Temporal and Social Context based Burst Detection from Folksonomies

Junjie Yao Bin Cui Yuxin Huang Xin Jin
Department of Computer Science
Key Lab of High Confidence Software Technologies (Ministry of Education)
Peking University
{junjie.yao, bin.cui, yuxinhuang, superjinxin}@pku.edu.cn

Abstract

Burst detection is an important topic in temporal stream analysis. Usually, only the textual features are used in burst detection. In the theme extraction from current prevailing social media content, it is necessary to consider not only textual features but also the pervasive collaborative context, e.g., resource lifetime, user activity.

This paper explores novel approaches to combine multiple sources of such indication for better burst extraction from social media content. We systematically investigate the characteristics of collaborative context, i.e., metadata frequency, topic coverage and user attractiveness. First, a robust state-based model is utilized to detect bursts from individual streams. We then propose a learning method to combine these burst pulses. Experiments on a large real dataset demonstrate the remarkable improvements over traditional methods.

1. Introduction

The proliferating social media fever has brought out lots of User Generated Content (UGC), such as blog posts, comments, tags, and tweets. Various types of data, e.g. text, photo, music and video are created, consumed and viewed. UGC is one of the main prevailing online trends (Baeza-Yates 2009).

UGC reflects prior viewpoint from an attendee’s perspective. Social media content is usually event-driven, and becomes an ideal source to reflect the real-world pulse, i.e., popularity of topics and events. Fig 1 presents the frequency change of some representative words from a social tagging website. It is shown that “bigbang” (an American sitcom) exposed two bursts in Jun 2008, 2009. “Android”, a mobile OS from Google first caught eyes in Sep 2008, the release of “Android” G1 phone, and in 2009, there is an increasing attention, because of more “Android” phones and the OS updates.

Recently, there exists an increasing attention to the real-time property of this social web (MacManus 2009). By identifying these events and the associated social media content, we can realize and improve various kinds of search and engagement experience, e.g., what are hot buzz words now, what are users’ sentiments about a company or product and how is a specific topic evolving.

A great deal of influential web applications, including Flikr, Youtube, Twitter and Delicious allow users to label posts with arbitrary keywords, also known as “tags”. Examples include content tag in Youtube and Delicious, geo tag in Flickr and hashtag in Twitter’s tweet. These tags facilitate easy description and annotation and enjoy dramatic increase, now coined as “folksonomy”.

Fig 2 illustrates this tagging interaction. Users, tags and posts are three key components. Over a timeline, users bookmark posts with tags and form a network by connection to others. As a concept/topic layer, social tags link the
user and content together. These social annotations are user-perspective descriptions of the web content, as well as good indicators of users' interests.

Though having high potential, UGC is inherently noisy and varies in quality. Compared with traditional corpus, UGC bars easy extraction of semantic meanings or underlying events. First, it is short in text description. Comments, community QnA, tweets are usually short sentences. Bookmark tags are even merely keywords. Second, it is inherently heterogenous in data types. Besides the textual data, photo, music and pages are also attached and connected. At last, as a collaborative environment, spamming and cheating are unavoidable. In social systems, things are popular, so frequency is not always the best thing to indicate the content quality or other evaluation.

While bringing great challenges, it also exhibits rich associated context not found previously. Specifically, social media content has a wealth of surrounding features, e.g., temporal date, user network and contributed annotations. In this paper, we are interested in how to effectively detect events from social media content, especially “folksonomy” data.

Our work in this paper can be viewed as an extension of burst detection from temporal stream. In the traditional burst event detection tasks, the objective is to detect events from a temporally ordered stream of documents. Though there exist various previous algorithms and seminal work, multiple stream burst detection hasn’t been investigated. To best of our knowledge, the combined extraction of bursts from social media stream has not been discussed yet.

By incorporating the temporal and social context in the burst detection, the quality and novelty of detected bursts can be improved. Not only can we extract events more accurately, but also uncover relations between the detected events and the interaction between content and users. This brings out new opportunities, e.g., burst-aware correlation discovery and temporal related ranking retrieval. Our contributions in this paper are listed as follows:

- **Investigate dynamic characters of social context**: We discuss temporal social characters of social stream and their indications for burst detection.

