes over time. The old interests of authors might gradually fade away while the new interests might arise. Previous interests might slowly fade out of authors' preferences while new interests might arise. For example, as shown in Fig. 1 and Table 8, Michael Jordan's research interests were mainly focused on expert networks and dynamical model in the early years, then the interests are expanded to graphical models and reinforcement learning. The results can be justified from his homepage.

In summary, we identity three plausible hypotheses about the effect of semantics of context and the dynamics of author interests; (i) We can get better word distributions of topics by exploiting the semantics of context and the order of words. (ii) The dynamic author topic model can capture the evolution of a specific author's interest by integrating the authorship and temporal information into the topic model; (iii) The author interest information can help discover better topics in topic modeling in reverse.

Therefore, in this paper, we test our hypotheses by proposing an Ordering-sensitive and Semantic-aware Dynamic Author Topic Model (OSDATM), which integrates the authorship and temporal information into topic model to capture the dynamic evolution of interests of individual authors. Our model is inspired by the recent successes of neural probabilistic language models (NPLMs) [14,20–23]. In OSDATM, the topic model is represented as a Gaussian mixture model of vector representations that represent words, sentences, documents, and interests of authors. Each mixture component of the Gaussian mixture model is associated with a specific topic. The topic model and latent vector representations are learned jointly. Similar to NPLMs, the latent vector representations are learned to optimize the prediction of a word using its context. In our model, the word orders and the semantic regularities of documents are exploited to learn the topic of each word. It overcomes the limitations of previous topic models which do not utilize the word orders and the semantic regularities of documents.

The main contributions of this paper are as follows:

- We propose a novel Ordering-sensitive and Semantic-aware Dynamic Author Topic Model (OSDATM), which integrates authorship and temporal information into GMNTM [40] to explore the evolution of interests of authors from time-stamped documents. OSDATM can capture the fact that co-authors often share similar interests, by making their interest vectors positively correlated.
- We propose an efficient convex optimization algorithm to estimate the latent vector representations of words, sentences, documents, and author interests in OSDATM.
- We conduct extensive experiments to justify the effectiveness of OSDATM on three widely used text datasets. The experiment results indicate that OSDATM significantly outperforms the state-of-the-art topic models from both quantitative and qualitative perspectives.

This manuscript is an extension of our preliminary work [40,41]. First, we extend the GMNTM in [40] to a dynamic author topic model by integrating authorship and temporal information. The goal of OSDATM is to deal with the challenge of modeling dynamic author interest. These challenges include: (1) how to model author's interest and incorporate it as a part of the language model; (2) how to model the evolution of a specific author's interest; (3) How to model the correlation between coauthor's interest; (4) how to model textual information beyond the "bag-of-words" assumption to take the ordering and the semantics of words into consideration; and (5) how to find an efficient algorithm for learning the composite model. Second, compared to [41] which uses stochastic gradient descent to estimate both the linear transformation coefficients and the latent vector representations, we propose an efficient convex algorithm to estimate the latent vector representations. Our algorithm ensures that all local optima are globally optimal, and it is more efficient than [41]. In addition, we also conduct extensive experiments to evaluate the effectiveness of our model on monitoring author interest evolution, which are not included in our preliminary work.

The rest of this manuscript is organized as follows. In Section 2, we review the related work. In Section 3, we present the OSDATM in detail. Section 4 describes the experiments. In Section 5, we conclude this manuscript and indicate the future work.

2. Related work

Topic models, such as Latent Dirichlet Allocation (LDA) [5], have achieved great success in discovering thematic structure from large document collections. As the interests of authors showing increasing importance for the development of personalized and user-centric applications, variety of LDA extensions have been proposed to incorporate authorship information into the text. The Author-Topic model [26] was the first generative model that simultaneously modeled the content of documents and the interests of authors. McCallum et al. [19] proposed the Author-Recipient-Topic model for social network analysis, which learned topic distributions based on the direction-sensitive messages sent between entities. Kawamae [12] proposed the Author-Interest-Topic model, introducing a latent variable with a separate probability distribution over topics into each document. However, the aforementioned models are devoted to discovering static latent topics and author's interests. In fact, many of the large data sets do not have static co-occurrence patterns, they are instead dynamic.

To characterize topics and their changes over time, there are also some approaches which use the information of the timestamps. Griffiths and Steyvers [9] proposed a method using post-hoc analysis to identify hot and cold topics based on the examination of topic mixtures estimated from an LDA model. Wang et al. [37] pre-divided the data into discrete time slices, and fitted a separate topic model in each slice. Blei and Lafferty [4] proposed a dynamic topic model (DTM) which jointly modeled word co-occurrence and time. In their model, the alignment among topics across time steps was captured by a Kalman filter. Wang and McCallum [36] proposed a non-Markov continuous-time model, called ToT. For each generated document, the mixture distribution over topics was influenced by both word co-occurrences and the timestamps of the document. Wang et al. [34] proposed a continuous time dynamic topic model (cDTM), which was an extension of the discrete-time based DTM model. These models are able to capture the evolution of topics, but they do not consider the authorship information.

There are recent studies taking both the timestamp and the authorship of documents into account, including the Temporal-Author-Topic (TAT) model [7], the Author-topic over time (AToT) model [39] and Author-Time-Topic (ATT) model [29]. These models utilize two facts: (i) every author is interested in a small subset of topics and (ii) the overall popularity of topics changes over time. Nevertheless, these models do not characterize the drift of the interests of individual authors. As we show in the experiments, the interests of individual authors may substantially change over time.

Moreover, all of the aforementioned topic models employ the bag-of-words assumption which is rarely true in practice. The bag-of-words assumption loses the ordering of the words and ignores the semantics of the context. There are several previous studies taking the ordering of words into account. Griffiths et al. [10] introduced a generative model that integrated topics and syntax. It contained a latent variable per each word that stood for syntactic classes. The model posited that the words were generated either from topics that were randomly drawn from the topic mixture of the document or from the syntactic classes that were drawn from the previous syntactic class. Wallach [33] explored a hierarchical generative probabilistic model that incorporated both n-gram statistics and latent topic variables. They extended a unigram topic model so that it could reflect properties of a hierarchical Dirichlet bigram model. Gruber et al. [11] modeled the topic of words as a Markov chain. They assumed that all words in the same sentence had the same topic, and successive sentences were more likely to have the same topics. Yang et al. [40] proposed a generative topic model that represented each topic as a cluster of multi-dimensional vectors and embed the corpus into a collection of vectors generated by the Gaussian mixture model. Nevertheless, none of these models uses information of the timestamps or models the interests of authors.

On the other hand, there are some studies using conventional statistical methods to extract topics. Velden et al. [32] described most of the existing techniques for topic extraction by using statistics information. For example, Chen et al. [6] extracted hot topics from the textual documents published in a given time period. Hot terms were first extracted by considering variations in the frequency that terms were used over time. Based on the extracted hot terms, key sentences were identified and then grouped into clusters that represent hot topics by using multidimensional sentence vectors. Wang and Koopman [35] built a valid semantic representation for each article from all the entities they are associated with and identified article clusters using standard methods based on such a semantic representation. Osborne and Motta [24] developed

Klink-2 that employed semantic web technologies and network analysis for modeling the dynamics of the evolution of topics (and their relations) in the scholarly domain.

The focus of our OSDATM model is different from all of the above. OSDATM explicitly models the evolution of author interest, instead of adding authorship as a simple feature. It is the first model that utilizes the interaction between the authorship information and the timestamp information, which results in very interesting discoveries. Simultaneously, the ordering and semantics of words are taken into consideration based on the context.

3. Ordering-sensitive and semantic-aware dynamic author topic model (OSDATM)

In this section, we describe the OSDATM model as a generative model. After that, we illustrate the inference algorithm for estimating the model parameters.

3.1. Generative model

We assume that there are W different words in the vocabulary and there are D documents in corpus. In addition, these documents belong to T topics, where T is a hyper-parameter specified by the user. For a specific document d, suppose that it has m authors a_1, ..., a_m. We use a p-dimensional vectorized representation (interest vector) $\text{vec}(a_i, d) \in \mathbb{R}^p$ to represent the interest of author a_i when she/he writes document d. Two authors have similar interest if the distance between their interest vectors is small. Our goal is to learn these interest vectors from the content of the corpus. It is clear that the interests of co-authors should be correlated. The interest of the same author in writing different documents should be consistent. Thus, we define a generative model to characterize the correlation between these dependent interests.

