The VLDB Journal (2014) 23:381–400
DOI 10.1007/s00778-013-0320-3

REGULAR PAPER

Event detection over twitter social media streams

Xiangmin Zhou · Lei Chen

Received: 8 November 2012 / Revised: 14 May 2013 / Accepted: 17 May 2013 / Published online: 19 July 2013

Abstract In recent years, microblogs have become an important source for reporting real-world events. A real-world occurrence reported in microblogs is also called a social event. Social events may hold critical materials that describe the situations during a crisis. In real applications, such as crisis management and decision making, monitoring the critical events over social streams will enable watch officers to analyze a whole situation that is a composite event, and make the right decision based on the detailed contexts such as what is happening, where an event is happening, and who are involved. Although there has been significant research effort on detecting a target event in social networks based on a single source, in crisis, we often want to analyze the composite events contributed by different social users. So far, the problem of integrating ambiguous views from different users is not well investigated. To address this issue, we propose a novel framework to detect composite social events over streams, which fully exploits the information of social data over multiple dimensions. Specifically, we first propose a graphical model called location-time constrained topic (LTT) to capture the content, time, and location of social messages. Using LTT, a social message is represented as a probability distribution over a set of topics by inference, and the similarity between two messages is measured by the distance between their distributions. Then, the events are identified by conducting efficient similarity joins over social media streams. To accelerate the similarity join, we also propose a variable dimensional extendible hash over social streams. We have conducted extensive experiments to prove the high effectiveness and efficiency of the proposed approach.

Keywords Graphical model · Location-time constrained topic · Variable dimensional extendible hash · Social streams · Social event detection

X. Zhou (✉)
ICT Center, CSIRO, Canberra, Australia
e-mail: xiangminemilyzhou@gmail.com; xiangmin.zhou@csiro.au

L. Chen
Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Hongkong, China
e-mail: leichen@cse.ust.hk

1 Introduction

The recent years have witnessed a rapid advancement of online social media sites, such as Twitter and Sina Weibo, which provide a convenient platform for users to report and share what is happening. The availability of these microblogging services has pushed forward the explosion of social data. According to a recent report on Twitter, about 200 million users used this social networking service in 2011, generating over 200 million tweets and handling over 1.6 billion queries per day [11]. As a result, a large amount of data are spreading over social networks, which provides important clues about specific situations. Monitoring events over social streams has many applications such as crisis management and decision making. Consider an application of social event monitoring in crisis management. When Queensland floods happened, messages on this event were reported in real time to twitter, describing the whole situation from various aspects such as what was happening, where an event was happening, who were involved, the effects on surrounding. While some messages reported a yacht sinking on Brisbane River, and the reopen of the port, others described a bull shark on a flooded street, or the blackout of some offices etc. Monitoring the whole crisis situation is helpful for first-aid officers to make right response on time, reducing the loss from disaster events. Therefore, how to effectively and efficiently monitor events

over social streams has become an important research problem.

We study effective and efficient solutions for social event monitoring over microblogs. This problem requires a meaningful definition of a social event. The literature contains several definitions on social events such as the relations between users [36], the information flow between users [37], or the arbitrary classification of a space and time region [19]. However, while some definitions ignore the locations of social messages [36,37], the arbitrary classification-based approach only detects a predefined target event. As a result, these definitions are all limited for the security-related applications such as crisis management where the location information is a vital factor contributing to a situation and multiple unknown events may appear in social networks at any time. Meanwhile, as many social messages describe the details of a situation from various aspects, it is important to find these details for making the right decision on certain situation. For example, during Queensland floods, it was reported that a yacht was sinking on Brisbane River. While the social relations and link-based approaches [36,37] consider it as a new event, the arbitrary classification ignores it directly since it only cares about whether flooding is happening in a place. While in security-related applications, the yacht sinking event is actually a part of the flood situation, where its effect on the surrounding was reported. To overcome these limitations, we introduce a novel definition that considers each social message as an event element and defines an event as a composition of multiple event elements over topic, time, location, and social dimensions. Compared to the existing attempts, the advantages of this definition mainly exist in two aspects. For one thing, an event is described as a complete view about a situation, which is appropriate to making right decision in crisis. For another, multiple unknown events can be detected simultaneously, thus correctly predicting various critical situations. Though event detection has been studied in information retrieval domain [1,27,35], there are still several challenges in microblogging scenario due to the special characteristics of social data in contrast to general text streams.

- Social media data are complex. Each message is not only a set of words but also consists of the information over multiple dimensions including the text content of the message, its posting time, the location of its social user, and the connection between users.
- Social messages are highly uncertain. Due to the limit of 140 characters on the message length, an instant message may be incomplete for describing an event. Meanwhile, as messages are input by users worldwide with various background, which unavoidably introduces more ambiguity in text content. In addition, possible time delays or location changes may be involved when a message is posted, leading to the high uncertainty of time and location.
- The amount of social messages is huge. Due to the popularity of social networking services, social users get connection via these microblogging platforms, discuss on their interested topics, and report the events around them. The increasing social activities of people on networking platforms have produced a large volume of social data.

Considering these factors, to identify social events effectively and efficiently, we need to well address three challenges. First, we need to construct a robust data representation model which captures the social media information over multiple attributes. This issue is important as the social contexts such as time, location, and social connection are inalienable parts of a message. Ignoring these contexts may make the corresponding messages less distinguishable. For example, the information on Victoria bushfires in Feb. 2009 may not be distinguished from those on the Roleystone Kelmscott bushfire in Western Australia in Feb. 2011. Second, an advanced technique should be designed to handle the uncertainty of social data. We should be able to infer the incomplete or uncertain information based on the occurrence of words in messages, and the distributions of time and locations. Finally, we need to design a set of efficient query processing techniques to accelerate the social data understanding. Pair-wise message comparison is clearly not acceptable for time critical online event detection.

In this paper, we propose a novel graphical model–based framework for effective and efficient social event detection. Specifically, we first propose a novel graphical model called location-time constrained topic (LTT) to capture the social data information over content, time, and location. To measure the similarity between two messages, we first define a KL divergence-based measure to decide the similarity of uncertain media content. Then, we present a longest common subsequence (LCS)-based measure for the link similarity between two certain user series. Finally, we aggregate these two measures by augmenting weights as a whole measure for the message similarity. We detect social events according to the similarity among messages over social streams and speed up the detection process by using a novel hash-based index scheme. The main contributions of this work are summarized as follows:

1. We propose a novel graphical model called location-time constrained topic (LTT) to capture the social data information over content, time, and location, and describe each message as a probability distribution over a number of topics. As a Bayesian-based model, the LTT well handles the uncertainty of social data over each dimension.

2. We propose a complementary measure that embeds the content similarity with time and location constraints, and the link similarity with time constraint. The content similarity takes into account the information from multiple attributes and the uncertainty of social data over content, time, and location, while the link similarity embeds the information flow with a conversation.
3. We detect the social events by conducting similarity join over tweet streams and design a novel hash-based index scheme to improve the event detection efficiency. The similarity join processing is further improved by a proposed nontrivial lower-bounding technique.
4. We have conducted extensive experiments over two large real social media stream datasets collected from Twitter platform. The experimental results prove the effectiveness and efficiency of our approach.

The remainder of the paper is organized as follows. Section 2 provides an overview of the related work in social event detection and topic model-based document summarization methods. Section 3 formally formulates our social event detection problem. Section 4 presents our new social data modeling approach together with our similarity measure on the proposed representation model. We present our new approach to social event detection and hash-based index scheme in Sect. 5. Extensive experimental results are reported in Sect. 6. Finally, Sect. 7 concludes the whole paper.

2 Related work

This section reviews the existing studies on detecting social events from two aspects, social event detection, and text document representation. The notations used in this paper are summarized in Table 1.

2.1 Social event detection

Event detection in social networks has received considerable attention in the fields of data mining and knowledge discovery. Generally, depending on the definition of an *event* in specific application context, different data features are extracted from social media streams and summarized for event detection tasks. Solutions have been proposed to detect social events by exploiting the information over content, temporal, and social dimensions. In [36], Zhao et al. define an event as a set of relations between social users on a specific topic over a time period. A social text stream is represented as a multigraph. Events are extracted from social streams by three steps, the text-based clustering, the temporal segmentation, and the graph cuts of social networks. In [37], to enhance the performance of event detection, Zhao et al. proposed a temporal information flow-based method. An event is described by the information flow between a group of social actors on a specific topic over a certain time slot. Unlike [36], information flow-based method divides the graph to a topic obtained from text-based clustering into a series of intervals by temporal intensity-based segmentation. The results are further optimized using information patterns and dynamic time wrapping measure. In [28], Yao et al. proposed to extract bursts from multiple social media sources. A state-based model is first used to detect bursts from each single stream, and then, the bursts identified from various sources are combined to form the final results. Using this approach, the relations between the detected events and the interaction between content and users can be fully uncovered.

Table 1 Notation

Notation	Meaning	Defined in (section)
$\mathcal{M}$	A social media stream	3
$\mathcal{D}$, $\mathcal{Q}$	A social message	4.1
T	The number of topics	4.1
z_n	A topic	4.1
w_n	A word	4.1
t_n	A time stamp	4.1
la_n /lo_n	The latitude/longitude of a location	4.1
N_d	Number of word tokens in document d	4.1
D, Q	A social link	4.2
τ	A time threshold	4.2
ϵ	The social message similarity threshold	3
ω	The weight parameter of two parts in social similarity	4.3
$\tilde{t}_1$, $\tilde{t}_2$	An uncertain time stamp	4.2
S_E	A set of social events	5
S_D	A set of social messages in a time window	5

Later, the same authors proposed an indexing structure to support fast event detection in micro-blog data [29]. In [15], topic models are trained using hashtags in the tweet stream for tracking broad topics, such as "baseball", and unigram language models show a good balance between maintaining recency and combating sparsity. Exploiting the information from three dimensions improves the accuracy of event detection. However, these approaches neglect the locations of social events. Moreover, an important assumption of these approaches is that messages on an event share some common keywords. Thus, the multiple event views without common keywords cannot be discriminated. Although recent works consider the time and geographical information for collecting situation information from twitter [31,32] or annotating events from online media sharing sites such as Flickr [18,24,40], they did not considered the uncertainty issues that are very common in social networks.

Link-based detection identifies abnormal events from email communication data [21]. In this application, the email contents are usually protected by privacy, thus unavailable to us. The only available data source is linkages. Wan et al. [21] proposed to identify abnormal email communication patterns in the email network caused by real-world events. The whole email communication network is represented as a graph, where each vertex is an email account and each edge is an email communication. Abnormal events are detected by checking the individual deviation and cluster deviation based on the individual and neighborhood features. This method provides an event detection solution for email data. However, since it exploits the information from single source, the accuracy of detection cannot be guaranteed for general social applications. In [19], approach has been proposed to detect a target event by monitoring tweets in Twitter. Each target event is defined as an arbitrary classification of a space/time region. Tweets are searched and classified using a support vector machine. A target event is detected by a temporal model which is constructed as a probability model. Location of a certain event is estimated by the Bayesian filters. This method is designed for a certain event and customized for earthquake application. It targets disaster location prediction, while not delivering unknown events or complete detailed crisis situations.

2.2 Document representation

Several approaches have been proposed to represent text documents. A popular representation method is to describe a document using its keywords [8]. Recently, applying topic models for document representation has become a new interest in machine learning and information retrieval. Typical topic models include latent Dirichlet allocation (LDA) [4], topic over time (TOT) [23], online LDA (OLDA) [2], GeoFolk [20], and latent geographical topic analysis (LGTA) [33] etc. Instead of directly comparing keywords of documents, these topic models represent a document as a probabilistic distribution over multiple topics.

