A Noise-Filtering Approach for Spatio-temporal Event Detection in Social Media

Yuan Liang, James Caverlee, and Cheng Cao

Department of Computer Science and Engineering, Texas A&M University
College Station, Texas, USA
{yliang,caverlee,chengcao}@cse.tamu.edu

Abstract. We propose an iterative spatial-temporal mining algorithm for identifying and extracting events from social media. One of the key aspects of the proposed algorithm is a signal processing-inspired approach for viewing spatial-temporal term occurrences as signals, analyzing the noise contained in the signals, and applying noise filters to improve the quality of event extraction from these signals. The iterative event mining algorithm alternately clusters terms and then generates new filters based on the results of clustering. Through experiments on ten Twitter data sets, we find improved event retrieval compared to two baselines.

1 Introduction

As users of services like Twitter and Facebook react to and report on their experiences – like political debates, earthquakes, and other real-world events – there is an opportunity for large-scale mining of these *socially sensed* events, leading to services that support intelligent emergency monitoring, finding nearby activities (e.g., rallies), and improving access to online content [5,14,20,29]. While there has been a long history of *event extraction* from traditional media like news articles, e.g., [1,25], the growth of user-contributed and often *on-the-ground* reaction by regular social media users has begun to spark new approaches.

In general, existing event detection methods can be categorized into two types: *document-pivot* approaches and *feature-pivot* approaches [7]. Document-pivot approaches identify events by clustering documents (e.g., news articles) based on semantic similarity, and then treating each cluster as an event. A series of works like [9,23] have shown the effectiveness of this method over long-form documents like news articles, which typically provide a rich source of context for event detection. Social media content, in contrast, often provides only a short description, title, or tags, (and thereby little textual narrative) limiting the effectiveness of semantic similarity based event detection techniques. As a result, many social media event detection algorithms have relied on *feature-pivot approaches*, which group similar event-related terms, for example by finding terms with a similar temporal distribution. In this way, event-related terms may be clustered together based on these common signals (treating each term as a frequency function over either time or space). These feature-pivot approaches,

A. Hanbury et al. (Eds.): ECIR 2015, LNCS 9022, pp. 233–244, 2015.

e.g., [3,4,29], have shown the potential of this approach for scaling to event detection over user-contributed social media posts.

While encouraging, these approaches may be susceptible to *noise* in both the temporal and spatial signals they use, which can hinder the quality of event detection. For example, topics not directly related to a specific event may introduce noise (e.g., discussion of a political candidate that is unrelated to a specific rally), as well as related but different events (e.g., reports of tornados in one city may pollute the signal of tornados in another city), and by data sparsity, in-correct timestamps or locations, mislabeled geo-coordinates, and so on.

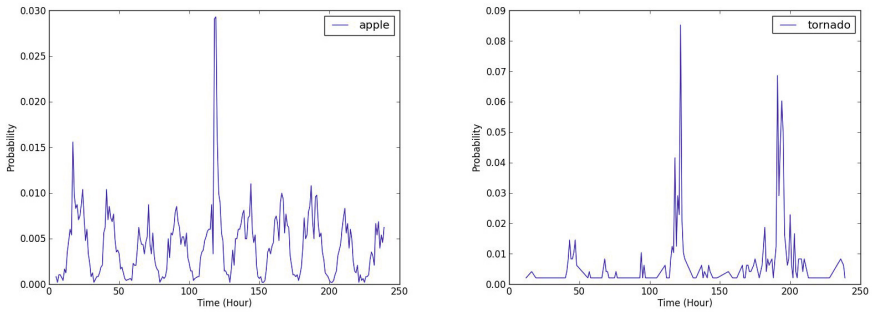
Hence, we explore in this paper a *signal-processing* inspired event detection framework designed to target these sources of noise. We view spatial-temporal term occurrences as signals, analyze their noise, and apply filters to improve the quality of event extraction from these signals. We incorporate this noise-filtering approach into an iterative spatial-temporal event mining algorithm for identifying and extracting events from social media. This approach alternately clusters terms using their filtered signals, and then generates new filters based on the results of clustering. Over ten Twitter-based event datasets – we find that the noise filtering method results in a 7-10% improvement versus alternatives.