- **Utilize a robust burst detection model**: We apply a seminl burst detection algorithm, and extend it into the social media stadium.

- **Propose a learning based burst detection framework**: We investigate how to incorporate various indications of bursts into a learning model.

- **Experiment on a large real-world dataset**: We conduct experiments on a large dataset, demonstrating the effectiveness and applicability of our proposed approach.

The rest of this paper is organized as follows. We first discuss the problem definition and data characters in Section 2. Section 3 presents the burst detection model. Empirical result is shown in Section 4. We review related work in Section 5 and finally conclude this paper.

### 2. Temporal and Social Characters

#### 2.1 Preliminaries

We begin with a brief feature definition in folksonomy used in this paper. In tagging systems, a tagging action could be represented as a quar-partite structure $< u, t, p, d >$, where user $u$ bookmarked a post $p$ with tag $t$ at date $d$. There could be multiple tags $t$, associated with a post in a post action.

Various types of data are hosted in the tagging systems and a large amount of tags are created. There’s some recent work, utilizing tags to profile the temporal dynamics of folksonomy and social media content (Dubinko et al. 2006; Rattenbury, Good, and Naaman 2007). In a sequence of non-overlapping time intervals, $(x_0, x_1, \ldots, x_N)$, $freq_i$ is usually selected to identify the bursty tags and further events.

As we discussed in Section 1, this kind of single stream isn’t enough in social media content. There are also interactions among tags, posts and users (Fig 2). Posts have temporal attached information, e.g., it is first bookmarked or has been posted for several times. Users follow others based on friendship or common interests, forming a user community. We utilize user and post information to illustrate tag dynamics.

#### 2.2 Time-aware Post Coverage

A post has its lifetime. In its initial stage, it is fresh and may be attractive. Gradually, pages decay and lose interest. For a tag $t$, in the specific time interval $x_i$, there are totally $n$ posts tagged with $t$ by some users. $n(x_i|t)$ measures the post coverage of $t$ at $x_i$. Usually, $n(x_i|t)$ could be measured by all the posts, ignoring each post’s lifetime and freshness. Simply counting the posts of a tag cannot include posts’ temporal information. Posts of a tag should have different weighting schemes, based on their frequency or freshness.

Here we add the time aspect into the post coverage measurement. We note the original posts firstly posted in time interval $x_i$ as $new_i(t)$. More recently created posts should have a high priority, in contract to older ones with a low score. We utilize an decaying equation to include the previous $m$ intervals of tag $t$ in the following equation:

$$cov_i(t) = \beta \times new_i(t) + (1 - \beta) \times new_{i-1}(t) + \ldots + (1 - \beta)^m \times new_{i-m}(t)$$  

(1)

For all posts tagged by $t$ in interval $i$, we weight them by their first posted date. Observe that the above equation incorporates an exponentially decaying average of posts. This tracks the time changing behavior of a tag through the life span of all tagged posts, parameter $\beta$ is used to retain enough old post information.

By incorporating this time-aware coverage, we identify fresh content from old ones and also maintain old but important last longing posts. The above equation is general, and we could easily modify it to support other time drifting techniques.

#### 2.3 Expertise-based User Attractiveness

Though social media promotes a flat world and collaborative community, users are not same. There’re spamming or
noisy users to hazard the system and there’re also authoritative users who has lots of followers. The user network enables the information flow from the discoverer to followers. (Noll et al. 2009) reported that popular users bookmark frequently and tend to be one of the first users to bookmark a web page.

A tag could be posted by various users. In a specific time interval, if a tag is used by more authoritative users, this tag may exhibit an attractive burst. Because the follower are “waiting” to bookmark this tag later. From the perspective of user, we call this $\text{attr}_t(t)$ as the temporal attractiveness of tag $t$ at $i$. To evaluate tag temporal dynamics from user perspective, we resort to take the user’s authority information into consideration.

As there are also spamming users, simple posting active-ness or fans count won’t guarantee the quality of users. An intuitive choice is resorting to a network authority method. Here we utilize the user’s natural following network to assess user expertise.