We now specify the generative model for vectors $\text{vec}(a_i, d)$ $(i = 1, \ldots, m)$, conditioning on the interest vectors of all earlier documents. This defines a valid model, since interest vectors can be generated sequentially from the earliest document to the latest document. Let t be the time that document d was written. Let d_i' be the last document that author a_i has written before she writes the current document, and let the timestamps for d_i' be t_i'. We have $t_i' \le t$ for all $i = 1, \ldots, m$. We define a joint distribution on the vectors $\text{vec}(a, d) := (\text{vec}(a_1, d), \ldots, \text{vec}(a_m, d))$. It is a multivariate normal distribution on $\mathbb{R}^{m\times p}$ taking the form $N(\mu_d, \Sigma_d)$. First, we specify the mean. Let

$$\mu_d := (\text{vec}(a_1,d_1'), \ldots, \text{vec}(a_m,d_m')), \tag{1}$$

where $\text{vec}(a_i; d_i')$ is the interest vector for author a_i when she wrote document d_i'. This definition implies that the new interests of authors have connections to the history.

Second, we define the covariance matrix Σ_d. Note that Σ_d is an $mp \times mp$ matrix, so that it can be partitioned into m^2 sub-matrices, each with dimension $p \times p$. Let $\Sigma_{ij} \in \mathbb{R}^{p\times p}$ be the sub-matrix on the ith row and jth column. If $i = j$, then the submatrix characterize the covariance of the author i's interest. Let

$$\Sigma_{ii} := \sigma^2(t - t_i')I, \tag{2}$$

where $\sigma(x)$ is an increasing function of x. It means that as more time passed, the covariance matrix entries get bigger, indicating that the interest of author i is less likely to concentrate on its mean – the interest vector when he/she wrote the earlier document d'. More concretely, we adopt a linear function $\sigma(x) = \alpha + \beta x$ with hyper-parameters $\alpha > 0$ and $\beta > 0$. The values of α and β are chosen by cross-validation.

If $i \neq j$, then the submatrix Σ_{ij} characterizes the correlation between the interest of the author i and the author j. Let

$$\Sigma_{ij} := \rho\sigma(t - t_i')\sigma(t - t_j')I, \tag{3}$$

where $\rho \in [0, 1]$ is another hyper-parameter measuring the correlation of the interests of co-authors. if $\rho = 0$, then $\text{vec}(a_1, d)$, ...,$\text{vec}(a_m, d)$ are mutually independent, meaning that there is no correlation between co-authors. If $\rho = 1$, then the drift vectors $\text{vec}(a_i, d) - \text{vec}(a_i, d_i')$ are in the same direction for all co-authors, meaning that the interest drift is perfectly correlated. One typically chooses the value of ρ between zero and one. In summary, the density function of vector $\text{vec}(a, d)$ is defined by

$$p(\text{vec}(a, d) = v) \propto \exp\left(-\frac{(v - \mu_d)^T \Sigma_d^{-1}(v - \mu_d)}{2}\right) \tag{4}$$

The vector μ_d and the matrix Σ_d follow the definition above.

Given that the author interests are vectorized, we use vectors to represent words, sentences and documents, so that their semantic representations are naturally connected with that of the author interest. For each word $w \in \{1, \ldots, W\}$, there is an associated p-dimensional vectorized representation $\text{vec}(w) \in \mathbb{R}^p$. Each document with index $d \in \{1, \ldots, D\}$ has a vector representation $\text{vec}(d) \in \mathbb{R}^p$. There are S sentences in the corpus, each sentence indexed by $s \in \{1, \ldots, S\}$. The sentence with index s is associated with a vector representation $\text{vec}(s) \in \mathbb{R}^p$.

We assume that there are T topics. For each topic, we define a generative model for the vectors of words, sentences and documents. We assume that all these vectors are generated from a multi-variate normal distribution. That means, given a topic k, the conditional distribution of $\text{vec}(w)$, $\text{vec}(s)$ and $\text{vec}(d)$ are defined by

$$\text{vec}(w) \sim N\left(\mu_k^{\text{word}}, \Sigma_k^{\text{word}}\right) \tag{5}$$

$$\text{vec}(s) \sim N\left(\mu_k^{\text{sen}}, \Sigma_k^{\text{sen}}\right) \tag{6}$$

$$\text{vec}(d) \sim N\left(\mu_k^{\text{doc}}, \Sigma_k^{\text{doc}}\right), \tag{7}$$

where μ_k and Σ_k are the mean and the variance of the multi-variate normal distribution. The superscripts "word", "sen" and "doc" indicates that the conditional distributions for words, sentences and documents might be different. It allows learning distinct models for each layer of the semantic abstraction.

To complete the definition, it remains to define the prior distribution on the topic k. We adopt a simple multivariate prior, where the probability of choosing topic k is equal to π_k. To make it a probability distribution, we impose constraint such that $\sum_{k=1}^{T} \pi_k = 1$. Combining these definitions, it is easy to see that the vectorized words, sentences and documents are generated from a Gaussian mixture model with unknown mixture weights and unknown Gaussian parameters. We use Ψ to represent the set of these unknown parameters. Given that the word vectors, the sentence vectors and the document vectors are known, the parameters Ψ can be estimated by applying the classical Expectation Maximization (EM) algorithm. The estimates on Ψ may further help re-estimating the vectors $\text{vec}(w)$, $\text{vec}(s)$ and $\text{vec}(d)$. This naturally leads to an iterative estimation algorithm, which we will specify in details in Section 3.3.

Given the generative model for vectorized author interests, words, sentences and documents, we now describe the procedure that the actual word realizations are generated. Let V be the collection of all latent vectors:

$$V := \{\text{vec}(w)\} \cup \{\text{vec}(d)\} \cup \{\text{vec}(s)\} \cup \{\text{vec}(a, d)\} \tag{8}$$

For each word slot in the sentence, its word realization is generated according to the author's interest vector $\text{vec}(a, d)$, the document's vector $\text{vec}(d)$, the current sentence's vector $\text{vec}(s)$ as well as vectors of the n previous words in the same sentence. The connection between the word realization and these vectors might be arbitrary, but we found in practice that a linear prediction model works well enough. Formally, for the ith word in the sentence, we represent its word realization by w_i. The probability distribution of w_i being generated is defined by

$$\begin{aligned} &p(w_i = w | d, a, s, w_{i-n}, \ldots, w_{i-1}) \\ &\propto \exp\left(a_{\text{author}}^w + a_{\text{doc}}^w + a_{\text{sen}}^w + \sum_{t=1}^{n} a_t^w\right), \end{aligned} \tag{9}$$

where a_{author}^w, a_{doc}^w, a_{sen}^w and a_t^w are linear transformations of the vectors of the author interest, the document, the sentence and the previous n words. These linear transformations are defined by

$$a_{\text{author}}^w = \left\langle u_{\text{author}}^w, \text{vec}(a, d) \right\rangle \tag{10}$$

$$a_{\text{doc}}^w = \left\langle u_{\text{doc}}^w, \text{vec}(d) \right\rangle \tag{11}$$

$$a_{\text{sen}}^w = \langle u_{\text{sen}}^w, \text{vec}(s) \rangle \tag{12}$$

$$a_t^w = \langle u_t^w, \text{vec}(w_{i-t}) \rangle, \tag{13}$$

where $u_{author}^w, u_{\text{doc}}^w, u_{\text{sen}}^w, u_t^w \in \mathbb{R}^p$ are transformation coefficient. The transformation coefficients are unknown parameters of the model, but they are shared across all word slots in the corpus. We use U to represent this collection of these coefficients.

We complete the model specification by summarizing the set of unknown parameters that the model needs to estimate. There are three sets of unknown parameters: the set V containing the vector of author interests, words, sentences and documents, the set U containing the unknown linear transformation coefficients, and the set Ψ containing parameters that defines the Gaussian mixture model. We will see in Section 3.3 that given arbitrary two of these sets, the remaining set of parameters can be solved by efficient algorithms.