2 Related Work

Event detection refers to the discovery of a specific activity that happens at a certain time and in a certain place. Event detection is typically categorized into two types: retrospective detection and on-line detection [25]. The former is to detect events from collected historical documents [15,13], and the latter tries to extract events from real-time documents [1,24,8]. Early detection approaches usually adopted clustering methods based on document similarity, e.g., [1] used a modified version of TF/IDF to measure the distance of documents. [25] added a time window and a decay factor for the similarity measurement between documents. In this paper, we focus on retrospective detection where the collection consists of user-generated content in social media.

User-generated content in social media has different characteristics from traditional document collections, so many clustering approaches have considered event-related metadata rather than directly measuring semantic relatedness. For instance, the work in [30] detects events from click-through web data by considering each event as a set of query-page pairs. In [14], a tweet is segmented into pieces and Wikipedia is exploited for identifying events. Via co-occurrence, [18] and [23] measured closeness of tags for landmark detection and tag recommendation. [21] constructed a keyword graph where co-occurrence frequency was used to assign weights on edges and then applied a shortest path based scheme to do community detection. [2] considered graph structure to bind all associated heterogeneous metadata, and proposed a co-clustering scheme to partition them into different events.

Separately, many approaches have adopted learning-based methods, including [4,5,11] or focused on temporal and spatial features. [17] utilized temporal



(a) Temporal Distribution for "apple" (b) Temporal Distribution for "Tornado"

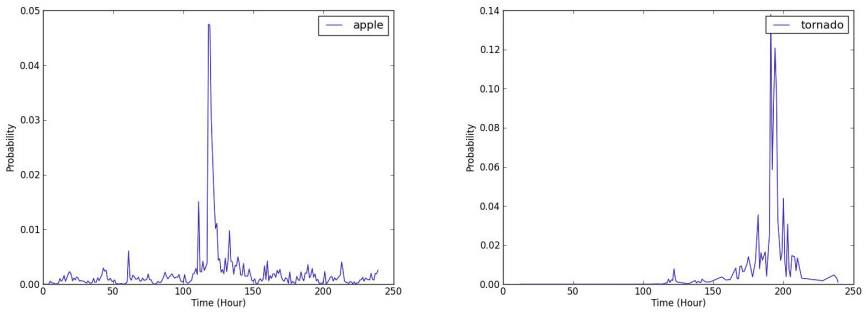
Fig. 1. Examples of Noise in Term Temporal Signals

information to determine a set of bursty features in different time windows, and then detected bursty events based on the feature distributions. [7] observed the spatial-temporal patterns for tags, and adopted a wavelet transform-based method to find tags with significant peaks in spatial-temporal distribution. Similarly, [19] looked for tags with bursts in temporal and spatial patterns for event detection. [29] compared spatial-temporal distributions between terms as the measurement of the closeness of different terms, and clustered terms based on the distances to extract events. At the same time, efforts such as [12,16,22,26,27,28] integrate geo-location information, showing the potential of spatial features.

3 Noise-Aware Event Detection

Given a collection of user-contributed social media documents $D = \{d_1, d_2, ..., d_T\}$, each document d_i can be viewed as $\langle W, t, l \rangle$, where W is a list of terms from vocabulary V, t is a timestamp indicating when d_i was posted, and $l = (la, lo)$ is the associated geo-location, consisting of latitude and longitude coordinates. We assume that there are K events $\theta = \{\theta_1, ..., \theta_K\}$ hidden in D and each document belongs to one of these events. Our goal is to detect these K hidden events from the observed documents. For our purposes, an *event* refers to a specific activity that happens in a specific time and place [7]. Therefore, given a group of terms, if it represents an event, the group of terms should satisfy three constraints: 1) the terms are semantically consistent, 2) the terms should happen in the same time period, and 3) the terms should appear in similar locations. Hence, we define event detection as: given a set of terms S, to detect subsets from S so that each subset $S_k \in S$ is a set of terms satisfying these constraints.