One user following another in social media is analogous to one page linking to another on the Web. Both are a form of recommendation. The following relationship earn reputation and then give reputation. More fans of a user, more authority he/she is. More authoritative of his/her fans, more authoritative he/she is. We utilize HITS algorithm (Kleinberg 1999; Noll et al. 2009) to extract the authority of a user $\text{auth}(u)$ based on the user network in the social community.

The attractiveness of a tag $t$ is measured below:

$$\text{attr}_t(t) = \sum_{u=1}^{U} \text{auth}(u|t).$$

For simplification, we let $\text{auth}(u|t) = \text{auth}(u)$. A global user authority is used to measure every tags. A tag personalized user authority is an ongoing work.

### 3. Burst Detection Framework

After the discussion of several features in social burst detection, here we present the proposed burst detection framework. Our motivation in this paper is to combine multiple burst indications to better detect burst events. The proposed burst detection approach follows a learning based ensemble.

Given features extracted as input, we divide the detection task as a two-step approach. We first identify bursts from each temporal feature separately, which copes with the inherent nature of social media well. Here we describe a burst detection method using a seminal Hidden Markov Model. And then we employ a guided learning model to merge these preliminary results from all feature sources.

The reason we choose this two-step approach is that, it is explicitly intuitive and easy to guide the training/evaluation process. The deployment of this model is guaranteed.

#### 3.1 Robust State based burst detection

Given a temporal tag stream, mining burst or anomaly intervals of this tag is an important work in sequential mining or statistical analysis. Inspired by Kleinberg’s seminal work (Kleinberg 2002), we translate this problem as an optimal state learning problem. It profiles a slower base state corresponding to the average rate of appearance of the word, while a burst state corresponds to a faster burst rate.

For a specific tag $t$, assuming there are $N$ time intervals in total, with tag frequency $X = (x_1, x_2, \ldots, x_N)$, we need to find an optimal state sequence $q = (q_1, q_2, \ldots, q_N)$. $q_i$ represents whether or not interval $i$ is in burst.

In a binary state model, two states are used: “stable” $q_0$ and “burst” $q_1$ respectively. When a state model $A$ is in state $q_0$, tags are posted in a slow rate, with gaps $x$ between consecutive tags posted independently according to a density function $f_0(x) = \beta_0 e^{-\beta_0 x}$. This density function follows the common Poison distribution.

In state $q_1$, tags are posted in a faster rate, $f_1(x) = \beta_1 e^{-\beta_1 x}$, where $\beta_1 > \beta_0$. A changes state with probability $p$, remaining in its current state with $1-p$. This state change is independently of previous tag posting actions, thus Markov memoryless.

A sequence $q$ induces a density function $f_q$ over sequences of gaps, which has the form: $f_q(x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N} f_i(x_i)$. Due to the space limit, we omit the proof here. The optimization problem is equivalent to finding a state sequence $q$ that minimizes the following cost function:

$$c(q|x) = b \ln \left(1 - \frac{p}{p}\right) + \sum_{t=1}^{n} -\ln f_i(x_t)$$

where $b$ denotes the number of state transition in $q$, i.e., the number of indices $i$, so $q_i \neq q_{i+1}$.

By minimizing the above cost function, we achieve the goal that both let the state sequence fit well to the tag posting rate and minimize the change cost from one state to another. The dynamic programming algorithm could be used to derive the optimal state sequence which minimizes the overall state cost.

The burst state extraction process is mainly composed of two stages: a forward step to calculate all possible $f_{ij}$, $path_{ij}$, and a backward one to retrieve the optimal state values for each interval, where $f_{ij}$ is the current minimum value of $c(q|x)$ when interval $i$ is in state $j (j(0, 1)$ and $path_{ij}$ records the previous interval$(i-1)$’s state when current state is $j$.

Though (Kleinberg 2002) provided a hierarchical multi-state model, this is complex and computing expensive. To fit our problem, we choose the basic two-state burst model and add an external step to compute the burst degree $\text{conf}_i$, i.e., for a specific interval, how confidence do we have about its burstness?

