Embedded Event and Trend Diagnostics to extract LDA Topic Models on Real Time Multi-Data Streams

WALISA ROMSAIYUD[1] AND WICHIAN PREMCHAISWADI[2]
Graduate School of Information Technology
Siam University
38 Petkasem rd., Prasri-charoen, Bangkok, 10160
THAILAND
walisar@gmail.com[1], wichian.prem@siam.edu[2]

Abstract: - Existing latent dirichlet allocation (LDA) methods make use of random mixtures over latent topics and each topic is characterized by a distribution over words from both batch and continuous streams over time. However, it is nontrivial to explore the correlation with the existence of different among multiple data streams, i.e., documents from different multiple data streams about the same topic may have different time stamps. This paper introduces a new novel algorithm based on the latent dirichlet allocation (LDA) topic model. The algorithm includes two main methods. The first method introduces a principled approach to detecting surprising events in documents. The embedded events and trends of the model parameters are used for filtering surprising events and preprocessing documents in an associated time sequence. The second method suits real time monitoring and control of the process from multiple asynchronous text streams. In the experiment, these two methods were alternatively executed and after iterations a monotonic convergence can be guaranteed. The advantages of our approach were justified through extensive empirical studies on two real data sets from three news and micro-blogging respectively.

Key-Words: - Latent Dirichlet allocation (LDA), Topic model, Asynchronous Text Stream, Time-Stamped Documents, Fuzzy K-Mean Clustering, Semantic Analysis

1 Introduction

Tremendous and potentially infinite volumes of data streams are often generated by real-time surveillance systems, micro-blogging (twitter, facebook or google+), weather data, weblog articles, news or mail list and other dynamic environments. The stream data flow in and out of a computer system continuously and with varying update rates. Moreover, the effective processing of stream data, new data structures, techniques, and algorithms are needs for extracting valuable knowledge tasks are more complex when processing text documents that arrive in discrete or continuous forms of asynchronous data in real-time.

A noticeable amount of work is conducted on the topic of latent dirichlet allocation (LDA) modeling, Blei [7] with the main objective to enhance the descriptive and/or predictive model of the data's thematic structure based on the embedded prior knowledge about the domain's semantics. The basic idea in LDA is that documents are represented by a mixture of topics where each topic is a latent multinomial variable characterized by a distribution over a fixed vocabulary of words. The evolution of the LDA's generative process for documents is achieved by considering Dirichlet priors on the document distributions over topics and on the topic distributions over words. This emerging approach has been effectively applied to discover useful structures in many kinds of documents including emails, the scientific literature [13], libraries of digital books [33], and news records [30]. Furthermore, OLDA is an online version of the LDA model that is capable of processing text streams [2]. The OLDA model considers the temporal ordering information and presupposes that the documents arrive in discrete time slices. At each time slice t of a fixed size ε, e.g. an hour, a day, or a year, a stream of documents, $S^t = \{d_1, . . . , d_D{}^t\}$, of variable size, D^t , is received and ready to be processed. A document d received at time t is represented as a vector of word tokens, $w^t{}_d = \{w^t{}_{d1}, . . . ,w^t{}_{dNd}\ \}$. Then, an LDA topic model with K components is used to model the recently arrived documents. The generated model, at a given time, is used as a prior for LDA at the following time slice, when a new data stream is available for processing. Then, it was assumed that given a document in the sequence, the time stamp of the document was generated provisionally independently from word. In Blei et al.,[7], the authors introduced hyper-parameters that advance over time in state transfer models in the sequence. For each time slice, a hyper-parameter is allocated with a state by a probability distribution, given the state on the prior time slice. In Mei Q. [22], the time dimension of the sequence was cut into time slices and topics were discovered from documents in each slice independently. Consequently, in multiple-sequence cases, topics in each sequence can only be estimated separately and potential correlation between topics in different sequences, both semantically and temporally, could not be fully explored. In [7, 20], the semantic correlation between different topics in static text collections was considered. Similarly, Zhai et al.,[32] explored common topics in multiple static text collections. Wang et al.,[28] studied a generalized asynchronous distributed learning scheme with applications in topic mining. However, in their work the term “asynchronous” means set independent Gibbs samplers which communicate with each other in an asynchronous manner. Therefore, their problem statement is basically different from ours. In this paper, we address the problem of nontrivial to investigate the correlation with the existence of different among multiple data stream, i.e., documents from different multiple Text Streams about the same topic may have different time stamps and propose a helpful method to solve it.