We propose to tackle event detection from a signal-processing perspective, where terms may be viewed as signals. For example, we could view a single term as a sequence of (normalized) counts for every minute of the day, resulting in a *temporal time signal*. That is, term w_i is represented by a temporal sequence of counts: $F_{t,w_i} = \{f_{i,1}, f_{i,2}, ..., f_{i,T}\}$, where t denotes the *temporal* signal domain.



(a) Temporal Distribution for "apple" (b) Temporal Distribution for "Tornado"

Fig. 2. Temporal Signals After Filtering Using the Proposed Method

Similarly, we could view a term as a two-dimensional *spatial term signal* by bucketing terms into a grid over the latitude-longitude space (denoted as F_{l,w_i} for a term w_i in the *location* signal domain). Both perspectives can additionally be merged into a three-dimensional *spatial-temporal term signal*, denoted by F_{t,l,w_i}. Together, we view the overall event signal corresponding to event θ_k as an aggregation of the signals of the terms belong to event θ_k. Hence, given a set of terms S_k associated with event θ_k, the event signal is:

$$F_{t,l,\theta_k} = \sum_{E(w_i)=\theta_k} F_{t,l,w_i} \lambda_{w_i,\theta_k} \tag{1}$$

where $E(w_i)$ refers to the corresponding event of w_i and $\lambda_{w_i,\theta}$ is the weight of w_i. Unfortunately, these event signals are necessarily *noisy*, meaning the detection faces significant challenges. We broadly classify three prominent types of noise:
Background-topic noise refers to the signals caused by unrelated topics to the event of interest, but that may overlap with the event of interest. For example, background discussion of "apple" as in Figure 1(a), which is unrelated to a major Apple announcement (the spike of attention).
Multi-event noise refers to the burst signal caused by other unrelated events. A term w_i can belong to multiple events, so its spatial-temporal signals are actually the combination of signals from multiple events, i.e., $F_{t,l,w_i} = \sum_k F_{t,l,w_i,\theta_k}$. For example, Figure 1(b) shows two tornado events.
Random noise refers to the random signals introduced by the sparsity of data, in-correct timestamps or locations, mislabeled geo-coordinates, and so on.

3.1 An Iterative Event Extraction Method

With these challenges in mind, we propose an iterative noise-aware event extraction method that seeks to limit the impact of noise. Concretely, we view that the term signals F_{t,l,w_i} for w_i are comprised of three components: (i) the event signal of interest F_{t,l,w_i,θ_e}; (ii) random noise F_{t,l,w_i,θ_r}; and (iii) event noise $F_{t,l,w_i,\theta_{S-e}}$, where S is the set of all the events: $F_{t,l,w_i} = F_{t,l,w_i,\theta_e} + F_{t,l,w_i,\theta_{S-e}} + F_{t,l,w_i,\theta_r}$

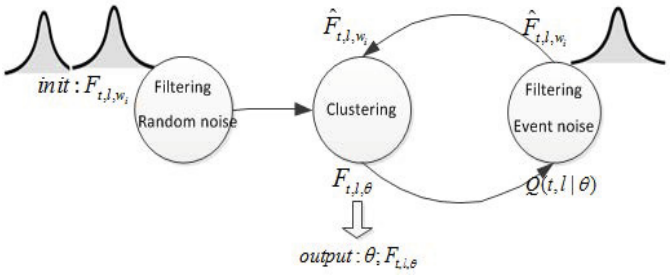


Fig. 3. Structure of Iterative Event Extraction Method

Our goal is to estimate the event signals F_{t,l,w_i,θ_e}, in effect cleaning the signal to focus primarily on the event of interest as illustrated in Figure 2. The overall approach is shown in Figure 3, where term signals are first filtered of random noise and then the signals are repeatedly clustered and filtered of event noise, until a final set of events is identified.