In [4], Blei et al. proposed a three-level hierarchical Bayesian model LDA for document representation. The key idea of LDA is to represent a set of documents as random mixtures over an underlying set of topics, where a topic is described as a probability distribution over a number of words in a predefined vocabulary. It assumes that the words in a document are unordered. The process of generating a corpus with LDA is performed by four steps. For each document $\mathcal{D}$, a multinomial distribution θ over topics is chosen from a Dirichlet with parameter α. A topic is then chosen from this topic distribution. For a specific topic z_{di}, a word distribution $\phi_{z_{di}}$ is selected to produce a specific word by randomly sampling word from it. As a Bayesian-based model, LDA well handles the incompleteness and uncertainty in documents. Because of its high flexibility in document generation, LDA has been successfully applied in document summarization in information retrieval [5,22,25,30].

To adapt LDA to dynamic environment, approaches have been proposed to model time jointly with word co-occurrence pattern of documents [23] or update the model incrementally based on the information inferred from the new stream of data [2]. In [23], TOT model is presented to attach each topic with a continuous distribution over time. It extends the LDA by enabling topics to generate both words and time stamps. In TOT, the mixture distribution over topics of a generated document is decided not only by word co-occurrences as in LDA, but also by the document's time information. Using TOT, the time stamp of a document is well combined with word co-occurrence as a constraint in document generation. In [2], AlSumait et al. proposed OLDA that extends LDA from offline to online version. In OLDA, the learned topics are incrementally adjusted according to the dynamic changes in the data, in which words in the incoming documents are sampled based on the latest presented distribution. Though these models presented solutions for dynamic document modeling, they are inapplicable to social media-based crisis applications as the locations of data are not considered.

Models have been proposed to integrate spatial information with text in social networks like Flickr [20,33]. In [20], each tag is generated by selecting a topic from a topic distribution and then drawing a tag from a topic-specific multinomial distribution. Meanwhile, the latitude and longitude of this document are generated from two topic-specific Gaussian distributions. The model was successfully applied to three realistic scenarios for social media including content classification, content clustering, and tag recommendation. In [33], Yin et al. proposed LGTA that generate topics from regions for geographical topic discovery and comparison of the topics across different locations. The regions are identified with respect to both location and text information, and

the geographical topics are discovered based on the identified geographical regions. The geographical distribution of each region follows a Gaussian distribution. With this model, each topic can be related to several regions and the topics with complex shapes can be well handled. Although these two models well embedded geographical information into text content, they were proposed for static social media sources, while ignore the time information in streams.

3 Problem formulation

In this section, we formally define the problem of social event detection. Before proceeding to the problem formulation, we first introduce two vital concepts, *Event Element* and *Event*.

Definition 1 In social networks, an input message is an observation from a user. An *event element* is defined as an observation on a real-world occurrence happening at a certain location and time. Given a set of event elements $E =< \mathcal{D}_i >$, an *event* is defined as a subset E_i of E, such that all the event elements in E_i are related to a specific real-world occurrence over a location and time range, and describe its multiple attributes such as what is happening, who are involved, where it is happening, and its effect to the surrounding.

Generally, an event consists of four parts, message content, location, time, and social connection. Content information describes what is happening, while location and time serve as a constraint for the prediction of where and when an event is happening. Naturally, messages with similar content and sent from the neighboring locations in close time periods describe the views of the same event. However, practically, in Twitter, the conversation about the same event between two users may be less similar in content. In this case, it is not enough to judge if two messages describe the same event based on their content. Embedding the social links of different users will be promising for the information compensation in event detection. We extract four types of features to describe a social message.

- Content: The textual content description on a specific message or a number of related messages, normally referring to the keywords of social messages after the specific stop words are excluded from them.
- Location: The location associated with the profile of each user, such as city name, country name, suburb name, postcode. A location is mapped into a point (la, lo), where la denotes its latitude and lo is its longitude.
- Temporal information: The time stamp attached to each message. It shows the posted time of the messages, indicating the approximate time of an event.
- Social information: The followers related to the message. The social information indicates the links between the current users and their followers, and the connections between their messages.

We address the problem of detecting social events from streams by taking into account four types of features in messages. Social event detection is generally related to similarity retrieval tasks, where the key point is how to conduct the similarity join between messages with rich social data features. The social event monitoring is formally defined as follows:

Definition 2 Given a social stream $\mathcal{M}$, a similarity threshold ϵ, and a distance function $Dist$, social event detection automatically returns a set of composite social events $< E_i >$, where $\forall D_i \in E_i, \exists D_j \in E_i$ such that $Dist(D_i, D_j) \leq \epsilon$.

A composite event includes all the aspects of an occurrence, such as what is happening, who are involved (e.g., people, animals, roads, or other objects etc.), effects to the surrounding. Figure 1 shows the information of the event, QLDflood-Darling Downs, over the stream from 10/01/2012 7:05am to 10/01/2012 7:16am. It describes the flood in Darling Downs area from different aspects, such as the water contamination, road close, film scene, footage flood, car damage, person dead, and miss. Our work is to address the problem of effective and efficient event detection over social streams. In the next section, we will give details on how to perform similarity-based event detection over social streams using the

<id> 24361112182984704 </id> <l> -27.57 151.94</l> <t> 10 jan 2011 7 5 16 </t> toowoomba disast control centr confirm contamin water suppli doubt boil water
<id> 24361215278981121 </id> <l> -27.47 153.02</l> <t> 10 jan 2011 7 5 40 </t> eeep hope friend toowoomba go flood road rang lowood close
<id> 24361219389390849 </id> <l> -27.47 153.02</l> <t> 10 jan 2011 7 5 41 </t> toowoomba smash massiv downpour surg seriou damag film scene insan pic
<id> 24361432493596672 </id> <l> -27.47 153.02</l> <t> 10 jan 2011 7 6 32 </t> mini crazi video footag toowoomba flood fb
<id> 24361740259041280 </id> <l> -27.47 153.02</l> <t> 10 jan 2011 7 7 46 </t> love insan car float rapid toowoomba qldflood
<id> 24362182334484480 </id> <l> -28.00 153.42</l> <t> 10 jan 2011 7 9 31 </t> flash flood toowoomba wipe car water bad mkai
<id> 24362401658830848 </id> <l> -27.61 152.75</l> <t> 10 jan 2011 7 10 23 </t> flash flood toowoomba scari person dead miss qldflood
<id> 24362440418398208 </id> <l> -26.80 153.12</l> <t> 10 jan 2011 7 10 33 </t> heard person confirm dead toowoomba flashflood sad qldflood
<id> 24362692298936320 </id> <l> -27.57 151.94</l> <t> 10 jan 2011 7 11 33 </t> frighten abc new footag flash flood toowoomba
<id> 24363073024294912 </id> <l> -27.47 153.02</l> <t> 10 jan 2011 7 13 3 </t> love insan car float rapid toowoomba qldflood
<id> 24363228209348608 </id> <l> -28.00 153.42</l> <t> 10 jan 2011 7 13 40 </t> crisi land slide toowoomba esk water stanthorp evacu
<id> 24363721610493953 </id> <l> -26.80 153.12</l> <t> 10 jan 2011 7 15 38 </t> heard person confirm dead toowoomba flashflood sad qldflood
<id> 24363753814360065 </id> <l> -27.47 153.02</l> <t> 10 jan 2011 7 15 46 </t> jeez hope ppl toowoomba alright look pretti bad water hit

Fig. 1 An example of a composite event

content, temporal, location, and social information. We first propose a novel graphical model over multiple attributes of messages, including content, time, and location. Using this model, each twitter message (also called *tweet*) is represented as a probabilistic distribution over a number of topics, and the content similarity between two messages is measured over their distributions. Then, we propose the link similarity to capture the social contacts of users over an event. Based on the content similarity and link similarity, the global social media similarity is defined to detect the complete view of an event. Finally, based on this similarity model, we further design a hash-based index scheme which improves the efficiency of online similarity join for social event detection over streams.

4 Social media modeling

In this section, we present our data model together with our similarity measure for social messages. We will first present our location-time constrained topic model with the content similarity based on it. Then, we will propose our link similarity to capture the social contacts within a user conversation. We finally formulate our complementary measure by embedding the link similarity into the content similarity.

4.1 Location-time constrained topic model

As introduced previously, a social message can be described as a set of keywords. Given a corpus of keyword sets, various models such as pLSI [12] and LDA [4] can be used for document representation and topic detection. Among these models, LDA has received great attention due to its advantage of processing unknown document patterns. LDA is a Bayesian network that generates a document using a mixture of topics. In our application, an event cares not only what happens but also the location and time of its happening. Thus, when we define a model for document representation, we should consider the constraints of location and time on the message content.

Since social messages are input by different users manually, a large amount of noise and incomplete or misspelling words exist in them, which makes the message contents usually uncertain. Meanwhile, given a message, its post time may not be the exact time of an event. A message may be posted several minutes after the event. The location of a user may not be the actual location of an event. The user may register account at one place, but travel to some other places nearby and send messages to report an event other than his register location. Accordingly, the temporal and location information obtained from the social networks can be uncertain as well. Thus, when we build a model for social messages, we have to consider their uncertainty property with respect to all these factors. We propose a location-time constrained topic (LTT) model which extends the LDA on traditional text documents to social texts. Unlike the TOT model which extends the LDA by considering the time of a document [23], LTT considers both the time and location of each message as additional variables. Meanwhile, our LTT takes advantage of the OLDA by constructing the model over each subset within a time slot and uses the model for a time period to infer social messages in its following time slot [2]. The generative process of a LTT model for a message $\mathcal{D}$ in a stream $\mathcal{M}$ is as follows:

1. Choose T multinomials ϕ_z from a Dirichlet prior β, one for each topic z_n
2. For each message $\mathcal{D}$, draw a multinomial θ from a Dirichlet Prior α.
3. For each of the words w_n in the message:
   - Choose a topic $z_n \sim Multinomial(\theta)$.
   - Choose a word $w_n \sim Multinomial(\phi_{z_n})$.
   - Choose a time stamp $t_n \sim Beta(\psi_{z_n})$, a Beta distribution conditioned on the topic z_n.
   - Choose a location latitude $la_n \sim Beta(\delta_{z_n})$, a Beta distribution conditioned on the topic z_n.
   - Choose a location longitude $lo_n \sim Beta(\gamma_{z_n})$, a Beta distribution conditioned on the topic z_n.