As the path selection in dynamic programming is backward, we need to know the state selection of interval $i+1$ before we determine the state of interval $i$. Thus, it is reasonable for us to define the following intuitive metric:

$$\text{conf}_i = \begin{cases} \frac{c_i}{c_0 + c_1}, & c_0 > c_1 \\ 0, & \text{otherwise} \end{cases}$$

where $c_i$ is the cost value of interval $i+1$ when interval $i$ is in state $j$. Our consideration is that in the backward procedure, when state $i+1$ is determined, state $i$ is to be determined according to the difference of $c_0$ and $c_1$. Therefore, we normalize the difference of $c_0$ and $c_1$ to represent...
our confidence to say state $i$ is burst. In the backward step of dynamic programming, every interval’s burst weight is also calculated.

The outline of extraction process is listed in Algorithm 1. In the whole process, three parameters need to be adjusted manually: the low tag posting rate $\beta_0$, the high one $\beta_1$, and the state change probability $p$.

**Algorithm 1 Optimal Burst State Extraction**

**Input:** a specific tag frequency sequence in $N$ time intervals

**Input:** parameter $\beta_0$, $\beta_1$ and state change cost $p$

**Output:** burst weight sequence: $conf_1, conf_2, \ldots, conf_N$

**Step 1:** for each interval $i$ between 0 and $N$:
- calculate $f_{ij}$ and $path_{ij}$ iteratively.
- end for

**Step 2:** determine $burst_N$ and the burst confidence $conf_N$

**Step 3:** from interval $N-1$ to 0, for each $i$:
- determine the burst/stable state $burst_i$, also output the burst confidence $conf_i$
- end for

return the burst weight sequence.

The algorithm is robust and able to persist through noise and unstable situation. The added confidence weight also improves the mixture framework which we will discuss later.

### 3.2 Mixture model from multiple burst sources

In Section 2, we investigated some features (tag, user and resource) to measure a tag’s temporal variation and Section 3.1 present a burst detection approach for each single stream separately. Here we discuss how to combine these indicators together to improve the overall accuracy of burst detection.

Integrating multiple sources of feature is an important problem. There exist lots of manually tuning or unsupervised parameter tuning methods to resolve this. Due to the heterogenous nature of social media, here we select a learning based mixture model to combine these pulse information.

Guided by an input ground truth, Rankboost (Freund et al. 2003) is a method of producing prediction rules by combining many “weak” rules which may be only moderately accurate. For each individual ranker, a function $f_i$ is generated to map an instance $x_i$ to $R$. These given mappings are called ranking features. Here, all time intervals of a specific tag form an instance space $\mathcal{X}$ and $R$ is regarded as the ranking space. We regard the three features: tag, user and post as “weak” ranking features. From these preliminary burst detection results, we’d like to get a mixture result.

For a specific tag $t$, with the given burst truth, the detection loss is defined as follows:

$$ rloss_D(H,t) = \sum_{x_0, x_1} D_t(x_0, x_1) \delta(H_t(x_1) \leq H_t(x_0)) $$

$$ D_t(x_0, x_1) = c \cdot \max(0, \Phi_t(x_0, x_1)) $$

where $H_t$ is the sum combination of all individual rankers, and $\delta(\pi)$ is 1 if $\pi$ holds and 0 otherwise. $\Phi(\cdot)$ is the feedback function, $\Phi : \mathcal{X} \times \mathcal{X} \rightarrow R$, usually generated by ground truth or user labeling. If time interval $x_1$ is more bursty than $x_0$, then $\Phi(x_0, x_1) > 0$.

As proved in (Freund et al. 2003), the ranking loss of $H$ satisfies:

$$ rloss_D(H) \leq \prod_{t=1}^{T} Z_t $$

$$ Z_t = \sum_{x_0, x_1} D_t(x_0, x_1) \exp(\alpha_t(h_t(x_0) - h_t(x_1))) $$

where $h_t$ is the output of the $t^{th}$ individual ranker.

For each training tag instance with its corresponding labeled burst time interval sequences, we get a combination of weighting parameters $\alpha_1, \ldots, \alpha_K$ to tag, user and post separately. The learning process is described in Algorithm 2.

By averaging over all the tags from the training set, we get an overview mixture burst detection model with learned parameters. In the next section, we describe the experiment study showing the advantage of this approach.