Filtering Random Noise. We begin with the first filter, for reducing random noise from the term signals. In speech and image processing, the *mean filter* is an effective way to smooth the signal and reduce un-correlated random noise [10]. In our context, we also assume that the random noise contained in the term signals are un-correlated, and therefore we can directly apply the mean filter to the signals. The key point of a mean filter is using the neighbors to average the signal values. For every point in the signals, the value is smoothed by:

$$F'_{t,l,w_i} = \sum_{t' \in N(t), l' \in N(l)} F_{t,l,w_i} Q(t', l') \tag{2}$$

For the mean filter, $Q(t', l')$ is set with $1/M$, where M is the number of neighbors, $N(t)$ refers to the set of neighbor points of t. A neighbor here is the point with adjacent time unit to t and close location to $l = (la, lo)$. For example, if we define $N(t) = [t-2, t+2]$ and $N(l) = [l-2, l+2]$, then all the points within 2 time units and 2 "distance" units (which could correspond to kilometers) at (t, la, lo) are regarded as the neighbors of the unit of (t, l).

Filtering Event Noise. After filtering random noise, we alternately cluster terms using their filtered signals, and then generate new filters based on the results of clustering, toward identifying groups of event-related terms. For the initial clustering, we adopt an existing co-occurrence based method [6] to group related term signals; alternately, other clustering methods could also be applied. These clusters could be immediately viewed as events, but for the inherent multi-event and background noise in the signals. Hence, we adopt a *band-pass filter* to limit the impact of these sources of noise. The intuition of the band-pass filter is to pass the signals in a Region-of-Interest, but filter or reduce the signals in other regions. After applying the band-pass filter, the cleaned term signals are clustered again. This iterative clustering and noise filtering proceed until

the clusters of terms do not change or the iteration count reaches a threshold. Finally, we output the clusters as the detected events.

The key issues are how to find the Region-of-Interest for a particular event, and how to estimate the band-pass filter $Q(t, l|\theta_k)$ based on the detected Region-of-Interest. Once the filter $Q(t, l|\theta_k)$ is estimated, we can use F_{t,l,w_i} and $Q(t, l|\theta_k)$ to retrieve the signals belonging to θ_k with Equation 3:

$$F_{t,l,w_i,\theta_k} = F_{t,l,w_i} Q(t, l|\theta_k) \tag{3}$$

where $Q(t, l|\theta_k)$ is the band-pass filter for θ_k in the spatial-temporal domain.

To detect the Region-of-Interest for a certain event θ_k, we propose to aggregate all the signals of the terms belonging to event θ_k, and then label the region which contains the strongest signals as the Region-of-Interest. The idea behind this method is to use the neighbors to filter un-correlated noises and strengthen the signals belonging to θ_k. In signal processing, mean filtering is used to sum multiple polluted signals. For example, if $s_1, s_2, ..., s_K$ are K different samples of the signal s polluted by noise, then the mean filter uses $\lambda_1 s_1 + \lambda_2 s_2 + ... + \lambda_K s_K$, $(\lambda_1 + \lambda_2 + ... + \lambda_K = 1)$ to find the un-polluted signal s. If the noise and signal are un-correlated, then by increasing K, the strength of the noise will be reduced to $1/\sqrt{K}$ [10]. Here, since individual terms can be polluted by some event noises which are usually uncorrelated, by averaging the signals of term w_i with the signals of its neighbors, the noise introduced by different events will be reduced.

Unlike the neighbors for random noise filtering which are found based on the adjacent time unit or spatial grid, the neighbors here refer to the terms belonging to the same event as determined by the clustering component. We first use a clustering method to find the neighbors for term w_i, then the signals belonging to the same cluster are averaged using Equation 1 to arrive at the estimated event signals. Regarding the clustering method, *k-means* is adopted in this paper if the number of actual clusters is already known, and *Affinity Propagation* is used if it is unknown.

We consider several different band-pass filters to explore their appropriateness for event detection from social media: a Gaussian band-pass filter, an Ideal band-pass filter, and an average band-pass filter.

Gaussian Band-Pass Filter: In the Gaussian filter, we assume that $Q(t, l|\theta_k)$ for θ_k can be represented as a single Gaussian. Then we use the event signals F_{t,l,θ_k} to train the parameters of $Q(t, l|\theta_k)$ where x is the vector of $\langle t, l \rangle$:

$$Q(t, l|\theta_k) = \frac{1}{\sigma\sqrt{2\pi}} exp\{-\frac{(x-\mu)^2}{2\sigma^2}\} \tag{4}$$

Ideal Band-Pass Filter: In the Ideal filter, we assume each point in the region (where the center is the point with strongest signal) has a weight much larger than points outside the region.

$$Q(t, l|\theta_k) = \begin{cases} \frac{\lambda}{r} & x \in [x_u, x_d] \\ \eta * \frac{1-\lambda}{R-r} & else \end{cases} \tag{5}$$

where λ is the cumulative frequency probability of the region $[x_u, x_d]$, x_u and x_d are the left-up and right-down coordinators respectively. r is the area of the region, R is the whole area of the boundary, and $\eta = 0.1$ is a penalty factor.