The general idea in of our method is to distinguish events and trends that evolving content in text streams and to build up a shared reinforcement process. We used a web crawler for collecting the heterogeneous data for changes within the content of that data stream. The fuzzy k-mean used for clustering group of all documents that similarity distance in each time stamp. In each cluster, we extracted keyword/term from data stream by counting the number of documents that contain a given term/keyword. For each document, a time stamp is identified, allowing our analysis to be temporally. However, information analysis professionals often seek to discover and track surprising events and emerging trends over time and in a timely fashion. For example, text that contains e-mail messages may provide useful feedback about products, so that you could use the Term Extraction transformation to extract the topics of discussion in the messages, as a way of analyzing the feedback.

We began with embedding an objective function in the inference process of the LDA model in order to boost the discovered topic from the asynchronous text streams, which will be referred to hereafter as the test document. Our major contributions are the following:

1. We address the problem of LDA topic models from multiple asynchronous text streams. To the extent of our knowledge, this is the initial effort to solve this problem.
2. We applied the events and trends detection in text streams for monitoring a stream of text or messages for changes within the content of that data streams.
3. We formalize our problems by introducing a principled probabilistic framework and propose an objective function for to solve our problems.
4. We develop a novel interchange optimization algorithm to maximize the objective function with a theoretically guaranteed (local) optimum.

The effectiveness and advantage of our methods are validated by a wide-ranging empirical study on two real-world data sets (Reuters-21578 and twitter dataset).

The rest of the paper is structured as follows: In Section 2, we address some supplementary detail about a variety of Latent Dirichlet allocation (LDA) and event detection. We discussed the associated work in Section 3. Section 4 proposes extensions of our model and algorithm. Section 5 presents the empirical results. Concluding remarks and future extensions are discussed in Section 6.

2 Background

There are a number of theories concerning LDA topic models and Real-time event detection, but this article only discusses a subset of them. The specific theories discussed have been chosen because they appear to be most applicable to the analysis and design of detection the event and trend of text streams in real-time situation.

2.1 Latent Dirichlet Allocation (LDA)

Blei et al.,[7] proposed a topic model called Latent Dirichlet allocation (LDA). The basic idea of LDA is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. A process flow of LDA is illustrated in Fig.1.

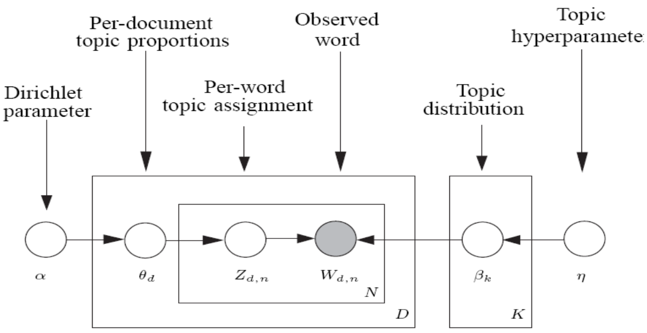

Fig. 1. A Process Flow of Latent Dirichlet allocation (LDA)

In Fig. 1, the documents (θ) are not directly linked to the words (w). However, this relationship is governed by additional latent variables, z, introduced to represent the responsibility of a particular topic in using that word in the document, i.e. the topic(s) that the document is focused on. By introducing the Dirichlet priors α and β over the document and topic distributions, respectively, the generative model of LDA is complete and is capable of processing unseen documents.

LDA assumes the following generative process for each document **w** in a corpus *D*:

(1) Choose $N \sim Poisson(\xi)$.

(2) Choose $\theta \sim Dir(\alpha)$.

(3) For each of the *N* words w_n:

(a) Choose a topic $z_n \sim Multinomial\ (\theta)$.

(b) Choose a word w_n from $p(w_n\,|z_n,\beta)$, a multinomial probability conditioned on the topic z_n.

