Discovery of Rare Sequential Topic Patterns in Document Stream*

Zhongyi Hu [‡] Hongan Wang [†‡] Jiaqi Zhu [‡¶] Maozhen Li [§] Ying Qiao [‡]
Changzhi Deng [‡]

Abstract

Plain text documents created and distributed on the Internet are ever changing in various forms. Mining topics of these documents has significant applications in many domains. Most of the literature is devoted to topic modeling, while sequential patterns of topics in document streams are ignored. Moreover, traditional sequential pattern mining algorithms mainly focused on frequent patterns for deterministic data sets, and thus not suitable for document streams with topic uncertainty and rare patterns. In this paper, we formulate and handle the mining problem of rare Sequential Topic Patterns (STPs) for Internet document streams, which are rare on the whole but relatively often for specific users, so also interesting. Since this type of rare STPs reflects users' specific behaviors, our work can be applied in many fields, such as personalized context-aware recommendation and real-time monitoring on abnormal user behaviors on the Internet. We propose a novel approach to discovering user-related rare STPs based on the temporal and probabilistic information of concerned topics. After extracting topics from documents by LDA and sorting the document stream into sessions for different users during different time periods, the proposed algorithms discover rare STPs by (1) mining STP candidates for each user through an efficient algorithm based on pattern-growth, and (2) generating user-related rare STPs by pattern rarity analysis. Experiments on both synthetic and real data sets show that our approach can discover interesting rare STPs very effectively and efficiently.

*This work is supported by the National Key Basic Research Program of China (973 Program, No.2013CB329305), the State Key Program of National Natural Science Foundation of China (NSFC-61232013), National Natural Science Foundation of China (NSFC-61202217), the National High Technology Research and Development Program of China (863 Program, No.2012AA040904), the National Key Technology R&D Program (2012BAK02B00), and the program from Institute of Software, Chinese Academy of Sciences (ISCAS2009-JQ03).

†State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences.

‡Beijing Key Laboratory of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences.

§School of Engineering and Design, Brunel University.

¶The corresponding author. Email: zhujq@ios.ac.cn

1 Introduction.

Document streams are generated in various forms on the Internet, such as news streams, emails, microblog articles, instant messages, research paper archives, web forum discussion threads, and so forth. These document streams generally concentrate on specific topics. For example, people in the same social community may talk about some common topics or discuss some public or private events on the web. So far, most of text mining research focused on finding topics in document streams. Topics can be extracted from the stream involving both semantic and temporal information by various topic modeling methods [5, 6, 18, 24]. Apparently, there may be some correlations among these obtained topics in successive documents for a specific user, and these correlations could be described by ***Sequential Topic Patterns (STPs)***. Since capturing both topic combinations and their orders, STPs serve well as discriminative units of semantic association in ambiguous situations. Moreover, the abstract and probabilistic description of topics can help to solve the cold start problem and reach high confidence level in pattern matching.

Some STPs occur frequently in a document stream and thus reflect common behaviors of users. Besides, there are still some others which are rare for the general population, but occur relatively often for some specific user or some specific group of users. Compared to frequent ones, mining these ***user-related rare STPs*** is more interesting. Theoretically, it defines a new kind of patterns for event mining, which can characterize those individual and personalized behaviors in a certain context. Practically, it can be applied in many real-life scenarios, as illustrated in the following two examples.

Example 1. ***Personalized context-aware recommendation.*** *Traditional recommendation systems have been extensively used to make recommendations based on users' history of preferences. However, in some applications, they failed to consider users' current situations and thus neglected the different preferences of users in different contexts. For example, when a user visits a web site, the context is reflected in the sequence of documents which the user has clicked and read in his/her*

current interaction. That can be rightly characterized by STPs. Moreover, mining sequential patterns of topics instead of documents is significant in capturing general characteristics of documents. In this way, users' interests can be found at a more abstract level and it gets easier to track and detect changes in users' preferences. For instance, $<$ earthquake $\rightarrow$ tectonic structures $\rightarrow$ crustal movement $>$ and $<$ earthquake $\rightarrow$ geologic hazard $\rightarrow$ self-help skills $>$ are two different STPs about earthquake, and they reflect completely different needs or interests of users in specific interactions. The discovered STPs can be used to predict the topic of the next document the user would like to read in his future interaction. Furthermore, the predicted topics can be used to post-filter the initial ranking produced by a traditional recommendation algorithm to perform context-aware recommendation.