Average Band-Pass Filter: In the Average filter, $Q(t, l|\theta_k)$ the λ_{w_i,θ_k} in Equation 1 is set with $1/N$, where N is the number of terms belonging to θ_k.

4 Experiments

In this section, we evaluate the effectiveness of the proposed filter-based method for event extraction. We first investigate the impact of noise filtering and then compare the quality of the proposed approach versus two alternatives.

4.1 Data Collection

Our experiments are over ten different tweet datasets containing multiple events each (as shown in Table 1). We manually selected 20 events from Wikipedia between February 2011 to February 2013 and grouped them into six categories: *seasonal*, *burst*, *long-term*, *short-term*, *global area* and *local area*. An event may belong to more than one category, e.g., Christmas Eve can be in seasonal, short-term, and global. For each category, we manually select 10 hashtags that reflect the events in the category; collect all of the co-occurring hashtags; and finally rank by co-occurrence frequency. The top 10 hashtags are assumed as *relevant* for representing these events. We augment this group of six event categories with four additional datasets with a narrower geographic scope by (i) determining keywords that best describe an event; and (ii) using selected keywords to retrieve tweets for the event. We start with identifying one or two obvious keywords for an event, e.g., Irene for *Hurricane Irene*. Then we go through our tweets and find those terms that frequently appear together with our selected keyword(s). We select the top 15 terms to expand our keywords for each event, and retrieve the tweets containing the selected words.

4.2 Parameter Setup

For each selected term in the dataset, we first compute the temporal and spatial signals for them and measure the distance between each pair of terms based on the extracted signals as follows:

Temporal Distance: Given a complete time span, all the timestamps for each term w_i can be bucketed into bins: $\langle F_{t_1,w_i}, F_{t_2,w_i}, ..., F_{t_n,w_i} \rangle$. Then these temporal frequencies are normalized and used as the temporal signals. The width of each bin is set as 1 hour. The temporal distances based on F_{t,w_i} between w_i and w_j is defined as:

$$D_t(w_i, w_j) = \sum_t |F_{t,w_i} - F_{t,w_j}| \tag{6}$$

Table 1. Event Dataset

Dataset	Events	Period	Bounding[1]
SEASON	NBA, NFL, MLB, UEFA, Thanksgiving, Christmas, Halloween	02/01/2011 -02/01/2013	(0, 0) (90, 180)
BURST	Japan Tohoku earthquake 2011, Irene Hurricane 2011, Royal Wedding 2011, Sandy Hurricane 2012, London Olympics 2012, Arab Spring (2011–2012), US presidential election 2012	02/01/2011 -02/01/2013	(0, 0) (90, 180)
LONG	NBA, NFL, MLB, UEFA,Arab Spring (2011–2012), London Olympics 2012, US presidential election 2012	02/01/2011 -02/01/2013	(0, 0) (90, 180)
SHORT	Irene Hurricane 2011, Japan Tohoku Earthquake 2011 Royal Wedding 2011, Sandy Hurricane 2011, the Oscars 2013, the Cannes 2013, Steve Jobs' dearth 2011	02/01/2011 -02/01/2013	(0, 0) (90, 180)
GLOBAL	Arab Spring (2011-2012), London Olympics 2012, the Oscars 2013, the Cannes 2013, UEFA	02/01/2011 -02/01/2013	(0, 0) (90, 180)
LOCAL	Oktoberfest Beer Festival 2012, the Super bowl 2012, Memphis In May International Festival 2012	02/01/2011 -02/01/2013	(0, 0) (90, 180)
IRENE	Irene Hurricane 2011, Steve Jobs' resignation 2011, US Virginia earthquake 2011	08/20/2011 -08/30/2011	(29.6, -125.5) (49.1, -69.3)
JPEQ	Fire, Transportation, Asylum, Nuclear, General information of Tohoku Earthquake	03/11/2011 -03/20/2011	(30.4, 129.5) (45.4, 147.0)
MARCH	Japan Tohoku Earthquake 2011, Arab Spring (2011), New Zealand Christchurch earthquake 2011, Federal shutdown March 2011, background topic	03/01/2011 -03/30/2011	(29.6, -125.5) (49.1, -69.3)
AUGUST	Irene Hurricane 2011, Steve Jobs' resignation 2011, US Virginia earthquake 2011, Arab Spring (2011), background topic	08/01/2011 -08/30/2011	(29.6, -125.5) (49.1, -69.3)