Several simplifying assumptions are made in this basic model, some of which we remove in subsequent sections. First, the dimensionality k of the Dirichlet distribution (and thus the dimensionality of the topic variable z) is assumed known and fixed. Second, the word probabilities are parameterized by a k_V matrix β where $\beta_{ij} = p(w_j = 1_{jzi} = 1)$, which for now we treat as a fixed quantity that is to be estimated. Finally, the Poisson assumption is not critical to anything that follows and more realistic document length distributions can be used as needed. Furthermore, note that N is independent of all the other data generating variables (θ and z). It is thus an ancillary variable and we will generally ignore its randomness in the subsequent development. A k-dimensional Dirichlet random variable θ can take values in the $(k-1)$-simplex (a k-vector θ lies in the $(k-1)$-simplex if $\theta_i >= 0$, $\sum k_{\,i=1}\ \theta_i = 1$), and has the following probability density on this simplex:

$$p(\theta\,|\infty) = \frac{(\sum_{i=1}^{k} \infty i)}{\prod_{i=1}^{k} \quad (\infty i)}\ \theta_1^{\alpha i-1} \ldots . \theta_k^{\alpha k-1} \qquad (1)$$

Where the parameter ∞ is a k-vector with components $\alpha i > 0$; $\Gamma(x)$ is the Gamma function.

2.2 Event detection

Event detection is conducted on formal document collections that shows somehow related to a number of undiscovered events. However, finding these events/terms in a timely fashion is not an easy task. A surprise event/term is an arbitrary classification of a space/time region. The statistic looks for deviations in the number of occurrences of a specific term normalized by the total number of documents (within the same time interval). The formula used for the chi-square statistic is

$$\mathrm{x}^2 = \frac{n \ldots \left(|n11n22 - n12n21| - \frac{1}{2}yn\right)2}{n1.n2.n1.n2} \qquad (2)$$

From equation (2), y is either 0 or 1. If y is 1, the Yates continuity correction is applied for the low sample size in which the count in at least one cell is less than or equal to 5.

The Gaussian statistic is based on comparing the observed value x_i to the average over the previous values $(1/np)$ $\sum x_j$, normalized by the standard deviation of these previous values.

$$\mathrm{G} = \frac{xi - \frac{1}{np} \sum_{j=1-np}^{i-1} xj}{s.(1 + \frac{1}{np})} \qquad (3)$$

From equation (3), np is the number of time intervals in the previous time windows and s is the standard deviation.

Engel et al combine the previous algorithms (chi-square and Gaussian) for the surprise statistic [Engel et al 2010], as follows:

$$C_{surprise} = \sqrt{X2} + |\,\mathrm{G}\,| \qquad (4)$$

We focus on the event and term for calculating the surprise statistic using ratio of the maximum value of the likelihood functions over the entire parameter space. An event might have actively participating agents, passive factors, products, and a location in space and time [4, 6]. We interest in using the events obtained from news and tweets that represent the large collection of time-stamped text sequence. These events have several properties: i) they are of large scale (many users experience the event), ii) they particularly influence people's daily life (for that reason, they are induced to tweet about it or reports on news), and iii) they have both spatial and temporal regions (so that real-time location estimation would be possible). Such events include social events such as large parties, sports events, exhibitions, accidents, and political campaigns.

3 Related Work

Topic mining has been extensively studied in the literature, starting with the Topic detection and Tracking (TDT) project [20, 31], which aimed to find and track topics (events) in news with clustering-based techniques.

Later on, probabilistic generative models were introduced into use, such as Probabilistic Latent Semantic Analysis (PLSA) [12], Latent Dirichlet Allocation (LDA) [3], and their derivatives [4-6]. In many real applications, text collections carry generic temporal information and, thus, can be considered as text sequences. To capture the temporal dynamics of topics, various methods have been proposed to discover topics over time in text sequences [1, 15-18, 21, 25, 28]. However, these methods were only designed to extract topics from a single sequence. For example, in Ting et al. [27] and Paradimitrious et al., [23], which adopted the generative model, time stamps of individual documents were modelled with a random variable, either discrete or continuous. Then, it was assumed that given a document in the sequence, the time stamp of the document was generated conditionally independently from word. In Williams G.,[32], the authors introduced hyper-parameters that evolve over time in state transfer models in the sequence.

A very recent work by [Wang et al. 2012] first proposed a topic mining method that aimed to discover common (bursty) topics over multiple text sequences. They find topics that shared common time distribution over different sequences by assuming that the sequences were synchronous, or coordinated. Based on this premise, documents with same time stamps are combined together over different sequences so that the word distributions of topics in individual sequences can be discovered. As a contrast, in our proposed, we aim to find topics that are common in semantics, while having asynchronous time distributions in different sequences.