The LTT graphical model is shown in Fig. 2. Similar to the LDA model [4], the LTT representation contains three levels: corpus-level parameters, document-level variables, and word-level variables. Unlike the LDA that only considers a single token at the word level, the LTT model attaches the location longitude, latitude, and time variables with each token. As such, the uncertain time, location, and social content information can be fused together. As shown in Fig. 2, the posterior distribution of topics depends on the information from three types of attributes, content, location, and time. Comparing with the LDA model, the LTT model contains three more parameters, t_n, la_n, and lo_n. Since an inference approach cannot be found in this model, we adopt Gibbs sampling to perform approximate inference following the

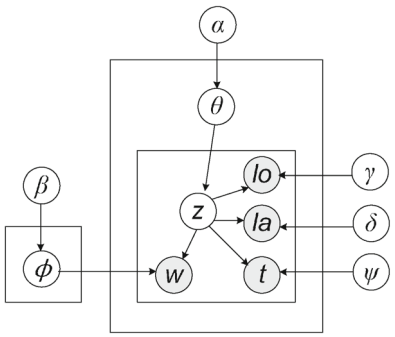

Fig. 2 A LTT model example

suggestion in LDA [4]. The α and β can be estimated from data. Following the same setting given by the existing works [4,23], we use fixed symmetric Dirichlet distributions with $\alpha = 50/T$ and $\beta = 0.1$ for simplicity. We choose Beta distribution for time and location dimensions because of its flexibility in representing various skewed shapes. The Beta distributions ψ_z, δ_z, and γ_z are estimated by the method of moments[1] based on the data of topics in every iteration of Gibbs sampling. Although the involvement of location and time in LTT model causes higher time cost in each training comparing with the LDA model, it greatly speeds up the convergence speed in each training at the same time, thus keeping the high efficiency of the whole learning process. In the Gibbs sampling, we need to calculate the conditional distribution $P(z_{di}|w, t, la, lo, z\_{di}, \alpha, \beta, \psi, \delta, \gamma)$, where z_{di} is the topic assignment for all tokens except w_{di}. We start with the joint distribution $P(w, t, z, la, lo|\alpha, \beta, \psi, \delta, \gamma)$. Based on the joint probability of a dataset and the chain rule, the conditional probability is obtained as Eq. 2.

$$
\begin{aligned}
&P(w,t,z,la,lo|\alpha,\beta,\Psi,\delta,\gamma)\\
&= P(w|z,\beta)p(t|\psi,z)p(la|\delta,z)p(lo|\gamma,z)P(z|\alpha)\\
&= \int P(w|\Phi,z)p(\Phi|\beta)\mathrm{d}\Phi p(t|\Psi,z)p(la|\delta,z)p(lo|\gamma,z)\\
&\quad \int P(z|\Theta)p(\Theta|\alpha)\mathrm{d}\Theta\\
&= \int \prod_{d=1}^{|S_D|}\prod_{i=1}^{N_d} P(w_{di}|\phi_{z_{di}}) \prod_{z=1}^{T} p(\phi_z|\beta)\mathrm{d}\Phi\\
&\quad \prod_{d=1}^{|S_D|}\prod_{i=1}^{N_d} p(t_{di}|\psi_{z_{di}}) \prod_{d=1}^{|S_D|}\prod_{i=1}^{N_d} p(la_{di}|\delta_{z_{di}}) \prod_{d=1}^{|S_D|}\prod_{i=1}^{N_d} p(lo_{di}|\gamma_{z_{di}})\\
&\quad \times \int \prod_{d=1}^{|S_D|}(\prod_{i=1}^{N_d} P(z_{di}|\theta_d)p(\theta_d|\alpha))\mathrm{d}\Theta \qquad (1)\\
&P(z_{di}|w,t,la,lo,z\_{di},\alpha,\beta,\Psi,\delta,\gamma)\\
&= \frac{P(z_{di},w_{di},t_{di},la_{di},lo_{di}|w\_{di},t\_{di},z\_{di},la\_{di},lo\_{di},\alpha,\beta,\Psi,\delta,\gamma)}{P(w_{di},t_{di},la_{di},lo_{di}|w\_{di},t\_{di},z\_{di},\alpha,\beta,\Psi,\delta,\gamma)}\\
&\propto \frac{P(w,t,z,la,lo|\alpha,\beta,\Psi,\delta,\gamma)}{P(w\_{di},t\_{di},z\_{di},la\_{di},lo\_{di}|\alpha,\beta,\Psi,\delta,\gamma)}\\
&\propto (m_{dz_{di}}+\alpha_{dz_{di}-1})\frac{n_{z_{di}w_{di}}+\beta_{w_{di}}-1}{\Sigma_{v=1}^{V}(n_{z_{di}v}+\beta_v)-1}\\
&\quad \times \frac{(1-t_{di})^{\psi_{z_{di}1}-1}t_{di}^{\psi_{z_{di}2}-1}}{B(\psi_{z_{di}1},\psi_{z_{di}2})} \frac{(1-la_{di})^{\delta_{z_{di}1}-1}la_{di}^{\delta_{z_{di}2}-1}}{B(\delta_{z_{di}1},\delta_{z_{di}2})}\\
&\quad \times \frac{(1-lo_{di})^{\gamma_{z_{di}1}-1}lo_{di}^{\gamma_{z_{di}2}-1}}{B(\gamma_{z_{di}1},\gamma_{z_{di}2})} \qquad (2)
\end{aligned}
$$

where n_{zv} is the number of tokens of word v assigned to topic z, m_{dz} that in document d assigned to topic z. The parameters, Ψ, δ, and γ, are updated after each Gibbs sample by the equations below:

$$\hat{\psi}_{z1} = \bar{t_z}\left(\frac{\bar{t_z}(1-\bar{t_z})}{s_{zt}^2} - 1\right) \qquad (3)$$

$$\hat{\psi}_{z2} = (1-\bar{t_z})\left(\frac{\bar{t_z}(1-\bar{t_z})}{s_{zt}^2} - 1\right) \qquad (4)$$

$$\hat{\delta}_{z1} = l\bar{a}_z\left(\frac{l\bar{a}_z(1-l\bar{a}_z)}{s_{zla}^2} - 1\right) \qquad (5)$$

$$\hat{\delta}_{z2} = (1-l\bar{a}_z)\left(\frac{l\bar{a}_z(1-l\bar{a}_z)}{s_{zla}^2} - 1\right) \qquad (6)$$

$$\hat{\gamma}_{z1} = l\bar{o}_z\left(\frac{l\bar{o}_z(1-l\bar{o}_z)}{s_{zlo}^2} - 1\right) \qquad (7)$$

$$\hat{\gamma}_{z2} = (1-l\bar{o}_z)\left(\frac{l\bar{o}_z(1-l\bar{o}_z)}{s_{zlo}^2} - 1\right) \qquad (8)$$

After applying the LTT model, a social message is described as a vector of probabilities over the space of topics which depend on the words, time stamps, and locations of messages. We then need a "good" function to measure the dissimilarity between two distributions. We choose the Kullback–Leibler (KL) divergence for our message similarity considering its advantages, such as well-defined for continuous distributions, and invariant under parameter transformations [10]. Given two probability distributions of two messages, $\mathcal{D}$ and $\mathcal{Q}$, the KL divergence measures the expected number of extra bits required to code samples from $\mathcal{D}$ when using a code based on $\mathcal{Q}$. For probability distributions $\mathcal{D}$ and $\mathcal{Q}$ of a discrete random variable i over topics, their KL divergence is defined as below:

$$D_{\mathrm{KL}}(\mathcal{D}||\mathcal{Q}) = \sum_i \mathcal{D}(i)\log\frac{\mathcal{D}(i)}{\mathcal{Q}(i)} \qquad (9)$$

The KL divergence is not a true metric, since it does not meet the property of symmetry. Thus, we define the following real distance function based on it.

$$D_{\mathrm{LTT}}(\mathcal{D},\mathcal{Q}) = \frac{1}{2}(D_{\mathrm{KL}}(\mathcal{D}||\mathcal{Q}) + D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{D})) \qquad (10)$$

The D_{LTT} is symmetric, while preserves the advantages of KL divergence. As the LTT model fuses the information from content, time and location, our D_{LTT} measure captures the dissimilarity between two social messages over these three types of attributes.

4.2 Time constrained link similarity

The link information between users has been used as an effective way of event identification in email communications [21]. Using the linkages, messages on the same conversation may be found and clustered together. When messages are posted and replied via twitter, the user pair related to a given message, the current social user and its followers, is the only information that can be obtained directly. We believe a social message is applied within a limited time period and define

[1] http://en.wikipedia.org/wiki/Beta_distribution.

Fig. 3 A tweet message example

a function to measure the link similarity between two messages with time constraint. Two social users associated with a message indicate a link between them on a conversation. Given a social message, we obtain its current and following user accounts together with its post time. Each user account is a record described as a single *id*. Given a message $\mathcal{D}$, its social link is described as a series of *ids* related to it, where the order of *ids* indicates the hierarchy in the conversation. Figure 3 shows an example of social message. Suppose the ids of "carleejones," "kaz2230," "Chaslicc," and "jasonelcher" are 1, 2, 3, 4, respectively. Then, the social link of the tweet in Fig. 3 is described as a string "1234". The common *id* subseries of two *id* series reflects the connection between two user messages. Given two series, $D =< d_1, \ldots, d_m >$ and $Q =< q_1, \ldots, q_n >$, of *ids* describing the social links of two messages, we measure their similarity by the *Longest Common Subseries* between them. Let $\mathrm{LCS}(D_i, Q_j)$ represent the set of longest common subsequences of prefixes D_i and Q_j. The similarity between sequences is computed by the following equation.

$$\mathrm{LCS}(D_i, Q_j) = \begin{cases} 0 & i = 0 \; or \; j = 0 \\ \mathrm{LCS}(D_{i-1}, Q_{j-1}) + 1 & d_i = q_j \\ \max\{\mathrm{LCS}(D_i, Q_{j-1}), \mathrm{LCS}(D_{i-1}, Q_j)\} & d_i \neq q_j \end{cases} \quad (11)$$

We normalize the LCS similarity by considering the sizes of the element union in two series. Given two series D and Q, the similarity between them is measured by:

$$\mathrm{LCS}_l(D, Q) = \frac{\mathrm{LCS}(D_m, Q_n)}{|D_m \bigcup Q_n|} \quad (12)$$

Given that the messages we process are instant, and two messages on the same event should be posted within a time threshold τ minutes. Two messages with a smaller time gap are more likely about the same event. Since the time information is uncertain, we use the probabilistic similarity between two uncertain time stamps to measure the difference of their time. Given two uncertain time stamps, $\tilde{t}_1$ and $\tilde{t}_2$, we treat $\tilde{t}_1$ and $\tilde{t}_2$ as random variables with arbitrary distribution. Given a time threshold τ, the possibility of the distance between $\tilde{t}_1$ and $\tilde{t}_2$ smaller than τ is computed by

$$\mathcal{T} = \Pr(\mathrm{dst}(\tilde{t}_1, \tilde{t}_2) \leq \tau) \quad (13)$$

where Pr(.) denotes the probability of an event, dst is the Euclidean distance. Given $\tilde{t}_1$ and $\tilde{t}_2$, $\mathcal{T}$ can be easily obtained by first sampling points from their distributions and then computing the distance over their samples. We define the link similarity between two messages by embedding the uncertain time constraint into the similarity between social user series. Given two messages $\mathcal{D}$ and $\mathcal{Q}$, let D and Q be their social user series, $\tilde{t}_D$ and $\tilde{t}_Q$ be their time variables, respectively, the link similarity between them is defined as:

$$\mathrm{Sim}_{wl}(\mathcal{D}, \mathcal{Q}) = \mathrm{LCS}_l * \mathcal{T} \quad (14)$$

We derive the link difference between two messages from their similarity as below:

$$D_{wl} = 1 - \mathrm{Sim}_{wl}(\mathcal{D}, \mathcal{Q}) \quad (15)$$

D_{wl} models the temporal social conversion among a group of users that could not be reflected in the LTT model. The time constraint in D_{wl} requires the messages on one conversion be posted in a time period, while the time in LTT model indicates the approximate time of the event element to a message.

4.3 Social media similarity

Having defined the content similarity and link similarity between two messages, we can integrate them to formalize their similarity globally. We believe that the overall difference between two messages is affected by their content difference and link difference to different extent. Given two messages $\mathcal{D}$ and $\mathcal{Q}$, their global social media similarity is defined as:

$$D_G(\mathcal{D}, \mathcal{Q}) = (1 - \omega) D_{\mathrm{LTT}}(\mathcal{D}, \mathcal{Q}) + \omega D_{wl}(\mathcal{D}, \mathcal{Q}) \quad (16)$$

where ω is a parameter related to the weights of the similarity components. We investigate the effect of two similarity components by varying their weights. D_G is a complementary distance which considers the content and link differences between two messages. Using this complementary distance, the difference between two messages is captured over four attributes: content, location, time, and social connection. Meanwhile, the ambiguity and uncertainty of social messages are taken into consideration. The time cost of D_G depends on the number of topics in the LTT model and the lengths of compared links. Suppose that the topic number of the LTT model is T, the lengths of two compared links are m and n, respectively, then the time cost of D_G would be $O(T + m * n)$.