3.2. Topic inference

Given the model parameters U, Ψ and the vectors V, we can infer the posterior probability distribution of topics. In particular, for a word w with vector representation $\text{vec}(w)$, the posterior distribution of its topic, namely $q(z(w))$, is easy to derive given the generative model. For any topic $z \in 1, 2, \ldots, T$, we have

$$q(z(w) = z) = \frac{\pi_z \mathcal{N}(\text{vec}(w) | \mu_z, \Sigma_z)}{\sum_{k=1}^{T} \pi_k \mathcal{N}(\text{vec}(w) | \mu_k, \Sigma_k)}. \tag{14}$$

Similarly, for each sentence s in the document d, the posterior distribution of its topic is

$$q(z(s) = z) = \frac{\pi_z \mathcal{N}(\text{vec}(s)|\mu_z, \Sigma_z)}{\sum_{k=1}^{T} \pi_k \mathcal{N}(\text{vec}(s)|\mu_k, \Sigma_k)}. \tag{15}$$

For a document d, the posterior distribution of its topic is

$$q(z(w) = z) = \frac{\pi_z \mathcal{N}(\text{vec}(d)|\mu_z, \Sigma_z)}{\sum_{k=1}^{T} \pi_k \mathcal{N}(\text{vec}(d)|\mu_k, \Sigma_k)} \tag{16}$$

For authors, there could be topics associated with the author's interest. By definition, the author interest vectors are not generated by the Gaussian mixture model induced by topics. Nevertheless, since these vectors are of the same dimension as other vectors, we can use the same formula for estimating their connection to the topics, i.e. by

$$q(z(a, d) = z) = \frac{\pi_z \mathcal{N}(\text{vec}(a, d)|\mu_z, \Sigma_z)}{\sum_{k=1}^{T} \pi_k \mathcal{N}(\text{vec}(a, d)|\mu_k, \Sigma_k)}. \tag{17}$$

Here, the term $q(z(a, d) = z)$ reflects the ratio of the author a's interest in topic z, when she writes document d. In the experiment section, we demonstrate that such a scheme yields reasonable representation for the author's interest.

We also define topic for the context around a specific location. For the context around a word slot, the topic is affected by its neighboring words, the sentence/document it belongs to as well as the author interest, We define the probability that it belongs to topic z proportional to the product of $q(z(w) = z)$, $q(z(s) = z)$, $q(z(a, d) = z)$, and $q(z(d) = z)$, where w, s, a and d are the word, the sentence, the authors and the document that this word slot associates with.

3.3. Estimating model parameters

We estimate the model parameters and latent vectors Ψ, U and V by maximizing a posteriori (MAP) of the generative model. The parameter estimation consists of three stages. In Stage I, we maximize the likelihood of the model with respect to Ψ. Since Ψ characterizes a Gaussian mixture model, this procedure can be implemented by the Expectation Maximization (EM) algorithm. In Stage II, we maximize the likelihood with respect to U. This is a standard logistic regression problem, which can be solved by efficient convex optimization algorithm. In Stage III, we maximize a posteriori with respect to V. We demonstrate that it is also a convex optimization problem, thus can be efficiently solved. We alternatively execute Stage I, II and III until the parameters converge. The inference of OSDATM is summarized in Algorithm 1.

Algorithm 1: Inference of OSDATM model.

Input : Corpus C
Output: latent vectors V, transformation coefficients U, Gaussian mixture model parameters Ψ

```
// initialization
V ← randomly initialized from [−1, 1] ;
// Initialize parameters U with all-zero vectors
U ← 0 ;
// Initialize Ψ with the standard normal distribution
Ψ ← N(0, diag(1)) ;
repeat
    update Ψ with EM algorithm, fixing V and U ;
    Updata U by maximizing Eq. (9) and fixing V, Ψ;
    update V by minimizing Eq. (19) and fixing Ψ and U ;
until coverage;
```

3.3.1. Stage I: estimating Gaussian mixture components Ψ

In this stage, the latent vector of words, sentences and documents are given. If it is the first iteration, then these vectors are randomly initialized. We estimate the parameters of the Gaussian mixture model $\Psi = \{\pi_k, \mu_k, \Sigma_k\}$. This is a classical non-convex estimation problem. Although it is difficult to find the global optimal solution, the EM algorithm guarantees to converge to a locally optimal solution, which is often good enough in practice. The reader can refer to the book [2] for the implementation of the EM algorithm.

3.3.2. Stage II: estimating linear transformation coefficients U

In this stage, the latent vectors V are given. We estimate the linear transformation coefficients U such that the likelihood of the prediction is maximized. Since the prediction model is a multi-class logistic regression model, the estimation of coefficient U is reduced to solving a multi-class logistic regression problem. There is a broad class of algorithms for solving this problem. The user can choose stochastic gradient descent or its variants.

3.3.3. Stage III: estimating latent vectors V

In this stage, the Gaussian mixture components Ψ and the linear transformation coefficients U are given. We estimate latent vectors V by maximizing its posterior probability. Recall that the posterior probability is proportional to the product of prior and likelihood. This is equivalent to minimizing the negative log-prior plus the negative log-likelihood. The negative log-likelihood is determined by the prediction model. Formally, it takes the form

$$\ell(V) = -\sum_{w}\left(a^{w}_{\text{author}} + a^{w}_{\text{doc}} + a^{w}_{\text{sen}} + \sum_{t=1}^{m} a^{w}_{t}\right. \tag{18}$$

$$\left. + \log\left(\sum_{w'}\exp\left(a^{w'}_{\text{author}} + a^{w'}_{\text{doc}} + a^{w'}_{\text{sen}} + \sum_{t=1}^{m} a^{w'}_{t}\right)\right)\right), \tag{19}$$

where w enumerates over every word in the corpus. The term $a^{w}_{\text{author}} + a + a^{w}_{\text{sen}} + \sum_{t=1}^{m} a^{w}_{t}$ is linear function of V. Thus, the function $\ell(V)$ is convex with respect to V.

It remains to look at the negative log-prior of vectors V. For author interest vectors, the definition of Eq. (4) yields that the negative log-prior is equal to

$$-\log(p(\text{vec}(a,d))) = \frac{1}{2}(\text{vec}(a,d) - \mu_d)^T \Sigma_d^{-1}(\text{vec}(a,d) - \mu_d) + C, \tag{20}$$

where C is a constant independent of vec(a, d). Recall that $\mu_d = \text{vec}(a,d')$, which is the concatenation of interest vectors of authors when they wrote the previous article. Thus, the term inside the sum is a quadratic function of vec(a, d) and vec(a, d'). According to the definition in Section 3.1, the matrix Σ_d^{-1} is positive semi-definite. Thus, the negative log-prior function is convex with respect to V.

For word vectors, sentence vectors and document vectors, their negative log-priors are derived from the Gaussian mixture model. Taking word w as an example, the negative log-prior of vec(w) is equal to

$$-\log\left(p(\text{vec}(w))\right) = -\log\left(\sum_{k=1}^{T}\pi_k \exp\left(-\frac{1}{2}(\text{vec}(w) - \mu_k^{\text{word}})^T \times \left(\Sigma_k^{\text{word}}\right)^{-1}(\text{vec}(w) - \mu_k^{\text{word}})\right)\right) + C \tag{21}$$

Unfortunately, Eq. (21) is no longer a convex function of vec(w). To approximate it by a convex function, we randomly sample the topic of word w from the multinomial distribution $(\pi_1, \ldots, \pi_T)$. Denote by z the randomly sampled topic. Conditional on the topic, the negative log-prior of vec(w) is equal to

$$-\log(p(\text{vec}(w)|z)) = \frac{1}{2}\left(\text{vec}(w) - \mu_z^{\text{word}}\right)^T \left(\Sigma_z^{\text{word}}\right)^{-1} \cdot \left(\text{vec}(w) - \mu_z^{\text{word}}\right) + C \tag{22}$$

which is a convex function of vec(w). Since the topic is randomly sampled, we take expectation with respect to z, then use

$$\mathbb{E}[-\log(p(\text{vec}(w)|z))] = \sum_{k=1}^{T}\frac{\pi_k}{2}\left(\text{vec}(w) - \mu_k^{\text{word}}\right)^T \cdot \left(\Sigma_k^{\text{word}}\right)^{-1}\left(\text{vec}(w) - \mu_k^{\text{word}}\right) + C \tag{23}$$

as the log-prior of vector vec(w). By doing the same trick on the log-prior of vec(s) and vec(d), both the log-prior functions and the log-likelihood function are convex with respect to V. It allows us to employ efficient convex optimization algorithms to solve V. Since the log-prior functions cannot be decomposed into a sum of sample-based loss functions, it is difficult to employ stochastic gradient descent. Instead, we recommend using quasi-Newton methods, in particular the L-BFGS algorithm, for solving the problem. We observe in practice that L-BFGS is sufficiently fast in solving this problem.