Sakaki et al.,[25] investigate the real-time interaction of events such as earthquakes, in Twitter, and propose an algorithm to monitor tweets and to detect a target event. They consider each Twitter user as a sensor and apply Kalman filtering and particle filtering, which are widely used for location estimation in ubiquitous/pervasive computing. As a contrast, in our proposed, we detect surprising event/term from the unit of calculation that represent as term or keyword that can be a single word or multiple words and pre-processing of the time-sequence documents.

The work presented in this paper is an extension of the previous studies AlSumait et al.,[3]; Engel et al.,[12]; Sakaki et al.,[25] and Whitney et al.,[31] in the following ways. First, common topics are extracted from multiple sequences based on the adjusted time-stamps that update the time stamps of documents in all sequences by assigning them to most relevant topics. ~~Seconds~~, we developed and tested a new programming model that can be embedded in the inference process of the LDA model. It can enhance the discovered surprise event from the text data (test documents). Third, the results obtained from our model was evaluated and compared with of other models using real-world data on a large volume of dataset.

4 Objective Function and Algorithm

In this section, we formally define our algorithm for extracting common topics from multiple asynchronous text sequences.

4.1 Objective Function

We derive out objective function, which is to maximize the likelihood estimation subject to certain constraints. The main symbols are illustrated in Table 1.

Table I. The symbols and meanings

Symbols	Description
d	Document
t	Timestamp
w	Word
z	topic
M	Number of text sequences
T	Length of sequences
V	Number of distinct words
K	Number of topics (given by users)

We define text sequence as follows:

—Definition 1 (Text Sequence)

S is a sequence of N documents (d_1,…, d_n). Each document d is a collection of words over vocabulary V and indexed by a unique time stamp $t \in \{1,\ldots,T\}$

—Definition 2 (Common Topic)

The documents $\{d \in S_m : 1 <= m <= M\}$ are modeled by a discrete random variable d. The words are modeled by a discrete random variable w over vocabulary V. The time stamps are modeled by a discrete random variable t over {1,…,T}, At last, the common topics Z are encoded by a discrete random variable $z \in 2\ \{1,2,\ldots,K\}$

The generating process is as follows:

(1) Pick a document d with probability p(d).
(2) Given the document d, pick a time stamp t with probability p(t|d), where p(t=t|d) = 1

for some t. This means that a given document only has one time stamp.

(3) Given the time stamp t, pick a common topic z with probability p(z | t) ~ Mult(□).

(4) Given the topic z, pick a word w with probability p(w | z) ~ Mult(φ).

The according to the generative process, the probability of word w in document d is:

$$p(w \mid d) = \sum_{t,z} p(d)p(t|d)p(z|t)\, p(w|z) \qquad (5)$$

—Definition 3 (Asynchronism)

Given M text sequences {Sm : 1 <= m <= M}, in which documents are indexed by time stamps {t:1 <= t<=T}, asynchronism means that the time stamps of the documents sharing the same topic in different sequences are not properly aligned.

arg max □ = $p(t|d), p(z|t), p(w|z)$
(6)

We calculate the maximize the likelihood function □ by adjusting $p(z|t)$ and $p(w|z)$ as well as $p(t|d)$ subject to the constraint of preserving temporal order within sequence.

—Definition 4 (Term Frequency*Inverse Document Frequency)

The term count in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term within the particular document. Thus one can calculate the term frequency for the word as the ratio of number of times the word occurs in the document to the total number of words in the document. The inverse document frequency is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$\mathrm{Idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \qquad (7)$$

From equation (7), we assume a document containing 100 words wherein the word cow appears 3 times. Following the previously defined formulas, the term frequency (TF) for cow is then (3 / 100) = 0.03. Now, assume we have 10 million documents and cow appears in one thousand of these. Then, the inverse document frequency is calculated as log(10 000 000 / 1 000) = 4. The tf*idf score is the product of these quantities: 0.03 × 4 = 0.12.

—Definition 5 (Cosine Similarity)

When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, the so-called cosine similarity.