Example 2. ***Real-time monitoring on abnormal user behaviors on the Internet.*** *Recently, micro-blogs such as Twitter are attracting more and more attention all over the world. Micro-blog messages are real-time, spontaneous reports of what the users are feeling, thinking, and doing. However, the real intentions of users are hard to be found out directly from a single micro-blog message. For example, some people take advantage of the characteristics of micro-blogs to spread information widely for the purpose of fraud, such as lottery fraud. This kind of behaviors usually consists of the following steps by posting messages: 1)make award temptations; 2)diddle other users' information; 3)obtain various fees by cheating; 4)take illegal intimidation if their requests are denied. Detecting such abnormal user intention or behavior is significant for applications such as social security surveillance, which needs both content information and temporal relationships of the messages. STP mining happens to cover the both two aspects and is thus a good technique for real-time user behavioral monitoring on the Internet.*

In this paper, we investigate the unsolved problem of mining sequential patterns of topics with uncertainty in document streams. The main contributions of our work are on the following aspects:

1. we give formal definitions of STPs, rare STPs, and propose the problem of mining user-related rare STPs in document streams, which reflects personalized behaviors of specific users;
2. we design a group of algorithms, including a pattern-growth based algorithm for STP mining in uncertain environment, and a (user-related) rare STPs discovery algorithm with rarity analysis;
3. we validate our approach by experiments on both synthetic and real datasets.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 defines the problem of mining user-related rare STPs from a document stream. Section 4 presents the proposed mining algorithms in detail. Section 5 shows the experimental results on both synthetic and real datasets, and Section 6 concludes the paper and discusses future directions.

2 Related Work.

Topic mining has been extensively studied in the literature. Topic Detection and Tracking (TDT) task [3, 25] aimed to detect and track topics (events) in news streams with clustering-based techniques. Many generative topic models were also proposed, such as Probabilistic Latent Semantic Analysis (PLSA) [11], Latent Dirichlet Allocation (LDA) [5] and their extensions [4, 15, 19].

In many real applications, text collections carry generic temporal information and therefore can be considered as a text stream. To obtain the temporal dynamics of topics, various dynamic topic modeling methods have been proposed to discover topics over time in document streams [6, 18, 14, 24, 27]. However, these methods were designed to extract the evolution model of individual topics from a document stream, rather than to analyze the relationship among extracted topics in successive documents for specific users.

Sequential pattern mining has been well studied in the literature in the context of deterministic data, but not for topics with uncertainty. The concept *support* [20] is the most popular criteria for mining sequential patterns. It evaluates frequency of a pattern and can be interpreted as occurrence probability of the pattern. Formally, it is defined as

$$supp(\alpha) = \frac{f(\alpha)}{N} \tag{2.1}$$

where α is a sequential pattern, $f(\alpha)$ is the frequency of the pattern α and N is the total number of sequential data. Many methods have been proposed to solve the problem of sequential pattern mining based on *support*, such as PrefixSpan [22], FreeSpan [9] and SPADE [26]. These methods were designed to discover frequent sequential patterns whose supports are not less than a user-defined threshold *minsupp*. However, the obtained patterns are not always interesting, because those rare but significant patterns are pruned for their low supports. Furthermore, the frequent sequential pattern mining from deterministic databases is completely different from the STP mining that handles uncertainty of topics.

Aggarwal et al. [1] and Chui et al. [8] both studied frequent itemset mining in probabilistic databases, but

few researches addressed the problem of sequential pattern mining on uncertain data. Muzammal and Raman [21] proposed a method to discover frequent sequential patterns from probabilistic databases and evaluated the frequency of a pattern based on the expected support. However, the data model cannot be applied to topic sequences. In addition, they focused on the frequent pattern mining and failed to discover interesting rare patterns for some users.

Hariri et al. [10] presented an approach for context-aware music recommendation based on sequential patterns of latent topics. They processed tag data of songs by the LDA topic modeling module [5] to obtain the topic distribution for songs. Each song is then represented as a set of topics with probabilities above a certain threshold and each playlist is represented as a sequence of topic sets. The frequent topic-based sequential patterns occurring among playlists are then discovered to predict the next song in a user's interaction session. However, an existing algorithm PrefixSpan [22] without uncertainty was used there for mining frequent patterns, so the method lost uncertainty degree of topics due to the approximation induced by the certain probability threshold.

As the preliminary work of this paper, a poster [12] proposed the framework of mining sequential patterns of topics, but it lacks the formal definition of the problem and no mature mining algorithms are presented. The original and simple algorithms there are not as effective and efficient as those in this paper.

3 Problem Statement and Preliminaries.

In Topic Detection and Tracking (TDT) task, a topic is defined as an event or an activity [7]. In topic modeling [5, 4, 11, 15, 19], a probability distribution of words in a vocabulary set is used to indicate a topic. We adopt the latter in order to maintain the uncertainty degree of topics.

A text document d consists of a sequence of words from a vocabulary set $V = \{w_1, w_2, \cdots, w_{|V|}\}$, and it can be converted into probabilistic mixture of topics through topic modeling algorithms, such as LDA [5]. In this work, each document is represented in this way and called a topic-level document.

DEFINITION 3.1. *(**Topic-level documents**). Given a topic set $T = \{z_1, z_2, \cdots, z_{|T|}\}$ consisting of $|T|$ topics, a topic-level document $td = \{(z_1, p_1), (z_2, p_2), \cdots, (z_K, p_K)\}$ is defined as a mixture of K independent topics with probabilities to specify their uncertainty. The K topics describing the document are extracted from the topic set T, with probability higher than or equal to a predefined threshold.*

In this way, document streams can also be defined at topic level as follows.