**Algorithm 2 Rankboost-based Multiple Feature Mixture Burst Detection Model for a Specific Tag**

**Input:** the user annotated burst ground truth for this tag $t$: $x_0, x_2, \ldots, x_0N$.

**Input:** preliminary burst detection results of $K = 3$ features (tag, user and post): $x_{11}, x_{12}, \ldots, x_{1N}$; $x_{21}, x_{22}, \ldots, x_{2N}$ and $x_{31}, x_{32}, \ldots, x_{3N}$.

**Output:** the final ranking list $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ with the $\alpha$ parameters.

**Initialize:** $D_1 = D$ (based on the ground truth $x_0$).

**Ensemble:** for $k = 1, \ldots, K$, do:
- train individual learner using distribution $D_k$.
- get individual ranking $h_k$ from the $k^{th}$ ranking features.
- update this individual ranking’s weight: $\alpha_k$ based on the third method in Sec 3.2 (Freund et al. 2003).
- update $D_{k+1}(t_1, t_2) = \frac{D_k(t_1, t_2) \exp(\alpha_k(h_k(t_1) - h_k(t_2)))}{Z_k}$

end for

return parameter list;

### 4. Experiments

Here, we compare the proposed approach with those commonly used algorithms in a real world dataset. Extensive experiments are conducted to evaluate the performance and extensibility of the detected algorithm.

#### 4.1 Data Collection

We use a Delicious.com corpus with 51 million bookmarks, crawled at our group. We randomly choose 0.2 million users and collect their complete tagging history in 2008, 2009. The users’ network/subscription information are also collected. Every post page’s information is fetched. These raw pages
are about 640G in size. We extract these records and bulk them in MySQL, Lucene.

To remove long tail tags in this dataset, we only consider tags with frequency larger than 20 in any time interval. Month and week are selected as basic time interval granularity. There are about 0.1 million tags in all.

4.2 Evaluation Method

The burst detection is to find all burst periods of a querying tag or present top burst tags in a specific time period.

There’s no common ground truth for burst detection problem. We resort to user study and choose 50 tags for evaluation. These tags are chosen based on the frequency and burst activity randomly. Some of the labeled bursts include tags in Fig 1 and “nobel” (Nobel Prize), “halloween”, “swsx” (music and film conference).

Three volunteers are involved in this manual judgments. Each of them was asked to label the burst periods of a specific tag, by referencing several real world repositories, i.e., Google Trends, Yahoo! Upcoming.

Each burst detection approach will generate a classification or ranked list of time intervals. Based on the above labeled ground truth, we propose some IR style measurements for qualitatively comparison between different methods, which are MAP, R-precision, and Top N precision (P@N).

4.3 Effectiveness

We compare the approach with four baseline methods to demonstrate the effectiveness. Three are single stream based state detection model discussed in Section 3.1 applied on different features: tag frequency, time-aware post weighting and user attractiveness. The last one is a linear combination of three features and then goes through the state detection model.

For the proposed methods, three weak detectors are trained by tag frequency \( f_{\text{freq}} \), time-aware coverage \( c_{\text{cov}} \) and user attractiveness \( a_{\text{attr}} \) in each state detection model separately. Then these weak detectors are combined by Rankboost based learning, given the user labeled truth.

In our proposed approach, we apply 5-fold cross-validation experiments. In each trial, the parameters \( \alpha_0 \), \( \alpha_1 \) and \( \alpha_2 \) are auto-generated in the Algorithm 2 with four of the five subsets as training-set and the remaining one as testing-set. The parameters in linear combination are tuned manually on the whole dataset and reported best performance.

The result is shown in Table 1. We can see that our approach by aggregating features performed better than any other methods in all measurements.

Comparison with single features In single feature group, Tag Frequency based burst detection performs well, as it’s intuitive to understand this common sense. It’s also surprising to find that the performance of our introduced new burst dimension, i.e., Post Coverage and User Attractiveness is also comparable. It proves the effectiveness points of view stated in Section 3.2,3.3. The following two combination methods both have improvements compared with single feature baselines. This indicates that, the three features don’t overlap much and contribute to burst detection.