Given two documents $\vec{t_a}$ and $\vec{t_b}$, their cosine similarity is;

$$\mathrm{SIM}_c(\vec{t_a}, \vec{t_b}) = \frac{\vec{ta} * \vec{tb}}{|\vec{ta}| * |\vec{tb}|} \qquad (8)$$

where $\vec{t_a}$ and $\vec{t_b}$ are m-dimensional vectors over the term set $T = \{t_1, \ldots, t_m\}$. Each dimension represents a term with its weight in the document, which is non-negative. As a result, the cosine similarity is non-negative and bounded between [0,1]. An important property of the cosine similarity is its independence of document length. For example, combining two identical copies of a document d to get a new pseudo document d^0, the cosine similarity between d and d^0 is 1, which means that these two documents are regarded to be identical. Meanwhile, given another document l, d and d^0 will have the same similarity value to l, that is, sim($\vec{t_d}$, $\vec{t_l}$) = sim($\vec{t_{d0}}$, $\vec{t_l}$). In other words, documents with the same composition but different totals will be treated identically. Strictly speaking, this does not satisfy the second condition of a metric, because after all the combination of two copies is a different object from the original document. However, in practice, when the term vectors are normalized to a unit length such as 1, and in this case the representation of d and d_0 is the same.

4.2 Our Algorithm

Our proposed algorithm is summarized in Algorithm 1. As illustrated, K is the number of topic specified by users. The initial values of $p(t|d)$ and $c(w, d, t)$ are counted from the original time stamps in the sequences.

Algorithm 1. The LDA Topic Models on Real Time Multi-Data Streams

Input: K, $(\boldsymbol{t}|\boldsymbol{d})$ $\boldsymbol{and}$ $\boldsymbol{c}(\boldsymbol{w}, \boldsymbol{d}, \boldsymbol{t})$;
Output: $(\boldsymbol{w}|\boldsymbol{z}), \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{t}), \boldsymbol{p}(\boldsymbol{t}|\boldsymbol{d})$;
Initialization: compute the weight distribution $D_w(x,y)$ from two types dataset;

```
Generate: generate a topic $\phi^{(t)}_k$ ~ Dirichlet(. | $\boldsymbol{\beta}^{(t)}_k$) ;
Repeat
    Initialize p(z|t) and p(w|z) with random values;
Repeat
    update $p(z|t)$ and $p(w|z)$ ;
Until convergence;
For each word token, $w_{di}$, in document $d$:
    Draw $z^{(t)}_i$ from multinomial $\phi^{(t)}_d$;
        $p(z^{(t)}_i \mid \alpha^{(t)}_d)$;
    Draw $w^{(t)}_{di}$ from multinomial $\phi^{(t)}_{zi;}$
        $p(w^{(t)}_{di} \mid z^{(t)}_{i,} \beta^{(t)}_{zi})$;
End
For m = 1 to M do
    For j = 1 to T do initialize H (1:1, 1:j);
        For i=2 to T do
            For j = 1 to T do
                //Count the standard deviation of
                  the counts in the previous time
                  windows
                $G=\frac{\frac{1}{nc}\sum_{j=1}^{i+nc} xj-\frac{1}{np}\sum_{j=1-np}^{i-1} xj}{\sqrt{\frac{si}{nc}+\frac{sj}{np}}}$
            End
        End
    End
End
Until convergence;
```

The computation complexity of the topic extraction step is approximately O(VMT^3) where V is the size of vocabulary, T the number of different time stamps, K the number of topics, and M the number of sequence.

5 Empirical Evaluation

We evaluated our method on two sets of real-world text sequences, a set of two news article feeds and tweets dataset. To achieve the aims, the following issues were determined.

(1) We explore the underlying asynchronism among text sequences and fix it with our time synchronization techniques;
(2) We try to find meaningful and discriminative common topics from multiple text sequences;
(3) We consistently outperform the baseline method (without time synchronization), but also an improved competitor with one-time synchronization process; and
(4) We embedded semantics from a source by enhancing the generative process of the model parameter to improve the performance.

5.1 Data Sets

We downloaded the Reuters-21578 dataset from David Lewis' page and used the standard "modApté" train/test split. These documents appeared on the Reuters newswire in 1987 and were manually classified by personnel from Reuters Ltd.

The class distribution for these documents can be divided into two sub-collections that usually considered for text categorization tasks as following:

(1) R10; The set of the 10 classes with the highest number of positive training examples.
(2) R90; The set of the 90 classes with at least one positive training and testing example.

Table 2. The Reuters-21578 dataset

Reuters 21578				
#Topics	#train docs	#test docs	#other	Total #docs
0	1828	280	8103	10211
1	6552	2581	361	9494
2	890	309	135	1334
3	191	64	55	310
4	62	32	10	104
5	39	14	8	61
6	21	6	3	30
7	7	4	0	11
8	4	2	0	6
9	4	2	0	6
10	3	1	0	4
11	0	1	1	2
12	1	1	0	2
13	0	0	0	0
14	0	2	0	2
15	0	0	0	0
16	1	0	0	1

We consider single-labeled datasets in table 2. All the documents with less than or with more than one topic were eliminated. For example, classes in R10 and R90 were left with no train or test documents.