[1] The geo-coordinates (latitude, longitude) of the left-up and right-down points of the rectangle bounding area.

Spatial Distance: The geographical bounding-boxes for terms are separated into $N * M$ mesh grids, and all the geo-coordinates for each term w_i are retrieved and bucketed into these grids: $\langle F_{l_1,w_i}, F_{l_2,w_i}, ..., F_{l_n,w_i} \rangle$. The N and M are set with 90 and 180 (1 degree for the width of grid). Based on the normalized spatial signals, the spatial distance between any w_i and w_j is defined as:

$$D_l(w_i, w_j) = \sum_l |F_{l,w_i} - F_{l,w_j}| \tag{7}$$

We then construct the noise filters as follows:

Average band-pass Filter: The weight λ in Equation 1 is set to $1/N$, where N is the size of the cluster.

Gaussian band-pass Filter: The μ in Equation 4 is estimated with the t with the highest term frequency (for temporal signals). σ is estimated with the d where $P((t-d) : (t+d)|\theta) = 0.68$. For spatial distributions, the μ in is estimated with the index of the grid l owning the highest term frequency, and the σ is estimated with the width of the square area, centered with μ, covering 68% percentage term frequencies.

Ideal band-pass Filter: The area $[x_u, x_d]$ in Equation 5 is computed via: 1) identify the center c by finding the bin with highest term frequency in temporal or spatial domain; 2) find the areas (1 dimension area in temporal domain, and 2 dimension square area in spatial domain) centered at c and covering 68% term frequencies. γ is set as 0.68 and λ is 0.1.

4.3 Results

To evaluate the effects of filters using our method, the first set of experiments is to separately test different filters considering both temporal features and spatial features. Concretely, we consider three filters: Average band-pass, Ideal, and Gaussian filters. K-means is used as the clustering method, and the average results of 5 times experiments are used for evaluation

Filtering Temporal Signals: To observe the effects of filters in temporal domain, the Average, Ideal and Gaussian band-pass filters are used on the temporal signals for terms, and temporal distance in Equation 6 is used to measure the similarity between terms. The clustering results using filtered signals and un-filtered signals are compared in Table 2. Table 2 indicates that generally the Event noise filters reduces the noises contained in temporal signal, resulting in better estimation of the distances, and thus achieves better clustering results. Compared with the method with un-filtered signals, the average purities on the 10 data sets using Average filter, Ideal filter and Gaussian band-pass filter are increased by 8.08%, 3.16%, and 1.95% on purity respectively. The probability-based filter – Average filter achieves the better results than the window-based filters (Gaussian and Ideal band-pass filter), most likely since the Gaussian and Ideal band-pass filters put large weights on the detected ROI region, which dramatically changes the power of the signals. If the ROI region is not detected correctly, it will incorrectly filter out the actual event signals.

Table 2. Purity Results for Filtering Temporal Signals

	Filter			
Dataset	No-filter	Average	Ideal	Gaussian
SEASON	0.662	**0.728**	0.693	0.680
BURST	0.749	0.774	0.753	**0.779**
LONG	0.722	**0.782**	0.733	0.760
SHORT	0.673	0.674	0.671	**0.678**
GLOBAL	0.683	0.648	0.693	**0.707**
LOCAL	0.604	**0.675**	0.582	0.496
IRENE	0.750	**0.813**	0.822	0.795
JPEQ	0.683	0.654	**0.706**	0.702
MARCH	0.400	**0.539**	0.426	0.427
AUGUST	0.429	**0.582**	0.477	0.455
Average	0.636	**0.687**	0.656	0.648

In addition, the improvements on March and August data sets by the noise-filters are more substantial than those on other data sets. These two datasets contain more noise corresponding to general topics due to the inclusion of common words like 'we' and 'like'. In an encouraging direction, we see that the proposed filters perform well in these cases of high noise.