DEFINITION 3.2. *(**Topic-level document streams**). A topic-level document stream is defined as a sequence $TDS = < (td_1, u_1, t_1), (td_2, u_2, t_2), \cdots, (td_i, u_i, t_i), \cdots >$, where td_i is a topic-level document browsed or published by user u_i at time t_i, $u_i \in U$, and $t_i \leq t_j$ for all $i \leq j$. Here, $U = \{u_1, u_2, \cdots, u_{|U|}\}$ is the user set of interest.*

Usually, each user can only read or write one document at a time point, so we can assume that at any time point, for any specific user, at most one document is involved.

An STP can be considered as a topic sequence that appears in order during a time period, as below.

DEFINITION 3.3. *(**Sequential Topic Patterns**). A sequential topic pattern (STP) α is defined as $< z_1, z_2, \cdots, z_n >$. Each $z_i \in T$ is a topic, and called an element of α. n is the number of topics included in α, and called the length of α. Such a pattern α is called an n-STP.*

Discovering STPs is significant for document sequences related to a specific user during a certain time period (session). Each session contains a subsequence of the document stream, and can be regarded as a series of correlated activities performed by a user during a time period when he/she was navigating through a given website or posting messages on Internet forums or micro-blog sites. We leave our method for session identification to the next section.

Pattern instance is the concretization of STPs with uncertainty, and can be defined ad follows.

DEFINITION 3.4. *(**Pattern Instances**). Given an n-STP $\alpha = \{z_1, z_2, \cdots, z_n\}$ and a session s for user u_c in the document stream TDS. If we can extract a sequence $s' = < (td_1, u_c, t_1), \cdots, (td_n, u_c, t_n) >$ from s, such that $t_1 < \cdots < t_n$, and for all $i = 1, \ldots, n$, $(z_i, p_i) \in td_i$ holds for some p_i, then we say $\bar{\alpha} = < (z_1, p_1), \cdots, (z_n, p_n) >$ is a pattern instance of α in the session s.*

In a specific session, there may be multiple pattern instances of an STP, and we can mine interesting STPs by finding and analyzing corresponding pattern instances.

The major tasks of (user-related) rare STP discovery from a topic-level document stream includes: 1) find all the STP candidates for each user; 2) evaluate the rarity of them and identify those rare on the whole, but relatively frequent for specific users. This task is challenging for two reasons. Firstly, each document at topic

level is a combination of several topics with probabilities to describe the uncertainty. Hence, existing techniques of sequential pattern discovery, which aimed to mine frequent sequential patterns based on deterministic item set [13, 20], cannot be applied directly to solve this problem. Secondly, how to define user-related rarity of patterns is another critical problem, so that it can effectively characterize personalized behaviors of different users.

4 Rare Sequential Topic Pattern Mining Method

In this section, we present a novel approach for discovering (user-related) rare STPs in document streams.

The main processing framework is shown in Figure 1. It consists of three phases. Firstly, documents are crawled from the Internet and constitute a stream in the order of browsing time or publishing time. Secondly, the original document stream is converted to a topic-level document stream and divided into different sessions as preprocessing. Finally, a group of algorithms are applied to discover user-related rare STPs.

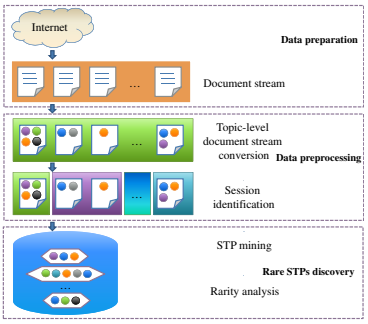

Figure 1: The processing framework of rare STPs mining.

4.1 Data Preprocessing Data in the original document stream about the concerns or comments of users cannot be directly used for the STP mining purpose. A preprocessing phase is necessary, which includes two steps. Firstly, topics in original documents are extracted and a topic-level document stream is obtained. We use the classical LDA module [5, 17] to complete this conversion. Secondly, all the sessions belonging to different users at different time periods are identified with a time-oriented heuristics [23]. Specifically, we assume that the duration of a session must not exceed a time-span threshold (h_{ts}), and there is no overlap among sessions.

Given a document stream TDS, for a specific user u_c, we at first extract all the documents related to u_c as a subsequence $TDS_c = <(td_1, u_c, t_1), \cdots, (td_i, u_c, t_i), \cdots>$, and then divide sessions in the following way. Firstly, we find the earliest document (td_s, u_c, t_s), which is unique due to the assumption above, and all the documents with time point t satisfying $t - t_s \leq h_{ts}$, which constitute the first session. For other documents in TDS_c, we repeat the same procedure until all the sessions are found. The result can be shown in Figure 2. The horizontal axis represents time, and the vertical axis represents users. Each ellipse represents a session, and all the sessions in each line constitute a document subsequence associated with an individual user.