5 Continuous event detection

We now present the details of our event detection approach. Intuitively, in social networks, messages are transferred over a time period among users. Consequently, the changes in social networks caused by an event happened in a location usually exhibit strong correlations. When an event occurs at a certain location, messages on this event are posted by certain users and spread to affect their followers in the next time slot. This observation indicates that the messages on the same event is likely to be found from the posts of social connected users, and one social user is likely to talk about the same event in consecutive time slices [14]. Thus, it is reasonable to detect event elements by the similarity join over social streams within consecutive timeslots, so an integrated global view of an event is formed by recursively merging matched pairs.

Given a set of social messages within a time window S_D, a distance function D_G, and a similarity threshold ϵ, the similarity join finds all pairs of messages, $< \mathcal{D}, \mathcal{Q} >$, such that the distance between them is no bigger than the given threshold ϵ, i.e., $D_G(\mathcal{D}, \mathcal{Q}) \leq \epsilon$. To perform the ϵ-similarity join of social messages, a naive method is to compute the similarity between each pair of messages. Given a set of n messages, the computation complexity of this naive method is $O(n^2)$, thus inappropriate to high speed social streams. Traditional high-dimensional index structures, such as R-tree and its variants [3,9], or B^+-tree-based high-dimensional data index techniques, such as iDistance [13,34,38] or Multiple B^+-trees [39], are not designed for online environment, thus inapplicable to our problem either. We propose a new detection procedure which compares an incoming message and each of the previous clusters to find the most similar cluster to this incoming message in short time. The previous clusters are created by grouping the similar social messages in the previous time periods. The cluster most similar to the incoming message is its destination composite event. Extracting matched messages of a certain cluster will be stopped in the next time period if it does not receive any matched social message in the current time period. A number of clusters to different events will be output by the end of the monitoring over tweet streams. A cluster to an event can be from a single time slot or span multiple time periods as well. The new procedure creates a dynamic hash structure for the detected messages of social events. Each bucket of the hash contains messages with high similarity. When a new message comes, we compare it with the previous social clusters to find its destination. After the new event decision is made, the current social message is inserted into the hash structure for later event detection. Next, we first present our variable dimensional extendible hash (VDEH) and then discuss the operations over this index structure, including the message insertion, deletion, the similarity join, and the query optimization over our VDEH.

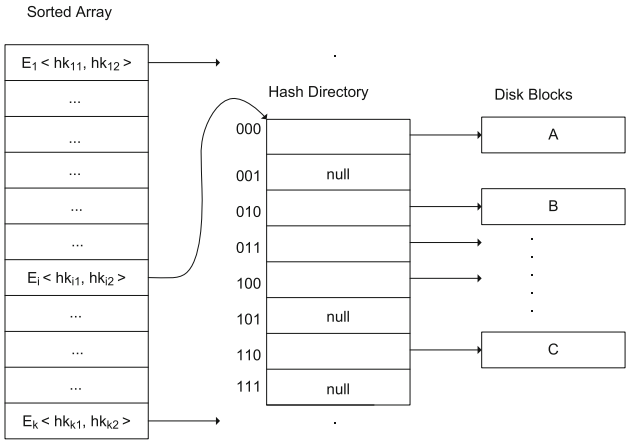

Fig. 4 VDEH index structure

5.1 Variable dimensional extendible hash

The new event detection creates a variable dimensional extendible hash dynamically. The structure includes: (1) a sorted array used to maintain the hash address of different events; (2) a number of hash tables pointing to different buckets containing social messages. To save the memory and CPU costs, we only store the messages in the latest time slot in this index and perform the similarity join over consecutive time periods. Considering the suggestion on the time slot size (one day, 1 h) in [21] and the huge amount of tweets in one day, we set the size of each time slot to 1 h in the detection. Figure 4 shows our index structure.

The sorted array is attached with a global codebook consisting of all topics and the words producing them in the latest time period. Using the LTT model, we fix the total number of topics during the detection, while permit the topic shift to fit the streaming environment. In other words, the kth topic in the latest time slot may be different from the kth one from its following period. To incorporate topic drift, our LTT model is refreshed after every block of tweets in an incoming time slot. The topic change over time is decided by the symmetrized KL divergence over their word distributions. Given two topics z_1 and z_2 belonging to continuous time slots, let P_1 and P_2 be the word distributions of z_1 and z_2, respectively, the topic drift between them is computed as follows.

$$D_{TD}(P_1, P_2) = \frac{1}{2}\left(\sum_i P_1(i) \log \frac{P_1(i)}{P_2(i)} + \sum_i P_2(i) \log \frac{P_2(i)}{P_1(i)}\right) \tag{17}$$

Equations 17 and 10 are similar, deriving from KL divergence in the same way. Here, we do not share equation 10 for topic drift to clarify that the distributions used in two equations have different meanings. Borrowing the idea for the assessment on topic similarity in [26], we randomly selected 150 pairs of consecutive topics from the Queensland flood

dataset and divided the topic pairs into two classes. The pairs with a D_{TD} no bigger than 0.2 are put into the similar class, while the rest of pairs are put into the dissimilar class. For each class, the topic pairs are labeled based on inter-subject agreement. We compare the classes based on topic similarity and the labeled results. The results showed 98 % of topic pairs have reached agreement under two classification methods. Thus, we use KL divergence for topic similarity assessment, and consider the pairs with D_{TD} bigger than 0.2 as dissimilar. Each entry of the array points to the starting address of the hash table to an event. Given an event E_i, the messages within the recent time slot are mapped into hash keys based on the global codebook, forming its hash range $< hk_{i1}, hk_{i2} >$. Each hash range is stored in an entry to it. Each event contains a set of messages and is organized using an extendible hash structure, which maintains the messages in the latest time slot, and is attached with a local codebook containing the topics producing this event.

Existing extendible hashing techniques handle data with fixed dimensionality [16]. However, for our application, since the topics over social streams within different time slots may change dynamically, the dimensionality of the data we process (i.e., the topics of an event in recent period) varies accordingly. Here, a topic is described as a distribution over a set of keywords, and an event may cover multiple topics. The changes on social data dimensionality include both the number of dimensions and the attribute of each single dimension. Figure 5 shows an example of topic evolvement on QLDflood–SouthEast over 30 h time periods. Borrowing the measure of topic strength over time in [17], we compute the strength of topic T_i over the sub-set of message with a given time stamp t by Eq. 18. Here, d_k is the kth document in collection $D(t)$, which is the set of documents at time t. $L(d_k)$ is the normalized length of document d_k, while $P(T_i|d_k)$ calculates the distribution of topic T_i in d_k. Some topics in the previous stream may disappear, while new topics may appear in the current time slot. Thus, we need to consider the dynamic change of the data dimensionality while expanding hash directory to fit the recent social events. To handle this problem, we propose a VDEH. Unlike the existing techniques that expand the hashing address by adding bits to the least/most significant position [16], our VDEH performs bidirectional address expansion. The hash directory growth is done by adding a bit to the most significant position if any of the buckets overflows. When a new topic is added into the code book to a certain hash table, the hash directory for this table is expanded by adding a bit to the least significant position. We use a mask track to record the topic change in an event. Following [16], another mask track is used for each hashing table to keep the track of the split direction in the bucket. As such, the directory of each hash table grows slowly.

$$S_i[t] = \sum_{d_k \in D(t)} L(d_k) \times P(T_i|d_k) \tag{18}$$

5.2 Insertion

We identify the elements of an event by checking the similarity between the recent social message and those stored in the VDEH, and inserting the incoming message to its matched event cluster. Given an incoming message $\mathcal{D}$, the insertion procedure is performed in two steps: (1) identify the destination cluster E_i; and (2) update the index by inserting $\mathcal{D}$ into the right bucket and updating the hash addresses in the index if necessary. Given a cluster E_i, the similarity between E_i and $\mathcal{D}$ is the density of matches in E_i, which is computed by Eq. 19. The destination cluster of message $\mathcal{D}$ is the one that has the maximal similarity with it. We check the status of the destination bucket B_i. The social message is inserted into

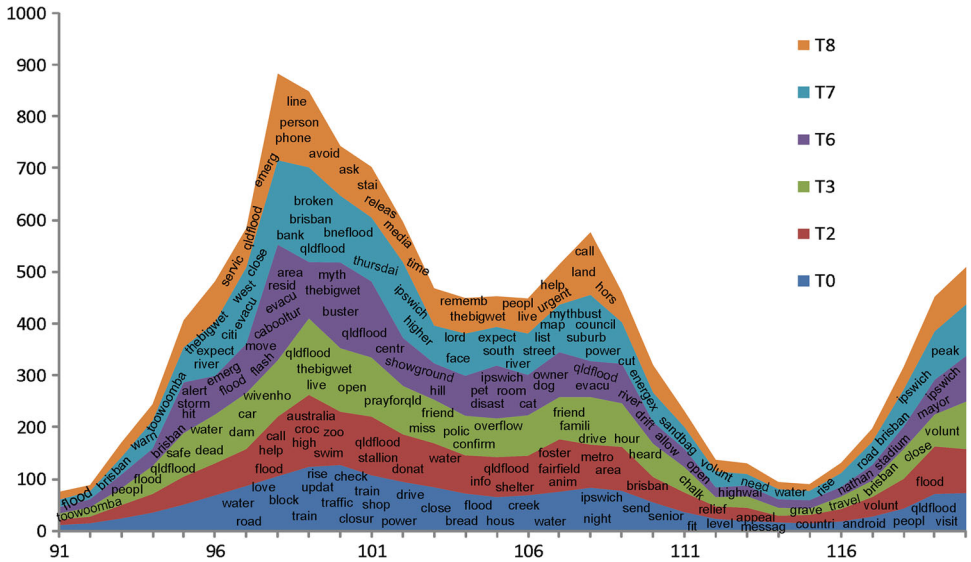

Fig. 5 Topic evolvement of the event QLDflood-Darling Downs

E_i directly if B_i is not full. Otherwise, the hash addresses increase and the bucket of B_i splits.

$$\text{Sim}_E(E_i, \mathcal{D}) = \frac{\text{Number of matches of } \mathcal{D} \text{ in } E_i}{\text{Number of messages in } E_i} \quad (19)$$

The detailed algorithm for social message insertion is shown as Fig. 6. First, we look for the cluster similar to the incoming message (line 1). If E_i is found, we identify the suitable position to store $\mathcal{D}$ (line 2–18). We check if a new topic has appeared with the new message and expand the hash index address by adding bits to the least significant positions (line 3–4). Then, we check the status of the destination bucket B_i. The social message is inserted into E_i directly if B_i is not full (line 5–7). The hash address space increases and the bucket of B_i splits in case that B_i overflows (line 8–14). The elements in B_i and the incoming one are redistributed between B_i and the new bucket B_j (line 10). A new bucket generated is inserted into the hash table directly if there is space in the hash directory to accommodate it (line 11–12). Otherwise, the directory is doubled for fitting the new bucket (line 13–14). If E_i cannot be found, a new event is identified and inserted into the index (line 15–17). A new entry is inserted into the sorted array and points to the hash table of this new event (line 16). When a new event is found, the topics of the new event are brand new for itself. In this case, we need to increase the hash address range of the new cluster by adding additional bits to the least significant position of its hash directories (line 17). Finally, the hash key ranges stored in the sorted array are updated, so each range reflects the new hash directory addresses of the corresponding event (line 18).

```
Procedure MessageInsertion(I, D, ε).
   input:  I - a hash index, D- a social message
           ε - social message similarity threshold
1. E_i ←FindDestinationCluster(I, D, ε)
2. if E_i is not null
3.    if D introduces new topics for E_i
4.       IncreaseLSHashAddress /*least significant position*/
5.    Let B_i be the destination bucket
6.    if B_i is not full
7.       B_i ← D
8.    else
9.       B_j ← BucketSplit   /*B_j is a new bucket*/
10.      ElementRedistribution(B_i, B_j, D)
11.      if hash directory is available
12.         InsertBucket(B_j)
13.      else
14.         IncreaseMSHashAddress
             /*most significant position*/
15. else      /*a new event is found*/
16.   I ←ProduceNewEvent
17.   IncreaseLSHashAddress
18. UpdateSortedArray
```

Fig. 6 Inserting a social message

5.3 Deletion

Since the social messages of an event usually appear in consecutive time periods, we only perform the similarity join over messages in consecutive time slots to save the time cost. Once all the messages in the current time slot are inserted into the hash index, we store the messages in the previous time slot in their cluster files and remove them from the hash table. Removing the expired messages from the index structure reduces the memory cost of event detection and the computational cost of similarity join over social messages. In case that a composite social event does not receive any similar event element from the current time slot, we believe the detection on this event has been finished, and should be output and removed from the hash index as well. Note that we only focus on the event detection in this paper, while leave outputting events to different users for future investigation on event recommendation. After the expired messages are deleted from a hash index, we check the corresponding local codebook and delete the expired topics. The bits to the deleted topics are deleted, and the hash address of the corresponding event is reduced. The blank buckets are released, and the neighboring buckets are merged to save the space cost. We update the hash address range of each affected event to reflect this deletion operation.