4. Experiments

In this section, we test the OSDATM model on three publicly available datasets. We compare our model with state-of-the-art topic models with both quantitative and qualitative evaluations.

4.1. Datasets

We use the NIPS paper data, The ArnetMiner academic search data and the Enron emails data in our experiments. Data preprocessing is executed before training the models. We first divide the text into sentences according to the delimiters as in [11]. Then we remove non-alphabet characters, numbers, pronouns, punctuations and stop words from the text. Finally, stemming is applied so as to reduce the vocabulary size and settle the issue of data spareness. To remove sensitivity to capitalization, all text is downcased except the words containing two or more capital letters. The detailed properties of the datasets are described as follow.

NIPS papers (NIPS): This dataset is a collection of papers from the NIPS conference between 1987 and 1999.[1] After data preprocessing, this dataset contains 1740 research papers with 2037 authors. Each timestamp is determined by the year of the proceedings. Following the preprocessing in [26], the 1740 papers are further divided into a training set of 1557 papers and a test set of 183 papers of which 102 are single-authored papers. We omit appearances of "e.g." and "i.e." in our preprocessing.

ArnetMiner abstracts (ArnetMiner): This dataset is the "Citation-network V1" dataset provided by Tsinghua University for their ArnetMiner academic search engine [30]. The Arnetminer dataset covers major publications in the area of computer science. It was collected by using a unified automatic extraction approach on researcher's profile pages from the Web and other online digital libraries. Currently, this dataset contains 629,814 publications, 12,609 conferences, and 595,740 authors covering the period of 2000–2010.[2] Each publication has the information about abstract, authors, year, venue, and title. The dataset is further divided into a training set of 529,814 abstracts and a test set of 100,000 abstracts of which 213 are single-authored.

Enron emails (Enron): This data was originally published by the Federal Energy Regulatory Commission during its investigation. After removing the emails from this collection when requested by the email's authors or recipients, there are 517,424 messages belonging to 150 users in the new version of Enron corpus.[3] We further clean the corpus by removing computer generated files from each user, as stated in [13]. We also remove "quoted original messages" in replies as in [19]. Finally, there are 75,686 messages in our Enron data set. We further divide the Enron data into a training set of 65,686 messages and a test set of 10,000 messages. Each author has at least one document in the training set.

4.2. Baseline methods

We describe the baseline methods used in this paper as follows:

- *Hot Topic Extraction (HTE).* This model extracted hot topics from textual document published in a given time period [6]. Hot terms were first extracted by considering variations in the frequency that terms were used over time. Based on the extracted hot terms, key sentences were identified and then grouped into clusters that represent hot topics by using multidimensional sentence vectors. Following the strategy in [6], we choose SL value as 100.
- *OCLC-31.* This model builds a valid semantic representation for each article from all the entities they are associated with and identifies article clusters using standard clustering methods such as *K*-means [35]. One of the most important parameters for *K*-Means is the choice of k. We tried the k from 5 to 100 and set $k = 80$ since it achieved best performance.
- *Latent Dirichlet Allocation (LDA).* LDA is a widely-used baseline method for topic modeling. In the LDA model [5], we use the online variational inference implementation of the Gensim toolkit.[4] We used the recommended parameter settings $\alpha = 1/T$ and $\beta = 0.01$, and 2000 iterations of the Gibbs sampler are adopted.
- *Author-Topic (AT).* This model is first proposed in [26], which extends LDA to include authorship information. We implement AT model using the publicly available code[5], with default setting for all the hyper-parameters. Here, we set $\alpha = 1/T$ and $\beta = 0.01$, and 2000 iterations of the Gibbs sampler are adopted.
- *Author-Time-Topic (ATT).* It is one of the first topic models that simultaneously modeling trends of changes in document contents and author interests [29]. α and β are set to 0.5 and 0.1, respectively. Gibbs sampling is run for 2000 iterations.
- *Metadata-incorporated Dynamic Topic Model (mDTM).* This model is proposed by Li et al. [15], which is developed to analyze the time evolution of topics in large document collections with the help of metadata. In this experiment, we take authors as metadata. We use the same settings for all hyper parameters as in [15]. f_γ is set as time-decay weighted average with $\kappa = 0.3$. Bayesian posterior evolution was used for g and f_β. The model is running with 600 sampling iterations, and we found 300 iterations were enough for valid inference.
- *Author-Topic over Time (AToT).* This is a dynamic user interest model presented in [39], combining Author-Topic model and Topic-over-Time model. We use the same settings for all hyper parameters as in [39]. The time range of the data is normalized to [0.01, 0.99]. α and β are set to 0.5 and 0.1, respectively. Gibbs sampling is run for 2000 iterations.
- *Gaussian Mixture Neural Topic Model (GMNTM).* This is our preliminary work [40], which jointly learns the topic model and the vector representations of words, sentences and documents. In the experiments, the learning rate is set to 0.025 and gradually reduced to 0.0001. For each word, at most 6 previous in the same sentence is used as the context. The word vector size is set to 100. The other parameters are the same as in [40].

[1] Available at http://www.cs.nyu.edu/~roweis/data.html.
[2] Available at http://aminer.org/billboard/citation.
[3] http://www.cs.cmu.edu/~enron/.
[4] http://radimrehurek.com/gensim/models/remodel.html.
[5] http://www.jmlr.org/papers/v13/zeng12a.html.

4.3. Implementation details

In OSDATM, we set the learning rate α to 0.05 and gradually reduce it to 0.0001. Specifically, in each epoch, we update the learning rate by the equation: LearningRate := LearningRate / (1 + decay). We initialize the LearningRate value as 0.05, and set the decay to 0.1. The value of ρ is set to 0.5. For each word, we only consider the previous six words ($m = 6$) within the same sentence. For easy comparison with baseline methods, the size of word vectors is $V = 100$. Increasing the size of word vectors can further boost the generated topics. We try 5, 10, 20, 40, 60, 80, 100, 200 as the topic number of the topic models for each dataset. We construct a restricted vocabulary by using the 20,000 most frequent words. The parameters U are initialized with all-zero vectors, and the Gaussian mixture model parameters are initialized with the standard normal distribution. All the words that are not included in the vocabulary are replaced by a special token "UNK" and are not counted into experiments. Documents are split into sentences and words using the NLTK toolkit [1].[6] The Gaussian mixture model is learnt using the variational inference algorithm in scikit-learn toolkit [25].[7]

4.4. Quantitative evaluation

4.4.1. Perplexity evaluation

To evaluate the predictive power of the proposed model, we compare test-set perplexity of our model with that of the baseline methods on both NIPS and Enron data sets. Perplexity is a widely used measure to estimate the predictive power of a generative model. Generally, perplexity monotonically decreases as log-likelihood increases, implying that a lower perplexity over the held-out document indicates the better generalization performance of a model. Note that we do not compare the perplexity with HTE, OCLC-31 and Klink-2 since these three baseline methods are not generative models.

With the estimated model parameters, the averaged test perplexity is computed as follows [26]:

$$\exp\left(-\frac{1}{N_{test}}\sum_{w}\log p(w_d|a_d, D_{train})\right),$$

where N_{test} are the total number of words in the held-out test documents and $p(w_d|a_d, D_{train})$ is the probability assigned to the words w_d in the test document, conditioned on the known authors a_d of the test document. Here we calculate the average perplexity of the 102 single-authored test documents in the NIPS corpus, and 1000 held-out documents that are sampled from the test sets in the Enron corpus.

Fig. 2 shows the perplexity plotted against the number of hidden topics K for NIPS, ArnetMiner, and Enron datasets. The LDA model is clearly worse than the other models, as illustrated by its relatively high perplexity score. This is because that the LDA model neglects the authorship information and the timestamps information. In addition, it does not use the ordering of the words. On the three datasets, the OSDATM model clearly outperform LDA and is slightly better than the mDTM, AT and AToT models in terms of perplexity. This verifies the strong predictive power of the proposed modeling approach.