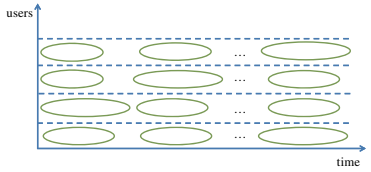

Figure 2: An example of session identification.

4.2 Rarity of sequential topic patterns Most researches of sequential pattern discovery focused on frequent patterns. However, some special behaviors of some users, which are not frequent, reflect the browsing or publishing habits of them. They can be specified by user-related rarity analysis of STPs.

For deterministic databases, pattern support $supp(\alpha)$ is defined in Equation 2.1. Nevertheless, the traditional support is not suitable for STP mining because of the uncertainty of topics. Instead, expected support has to be used to measure the frequency of uncertain itemsets [8]. We adopt and improve it for our STP mining problem, denoted still as *support* for simplicity.

Given a topic-level document stream TDS within a sliding window, a session set $S = \{s_1, s_2, \cdots, s_{|S|}\}$ is the set of identified sessions in TDS. Suppose STP $\alpha =< z_1, z_2, \cdots, z_n >$ is an n-STP, and $\Psi_i = \{\bar{\alpha}_1, \bar{\alpha}_2, \cdots, \bar{\alpha}_{|\Psi_i|}\}$ is the instance set of α in s_i, where $\bar{\alpha}_j$ is in the form of $< (z_{j1}, p_{j1}), \cdots, (z_{jn}, p_{jn}) >$. The support of α in TDS is defined as the average among all sessions, of the maximal occurrence probability of all pattern instances. It can be calculated by

$$supp(\alpha) = \frac{\sum_{i=1}^{|S|} max\{P(\bar{\alpha}_j) | \bar{\alpha}_j \in \Psi_i\}}{|S|} \tag{4.2}$$

where $P(\bar{\alpha}_j)$ is the probability of $\bar{\alpha}$ and can be calculated by $P(\bar{\alpha}_j) = \prod_{k=1}^{n} p_{jk}$. If there is no instance of α in s_i, i.e., $\Psi_i = \varnothing$, we assign the maximum probability for s_i with 0.

In order to compare supports of patterns with different lengths, we furthermore define *scaled support* as follows. It normalizes the values of the probability product to the same order of magnitude, so as to reduce the impact of the pattern length in the later comparison and unified computation.

$$scsupp(\alpha) = supp(\alpha)^{\frac{1}{|\alpha|}} \tag{4.3}$$

The support and the scaled support reflect the frequency of patterns on the whole. We can also analyze the frequency for a specific user by restricting the session set to that user, and get the support and the scaled support for a user u, denoted as $supp(\alpha)|_u$ and $scsupp(\alpha)|_u$ respectively.

Based on scaled support, we can evaluate user-related rarity of STPs by the following two concepts. Assume that $U = \{u_1, u_2, \cdots, u_{|U|}\}$ contains all users who browsed or published documents, and the discovered STP set is Φ. The *Absolute Rarity* of an STP α for a user u is used to describe the difference between the scaled support of α in u's sessions (local support) and that in all sessions (global support). It can be calculated by

$$AR(\alpha)|_u = scsupp(\alpha)|_u - scsupp(\alpha) \tag{4.4}$$

The *Relative Rarity* of α for u regularizes the absolute rarity by comparing that of α with the average among all patterns in Φ, and can be calculated by

$$RR(\alpha)|_u = AR(\alpha)|_u - \frac{\sum_{\beta \in \Phi} AR(\beta)|_u}{|\Phi|} \tag{4.5}$$

Now, we can define rare STPs, which is user-related.

DEFINITION 4.1. ***(User-Related Rare STPs).*** *Given a topic-level document stream TDS, a user-defined scaled support threshold h_{ss}, and a relative rarity threshold h_{rr}, an STP α related to u is rare if both $scsupp(\alpha) \leq h_{ss}$ and $RR(\alpha)|_u \geq h_{rr}$ hold.*

Notice that the first condition indicates the global rarity of α, and the second one assures its relatively high frequency for u.

4.3 Mining Algorithms The workflow of our approach is presented in Figure 3, and Algorithm 1 gives the pseudo-code of the main procedure. After preprocessing, the input document stream within a sliding window is transformed into a set of user-session pairs, denoted as $User\_Sess$. Its element is denoted by a Hash notation $< u : S_u >$, in which u is the key of the map and its value S_u is a set containing all the sessions for u. To be unified, all the sets of pairs used in our algorithms are denoted in this form. For each user-session pair, a thread is started and the subprocedure *UpsSTP* is recursively invoked to mine all the STP candidates for a specific u paired with their support values, and add the user-STP pair to the set $User\_STP$. These threads can be executed in parallel relying on the hardware environment. When all the threads have finished, the subprocedure *RSTPMiner* is called to make the rarity analysis and get the output set $User\_RSTP$, which contains all the pairs of users and the rare STPs related to them.