Table 3. The distribution of document per class for R8

R8			
Class	#train docs	#test docs	Total #docs
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
Interest	190	81	271
Money-fx	206	87	293
Ship	108	36	144
Trade	251	75	326
Total	**5485**	**2189**	**7674**

From table 3, we call these sets R8. Note that from R10 to R8 the classes corn and wheat, which are intimately related to the class grain, disappeared and this last class lost many of its documents.

Table 4. The example Tweet related with event situation

Twitter		
User	Message	DateTime
1	Happy 30th Birthday to Prince William #morebirthdays	09:10 AM, 20 Jun
2	Next 100 participants that register on a team will get a FREE Braves ticket for next Thursday's game!	9:52 AM, 25 Sep
3	Wants you to join us TOMORROW night at Flip Flops for KARAOKE!!!	About 24 hours ago from facebook
4	If you are hungry…check out John & Jack's Pancake Shack for some good eats!!!! Only $3 a plate	2 minutes ago from web
5	Yeah, I felt today's earthquake. But it just wasn't as good as last July's 3.6 quake; I prefer the earlier stuff.	14 minutes ago via Twitter for iPhone

From table 4, we select short twitter messages posted by five different users. For example, user 1 posts "the Birthday to Prince William on 20 Jun or last 14 minutes ago from time windows have an earthquake".

5.2 Evaluation Metrics

We evaluated the performance of our method using several different metrics. We have three main parameters, namely, $p(t|d)$, $p(z|t)$, and $p(w|z)$. Here, $p(t|d)$ gives the new time stamps of documents after adjustment, $p(z|t)$ indicates the time distribution of extracted topics while $p(w|z)$ gives the word distribution. These parameters are all to be examined in our experiments.

(1) For $p(w|z)$, we evaluate the meaningfulness of extracted topics by examining their top-ranked topical words. We also compute the pairwise *KL-divergence* between topics to evaluate how discriminative they are. (In practice, we normally expect meaningful topics that can be easily understood by human users and we want these topics to be as discriminative as possible, in order to avoid redundant information.)

(2) For p(z|t), we want to see if, after synchronization, our method is able to separate different topics along the time dimension, which would eventually improve the quality of extracted topics.

(3) For p(t|d), we demonstrate how our method adjusts documents' time stamps and fixes the synchronization among sequences.

We also computed the log-likelihood of our method and compared it to that of the baseline method.

Table 5. The top 5 topical word (sorted by probability)

Terms	Related words	Related documents	Related topics
Film	Episode	Film	Film, series, show
	Movie	Cinema of the united kingdom	City, large, area
	Television	History of film	Acid, form, wate
	Actor	Stanley Kubrick	#card, #make, #design
	Comic	B movie	Ship, engine, design
Disease	Blood	List of genetic disorders	Disease, patient, cell
	Patient	Crohn's disease	Specie, animal, plant
	Treatment	Infectious disease	Specie, animal, plant
	Disorder	Tay-sachs disease	Food, make, wine
	Brain	Multiple sclerosis	Country, population, people
Car	Vehicle	Brabhma	Car, race, vehicle
	Driver	Cable car (railway)	City, large, area)
	House	Grand prix legends	Build, building, house
	Wheel	Mini	Black, white, people
	Train	Stock car racing	War, force, army

In order to show the stability of our method against random initialization, we repeated our method more than 100 times and compared it to the baseline method under two different metrics: log-likelihood and pairwise KL-divergence between the words distributions of different topics.

5.3 Results on Literature Repositories

Twitter-influenced LDA was run on nine subsets of the Reuters dataset which correspond to the first nine streams. The perplexity of a model was computed using the successive stream as the test set.