Filtering Spatial Signals: In this experiment, the spatial distance in Equation 7 is used, and the Average, Ideal and Gaussian band-pass filters are compared in

spatial domain. Table 3 shows the clustering results on the 10 data sets using the spatial signals of terms. Compared with the methods with un-filtered spatial signals, the Average filter improves the clustering result by 3.73%, while the window-based methods degrade the clustering performance. One possible reason is that we assume the Gaussian window and rectangle window in the Gaussian and Ideal filters have only one center. However in the spatial domain, there are usually multiple centers for some events. For example, for the Irene event, there might exist multiple topic centers due to the transition of the center of hurricane. Therefore a single Gaussian or rectangle will incorrectly filter the real event signals, and thus degrade the clustering purities.

Also we can see that the filters have better performance in the temporal domain than the spatial domain. One possible reason could be that the spatial signals are more likely to be largely affected by the population density of different regions. If the ROI regions is incorrectly detected due to the population-affected tweet density, the filter will mistakenly filter out the actual event signals.

Table 3. Purity Results for Filtering Spatial Signals

	Filter			
Dataset	No-filter	Average	Ideal	Gaussian
SEASON	0.688	0.614	**0.731**	0.728
BURST	0.724	0.782	**0.811**	0.725
LONG	0.746	0.736	**0.782**	0.754
SHORT	0.667	0.659	0.635	**0.677**
GLOBAL	0.683	0.737	**0.844**	0.730
LOCAL	0.605	0.551	0.703	**0.735**
IRENE	0.681	**0.818**	0.590	0.727
JPEQ	0.662	**0.727**	0.246	0.246
MARCH	**0.375**	0.338	0.352	0.357
AUGUST	0.378	**0.479**	0.391	0.288
Average	0.621	**0.644**	0.609	0.597

Comparison with Baselines: Based on the results in the last section, we adopt the Average band-pass filter to filter noise in temporal and spatial signals. We combine the spatial and temporal distances into a unified distance as $D_{t,l,o}(w_i, w_j) = (D_o(w_i, w_j)+1)(D_t(w_i, w_j)+D_l(w_i, w_j))$, where $D_o(w_i, w_j)$ is a co-occurrence distance defined in [6]. As baselines we consider two alternatives: a co-occurrence based method [6] and a wavelet-based spatial-temporal method [7]. From Table 4, we observe that among three methods, the co-occurrence based and wavelet-based methods achieve comparable performances. Our proposed noise filtering method performs the best overall. On average, the proposed method has an improvement of 10.60% and 7.06% over the co-occurrence based and wavelet-based methods. The results indicate the proposed method is effective in filtering event-based noise, leading to higher quality event identification.

Table 4. Average Purity Comparison

	Methods		
Dataset	Co-occur	Wavelet	Proposed Method
SEASON	0.781	0.953	**0.984**
BURST	0.869	**0.920**	0.902
LONG	0.835	0.791	**0.851**
SHORT	0.828	0.714	**1.000**
GLOBAL	0.755	0.783	**0.857**
LOCAL	0.667	**0.836**	0.744
IRENE	0.718	0.773	**0.782**
JPEQ	0.734	0.716	**0.747**
MARCH	0.444	0.438	**0.450**
AUGUST	0.454	0.395	**0.519**
Average	0.709	0.732	**0.784**

5 Conclusion

The key insight of this paper is to view spatial-temporal term occurrences as signals, and then to apply noise filters to improve the quality of event extraction from these signals. The iterative event mining algorithm alternately clusters terms using their filtered signals, and then generates new filters based on the results of clustering. Over ten Twitter-based event datasets – we find that the noise filtering method results in a 7-10% improvement versus alternatives, suggesting the viability of noise-aware event detection.