When the number of topics is small, i.e., $K \leq 20$ for NIPS dataset, $K \leq 40$ for ArnetMiner dataset and $K \leq 20$ for Enron dataset, the perplexity of our model drops rapidly indicating that the OSDATM does not fit the unseen training data. It suggests that as the model gets richer in terms of the parameters, its generalization performance increases. However, after some point, increasing the number of topics does not reduce the perplexity of the model, while in fact the perplexity score remains constant or increases in a small scale. In the following experiments, we choose $K = 100$ for NIPS dataset, $K = 120$ for ArnetMiner dataset and $K = 80$ for Enron dataset.

4.4.2. Time complexity analysis

We analyze the time complexity of our model step by step. In stage 1 (refers to Section 3.3.1), we use EM algorithm to estimate the parameters of Gaussian mixture model. It requires $O(D^*K^2)$, where D is the number of documents in the corpus and K is the number of topics. Stage II and Stage III (refer to Sections 3.3.2 and 3.3.3) estimate the word and author interest vectors and the corresponding transformation coefficients, and require $O(D \times (n \times p + p \times W))$, where n is number of context words we used, p is the dimension of the word and author interest vector, W is size of the vocabulary.

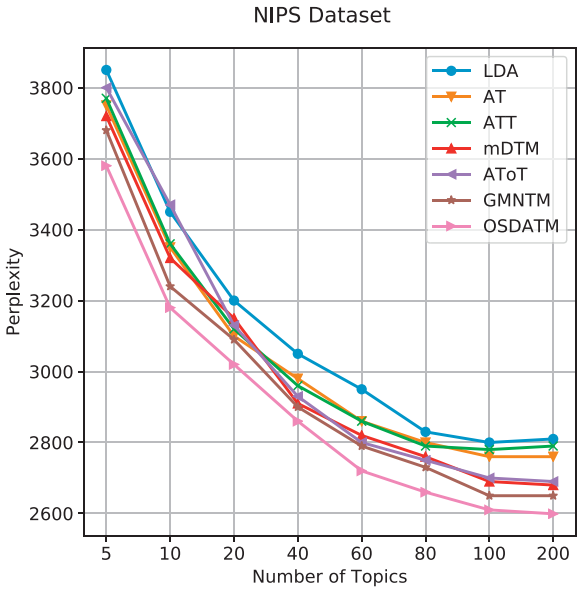
We also evaluate the efficiency of our model by calculating the average running time for each epoch during the training process (using one CPU only). The results are shown in Table 1 with the topic number of 100. For NIPS dataset, the average running time of one epoch of our model is 7.31 min, which is competitive with or better than AToT and GMNTM. LDA and AT are much faster than our model. However, the performances of LDA and AT are vastly inferior compared to our model. Similar results can be observed on the ArnetMiner and Enron datasets.

4.4.3. Document retrieval evaluation

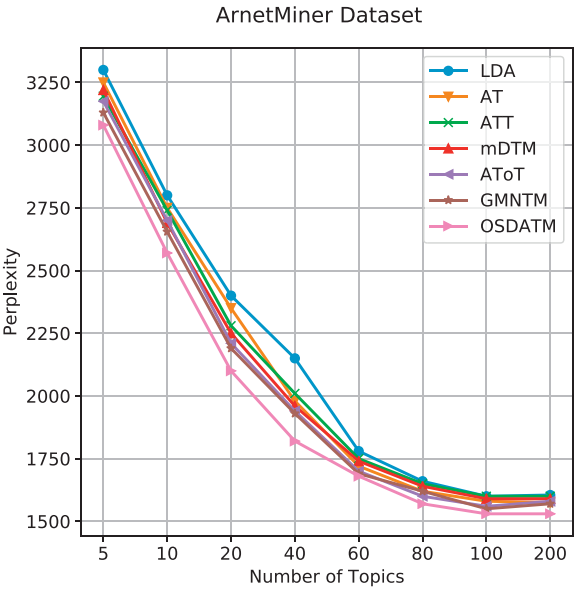
To measure the quality of the documents representations learned by our models, we conduct document retrieval experiments. Following the evaluation in [27], the documents in training sets are utilized as a database, while the documents

[6] http://www.nltk.org.

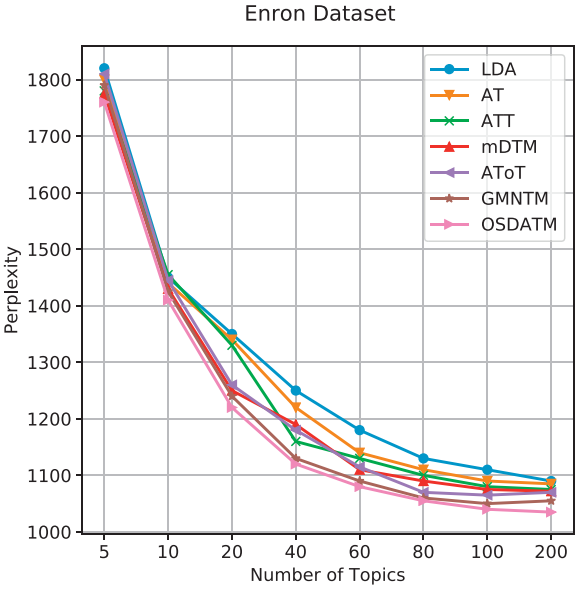
[7] http://scikit-learn.org.



(a) Results on the NIPS papers data set.



(b) Results on the ArnetMiner abstracts data set.



(c) Results on the Enron emails data set.

Fig. 2. Predictive perplexity as a function of the number of topics on there datasets: NIPS papers (top), ArnetMiner abstracts (middle) and Enron emails (bottom)

Table 1
Comparison of average training time per epoch (minute).

Data	LDA	AT	ATT	mDTM	AToT	GMNTM	OSDATM
NIPS	2.03	2.99	4.07	4.79	7.45	7.12	7.31
ArnetMiner	18.66	19.32	21.56	22.45	25.42	23.22	24.56
Enron	25.22	29.48	34.33	42.22	50.13	47.26	49.20