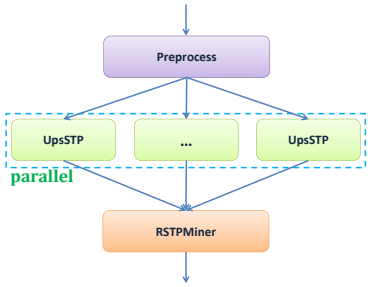

Figure 3: The workflow of rare STP mining.

Algorithm 1 $Main(DS, h_{ss}, h_{rr})$

Input: the original document stream within a sliding window $DS = \{(d_1, u_1, t_1), \cdots, (d_M, u_M, t_M)\}$, the scaled support threshold h_{ss} and the relative rarity threshold h_{rr}.
Output: a set $User\_RSTP$ of User-RSTP pairs.

```
1: User_Sess ← Preprocess(DS);
2: for all < u : S_u > in User_Sess do
3:     STP_Supp_u ← UpsSTP(∅, ∅, S_u, |S_u|);
4:     User_STP ← User_STP ∪ < u : STP_Supp_u >;
5: User_RSTP ← RSTPMiner(User_STP, h_ss, h_rr);
6: return User_RSTP;
```

The pseudo-code of *UpsSTP* is shown in Algorithm 2. It is a pattern-growth based algorithm, extending the classical sequential pattern mining algorithm PrefixSpan [22] with uncertainty calculation. When it is called, there is always a specific user u. It has four input parameters: α is the currently derived STP; $Pref_\alpha$ is a set of α-prefix triples of the form $< i, j, p >$, indicating that an instance of the STP α with probability p occurs in the prefix of the sequence in the ith session for u, and the instance ends at position j of the sequence; S_α is a set of subsequences, each of which is a suffix of a sequence in the sessions for u, and the corresponding prefix just contains an instance of α (ended with the last element of α); S is the number of sessions, which is unchanged for a fixed u.

Each execution of *UpsSTP* performs pattern-growth from the input STP α to a new one $\beta = \alpha z$, by appending an element (topic) z. Notice that α is $\varnothing$ for the first invocation. At first, we scan all the sequences in S_α to obtain the set E containing all the possible topics which can be appended to α, and then for each z in E and each sequence s_i in S_α, find all the instances of z in s_i, compute the maximum probability among all the instances of the new STP, and record new prefix triples in $Pref_\beta$. Specifically, for each instance of z, if $\alpha = \varnothing$, the instance of β in s_i is unique, and its probability can be directly obtained from the jth position of s_i, denoted by $P(z, j)$. Otherwise, there may be multiple instances of β in s_i. Taking advantage of the input information in $Pref_\alpha$, we can find all the instances of α in s_i previous to j, and choose the one with maximum probability to multiply with $P(z, j)$. After all the instances of β are handled, the support of β for u is computed according to Formula 4.2, and the pair $< \beta : supp_\beta >$ is added to the set STP_Supp. After that, we get β-projected sequences by projecting out z if there is from each α-projected sequence in S_α. That is similar to the algorithm PrefixSpan, so the details is omitted here. Finally, we recursively call *UpsSTP* with β as the current STP to mine longer STPs, and output all the STP-support pairs for u obtained here and from the return value of the next recursion.

Algorithm 2 $UpsSTP(\alpha, Pref_\alpha, S_\alpha, |S|)$

Input: an STP α, a set $Pref_\alpha$ of α-prefix triples, a set S_α of α-projected subsequences (sessions), and the number of sessions $|S|$ for the specific user.
Output: a set STP_Supp of STP-support pairs

$STP\_Supp \leftarrow \varnothing$;
find all the possible elements (topics) which can be appended to α to form a new STP, and record them in E.
for all $z \in E$ **do**
  $\beta \leftarrow \alpha z$;
  $supp_\beta \leftarrow 0$;
  for all $s_i \in S_\alpha$ **do**
    find all the instances of z in s_i, and record their positions j in I;
    for all $j \in I$ **do**
      if $Pref_\alpha == \varnothing$ **then**
        $p \leftarrow P(z, j)$;
      else
        $p \leftarrow P(z, j) \times \max\{\, p \mid < i, k, p > \in Pref_\alpha \wedge k < j\}$;
      $Pref_\beta \leftarrow Pref_\beta \cup < i, j, p >$;
    $supp_\beta \leftarrow supp_\beta + \max\{\, p \mid \exists j. < i, j, p > \in Pref_\beta\}/|S|$;
  $STP\_Supp \leftarrow STP\_Supp \cup < \beta : supp_\beta >$;
  $S_\beta \leftarrow Project(S_\alpha, z)$;
  $STP\_Supp \leftarrow STP\_Supp \cup UpsSTP(\beta, Pref_\beta, S_\beta, |S|)$;
return STP_Supp;

The subprocedure *RSTPMiner* in Algorithm 3 is designed for rarity analysis and rare STPs discovery. It transforms the set of User-STP pairs into a set of User-RSTP pairs. At first, we get the set of all STPs and globally rare ones according to the definition. Here, the global support of α can be obtained by computing the weighted average value of the supports of α for all users. Then, for each user u, we calculate the absolute rarity for all STPs and the relative rarity for those STPs globally rare and associated with u. The locally frequent STPs are selected to derive an STP-RR pair, which is then added to the set of User-RSTP pairs for u.