5.4 Similarity join

Using the VDEH over the recent historical social messages, we can find the matched composite event of an incoming message by simply performing similarity query over the index. Since a number of users may send instant messages to a social network at a certain time stamp, we need to simultaneously identify all the matched pairs $< \mathcal{D}, \mathcal{Q} >$, where $\mathcal{Q}$ is an incoming social message at a certain time, and $\mathcal{D}$ is a historical message belonging to a recent event. To do this, we perform similarity join over two social message sets that contain the incoming social messages and the historical ones, respectively. Suppose that m social messages come at a certain time, this similarity join operation can be performed by m similarity queries for them. Given an incoming social message $\mathcal{Q}$ and a distance threshold ϵ, a similarity query identifies all message pairs $< \mathcal{D}, \mathcal{Q} >$, such that $D_G(\mathcal{D}, \mathcal{Q}) \leq \epsilon$. We perform similarity query over VDEH to quickly find the matched message pairs over two social message sets.

To perform the similarity query of an incoming message, we compute the hash address ranges, which decide its potential matched events and further specific buckets. The similarity query algorithm will perform the search by three steps. First, it locates all the events whose regions overlap the search space based on the global codebook from the sorted array. Then, all the buckets overlapping it are identified from the corresponding hash tables. Finally, the contents of these

buckets are examined by the similarity measure between the incoming message and each historical social message in them. Since the similarity between messages is measured using a KL-based distance function, it is nontrivial to find out whether a search space overlaps an event hash region or a bucket pointed at by a hash directory. Next, we will deduce the candidate regions based on the KL-based distance.

Lemma 1 *Given a query $\mathcal{Q}$ and a message $\mathcal{D} \in \mathcal{I}$, where $\mathcal{I}$ is the hash table of the event containing $\mathcal{D}$, a similarity threshold ϵ, and a weight parameter ω, let $c = \frac{2\epsilon}{1-\omega}$, and $\mathcal{D}_{\text{mini}}$ and $\mathcal{D}_{\text{maxi}}$ be the minimal and maximal values of the messages in $\mathcal{I}$ over topic i, respectively. If $\|(\mathcal{D}(i) - \mathcal{Q}(i))\| \geq \frac{c}{\min(\|\log \mathcal{Q}_i - \log \mathcal{D}_{\text{maxi}}\|, \|\log \mathcal{Q}_i - \log \mathcal{D}_{\text{mini}}\|)}$, $\mathcal{Q}$ and $\mathcal{D}$ are unmatched under the constraint of $\mathcal{Q}(i) \notin [\mathcal{D}_{\text{mini}}, \mathcal{D}_{\text{maxi}}]$.*

Proof By Eqs. 9 and 10, if $\mathcal{Q}$ and $\mathcal{D}$ are matched, the following condition will hold.

$$\begin{aligned} 2D_{\text{LTT}}(\mathcal{D}, \mathcal{Q}) &= \sum_i (\mathcal{D}(i) \log \frac{\mathcal{D}(i)}{\mathcal{Q}(i)} + \mathcal{Q}(i) \log \frac{\mathcal{Q}(i)}{\mathcal{D}(i)}) \\ &= \sum_i (\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i)) \leq c \end{aligned} \tag{20}$$

We will check the space holding the following condition:

$$\forall (\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i)) \leq c \tag{21}$$

Since $(\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i)) \geq 0$, then

$$\|(\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i))\| \leq c \tag{22}$$

Since $\mathcal{Q}(i) \notin [\mathcal{D}_{\text{mini}}, \mathcal{D}_{\text{maxi}}]$, then

$$\begin{aligned} \|(\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i))\| &\geq \|(\mathcal{D}(i) - \mathcal{Q}(i))\| \\ &\times \min(\|\log \mathcal{Q}_i - \log \mathcal{D}_{\text{maxi}}\|, \|\log \mathcal{Q}_i - \log \mathcal{D}_{\text{mini}}\|) \end{aligned} \tag{23}$$

Thus, combining inequalities 22 and 23, we have

$$|(\mathcal{D}(i) - \mathcal{Q}(i))\| \leq \frac{c}{\min(\|\log \mathcal{Q}_i - \log \mathcal{D}_{\text{maxi}}\|, \|\log \mathcal{Q}_i - \log \mathcal{D}_{\text{mini}}\|)} \tag{24}$$

□

Thus, we conclude that $\mathcal{Q}$ and $\mathcal{D}$ are unmatched under the constraint of $\mathcal{Q}(i)$ if the inequality 24 holds.

We check the social messages that meet the conditions of inequality 24 and the constraint on $\mathcal{Q}(i)$. Any bucket that conflicts with these two conditions can be safely pruned without false dismissal. Based on the inequality 24 and the constraint on $\mathcal{Q}(i)$, we obtain our query range for finding the matches of a given message, so the similarity join is performed efficiently. The join process starts with computing the initial hash addresses of the historical events and buckets that overlap the search space. Suppose that we map $\frac{c}{\min(\|\log \mathcal{Q}_i - \log \mathcal{D}_{\text{maxi}}\|, \|\log \mathcal{Q}_i - \log \mathcal{D}_{\text{mini}}\|)}$ into a binary hash address h_c, $\mathcal{Q}(i)$ into h_q. The query range of $\mathcal{D}(i)$ is $h_q \pm h_c$. The same operation is performed over each of the topics in the global codebook to get all the clusters from the sorted array, and further, all the buckets containing potential matched messages in the corresponding hash tables.

5.5 Query optimization over VDEH

Using our VDEH scheme, we can reduce the number of distance computations during social message set similarity join based on the content difference between them. However, in our model, identifying the content similarity of messages is not the whole story of social media similarity measure. Further improvement can be done to reduce the cost of social message set similarity join by embedding both content and link-based filtering. In this section, we will propose two alternative pruning strategies that integrate the lower-bounding measures of content difference and link difference. Next, we first present the lower-bounding measures of our social media measure, and then go to the strategy of using them for similarity pruning.

As introduced in Lemma 1, the KL-based distance of two social messages over a single topic dimension is always a positive value. Accordingly, the social content difference between two messages in a topic subspace lower-bounds the true D_{LTT} distance between them in the whole topic space. Usually, the probability densities of a social message over different topics vary to large extent. We consider a topic with highest probability density in a social message as its dominant topic. Since the difference between two messages is mainly decided by their dominant topics, it is reasonable to choose a dominant topic as the lower-bound measure of the true D_{LTT} distance considering both the high filtering power and low filtering cost. Given two social messages $\mathcal{D}$ and $\mathcal{Q}$, let i be the selected dominant topic number of $\mathcal{D}$ or $\mathcal{Q}$, we define a lower-bound measure of them as below:

$$\text{DLB}_{\text{LTT}} = \frac{1}{2}(\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i)) \tag{25}$$

We now define a lower-bound, *user histogram difference* (HD), for our link similarity between social messages. The HD measure is defined by relaxing the time constraint of the conversation links. Given two links D and Q, let L_D and L_Q be the lengths of D and Q, respectively, the HD measure is obtained by first converting each link into a user histogram, which counts the number of each user's occurrence in a conversation. We represent the user histogram of link D as a vector, $V_D =< vd_1, \ldots, vd_n >$, where vd_i denotes the user occurrence frequency in this link. Likewise, the user histogram of link Q is constructed as a vector, $V_Q =< vq_1, \ldots, vq_n >$. Then, we define the HD lower-

bound measure as follows:

$$HD_{wl} = 1 - \frac{\sum \min(vd_i, vq_i)}{\sum \max(vd_i, vq_i)} \quad (26)$$

where $\min(vd_i, vq_i)$ is to get the smaller value of vd_i and vq_i, and $\max(vd_i, vq_i)$ returns the bigger value of vd_i and vq_i. We integrate $\mathrm{DLB}_{\mathrm{LTT}}$ with HD_{wl} and formulate a lower-bounding measure, DLB_{GD}, for our social dissimilarity D_G.

$$\mathrm{DLB}_{GD} = (1-\omega)\mathrm{DLB}_{\mathrm{LTT}} + \omega HD_{wl} \quad (27)$$

The integrated lower-bound measure permits the candidate filtering to be operated over two types of similarities in the overall social media similarity. Next, we will prove that the DLB_{GD} indeed lower-bounds the D_G distance.

Theorem 1 *Given any two messages $\mathcal{D}$ and $\mathcal{Q}$ with social links D and Q, respectively, the following inequality holds: $DLB_{GD} \leq D_G$*

Proof We will prove

$$\mathrm{DLB}_{\mathrm{LTT}}(\mathcal{D}, \mathcal{Q}) \leq D_{\mathrm{LTT}}(\mathcal{D}, \mathcal{Q}) \quad (28)$$

$$HD_{wl}(D, Q) \leq D_{wl}(D, Q) \quad (29)$$

As in Lemma 1, for any pair of $\mathcal{D}_i$ and $\mathcal{Q}_i$, we have the inequality $(\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i)) \geq 0$. Then we have $(\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i)) \leq \sum_i (\mathcal{D}(i) - \mathcal{Q}(i))(\log \mathcal{D}(i) - \log \mathcal{Q}(i))$. Thus, the inequality 28 holds.

Now, we consider the measure HD_{wl}. Suppose that D_m and Q_n are two sets consisting of the user ids of D and those of Q, respectively. We have $\sum \max(vd_i, vq_i) = |D_m \bigcup Q_n|$. Meanwhile, $\sum \min(vd_i, vq_i)$ is based on the user id comparison between two links by representing each link as a set of user ids. In this case, the temporal order of users in a conversation is ignored. Thus, any match in the LCS measure between two links is definitely a match between their sets. Thus, the LCS is upper bounded by $\sum \min(vd_i, vq_i)$. Accordingly, we have

$$\frac{\sum \min(vd_i, vq_i)}{\sum \max(vd_i, vq_i)} \geq \frac{\mathrm{LCS}(D_m, Q_n)}{|D_m \bigcup Q_n|}$$

We further obtain the following inequality:

$$\frac{\sum \min(vd_i, vq_i)}{\sum \max(vd_i, vq_i)} \geq \frac{\mathrm{LCS}(D_m, Q_n)}{|D_m \bigcup Q_n|} \geq \mathcal{T}\frac{\mathrm{LCS}(D_m, Q_n)}{|D_m \bigcup Q_n|}$$

Thus, the inequality 29 holds. Combining inequalities 28 and 29, we conclude: $\mathrm{DLB}_{GD} \leq D_G$. □

To avoid operating long sparse user histograms, we only consider the user ids in the compared two conversations. As such, the complexity of HD_{wl} is reduced to $O(m+n)$ compared with the high cost of $O(m*n)$ of the D_{wl}. The whole event detection algorithm is presented in Fig. 7.