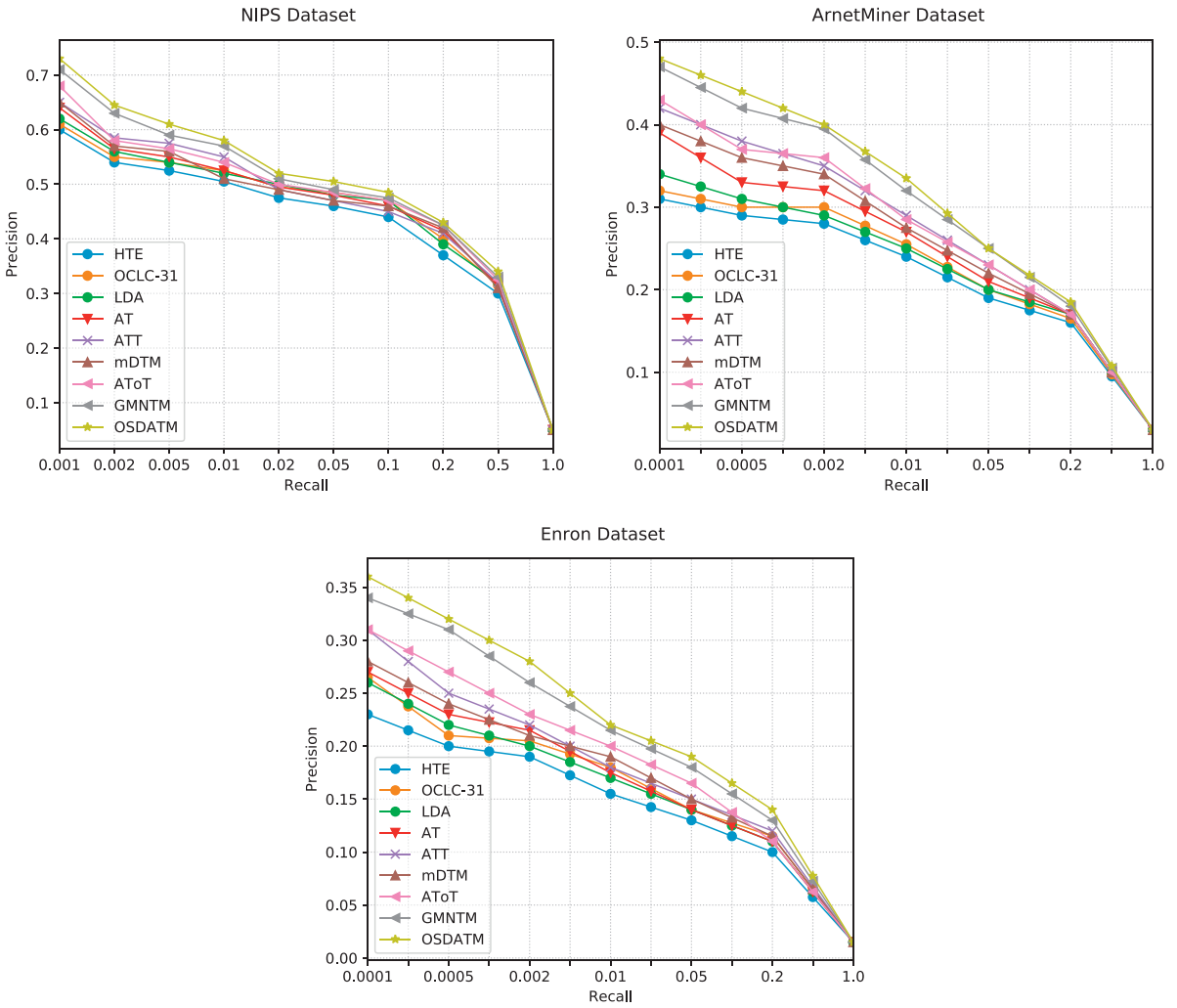


Fig. 3. Comparison of Precision-Recall curves for document retrieval on there datasets: NIPS papers (top), ArnetMiner abstracts (middle) and Enron emails (bottom).

Table 2
Average KL divergence between topics.

Data	HTE	OCLC-31	LDA	AT	ATT	mDTM	AToT	GMNTM	OSDATM
NIPS	0.5745	0.5445	0.5425	0.5695	0.5820	0.5723	0.5712	0.5782	0.5824
ArnetMiner	0.5456	0.5542	0.5631	0.5742	0.5688	0.5732	0.5709	0.5728	0.5793
Enron	0.5745	0.5821	0.5936	0.6042	0.6125	0.6223	0.6240	0.6251	0.6303

in test sets are treated as queries. For each query, documents in the database are ranked by using cosine distance as the similarity metric. The retrieval task is performed separately for each label, and the results are averaged. Fig. 3 illustrates the precision-recall curves with 100 topics for OSDATM and baseline methods. We see that for the NIPS dataset, the proposed model perform slightly better than the other models, while for the ArnetMiner and Enron datasets, OSDATM and GMNTM achieve a significant improvement. Since ArnetMiner and Enron datasets contain a greater amount of texts, OSDATM and GMNTM consider the ordering of words so that they are more powerful in capture the semantics of the documents. As expected, OSDATM achieves better results than GMNTM on all the datasets since OSDATM integrates the authorship and temporal information into GMNTM, which is beneficial to document modeling.

4.4.4. Topic distance

Distances between topics can also be measured numerically. Table 2 shows the average distance of word distributions between all pairs of topics, as measured by KL Divergence. In all three data sets, the topics generated by OSDATM are more distinct from each other. Partially because the OSDATM model strives to separate events that occur during different time

Table 3
Topic words discovered from the NIPS dataset.

OSDATM Topic words				AToT Topic Words			
Network	speech	bayesian	reinforcement	learning	HMM	model	state
Neural	recognition	posterior	policy	network	data	bayesian	learning
Weight	HMM	prior	belief	neural	speech	prior	belief
Activation	acoustic	network	action	input	mixture	monte	policy
Neurons	speaker	parameters	reward	model	figure	statistics	function
Non-linear	voice	sampling	agent	training	input	sample	states
Artificial	continuous	probability	MDP	data	suffix	posterior	action
Learning	phoneme	likelihood	optimal	figure	model	priors	reinforcement
Backpropagation	singer	inference	function	networks	acoustic	input	data
Training	gaussian	statistics	learning	function	saul	set	actions
GMNTM Topic words				**LDA Topic Words**			
Network	speech	bayesian	reinforcement	learning	speech	prior	action
Neural	HMM	posterior	learning	network	data	model	data
Weight	voice	likelihood	gradient	function	input	data	reinforcement
Function	context	distribution	belief	data	suffix	monte	decision
Neurons	recognition	parameters	policy	input	figure	bayesian	function
Non-linear	speaker	sampling	reward	network	model	sample	states
Training	time	training	optimal	neural	prior	network	learning
Layer	dynamic	prior	function	figure	voice	input	policy
Activation	continuous	probability	decision	set	speaker	statistics	data
Input	neural	learning	MDP	function	mixture	posterior	state

spans, and in real-world data, time and authorship differences are often correlated with word distribution differences that would have been more difficult to tease apart otherwise.

4.5. Qualitative evaluation

To evaluate the proposed model qualitatively, we present the topics discovered by the OSDATM model and analyze whether meaningful semantics have been captured. Due to the space limit, we only report four topics extracted by our model and compare them with topics from the three strong baseline methods (i.e., AToT, GMNTM, LDA). In each topic, we visualize it with the top 10 words which are most likely generated from the topic (see Eq. (14)).

The topics discovered by the OSDATM and AToT from the NIPS dataset are reported in Table 3. The 4 topics in the first row are quite specific representations of different topics that have been popular at the NIPS conference over the time-period 1987–99: *Neural networks, Speech recognition, Bayesian learning, and Reinforcement learning*. These topics can be easily interpreted according to the corresponding top words. Immediately, we see that the results of the two models are different in nature. OSDATM is able to discover the topics that consist of words having similar semantics. For example, in OSDATM, "speech" and "speaker" are in the same topic (*Speech recognition*), because they have strong semantic connections. The AToT model, which does not use the context information, is unable to put them into the same topic. In fact, AToT finds general words such as "data" and "figure", which are common and highly probable in many other topics as well.

Table 4 shows the topics discovered by the OSDATM and AToT from the ArnetMiner dataset. The 4 topics in the first row are *Software Engineering, Natural language processing, Information extraction and Wireless network*. Compared with the top words discovered by the AToT model, the top words obtained by the OSDATM model can better summarize the corresponding topics.

Table 5 shows similar types of results for the four selected topics (i.e., *Computer system, Business, California crisis, and Management*) from the Enron emails data set. Compered to the ArnetMiner and NIPS datasets, the Enron data is more informal that tends to have many misspellings, slang terms and shortened forms of words. We Our model works particularly well on this large-scale informal dataset. For example, the first topic in our model can be summarized by observing the words "database, system, install, server, ISO and SUN". It clearly gives the sense of a *Computer system* topic. On the other hand, because of the prevalence of generic words in AToT, some highly related words (e.g., "electricity" and "FERC" for *California crisis* topic) are not ranked high enough to be shown in the top 10 word list.

4.6. Topic distribution over time

Since the semantics of the topics can reflect authors' interests, the significant change of the topics usually indicates the occurrence and the end of events. Due to the limited space, we only demonstrate the change curve of chosen topics generated by OSDATM for Enron dataset, since there are several well-known events happening, which are related to Enron, and the identified dynamic evolution of the topics can be evaluated easily. From Fig. 4, we observe that OSDATM is able to capture the dynamic evolution of the topics. For example, as shown in Fig. 4, the Computer System topic in Enron dataset has its peak around October, 2000 and July, 2001. After examining relevant emails, it appears that there are two serious

Table 4
Topic words discovered from the ArnetMiner dataset.

OSDATM Topic words				AToT Topic Words			
Software	language	web	wireless	software	parsing	web	wireless
Architecture	parsing	semantic	protocol	design	model	search	sensor
Componet	natural	text	mobile	system	supervised	resources	networks
Design	processing	information	sensor	programming	text	extraction	model
Test	text	machine	networks	field	mining	data	technology
Engineering	sentiment	rank	cellular	method	data	evaluation	equipment
Oriented	analysis	knowledge	communication	engineering	language	information	station
Programming	discourse	search	capacity	computer	evaluation	model	mobile
System	syntax	website	connections	tools	resource	system	devices
Develop	grammar	unstructured	smartphones	report	corpus	document	system
GMNTM Topic words				**LDA Topic Words**			
System	processing	extraction	wireless	software	learning	information	model
Software	parsing	web	digital	report	text	web	mobile
Friendly	classification	text	industry	system	input	data	wireless
Design	text	information	remote	programming	parsing	internet	model
Programming	language	semantic	protocol	computer	language	page	history
Robust	grammar	rank	network	method	data	link	service
Oriented	model	knowledge	capacity	input	mining	model	network
Engineering	discourse	search	controls	test	evaluation	input	energy
Develop	natural	automatic	communication	data	natural	system	input
Architecture	syntax	unstructured	sensor	design	processing	search	system

Table 5
Topic words discovered from the Enron dataset.