Algorithm 3 $RSTPMiner(User\_STP, h_{ss}, h_{rr})$

Input: a set $User\_STP$ of User-STP pairs, the scaled support threshold h_{ss} and the relative rarity threshold h_{rr}.
Output: a set $User\_RSTP$ of User-RSTP pairs.

$User\_RR \leftarrow \varnothing$;
get the whole pattern set Φ from $User\_STP$;
$\Phi' \leftarrow \{\alpha | \alpha \in \Phi \wedge scsupp_\alpha \leq h_{rr}\}$; (Formulas 4.2, 4.3)
for all $< u : STP_u > \in User\_STP$ **do**
  for all $\alpha \in \Phi$ **do**
    $AR_\alpha \leftarrow$ the absolute rarity of α for u; (Formulas 4.3,4.4)
  $STP\_RR_u \leftarrow \varnothing$;
  for all $\alpha \in STP_u \cap \Phi'$ **do**
    $RR_\alpha \leftarrow$ the relative rarity of α for u; (Formula 4.5)
    if $RR_\alpha \geq h_{rr}$ **then**
      $STP\_RR_u \leftarrow STP\_RR_u \cup < \alpha, RR_\alpha >$;
  $User\_RSTP \leftarrow User\_RSTP \cup < u : STP\_RR_u >$;
return $User\_RSTP$;

5 Experiments

In this section, we evaluate the performance of our algorithms using both synthetic and real datasets. Since our work is innovative on mining user-related rare STPs, we will show its effectiveness by the three standard measures (precision, recall and F1-Measure [16]) on a synthetic dataset, and compare the performance of the subprocedure *UpsSTP* with other feasible algorithms for this problem. Then, we apply the proposed approach to a real Twitter data set for mining interesting user-related rare STPs. The results show that our method can effectively discover personalized special behaviors of users on the Internet.

All the experiments were run on a computer with Intel(R) Core(TM) i3 CPU and 2GB RAM. The algorithms were implemented in Java, and run in Eclipse on Windows 7 Enterprise.

5.1 Synthetic data set We use and improve the IBM data generator [2] to generate sequences of items, each of which can be regarded as a topic. Then, we assign probability to these items to get uncertain data sets. Specifically, the data in each session are simulated mainly with a common topic set for all users, supplemented with a few special topics for a subset of users individually. We assign probability to each

topic in the source sequence using a uniform distribution over $(0, 1)$, to obtain a collection of probabilistic topic sequences. The generated data conform to the topic-level document stream defined in this paper.

Initially, the average size of sessions (length of sequences) is 10 ($L = 10$), and the average number of sessions for each user is 100 ($|S| = 100$). The number of users varies in the range $[10, 100]$ and the number of topics $|T| = 20$ for 80% of users while $|T| = 30$ for the other 20% of users, in which special topics of their own are involved. The statistical results on precision, recall and F1-Measure of our mining algorithms with different user numbers is shown in Figure 4.

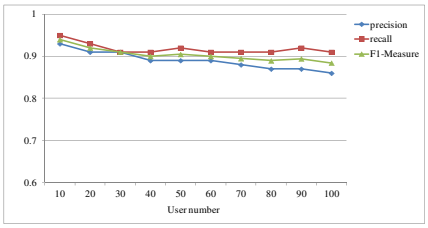

Figure 4: The values of precision, recall and F1-Measure with different user numbers.

We can see that the precision is between 0.86 and 0.93 and the recall is between 0.91 and 0.95. They are both high and thus compelling. As the number of users increases, recall maintains a very high value, which is consistent with our goal, as recall is the most important measure for the purpose of user-related rare STPs discovery. However, precision shows a moderate downward trend. The reason is that the patterns will become sparser with more users, and the discrepancy among users will get more obvious, so some insignificant user-related rare STPs will be found. F1-Measure shows the trade-off in maximizing both precision and recall. There is a downward trend in F1-Measure, which is mainly caused by the decreasing precision.