Procedure ContinousEventDetection($\mathcal{M}$, ϵ).
input: $\mathcal{M}$ - a social stream,
ϵ - social message similarity threshold
output: S_E - a set of social events
1. $S_E \leftarrow \emptyset$
2. Let S_{Dt} be a set of messages in timeslot t, n be the total number of timeslots.
3. Create a hash tree $\mathcal{I}$.
4. **for** each message set S_{Dt}, $t \in [1, n]$
5. **for** each message $\mathcal{D}_i$ in S_{Dt}
6. MessageInsertion($\mathcal{I}$, $\mathcal{D}_i$, ϵ)
7. **if** $t > 1$
8. WriteMSGsInTimeSlot($t-1$)
9. DeleteMSGsInTimeSlot($t-1$)
10. $S_E \leftarrow$FindAllFinishedClusters($\mathcal{I}$)
11. DeleteFinishedClusters($\mathcal{I}$)
/*process the clusters of the last timeslot*/
12. WriteMSGsInTimeSlot(n)
13. DeleteMSGsInTimeSlot(n)
14. $S_E \leftarrow$AllClustersInVDEH($\mathcal{I}$)
15. return S_E

Fig. 7 Continuous event detection over social streams

6 Experimental evaluation

In this section, we report our experimental results to demonstrate the high effectiveness and efficiency of our proposed approach for online social event detection over tweet streams.

6.1 Experimental setup

For experimental evaluation, we use data collected from Twitter, a popular microblogging service, during two severe disasters, Cyclone ULUI and flooding, in Queensland, Australia from 2010 to 2011. Among the two streams, the Cyclone ULUI data consist of 53.4M stream which captures all tweets within 1,000 km of Mackay from Mar 19, 2010 to Mar 22, 2010. During this three days period, the Cyclone ULUI passed over the Whitsunday region of the Queensland coast on 21 March. The Queensland flooding data were collected from Dec 14, 2010 to Jan 13, 2011 and include 1.16G tweets captured from Queensland within one month. We extract the keywords of tweets, remove the stop words, and stem the rest of keywords. Different from the existing work that only considers the location information based on city names [6], we extract more diverse location information including suburb name, postcode etc, as in Sect. 3 to avoid the sparsity problem of location at city level, and map each location to its latitude and longitude. We obtain the time stamps and conversation links among tweets. The four types of data attributes are used for our event detection task.

We use the whole Cyclone ULUI data set consisting of 72 h messages and a subset of Queensland flood data including the messages posted within 120 h from Jan 7, 2010 to Jan 11, 2011 for effectiveness evaluation. The first part of 36 h Cyclone ULUI data and that of 90 h Queensland flood data are selected for parameter tuning, and the rests of two datasets are used for the effectiveness comparison.

The ground truth is labeled based on inter-subject agreement on the relevance judgments. Three assessors participated in the user study with the direction of our social event definition. We only evaluate three-labeled crisis events in effectiveness evaluation, and many more events outside these are not evaluated. The sizes of ground truths labeled for three events: *Cyclone ULUI*, *Queensland flood–South East*, and *Queensland flood–Darling Downs*, are 506, 34495, and 1612, respectively. The following event detection approaches, including one state-of-the-art topic detection OLDA [2] and three proposed LTT-based alternatives, are used in the experiments.

- **OLDA** is the online LDA-based event detection [2].
- **LTT+Link** is our proposed model that integrates the content and link similarity for the social media similarity.
- **LTT** is our proposed model that only applies the content similarity to the matching.
- **LTT-L** is our proposed alternative that removes location information from our LTT model.

Three proposed LTT-based alternatives are compared to show the importance of locations and social links. The rest of event detection approaches mentioned in Sect. 2 is not used for performance comparison, as they were proposed for different specific applications and targeting data other than tweet streams, not extendible to our application.

6.2 Evaluation methodology

We evaluate the effectiveness of our system in terms of the probabilities of *missed detection* and *false alarm* errors (P_{Miss} and P_{Fa}). These two metrics have been widely used for the topic detection and tracking tasks [7]. A target event element can be correctly detected as an element of its composite event, or missed as a missed detection. A nontarget trial that is falsely detected is called a false alarm. A system with high effectiveness performance should have a good balance of P_{Miss} and P_{Fa}, i.e., smaller P_{Miss} and smaller P_{Fa}. The P_{Miss} and P_{Fa} are computed by:

$$P_{\text{Miss}} = \frac{\text{number of missed detections}}{\text{number of targets}} \tag{30}$$

$$P_{\text{Fa}} = \frac{\text{number of false alarms}}{\text{number of nontargets}} \tag{31}$$

Our effectiveness evaluation includes two parts: (1) parameter turning, (2) effectiveness comparison with existing detection approach. Parameter turning part tests the effect of four parameters, T, ϵ, τ, and ω, on the effectiveness of social event detection to get their optimal values by varying their values in the experiments. We aim to obtain an optimal parameter set to get the optimal effectiveness performance of the whole system. The effectiveness comparison part evaluates the superiority of our approach LTT+link over other competitors by reporting the detection results using four approaches, LTT+link, LTT, LTT-L, and OLDA.

We evaluate the efficiency of our proposed approach in terms of the overall time cost of event detection over tweet streams using our VDEH index. The social streams of 1.304G tweets are used for the efficiency test. Experiments are conducted on Windows XP platform with Intel Core 2 CPU (2.4 GHz) and 2 GB RAM.

6.3 Evaluation on effectiveness

We first test the effect of parameters on the effectiveness of social event detection using two data streams. Then, we compare our approach with the state-of-the-art OLDA-based approach in social event monitoring.

6.3.1 Effect of T

We evaluate the effect of topic number, T, on the probabilities of missed detections and false alarms in social event monitoring over two real tweet streams, the Cyclone ULUI and Queensland flood. In this test, we vary the T from 5 to 20. For each T, we change the ϵ, τ, and ω to obtain the best effectiveness at each T. Figure 8a, b shows the changes on the probabilities of missed detections and false alarms of three critical social events over two streams.

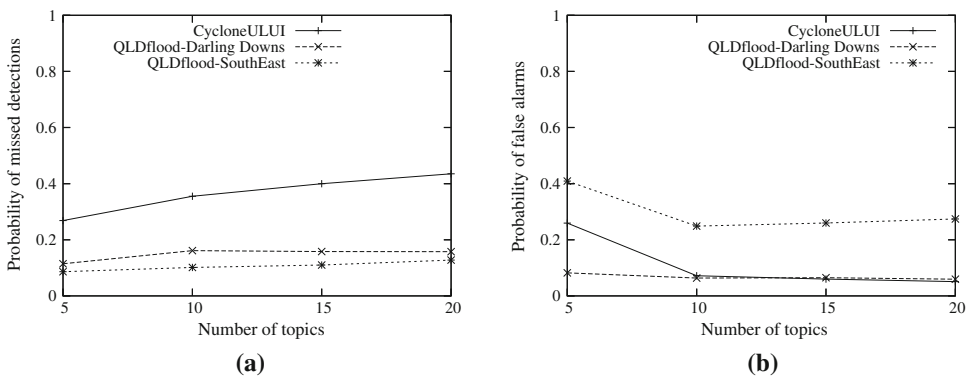

Fig. 8 Effect of T. **a** P_{Miss}. **b** P_{Fa}

As we can see, with the increasing of T, the probability of missed detections increases gradually and that of false alarms drops greatly from 5 to 10, and keeps steady with the further increasing of T. This is caused by two reasons. On the one hand, when we fix T to 5, due to the extremely small number of topics given for a time slot, a large volume of social data are used to extract topics to a very coarse level. Accordingly, the discrimination among messages is not distinguished, introducing more false alarms. On the other hand, with the increasing of topic number after 10, several relevant topics may be split to fit the big given topic number, producing a large number of redundant topics in a time slot. At the point of T=10, distinguished topics are extracted and less redundant topics are found via our LTT model from messages, leading to a good balance of effectiveness and efficiency on the detection. Thus, we set 10 as the default value of T.

6.3.2 Effect of ϵ

We evaluate the effect of message similarity threshold, ϵ, on the effectiveness of our approach by applying our LTT and link model. In this test, the ϵ is varied from 0.1 to 0.25, and the default T is applied. For each ϵ, we obtain the best effectiveness by turning the parameters τ and ω. Figure 9a, b shows the missed detection and false alarm probability change trends of our approach, respectively. Clearly, with the increasing of ϵ, the probability of missed detections decreases and that of false alarms increases for both social streams. Meanwhile, with the further increasing of ϵ after 0.15, the change speed of P_{Miss} reduces while the P_{Fa} increases quickly due to two reasons. For one thing, more candidates are allowed to get into a social cluster to their corresponding event due to the relaxation of ϵ. For another, a bigger ϵ introduces more false alarms. A good balance is obtained when ϵ is set to 0.15.

6.3.3 Effect of τ

We test the effect of the uncertain time stamp threshold, τ, by varying its value from 1 to 30. For each τ, the topic number T and the similarity threshold ϵ are set to their default values, and the ω is changed to get the best effectiveness with this τ. Figure 10a, b reports the probabilities of missed detections and false alarms, respectively.

As we can see, the probability of missed detections drops with the increase of τ, while that of false alarms increases. This is caused by two reasons. On the one hand, a relaxable time constraint helps detect more relevant social messages. On the other hand, a bigger time threshold has a weaker ability of rejecting irrelevant messages, introducing more false alarms. Meanwhile, while the probability of false alarms increases with the increasing of τ, that of missed detections drops slightly after τ is increased to 20. For the good balance of the detection system, we set the default value of τ to 20.