OSDATM Topic words				AToT Topic Words			
Database	buy	california	management	schedule	deal	power	company
System	business	gas	group	database	news	crisis	monday
Computer	sell	power	chairman	final	stock	capacity	business
Install	trade	energy	committee	friday	share	intent	energy
Server	invest	FERC	company	ISO	trade	gas	year
device	finance	crisis	business	device	year	california	CEO
ISO	energy	market	feedback	error	business	Enron	crisis
Memory	market	electricity	regulatory	computer	company	pipelines	Enron
SUN	stock	pipelines	Enron	system	would	would	chairman
Outage	securities	capacity	CEO	file	finance	energy	trade
GMNTM Topic words				**LDA Topic Words**			
Storage	market	gas	management	input	deal	trader	system
Device	business	crisis	Enron	database	data	energy	company
Schedule	power	power	employee	final	friday	california	data
Install	buy	sell	company	system	trade	system	energy
computer	share	market	computer	power	year	criss	management
Error	sell	california	regulatory	hour	buy	computer	year
Router	trade	nature	CEO	task	system	data	Enron
SUN	stock	energy	regulatory	time	finance	would	manage
Operation	billion	business	committee	system	would	price	California
Server	manage	company	group	file	employee	email	CEO

outage around October, 2000 and July, 2001, respectively. We can also infer these two events from the list of top words of the topic of Computer System in October, 2000 and July, 2001. The word “outage” has the highest priority in these timestamps. The California Crisis topic has its peak around January, 2001 and August, 2001, since California governor Davis declares a state of emergency in January and the energy prices normalized in September. The peaks of the Business topic indicates two events: first, Enron’s Board of Directors exempted CFO Fastow from the company’s code of ethics so that he can run a private equity fund – LJM1 that will raise money for and do deals with Enron in June, 1999. The LJM Funds become one of the key tools for Enron to manage its balance sheet. Second, in March 2001, Enron scheduled unusual analyst conference call to boost the stock. For the Management topic, we cannot explain the change curve in details since we have no idea about the personnel changes in Enron. Nevertheless, we know that Enron’s Board of Directors exempted CFO Fastow from the company’s code of ethics in June, 1999 and Lay retired as CEO and was replaced by Skilling in February, 2001. The change curve of Management topic precisely reflects these two event.

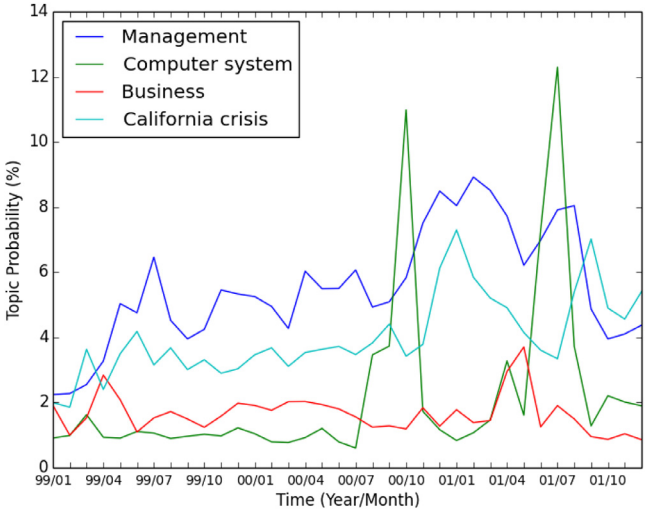


Fig. 4. Change curve of the selected topics for Enron dataset.

Table 6
Top five authors who are mostly interested in Neural networks topic over the period from 1990 to 1999.

Time	Authors who are interested in Neural networks topic
1990	Sejnowski_T, Waibel_A, Tesauro_G, Kailath_T, Goodman_R
1991	Giles_C, Waibel_A, Goodman_R, Bengio_Y, Lee_Y
1992	Goodman_R, Waibel_A, Ruppin_E, Chen_H, Sun_G
1993	Bengio_Y, Ruppin_E, LeCun_Y, Tresp_V, Giles_C
1994	Ruppin_E, Giles_C, Waibel_A, Bengio_Y, Spence_C
1995	Bengio_Y, Giles_C, Ruppin_E, LeCun_Y, Horn_D
1996	Bengio_Y, Giles_C, Horn_D, LeCun_Y, Tresp_V
1997	Bengio_Y, Tresp_V, Niranjan_M, LeCun_Y, Giles_C
1998	Bengio_Y, LeCun_Y, Tresp_V, Spence_C, Hinton_G
1999	Bengio_Y, Tresp_V, Bengio_S, Doucet_A, LeCun_Y

Table 7
Top five authors who are mostly interested in Speech recognition topic over the period from 1990 to 1999.

Time	Authors who are interested in Speech recognition topic
1990	Waibel_A, Lippmann_R, Bengio_Y, Bourlard_H, De-Mori_R
1991	Waibel_A, Bengio_Y, Lippmann_R, Tebelskis_J, De-Mori_R
1992	Waibel_A, Kawato_M, Tebelskis_J, Franco_H, Morgan_N
1993	Kawato_M, Hirayama_M, Waibel_A, Franco_H, Makhoul_J
1994	Makhoul_J, Zhao_Y, Waibel_A, Kawato_M, Hirayama_M
1995	Morgan_N, Hochberg_M, Waibel_A, Makhoul_J, Konig_Y
1996	Rigoll_G, Robel_A, Gray_M, Waibel_A, Sejnowski_T
1997	Rigoll_G, Houde_D, Willett_D, Gray_M, Waibel_A
1998	Saul_L, Rahim_M, Waibel_A, Rigoll_G, Kawato_M
1999	Yang_H, Hermansky_H, Rigoll_G, Niranjan_M, Waibel_A

4.7. Author interest evolution analysis

An interesting trait of OSDATM is its capability of discovering those authors who have specific interests. This interest-tracking trait is important in personalized recommendation and in assisting investigators finding or monitoring criminal activities. In order to analyze the dynamics of author interests, multiple documents written by the same author in different periods are required. However, in ArnetMiner and Enron datasets, the authors are sparse and most authors tend to write either a small number of documents or within a short time period. There are few authors continuously writing sufficient number of documents for their interest evolution to be analyzed. Therefore, in this experiment, we only evaluate OSDATM on NIPS dataset between 1990–1999. Due to the limited space, we only illustrate the results of two popular topics (i.e., Neural networks, Speech recognition). For each topic, we select the top five most likely authors according to Eq. (17). The results are illustrated in Tables 6 and 7 respectively. The author lists are quite sensible. For example, Bengio_Y is

Table 8
Top three topics in which Michael I. Jordan was mostly interested during the period from 1990 to 1999.

Time	Michael I. Jordan
1990	Dynamical model, Neural networks, Expert networks
1991	Expert networks, Supervised learning, Neural networks
1992	Dynamical model, Expert networks, Supervised learning
1993	Mixture model, Dynamical model, Neural networks
1994	Mixture model, Reinforcement learning, HMM
1995	Mixtures model, HMM, Reinforcement learning
1996	Belief networks, Graphical models, Mixture model
1997	Belief networks, Graphical models, EM
1998	EM, Mixture model, Unsupervised learning
1999	Graphical models, Bayesian learning, Mixture model

Table 9
Top three topics in which Terrence J. Sejnowski was mostly interested during the period from 1990 to 1999.

Time	Terrence J. Sejnowski
1990	Neural networks, Speech recognition, Vision
1991	Neural networks, Vision, Unsupervised learning
1992	Vision, Reinforcement learning, Unsupervised learning
1993	Reinforcement learning, Neural networks, Vision
1994	Reinforcement learning, Neural networks, EM
1995	Reinforcement learning, Vision, Neural networks
1996	Vision, Speech recognition, Bayesian learning
1997	Vision, ICA, Reinforcement learning
1998	Unsupervised learning, ICA, EM
1999	Mixture model, ICA, Vision

widely-known in the research area of Neural networks over the whole period, while Sejnowski_T made significant contribution to Neural networks topic in the early phase. Similarly, for the Speech recognition topic, almost all of the five most likely authors are frequent contributors of papers at NIPS conference on speech recognition each year.