Comparison with linear combination of multiple features Our approach shows an improvement over the linear combination, though not significantly. First, we report the best performance of linear combination, which is somewhat overfitting. Second, the learning based approach is guided and provides intuitive results. In contrast, the parameters in the fusion step are difficult to learn automatically. Real bursts only reflected from unique feature is more easily smoothed in the simple linear aggregation process than Rankboost.

### Table 1: Tag burst detection performance

<table>
<thead>
<tr>
<th>methods</th>
<th>P@10</th>
<th>R-Precision</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag Frequency</td>
<td>0.33</td>
<td>0.65</td>
<td>0.71</td>
</tr>
<tr>
<td>Post Coverage</td>
<td>0.32</td>
<td>0.54</td>
<td>0.64</td>
</tr>
<tr>
<td>User Attractiveness</td>
<td>0.32</td>
<td>0.6</td>
<td>0.68</td>
</tr>
<tr>
<td>Linear Combination</td>
<td>0.33</td>
<td>0.66</td>
<td>0.73</td>
</tr>
<tr>
<td>Our Approach</td>
<td>0.39</td>
<td>0.68</td>
<td>0.77</td>
</tr>
</tbody>
</table>

![Figure 3: Tag burst detected from three single features](image)

Qualitative Example We show a qualitative example to present the improvement of our approach in Fig 3. Bursts detected by three single features often varies, from the above chart, these three all detected the burst of “chrome” in Sep 2008 when google launched its Web Browser “chrome”. In
2009, several comparatively trivial events about ‘chrome’ were recorded on google trends, such like Google launched experiments to boast ‘chrome’ in March, chrome OS introduced on June and chrome 3.0 released on Sep. All features captured some events while losing others or even detected non-existing bursts. However, the mixture detection model in this paper effectively identifies the bursts.

4.4 Parameter Tuning

We discuss parameter tuning in each step of the approach. We try different interval granularity as month and week. It turns out that the smaller granularity can locate burst to more accurate level while more sensitively affected by noise. We also do experiments with different values for $\beta$ and the two-state automaton parameters. Finally, we set $\beta_0$ the total post number in a time interval divided by the total time spanned in the time interval. The total time is using unix time stamp. $\beta_1 = 1.5\beta_0$ and $p = 0.49$.

5. Related Work

The generation of UGC and temporal dynamics is of increasing research interest (Agichtein et al. 2008; Baeza-Yates 2009). The comparison between tags and query log are discussed in (Carman et al. 2009). Three properties of folksonomy, namely the categorization, keyword, and structure property, are explored to support search (Xu et al. 2008).

Temporal aspects of social media are also discussed, e.g. visualization of a single tag stream in (Dubinko et al. 2006), a spatial clustering based event extraction in (Rattenbury, Good, and Naaman 2007), and a multi-step clustering and partition approach (Zhao, Mitra, and Chen 2007). In our previous work (Yao et al. 2010), we discussed how to better detect tag burst from tag’s co-occurrence information. Work in this paper extends these and investigates additional characters.

There has been much work in both burst detection and temporal text streams. An common approach for event detection is to identify bursty features from a document stream. Features sharing similar bursty patterns in similar time periods are grouped together to describe events and determine the periods of the bursty events.

There’re usually threshold based and state based methods. The threshold method is efficient though not adaptive. Kleinberg’a ‘burst of activity model’ (Kleinberg 2002) uses a probabilistic infinite-state automaton to model the dynamic change. These traditional methods usually only consider one single stream, which is limited in social media content. Though there’re some recent work investigating multiple stream alignment (Wang et al. 2007), the noisy and heterogeneous social media features prohibits the application of these traditional models. The detection model used in this paper improves the seminal ones.

6. Conclusion

In this paper, we present a novel approach to detect burst by combining multiple burst features. By introducing more dimensions of features, not only the effectiveness of burst detection improves, but also the temporal correlated and proximity mining make possible.

Though the setting and empirical analysis in this paper is based on folksonomy data. The discovered characters and developed methods are general and applicable across other social media content.

7. Acknowledgments

This research was supported by the National Natural Science foundation of China under Grant No. 60933004 and 60811120098.

References


Yao, J.; Cui, B.; Huang, Y.; and Zhou, Y. 2010. Detecting burst events in collaborative tagging systems. In Proc. of ICDE.