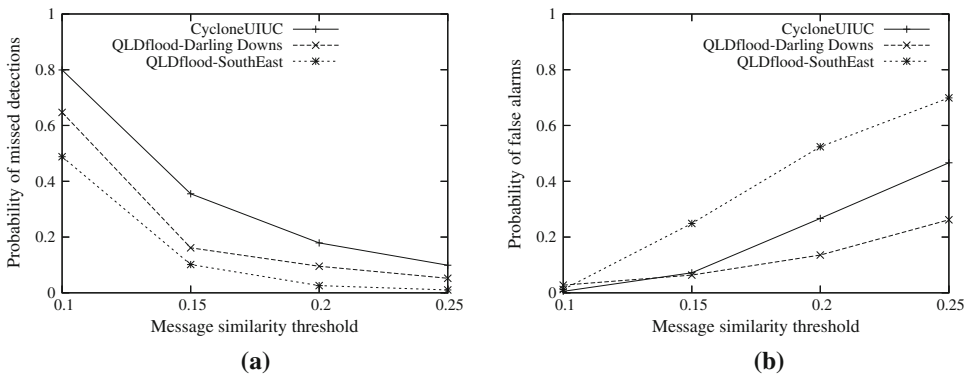

Fig. 9 Effect of ϵ. **a** P_{Miss}. **b** P_{Fa}

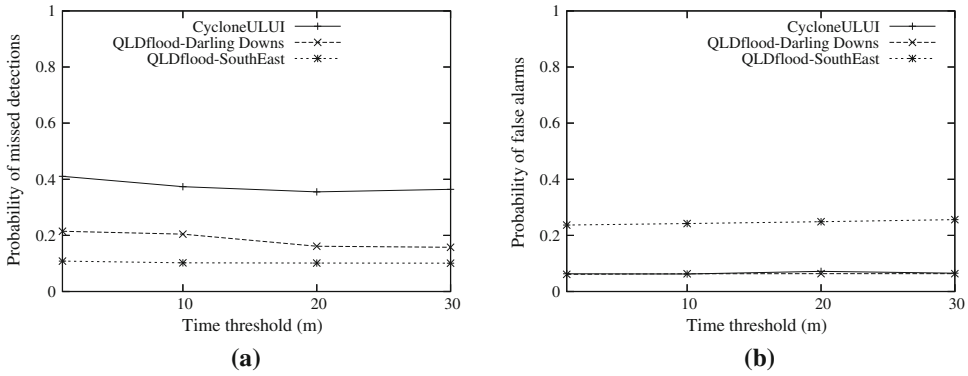

Fig. 10 Effect of τ. **a** P_{Miss}. **b** P_{Fa}

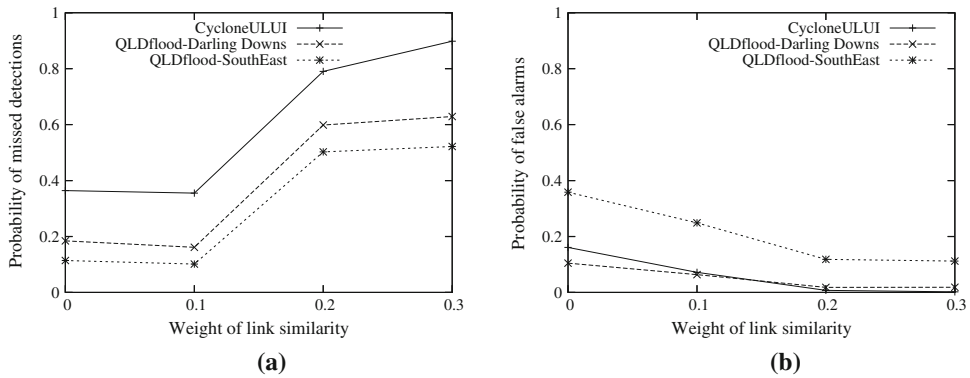

Fig. 11 Effect of ω. **a** P_{Miss}. **b** P_{Fa}

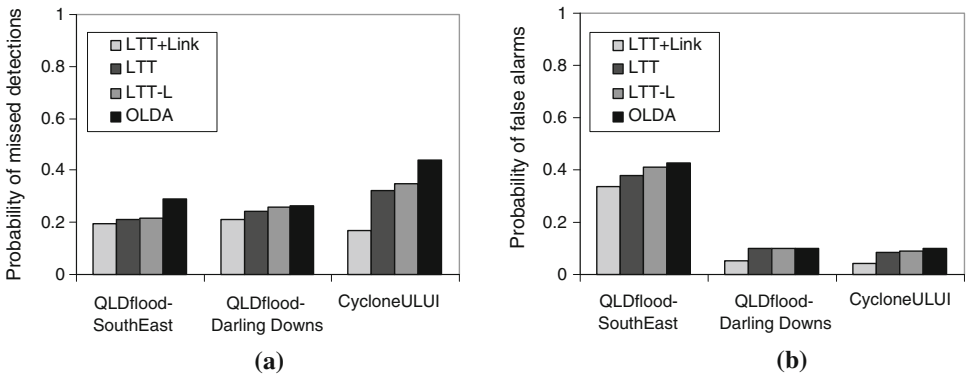

Fig. 12 Effectiveness comparison with OLDA. **a** P_{Miss}. **b** P_{Fa}

6.3.4 Effect of ω

We evaluate the effect of the two components in social similarity by varying the parameter ω from 0 to 0.3 and fixing the parameters, T, ϵ, and τ to their default values. Figure 11a, b shows the probability of missed detections and that of false alarms under different ω.

Clearly, both the missed detections and false alarms are reduced with the change of ω from 0 to 0.1, due to the link similarity compensation in social media matching. Meanwhile, with the further increasing of ω, the missed detections increase quickly, while the false alarms decrease slightly for the investigated events. This is caused by two reasons. For one thing, the link similarity enhances the ability of rejecting false alarms in the detection. For another, the extreme increase of ω weakens the effect of content similarity, thus more relevant messages are missed. Therefore, content similarity plays more important role in social media similarity, while link similarity effectively compensates the social information. A good balance of the weights can be obtained by setting ω to 0.1.

6.3.5 Effectiveness comparison

We perform experiments to compare the effectiveness of social data modeling and similarity matching for four approaches, including three proposed alternatives, LTT+Link, LTT, and LTT-L, and the existing competitor for topic detection OLDA, by performing event detection on twitter streams. For the OLDA, we report its best accuracy of social event detection. We use two real tweet streams for the event detection. Figure 12a, b shows the effectiveness comparison of four approaches over two streams in terms of P_{Miss} and P_{Fa}.

Clearly, the LTT+Link model produces the least missed detections and false alarms among three LTT-based social event monitoring approaches, followed by the LTT model. This is because the LTT+Link model fully exploits the information from the time, location, text content, and social links, which finds more relevant and rejects more irrelevant event elements. The LTT-L produces the worst detection results among three LTT-based approaches due to lacking of the location information, which makes the model less discriminative. Compared with the OLDA, our LTT+Link model obtains much higher performance because of the two reasons. For one thing, compared with the OLDA model that only captures the text content of each message, our LTT model fuses the time, location, and text content of a social message into an integrated representation, which captures the information of the message over multiple attributes and well handles the uncertainty in these attributes. Moreover, we exploit social links to capture the connections of messages, which helps reject false alarms effectively. For another, the performance gap of two approaches on the QLD flood is smaller compared to that on the Cyclone ULUI. This is because the Qld flood is a more serious disaster, which affected wider region, while

Table 2 The samples of relevant topics in Queensland flood dataset

Topics	Word distribution									
$T_{(91,3)}$	toowoomba	peopl	man	todai	hope	flash	head	food	thought	understand
	0.0125	0.0125	0.0102	0.0102	0.0080	0.0080	0.0080	0.0069	0.0069	0.0069
$T_{(91,7)}$	flood	brisban	qldflood	warn	river	lockyer	bremer	warril	wivenho	qld
	0.0380	0.0175	0.0155	0.0144	0.0144	0.0134	0.0103	0.0103	0.0103	0.0093
$T_{(92,3)}$	new	toowoomba	hope	flash	peopl	man	prai	yesterdai	todai	water
	0.0176	0.0151	0.0143	0.0126	0.0118	0.0109	0.0109	0.0092	0.0093	0.0084
$T_{(92,7)}$	flood	brisban	qldflood	qld	river	lockyer	warn	abc	hit	wivenho
	0.0498	0.0204	0.0164	0.0164	0.0147	0.0139	0.0139	0.0115	0.0115	0.0107
$T_{(93,3)}$	flood	new	qldflood	toowoomba	peopl	safe	todai	stai	miss	water
	0.0237	0.0197	0.0182	0.0172	0.0147	0.0142	0.0142	0.0127	0.0127	0.0116
$T_{(93,7)}$	flood	brisban	qldflood	australia	citi	suburb	qld	abc	hit	toowoomba
	0.0658	0.0404	0.0182	0.0166	0.0161	0.0151	0.0140	0.0140	0.0125	0.0120
$T_{(94,3)}$	flood	peopl	toowoomba	qldflood	stai	safe	queensland	new	qld	miss
	0.0552	0.0210	0.0207	0.0207	0.0182	0.0182	0.0167	0.0143	0.0122	0.0122
$T_{(94,7)}$	flood	brisban	qldflood	qld	river	thebigwet	suburb	list	australia	expect
	0.0828	0.0540	0.0372	0.0282	0.0258	0.0249	0.0204	0.0180	0.0168	0.0156
$T_{(95,3)}$	flood	toowoomba	peopl	stai	qldflood	safe	miss	dead	live	qld
	0.0648	0.0286	0.0274	0.0229	0.0225	0.0197	0.0187	0.0167	0.0147	0.0139
$T_{(95,6)}$	alert	storm	hit	emerg	brisban	flood	flash	hour	move	sever
	0.0753	0.0486	0.0460	0.0443	0.0443	0.0436	0.0423	0.0421	0.0421	0.0416
$T_{(95,7)}$	flood	qldflood	brisban	thebigwet	qld	river	warn	list	expect	citi
	0.0823	0.0606	0.0542	0.0346	0.0344	0.0325	0.0198	0.0166	0.0152	0.0141
$T_{(96,3)}$	flood	toowoomba	qldflood	peopl	stai	safe	dead	miss	live	water
	0.0607	0.0311	0.0264	0.0259	0.0215	0.0203	0.0191	0.0188	0.0164	0.0162
$T_{(96,6)}$	alert	flood	brisban	hour	hit	emerg	storm	move	start	flash
	0.0898	0.0646	0.0609	0.0597	0.0595	0.0592	0.0590	0.0538	0.0508	0.0498
$T_{(96,7)}$	qldflood	flood	brisban	thebigwet	river	warn	expect	citi	qld	council
	0.0738	0.0737	0.0504	0.0334	0.0313	0.0225	0.0205	0.0191	0.0191	0.0156

the Cyclone ULUI only affected the cities along the coast. Accordingly, there were more tweets on the flood, which weakens the performance gap between two techniques. To visualize these detected events, we further study the word distributions of relevant topics discovered in the investigated time slots and the topic distribution of each event. The sampled results over the first 6 h data are reported in Tables 2, 3, 4, and 5. Here, $T_{(i,j)}$ denotes the jth topic discovered in the ith hour stream. For the Queeensland flood dataset, we found 157 topics relevant to the QLDflood–SouthEast (QF–SE) and QLDflood–Darling Downs (QF–DD) events from the 30 h media stream (91th–120th time slots), and 14 relevant ones are found in the first 6 h data as shown in Fig. 2. The sampled topic distributions of these two events over the first 6 h data are shown in Table 4. Note that, for each event QF-SE/QF-DD, the sum of topic distribution values over all 157 relevant topics is 1. For the CycloneULUI dataset, we found 24 topics relevant to the CycloneULUI (ULUI) event from the 36 h tweet stream (37th–72th time slots), and 8 of them are from the first 6 h data. The sampled topic distribution of the event CyconeULUI is shown in Table 5. Likewise, the sum of topic distribution values of CycloneULUI event over all 24 relevant topics is 1.

6.4 Evaluation on efficiency

We evaluate the efficiency of our approach by first testing the effect of our VDEH scheme, and then compare our approach with the OLDA for the social event detection over tweet streams. Since the DLB-based optimization is actually the query processing over buckets in VDEH, we take the DLB filtering with the VDEH as a whole for efficiency evaluation.