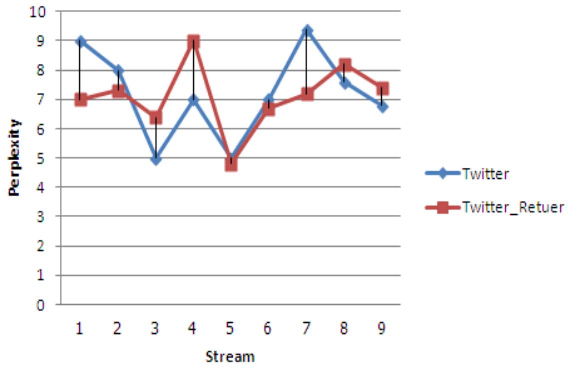

Fig. 2. LDA on Twitter (trained with Reuters articles)

Fig. 2 shows the Twitter-influenced LDA compared to the corresponding models that were trained on the Reuters documents only. It can be seen that the perplexity of LDA with Twitter articles is lower in five out of the nine models. It can be inferred that the higher perplexity in some cases with Twitter is due to the unstructured approach used to partition the data, which does not guarantee the representation of all the classes in each stream. Thus, any document in the test set that belongs to a new class would eventually increase the perplexity. However, when this factor is neutralized, incorporating external knowledge from Twitter does improve the performance.

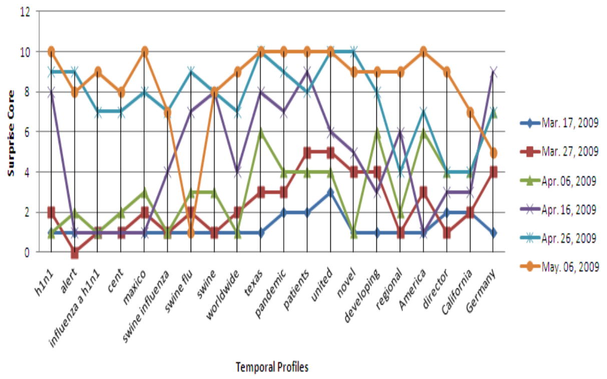

Fig. 3. Temporal profiles for the dataset, sorted by the Gausian emergence scored

Fig. 3 shows the temporal profiles for the top 20 surprising terms that come from twitter and Reuter news. From these plots, the main topic within this dataset becomes obvious (H1N1, the swine flu outbreak of 2009). On April 24, the surprise analysis starts to select terms that first appear about the swine flu outbreak (serious, vaccination, epidemic).

6 Conclusion

In this paper, we tackle the problem of mining common topics from multiple asynchronous text sequences. We propose a novel method which can automatically discover and fix potential asynchronism among sequences and consequentially extract better common topics. The key insight driving our approach is that we introduce a self-refinement process by utilizing correlation between the semantic and temporal information in the sequences. It performs topic extraction and time synchronization alternately to optimize a unified objective function. A local optimum is guaranteed by our algorithm. We justified the effectiveness of our method on two real-world data sets, with comparison to a baseline method. Empirical results suggest that our method 1) is able to find meaningful and discriminative topics from asynchronous text sequences; 2) significantly outperforms the baseline method, evaluated by using both qualitative and quantitative measures; 3) the performance of our method is robust and stable against different parameter settings and random initialization.

In addition, it can be extended to work in an online for mining text streams and using an evolving external knowledge. The effect of the embedded historic semantics on detecting emerging and/or periodic topics constitutes future work.

References:

[1] AGRESTI A, *Categorical Data Analysis*, 2 nd edn., John Wiley & sons. Inc, 2002.

[2] AISUMAIT L , BARBARA D AND DOMENICONI C, Online LDA: Adaptive topic model for mining text streams with application on topic detection and tracking, *Proceedings of the IEEE International Conference on Data Mining*, 2008.

[3] AISUMAIT L , WANG P , DOMENICONI C AND BARBARA D, *Text Mining: Applications and Theory*, John Wiley & Sons, Inc., 2010.

[4] ANDRZEJEWSKI D , ZHU X AND GRAVEN M, Incorporating domain knowledge into topic modeling via Dirichlet forest priors, *Proceedings of the International Conference on Machine Learning*, 2009.

[5] ARORA R AND RAVINDRAN B, *Latent Dirichlet Allocation Based Multi-Document Summarization*, 2008.

[6] BENHARDUS JAMES, Streaming Trend Detection in Twitter, *Uccsreu for artificial intelligence, natural language processing and information retrieval final report*, 2010.

[7] BIEI D , NG A AND JORDAN M, Latent Dirichlet allocation. *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 993 - 1022.

[8] BIRO I , SZABO J AND BENCZUR A, Latent Dirichlet Allocation in Web Spam Filtering, *Forth AIRWeb International Conference on Adversarial Information Retrieval on the Web*, 2008, pp. 29-32.