Acknowledgment. This work was supported in part by NSF grant IIS-1149383.

References

1. Allan, J., Papka, R., Lavrenko, V.: On-line new event detection and tracking. In: SIGIR (1998)
2. Bao, B.-K., Min, W., Lu, K., Xu, C.: Social event detection with robust high-order co-clustering. In: ICMR (2013)
3. Batal, I., et al.: Mining Recent Temporal Patterns for Event Detection in Multivariate Time Series Data. In: KDD (2012)
4. Becker, H., Iter, D., Naaman, M., Gravano, L.: Identifying content for planned events across social media sites. In: WSDM (2012)
5. Becker, H., Naaman, M., Gravano, L.: Beyond Trending Topics: Real-World Event Identification on Twitter. In: ICWSM (2011)
6. Begelman, G., Keller, P., Smadja, F.: Automated tag clustering: Improving search and exploration in the tag space. In: Collaborative Web Tagging Workshop (2006)
7. Chen, L., Roy, A.: Event Detection from Flickr Data through Wavelet-based Spatial Analysis. In: CIKM (2009)
8. Chen, Y., Amiri, H., Li, Z., Chua, T.-S.: Emerging topic detection for organizations from microblogs. In: SIGIR (2013)
9. Garg, N., Weber, I.: Personalized tag suggestion for Flickr. In: WWW (2008)

10. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice-Hall (2007)
11. He, Q., Chang, K., Lim, E.P.: Analyzing feature trajectories for event detection. In: SIGIR (2007)
12. Hong, L., Ahmed, A., Gurumurthy, S., Smola, A.J., Tsioutsiouliklis, K.: Discovering geographical topics in the twitter stream. In: WWW (2012)
13. Kleinberg, J.: Bursty and hierarchical structure in streams. Data Min. Knowl. Discov. (2003)
14. Li, C., Sun, A., Datta, A.: Twevent: Segment-based event detection from tweets. In: CIKM (2012)
15. Li, Z., Wang, B., Li, M., Ma, W.Y.: A probabilistic model for retrospective news event detection. In: SIGIR (2005)
16. Mei, Q., Liuy, C., Suz, H., Zhaiy, C.: A Probabilistic Approach to Spatiotemporal Theme Pattern Mining on Weblogs. In: WWW (2006)
17. Moxley, E., Kleban, J., Xu, J., Manjunath, B.S.: Not all tags are created equal: Learning Flickr tag semantics for global annotation. In: ICME (2009)
18. Papadopoulos, S., Zigkolis, C., Kompatsiaris, Y.: Cluster-Based Landmark and Event Detection for Tagged Photo Collections. In: Multimedia (2010)
19. Rattenbury, T., Good, N., Naaman, M.: Towards automatic extraction of event and place semantics from Flickr tags. In: SIGIR (2007)
20. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: WWW (2010)
21. Sayyadi, H., Hurst, M., Maykov, A.: Event Detection and Tracking in Social Streams. In: ICWSM (2009)
22. Sengstock, C., Gertz, M., Flatow, F., Abdelhaq, H.: A probablistic model for spatio-temporal signal extraction from social media. In: SIGSPATIAL (2013)
23. Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: WWW (2008)
24. Valkanas, G., Gunopulos, D.: How the live web feels about events. In: CIKM (2013)
25. Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: SIGIR (1998)
26. Yin, Z., Cao, L., Han, J., Zhai, C., Huang, T.: Geographical Topic Discovery and Comparison. In: WWW (2011)
27. Yin, Z., Cao, L., Han, J., Zhai, C., Huang, T.: LPTA: A Probabilistic Model for Latent Periodic Topic Analysis. In: ICDM (2011)
28. Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N.M.: Who, where, when and what: Discover spatio-temporal topics for twitter users. In: KDD (2013)
29. Zhang, H., Korayem, M., You, E., Crandall, D.J.: Beyond Co-occurrence: Discovering and Visualizing Tag Relationships from Geo-spatial and Temporal Similarities. In: WSDM (2012)
30. Zhao, Q., Liu, T.-Y., Bhowmick, S.S., Ma, W.-Y.: Event detection from evolution of click-through data. In: KDD (2006)