6.4.1 Effect of hashing index

We evaluate the effect of our VDEH scheme by conducting experiments over 5 weeks tweet streams. We compare the VDEH scheme which includes the index structure and

Table 3 The samples of relevant topics in Cyclone ULUI dataset

Topics	Word distribution									
$T_{(37,3)}$	ului	cyclon	friend	absolut	build	video	power	lost	hit	hous
	0.0686	0.0403	0.0144	0.012	0.012	0.012	0.012	0.012	0.01	0.01
$T_{(38,3)}$	ului	friend	hous	googl	peopl	tree	success	build	interior	door
	0.0589	0.0274	0.0229	0.0184	0.0184	0.0139	0.0094	0.0094	0.0094	0.0094
$T_{(40,2)}$	cyclon	call	sleep	woke	coast	control	van	cook	tengo	pressur
	0.0122	0.0092	0.0063	0.0063	0.0063	0.0063	0.0063	0.0063	0.0063	0.0063
$T_{(40,3)}$	ului	new	real	time	hope	interview	hand	march	estat	dai
	0.0188	0.0109	0.0109	0.0109	0.0082	0.0082	0.0082	0.0082	0.0082	0.0082
$T_{(41,2)}$	cyclon	light	man	coast	market	tree	love	design	fine	todai
	0.0247	0.0155	0.0125	0.0125	0.0125	0.0094	0.0094	0.0094	0.0064	0.0064
$T_{(41,3)}$	ului	feel	time	rememb	estat	dai	hope	real	new	night
	0.0198	0.0173	0.0173	0.01	0.01	0.01	0.01	0.01	0.01	0.01
$T_{(42,2)}$	cyclon	ului	market	tree	coast	light	man	live	love	hous
	0.0508	0.0234	0.0171	0.0129	0.0108	0.0108	0.0108	0.0087	0.0087	0.0087
$T_{(42,3)}$	time	feel	dai	peopl	new	week	hour	ului	danc	fat
	0.0262	0.0154	0.0154	0.0132	0.0089	0.0089	0.0089	0.0067	0.0067	0.0067

Table 4 The sampled topic distribution of events in Queensland flood dataset

Topics	QF–SE	QF–DD	Topics	QF–SE	QF–DD
$T_{(91,3)}$	0	0.0028	$T_{(91,7)}$	0.0011	0.0007
$T_{(92,3)}$	0.0006	0.0073	$T_{(92,7)}$	0.0007	0.0058
$T_{(93,3)}$	0.0018	0.0108	$T_{(93,7)}$	0.0029	0.0102
$T_{(94,3)}$	0.0035	0.0384	$T_{(94,7)}$	0.0055	0.018
$T_{(95,3)}$	0.0055	0.0606	$T_{(95,6)}$	0.0122	0.0049
$T_{(95,7)}$	0.0061	0.0159	$T_{(96,3)}$	0.0075	0.0512
$T_{(96,6)}$	0.0076	0.003	$T_{(96,7)}$	0.013	0.0091

Table 5 The sampled topic distribution of event in Cyclone ULUI dataset

Topics	ULUI	Topics	ULUI	Topics	ULUI
$T_{(37,3)}$	0.1132	$T_{(38,3)}$	0.0015	$T_{(40,2)}$	0.0116
$T_{(40,3)}$	0.0276	$T_{(41,2)}$	0.0218	$T_{(41,3)}$	0.0087
$T_{(42,2)}$	0.1393	$T_{(42,3)}$	0.0218		

lower-bound–based filtering and the social event monitoring with the lower-bound filtering only (DLB). The evaluation on two methods is performed by varying the lengths of tweet streams and reporting the overall time cost of the detection for each of them.

Figure 13a shows the time cost changes of continuous detection with two approaches. Clearly, with the support of our VDEH, the event detection cost over a social stream is reduced significantly due to the strong filtering over the index. Using our VDEH scheme, messages in many irrelevant buckets are removed from candidate set directly without any computation. Meanwhile, our VDEH scheme embeds the lower-bound–based filtering for each message in candidate sets, which further reduces the time cost of social event detection.

6.4.2 Time cost comparison

We compare our social event detection approach with the state-of-the-art technique in terms of the overall time cost for the efficiency evaluation. We test the time cost of detection over 1 to 5 weeks time periods.

Figure 13b compares our approach with the OLDA-based detection (OLDA+DLB) in terms of time cost used in social media detection. Here, OLDA+DLB applies our DLB lower-bounding technique to the OLDA to improve the efficiency of detection. Obviously, our approach is much faster than the improved OLDA-based approach, because of the strong filtering over VDEH index. With our VDEH scheme, the messages can be filtered out using hash directories and lower-bounding measure in social message similarity join over adjacent time slots. For OLDA+DLB, although it only operates on one attribute, the social text content, it suffers from high computation cost in similarity join processing among messages due to lacking of the efficient index scheme. With the increasing of social stream length, our approach obtains increasing improvement on the efficiency performance. This has further demonstrated the superiority of our VDEH scheme over the existing continuous detection approach.

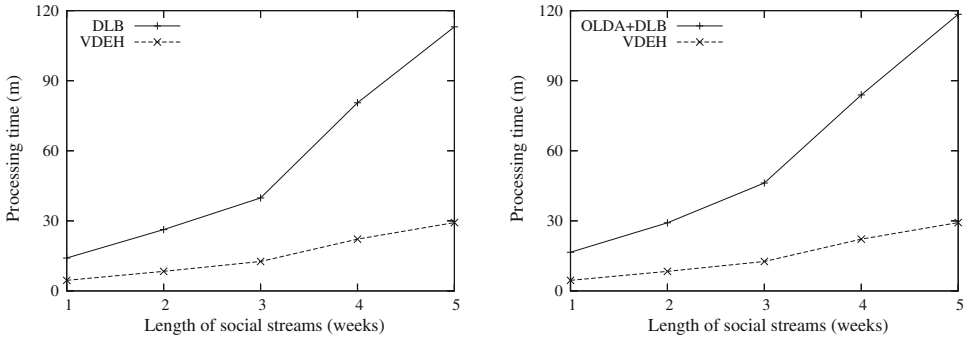

Fig. 13 Efficiency evaluation. **a** Effect of VDEH. **b** Comparison

7 Conclusions

In this paper, we study the problem of online social event monitoring over tweet streams for real applications like crisis management and decision making. We first propose a novel location-time constrained topic model that fuses social content, location and time information for tweet representation. The link of a message is modeled using a string of the user *ids* in a conversation. Then, the similarity of messages is captured by a complementary distance which considers the difference between two messages over four attributes including the content, location, time, and link. Finally, we propose a variable dimensional extendible hash scheme and the query optimization over this index for fast social event monitoring. We have conducted extensive experiments over long tweet streams during two crisis happened in Australia in recent years. The experimental results have verified the superiority of our proposed approach in terms of effectiveness and efficiency.

Acknowledgments Funding for this work was supported by the Hong Kong RGC GRF Project No. 611411, National Grand Fundamental Research 973 Program of China under Grant 2012-CB316200, Huawei Noah's Ark Lab under Project HWLB06-15C03212/13PN, HP IRP Project 2011, and Microsoft Research Asia Gift Grant.

References

1. Allan, J., Papka, R., Lavrenko, V.: On-line new event detection and tracking. In: SIGIR, pp. 37–45 (1998)
2. AlSumait, L., Barbará, D., Domeniconi, C.: On-line lda: adaptive topic models for mining text streams with applications to topic detection and tracking. In: ICDM, pp. 3–12 (2008)
3. Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The r*-tree: an efficient and robust access method for points and rectangles. In: SIGMOD, pp. 322–331 (1990)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
5. Chang, Y.-L., Chien, J.-T.: Latent dirichlet learning for document summarization. In: ICASSP, pp. 1689–1692 (2009)
6. Cheng, Z., Caverlee, J., Lee, K.: You are where you tweet: a content-based approach to geo-locating twitter users. In: CIKM, pp. 759–768 (2010)
7. Fiscus, J.G., Doddington, G.R.: Topic detection and tracking evaluation overview. In: Allan, J. (ed.) Topic detection and Tracking, pp. 17–31. Kluwer Academic Publishers, Norwell, USA (2002)
8. Fung, G.P.C., Yu, J.X., Yu, P.S., Lu, H.: Parameter free bursty events detection in text streams. In: VLDB, pp. 181–192 (2005)
9. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: SIGMOD, pp. 47–57 (1984)
10. http://en.wikipedia.org/wiki/kullback
11. http://en.wikipedia.org/wiki/twitter
12. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR, pp. 50–57 (1999)
13. Jagadish, H.V., Ooi, B.C., Tan, K.-L., Yu, C., Zhang, R.: iDistance: an adaptive b+-tree based indexing method for nearest neighbor search. TODS **30**(2), 364–397 (2005)
14. Lin, C.X., Mei, Q., Han, J., Jiang, Y., Danilevsky, M.: The joint inference of topic diffusion and evolution in social communities. In: ICDM, pp. 378–387 (2011)
15. Lin, J., Snow, R., Morgan, W.: Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In: KDD, pp. 422–429 (2011)
16. Lin, S., Özsu, M.T., Oria, V., Ng, R.T.: An extendible hash for multi-precision similarity querying of image databases. In: VLDB, pp. 221–230 (2001)
17. Liu, S., Zhou, M.X., Pan, S., Qian, W., Cai, W., Lian, X.: Interactive, topic-based visual text summarization and analysis. In: CIKM, pp. 543–552 (2009)
18. Rattenbury, T., Good, N., Naaman, M.: Towards automatic extraction of event and place semantics from flickr tags. In: SIGIR, pp. 103–110 (2007)
19. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: WWW, pp. 851–860 (2010)
20. Sizov, S.: Geofolk: latent spatial semantics in web 2.0 social media. In: WSDM, pp. 281–290 (2010)
21. Wan, X., Milios, E., Kalyaniwalla, N., Janssen, J.: Link-based event detection in email communication networks. In: SAC, pp. 1506–1510 (2009)
22. Wang, J., Zhao, Z., Zhou, J., Wang, H., Cui, B., Qi, G.: Recommending flickr groups with social topic model. Inf. Retr. **15**(3–4), 278–295 (2012)
23. Wang, X., McCallum, A.: Topics over time: a non-markov continuous-time model of topical trends. In: KDD, pp. 424–433 (2006)
24. Wang, Y., Sundaram, H., Xie, L.: Social event detection with interaction graph modeling. In: ACM Multimedia, pp. 865–868 (2012)
25. Wei, X., Croft, W.B.: Lda-based document models for ad-hoc retrieval. In: SIGIR, pp. 178–185 (2006)
26. White, R.W., Jose, J.M.: A study of topic similarity measures. In: SIGIR, pp. 520–521 (2004)

27. Yang, Y., Pierce, T., Carbonell, J.G.: A study of retrospective and on-line event detection. In: SIGIR, pp. 28–36 (1998)
28. Yao, J., Cui, B., Huang, Y., Jin, X.: Temporal and social context based burst detection from folksonomies. In: AAAI, pp. 1474–1479 (2010)
29. Yao, J., Cui, B., Xue, Z., Liu, Q.: Provenance-based indexing support in micro-blog platforms. In: ICDE, pp. 558–569 (2012)
30. Yin, H., Cui, B., Li, J., Yao, J., Chen, C.: Challenging the long tail recommendation. PVLDB **5**(9), 896–907 (2012)
31. Yin, H., Cui, B., Lu, H., Huang, Y., Yao, J.: A unified model for stable and temporal topic detection from social media data. In: ICDE, pp. 618–629 (2013)
32. Yin, J., Lampert, A., Cameron, M., Robinson, B., Power, R.: Using social media to enhance emergency situation awareness. IEEE Intell. Syst. **27**(6), 52–59 (2012)
33. Yin, Z., Cao, L., Han, J., Zhai, C., Huang, T.S.: Geographical topic discovery and comparison. In: WWW, pp. 247–256 (2011)
34. Yu, C., Ooi, B.C., Tan, K.-L., Jagadish, H.V.: Indexing the distance: an efficient method to knn processing. In: VLDB, pp. 421–430 (2001)
35. Zhang, K., Zi, J., Wu, L.G.: New event detection based on indexing-tree and named entity. In: SIGIR, pp. 215–222 (2007)
36. Zhao, Q., Mitra, P.: Event detection and visualization for social text streams. In: ICWSM (2007)
37. Zhao, Q., Mitra, P., Chen, B.: Temporal and information flow based event detection from social text streams. In: AAAI, pp. 1501–1506 (2007)
38. Zhou, X., Zhou, X., Chen, L., Bouguettaya, A.: Efficient subsequence matching over large video databases. VLDB J. **21**(4), 489–508 (2012)
39. Zhou, X., Zhou, X., Chen, L., Shu, Y., Bouguettaya, A., Taylor, J.A.: Adaptive subspace symbolization for content-based video detection. IEEE Trans. Knowl. Data Eng. **22**(10), 1372–1387 (2010)
40. Zunjarwad, A., Sundaram, H., Xie, L.: Contextual wisdom: social relations and correlations for multimedia event annotation. In: ACM Multimedia, pp. 615–624 (2007)