Moreover, we fix the user number as 50, and analyze the values of these measures with different scaled support thresholds and different relative rarity thresholds. Figures 5 and 6 show the respective results. In Figure 5, the support threshold varies in the range $[0.0001, 1]$ and the relative rarity threshold is assigned 0.01. It shows that a few patterns can be found only with small support threshold, so in that case the precision is high while the recall is low. However, recall increases significantly and precision shows a decreasing trend with the support threshold increasing. When the threshold is large enough, both precision and recall become stable. In Figure 6, the support threshold is assigned 1.0 and the relative rarity threshold varies in the range $[0.005, 0.5]$. Compared to Figure 5, an opposite result is obtained. Precision increases while recall decreases with the increase of the relative rarity threshold. From the F1-measure curves in the two figures, we can set the scaled support threshold to 0.5 and the relative rarity threshold to 0.01 for a good trade-off between precision and recall.

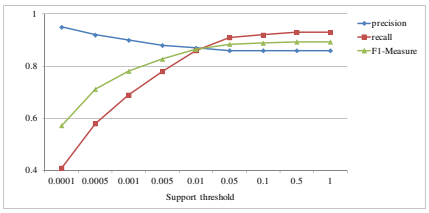

Figure 5: The values of precision, recall and F1-Measure with different support thresholds.

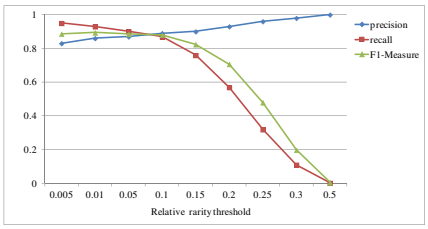

Figure 6: The values of precision, recall and F1-Measure with different relative rarity thresholds.

For the implementation of the subprocedure *UpsSTP*, *Depth-First Exploration* (denoted as *DFE*) and *Breadth-First Exploration* (denoted as *BFE*) are two alternative frequent sequential pattern mining algorithms [21] designed for probabilistic databases. As they cannot be used directly for mining STPs from data model like topic sequences with uncertainty, we rewrite them to accommodate this problem, and compare their performances with *UpsSTP*.

We get the synthetic data of sessions for one user through the previous method, and compare the execution time of the three algorithms in two situations with unchanged topic number $|T| = 20$. Firstly, we fix the average size of sessions as $L = 5$, but change the number of sessions $|S|$ from 100 to 10000. Secondly, we fix the session number as $|S| = 100$, but change the average size of sessions L from 4 to 12. The results are shown in Figure 7 and Figure 8. The former shows that the running time of the three algorithms are all augmented with the session number increasing. However, the performance of *UpsSTP* is almost stable, while for the others it declines sharply at $S = 5000$. From Figure 8, it can be seen that *UpsSTP* is not affected by the

change of average size of sessions, but the performances of both *DFE* and *BFE* decline obviously.

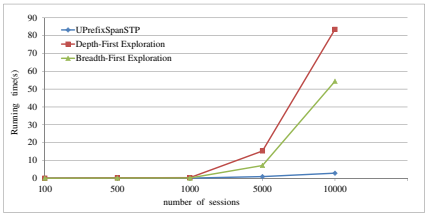

Figure 7: Performances of the three algorithms with different sessions numbers.

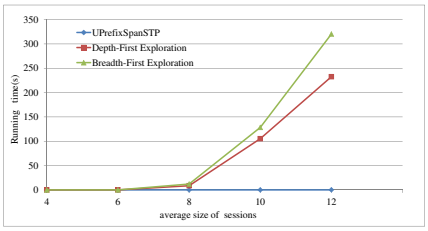

Figure 8: Performances of the three algorithms with different average sizes of sessions.

5.2 Real data set experiment For real data, we apply our method to Twitter messages. The data set totally contains 9082 users and 261580 tweets. The toolkit Mallet [17] is used to generate topics from the content of tweets with topic number $|T| = 15$. Then, we get a topic-level document stream, divide it into sessions, and mine STPs using the proposed algorithms. The result is recorded in a User-STP matrix, part of which is shown in Figure 9.

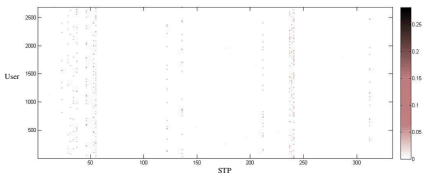

Figure 9: The User-STP matrix for real Twitter data set.

There are totally 2788 users and 362 STPs. As the right bar indicates, the grey level of points in the matrix represents the support of each STP for each user. We can see that the result is very sparse for real data. Therefore, traditional frequent sequential pattern mining algorithms are not applicable because there may be no frequent pattern found.

We apply our method to discover user-related rare STPs from the Twitter data set with a fixed scaled support threshold 0.1, and count the number of discovered rare STPs when the relative rarity threshold changes. Table 1 shows the experimental results. The change of rare STPs number is obvious with the change of relative rarity threshold. Moreover, we can indeed find some interesting and interpretable user-related rare STPs. Table 2 shows two instances of them, and Table 3 details the 9 top words of involved topics. They are similar to the result in [12], but with larger and thus more differentiable relative rarity.

Table 1: The number of discovered rare STPs with different relative rarity thresholds.