[9] CHIEN J AND CHUEH C, Topic-Based Hierarchical Segmentation, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No.10, 2012, pp. 55-66.

[10] CHUNDI P AND ROSENKRANTZ D , Constructing Time Decompositions for Analyzing Time Stamped Documents. *Forth SIAM International Conference on Data Mining*, 2004, pp. 57-68.

[11] CHUNDI P , ZHANG R AND CASTELLANOS M, Entropy Based Measure Functions for Analyzing Time Stamped Documents. Text Mining Workshop, *SIAM International Conference on Data Mining*, 2006.

[12] ENGEL D , WHITNEY P AND CRAMER N, *Text Mining: Applications and Theory*, John Wiley & Sons, Inc., 2010.

[13] GRIFFITHS T.L. AND STEYVERS M., A probabilistic approach to semantic representation, *International Conference on the 24th Annual Conference of the Cognitive Science Society*, 2002.

[14] HAN JIAWEI , KAMBER M AND PEI JIAN, *Data Mining Concepts and Techniques*, 3 rd edn Elsevier, 2011.

[15] HENDERSON K AND ELIASSI – RAD T, Applying Latent Dirichlet Allocation to Group Discovery in Large Graphs, *ACM SAC International Conference on Symposium on Applied Computing*, 2009, pp. 1456-1461

[16] HUANG S AND RENALS S, *Modeling Topic and Role Information in Meetings Using the Hierarchical Dirichlet Process*, 2008, pp. 214-225.

[17] KONG S AND LEE L, Semantic Analysis and Organization of Spoken Documents Based on Parameters Derived From Latent Topics, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 19, No.7, 2011, pp. 1875-1889.

[18] KONTOSTATHIS A , GALITSKY L , POTTENGER W , ROY S AND PHELPS D, *A survey of emerging trend detection in textual data mining. in: Survey of Text Mining: Clustering, Classification, and Retrieval. Springer*, 2003.

[19] KUMARAN G AND ALLAN J, Text classification and named entities for new event detection. *ACM SIGIR Conference*, 2004, pp. 297-304.

[20] LI W AND HUANG Y, *New Event Detect based on LDA and Correlation of Subject Terms* , 2011.

[21] LIU B, *Web Data Mining Exploring Hyperlinks, contents and Usage Data*, 2 nd edn Springer, 2011.

[22] MEI Q AND ZHAI C, Discovering evolutionary theme patterns from text: An exploration of temporal text mining. KDD, *11 th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005, pp. 198-207.

[23] PARADIMITRIOU S , SUN J AND FALOUTSOS C, Dimensionality Reduction and Forecasting on Streams. *In DATA STREAMS: Models and Algorithms* (ed. Charu C. aggarwal) edn Springer, 2007.

[24] POZDNOUKHOV A AND KAISER C, Space-Time Dynamics of Topics in Streaming Text . *Nineteen th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems*, 2011, pp. 1-8

[25] SAKAKI T , OKAZAKI M AND MATSUO Y, *Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors*, , 2011.

[26] SALTON G, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[27] TING H , LEE L , CHAN HO AND LAM T, *Approximating Frequent Items in Asynchronous Data Stream over a Sliding Window*, 2011, pp. 200-222.

[28] WANG X AND GAO X, *Dimensionality reduction with latent variable model*, Front. Electr. Electron. Eng., Vol.7, No.1, 2012, pp. 116-126.

[29] WANG X ZHANG K , JIN X AND SHEN D, Topic Mining over Asynchronous Text Sequences, *IEEE Transactions of Knowledge and Data Engineering*, Vol. 24, No.1, 2012, pp. 156-168.

[30] WEI X. AND W.B. CROFT, LDA-based document models for ad-hoc retrieval,

Proceedings of the 29th ACM SIGIR Conference on Research and Development in Information Retrieval, (CRDIR'06), Seattle, WA., 2006, pp: 178-185.

[31] WHITNEY P , ENGEL D AND CRAMER N, Mining for surprise events within text streams. *Ninth SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics*, 2009, pp. 617-627.

[32] WILLIAMS G, *Data Mining Theory Methodology Techniques* (ed. Simeon J. Simoff) Springer, 2006.

[33] YAO L., MIMNO D. AND MCCALLUM A, Efficient Methods for Topic Model Inference on Streaming Document Collections, *KDD, 15 th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 937-946.