We also extract the interests of individual authors over time from the results of OSDATM. We select two famous authors, i.e., Michael I. Jordan and Terrence J. Sejnowski, for this experiment. For each author, we show the top three topics in which they are mostly interested between 1990 and 1999 (by Eq. (17)), as shown in Tables 8 and 9 respectively. From these results, we can see that these topics are varying during the period. For instance, Jordan's research interests are mainly focused on*Expert networks* and *Dynamical model* in the early years, then it expanded to *Reinforcement learning* and *Graphical models*.This can be verified from his homepage. Similarly, by examining Sejnowski's homepage, most of the topics of interests to Sejnowski are also consistent with the truth.

5. Conclusion and future work

To alleviate the bag-of-words assumption and capture the semantic regularities in language, in this paper, we proposed an Ordering-sensitive and Semantic-aware Dynamic Author Topic Model (OSDATM) which combined the advantages of both topic models and neural language models. In OSDATM, the topic distributions of documents and the vector representations of words, sentences, documents and author interests are jointly learned. In contrast to the existing work, OSDATM was the first model that explicitly characterized the interest evolution of individual authors and took the word orders as well as semantic regularities in language into consideration. We conducted extensive experiments on three widely used publicly available datasets. The experimental results showed that by modeling the dynamic author interest, the proposed model could capture topics of words more accurately than most state-of-the-art topic models. We further showed that the proposed model effectively discovered the evolution of author's interest over time.

In future work, we plan to extend OSDATM to a non-parametric model that automatically detects the number of topics of the corpus. In addition, the proposed model has many potentially useful applications in natural language processing, such as entity recognition, information extraction and sentiment analysis. These applications deserve the further study. Another interesting extension to our model would be the ability to handle topical phrases and overcome the problem that only unigrams are allowed in the topic representation.

Acknowledgments

This work was also partially supported by the National Natural Science Foundation of China (Grant No. 61803249), the Shanghai Sailing Program (Grant No. 18YF1407700), the SIAT Innovation Program for Excellent Young Researchers (Grant No.

Y8G027), and the CAS Pioneer Hundred Talents Program (Grant No. Y84402). Min Yang was sponsored by CCF-Tencent Open Research Fund.

References

[1] S. Bird, Nltk: the natural language toolkit, in: Proceedings of the COLING/ACL on Interactive Presentation Sessions, Association for Computational Linguistics, 2006, pp. 69–72.
[2] C.M. Bishop, Pattern Recognition and Machine Learning, 1, Springer, New York, 2006.
[3] D.M. Blei, Probabilistic topic models, Commun. ACM 55 (4) (2012) 77–84.
[4] D.M. Blei, J.D. Lafferty, Dynamic topic models, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 113–120.
[5] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[6] K.-Y. Chen, L. Luesukprasert, T.C. Seng-cho, Hot topic extraction based on timeline analysis and multidimensional sentence modeling, IEEE Trans. Knowl. Data Eng. 19 (8) (2007).
[7] A. Daud, Using time topic modeling for semantics-based dynamic research interest finding, Knowl. Based Syst. 26 (2012) 154–163.
[8] D. Griffiths, M. Tenenbaum, Hierarchical topic models and the nested chinese restaurant process, Adv. Neural Inf. Process Syst. 16 (2004) 17.
[9] T.L. Griffiths, M. Steyvers, Finding scientific topics, Proc. Natl. Acad. Sci. 101 (suppl 1) (2004) 5228–5235.
[10] T.L. Griffiths, M. Steyvers, D.M. Blei, J.B. Tenenbaum, Integrating topics and syntax, in: Proceedings of the Advances in Neural Information Processing Systems, 2004, pp. 537–544.
[11] A. Gruber, Y. Weiss, M. Rosen-Zvi, Hidden topic Markov models, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2007, pp. 163–170.
[12] N. Kawamae, Author interest topic model, in: Proceedings of the 33rd international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2010, pp. 887–888.
[13] B. Klimt, Y. Yang, The enron corpus: a new dataset for email classification research, in: Machine Learning: ECML, Springer, 2004, pp. 217–226.
[14] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International conference on machine learning, 2014, pp. 1188–1196.
[15] T. Li, B. Kveton, Y. Wu, A. Kashyap, Incorporating metadata into dynamic topic analysis, in: Proceedings of the BMAW-12 Preface, 2012.
[16] C. Lin, Y. He, R. Everson, S. Ruger, Weakly supervised joint sentiment-topic detection from text, CIKM 24 (6) (2012) 1134–1145.
[17] B.M. Marlin, Modeling user rating profiles for collaborative filtering, in: Proceedings of the Advances in Neural Information Processing Systems, 2003.
[18] J.D. Mcauliffe, D.M. Blei, Supervised topic models, in: Proceedings of the Advances in Neural Information Processing systems, 2008, pp. 121–128.
[19] A. McCallum, X. Wang, A. Corrada-Emmanuel, Topic and role discovery in social networks with experiments on enron and academic email., J. Artif. Intell. Res.(JAIR) 30 (2007) 249–272.
[20] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR (2013) arXiv:1301.3781.
[21] A. Mnih, G.E. Hinton, A scalable hierarchical distributed language model, in: Proceedings of the Advances in Neural Information Processing Systems, 2009, pp. 1081–1088.
[22] A. Mnih, K. Kavukcuoglu, Learning word embeddings efficiently with noise-contrastive estimation, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 2265–2273.
[23] A. Mnih, Y.W. Teh, A fast and simple algorithm for training neural probabilistic language models, in: International Conference on Machine Learning, 2012.
[24] F. Osborne, E. Motta, Klink-2: integrating multiple web sources to generate semantic topic networks, in: Proceedings of the International Semantic Web Conference, Springer, 2015, pp. 408–424.
[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[26] M. Rosen-Zvi, T. Griffiths, M. Steyvers, P. Smyth, The author-topic model for authors and documents, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2004, pp. 487–494.
[27] N. Srivastava, R.R. Salakhutdinov, G.E. Hinton, Modeling documents with deep Boltzmann machines, in: The Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, 2013, pp. 616–624.
[28] M. Steyvers, T. Griffiths, Probabilistic topic models, in: Handbook of Latent Semantic Analysis, 427, Lawrence Erlbaum Associates, 2007, pp. 424–440.
[29] J. Tang, J. Zhang, Modeling the evolution of associated data, Data Knowl. Eng. (2010) 965–978.
[30] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: extraction and mining of academic social networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 990–998.
[31] Y.W. Teh, M.I. Jordan, M.J. Beal, D.M. Blei, Hierarchical dirichlet processes, J. Am. Stat. Assoc. 101 (476) (2006).
[32] T. Velden, K.W. Boyack, J. Gläser, R. Koopman, A. Scharnhorst, S. Wang, Comparison of topic extraction approaches and their results, Scientometrics 111 (2) (2017) 1169–1221.
[33] H.M. Wallach, Topic modeling: beyond bag-of-words, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 977–984.
[34] C. Wang, D. Blei, D. Heckerman, Continuous time dynamic topic models, in: The Twenty-Fourth Conference on Uncertainty in Artificial Intelligence, 2012, pp. 579–586.
[35] S. Wang, R. Koopman, Clustering articles based on semantic similarity, Scientometrics 111 (2) (2017) 1017–1031.
[36] X. Wang, 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, ACM, 2006, pp. 424–433.
[37] X. Wang, N. Mohanty, A. McCallum, Group and topic discovery from relations and text, in: Proceedings of the 3rd International Workshop on Link Discovery, ACM, 2005, pp. 28–35.
[38] X. Wei, W.B. Croft, Lda-based document models for ad-hoc retrieval, in: Proceedings of the 29th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2006, pp. 178–185.
[39] S. Xu, Q. Shi, X. Qiao, L. Zhu, H. Jung, S. Lee, S.-P. Choi, Author-topic over Time (Atot) a dynamic users interest model, in: Mobile, Ubiquitous, and Intelligent Computing, Springer, 2014, pp. 239–245.
[40] M. Yang, T. Cui, W. Tu, Ordering-sensitive and semantic-aware topic modeling, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-2015), 2015.
[41] M. Yang, J. Mei, F. Xu, W. Tu, Z. Lu, Discovering author interest evolution in topic modeling, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2016, pp. 801–804.