RR threshold	0.05	0.1	0.12	0.15	0.2
rare STP number	1133	1043	802	485	26

Table 2: Instances of user-related rare STPs.

user id	STP	support	relative rarity
125	$< 8, 14 >$	0.008	0.1882
207	$< 13, 2, 8 >$	0.0009	0.1526

Table 3: Instances and top words of topics in Table 2.

topic id	top words
2	world oil hours women goods
	call skin photo support
8	fan buy win stop care
	concert ball mobile part
13	love health body news cool
	pretty enjoy job great
14	day game happy things play
	class amazing weekend bit

From these results, we can estimate that the STP $< 8, 14 >$ could be related to buying and playing, and $< 13, 2, 8 >$ may be about healthy, product and buying. We check the respective original tweets and find the truth about the two patterns. 1) User 125 planed to buy a new baseball bat and asked his friends to play with him at weekend. 2) User 207 may be a cosmetic salesman, who at first published some tips about aromatherapy, then introduced the essential cosmetic product, and finally encouraged other users to buy his products. Therefore, these discovered rare STPs are indeed interesting and consistent with the real case. Although these two instances seem not particularly significant, if user 125 were about to do something criminal with others and user 207 were selling prohibited products, it is much more important to discover them. In these cases, our approach will be very effective and significant in real life.

6 Conclusions and Future Work

Mining user-related rare Sequential Topic Patterns (STPs) in document streams on the Internet is an innovative challenging problem. It formulates a new kind of patterns for uncertain complex event detection and inference, and has wide potential application fields, such as personalized context-aware recommendation and real-time monitoring on abnormal user behaviors on the Internet. In this paper, a group of new algorithms are proposed to solve this problem. The experiments based on both synthetic and real data sets show that the proposed approach is very effective and efficient for discovering interesting and interpretable rare STPs from document streams on the Internet.

In the future, we will refine the session identification process and the measures of user-related rarity, and improve the mining algorithms mainly on the degree of parallelism. Moreover, based on STPs, we will try to define more complex event patterns, such as imposing timing constraints on the sequential topics, and design corresponding mining algorithms. We are also interested in the dual problem, i.e., discovering patterns occurring frequent on the whole, but comparatively rare for specific users. What's more, we will apply our approach in more real-life mining problems and develop some practical tools.

References

[1] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang. Frequent pattern mining with uncertain data. In *KDD*, pages 29–38. ACM, 2009.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, pages 3–14, 1995.

[3] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *SIGIR*, pages 37–45. ACM, 1998.

[4] D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems 18*. MIT Press, 2006.

[5] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *The Journal of machine Learning research*, 3:993–1022, 2003.

[6] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120. ACM, 2006.

[7] K. Chen, L. Luesukprasert, and S. T. Chou. Hot topic extraction based on timeline analysis and multidimensional sentence modeling. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1016–1025, 2007.

[8] C. K. Chui and B. Kao. A decremental approach for mining frequent itemsets from uncertain data. In *PAKDD*, pages 64–75, 2008.

[9] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. Freespan: frequent pattern-projected sequential pattern mining. In *KDD*, pages 355–359. ACM, 2000.

[10] N. Hariri, B. Mobasher, and R. Burke. Context-aware music recommendation based on latenttopic sequential patterns. In *RecSys*, pages 131–138. ACM, 2012.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57. ACM, 1999.

[12] Z. Hu, H. Wang, and J. Zhu. The discovery of user related rare sequential patterns of topics in the internet document stream. Accepted in *SAC*, ACM, 2014.

[13] A. Koper and H. S. Nguyen. Sequential pattern mining from stream data. In *ADMA(2)*, pages 278–291. Springer, 2011.

[14] A. Krause, J. Leskovec, and C. Guestrin. Data association for topic intensity tracking. In *ICML*, volume 148, pages 497–504. ACM, 2006.

[15] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML*, volume 148, pages 577–584. ACM, 2006.

[16] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, 2008.

[17] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[18] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *WWW*, pages 533–542. ACM, 2006.

[19] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *ICML*, pages 633–640. ACM, 2007.

[20] C. H. Mooney and J. F. Roddick. Sequential pattern mining – approaches and algorithms. *ACM Computing Surveys*, 45(2):19:1–19:39, 2013.

[21] M. Muzammal and R. Raman. Mining sequential patterns from probabilistic databases. In *PAKDD(2)*, pages 210–221. Springer, 2011.

[22] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: Mining sequential patterns by prefix-projected growth. In *ICDE*, pages 215–224. IEEE Computer Society, 2001.

[23] M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *INFORMS Journal on Computing*, 15(2):171–190, 2003.

[24] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD*, pages 424–433. ACM, 2006.

[25] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *SIGIR*, pages 28–36. ACM, 1998.

[26] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42:31–60, 2001.

[27] Z. Zhang, Q. Li, and D. Zeng. Mining evolutionary topic patterns in community question answering systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 41:828–833, 2011.