Multiscale Event Detection in Social Media

Xiaowen Dong · Dimitrios Mavroeidis · Francesco Calabrese · Pascal Frossard

Abstract Event detection has been one of the most important research topics in social media analysis. Most of the traditional approaches detect events based on fixed temporal and spatial resolutions, while in reality events of different scales usually occur simultaneously, namely, they span different intervals in time and space. In this paper, we propose a novel approach towards multiscale event detection using social media data, which takes into account different temporal and spatial scales of events in the data. Specifically, we explore the properties of the wavelet transform, which is a well-developed multiscale transform in signal processing, to enable automatic handling of the interaction between temporal and spatial scales. We then propose a novel algorithm to compute a data similarity graph at appropriate scales and detect events of different scales simultaneously by a single graph-based clustering process. Furthermore, we present spatiotemporal statistical analysis of the noisy information present in the data stream, which allows us to define a novel term-filtering procedure for the proposed event detection algorithm and helps us study its behavior using simulated noisy data. Experimental results on both synthetically generated data and real world data collected from Twitter demonstrate the meaningfulness and effectiveness of the proposed approach. Our framework
further extends to numerous application domains that involve multiscale and multiresolution data analysis.

**Keywords** Multiscale event detection · Spatiotemporal analysis · Wavelet decomposition · Modularity-based clustering

1 Introduction

The last decade has seen rapid development of online social networks and social media platforms, which leads to an explosion of user-generated data posted on the Internet. The huge amount of such data enables the study of many research problems, and event detection is certainly one of the most popular and important topics in this novel research area. Social media platforms present several advantages for event detection. First, due to the real-time nature of online social services, the public awareness of real world happenings could be raised in a much quicker fashion than with the traditional media. Second, due to the large amount of users posting content online, more complete pictures of the real world events with descriptions from different angles are offered with fast and large-scale coverage. These advantages have attracted a significant amount of interest from the data mining communities in event detection problems. For instance, the MediaEval Workshop has open research task dedicated to event detection (Reuter et al., 2013), and numerous event detection approaches have been proposed recently in the literature (Sayyadi et al., 2009; Becker et al., 2009; Aggarwal and Subbian, 2012).

Events in social media platforms can be loosely defined as real world happenings that occur within similar time periods and geographical locations, and that have been mentioned by the online users in the forms of images, videos or texts. Different types of events are usually of different temporal and spatial scales or resolutions$^1$, meaning that they span different intervals in time and space. For example, discussions about the London 2012 Summer Olympic Games would span a temporal period of nearly one month and a spatial area of all over the world, while those regarding the 2012 concert of The Stone Roses in the Phoenix Park in Dublin may concentrate only on the date and at the location of the concert. Similarly, Fig. 1 illustrates discussions on Twitter about two events of different spatiotemporal scales in New York City. In the designs of event detection algorithms, it is thus important to take into account the different temporal and spatial scales of various kinds of events. This is challenging in the sense that: (i) Event detection approaches usually rely on classification or clustering algorithms with fixed temporal and spatial resolutions; This results in the detected events being of similar scales; (ii) It is not yet clear how multiple resolutions in time and in space interact with each other so that they can be analyzed simultaneously, even if it is relatively easier to take into account multiple resolutions in only one of these two dimensions; (iii) Data streams from social media platforms usually contain much

---

$^1$ Throughout the paper, we use “scales” and “resolutions” interchangeably.
noisy information irrelevant to the events of interest. It is thus important to understand how to attenuate the influence of the noise on detecting events of different scales. Efficient and robust multiscale event detection for solving the above challenges is exactly the objective of the present paper.

In this paper, we first introduce a baseline approach that detects events that are of similar scales and localized in both time and space, which serves as a first step towards the understanding of multiscale event detection. We then propose a novel approach towards the detection of events that are of different scales and localized either in time or in space but not necessarily in both simultaneously. To this end, we study the relationship between scales in the two dimensions and explore the properties of the wavelet transform to automatically and explicitly handle the interaction between different scales in time and space simultaneously. We propose an algorithm to compute a data similarity graph at appropriate scales, based on which we perform a graph-based clustering process to detect events of different spatiotemporal scales. Furthermore, we present spatiotemporal analysis of the distribution of noisy information in data streams, especially using notions from spatial statistics, which allows us to define a novel term-filtering procedure for the proposed multiscale event detection algorithm, and helps us study the behavior of the two approaches in this paper using simulated noisy data.

We compare the proposed multiscale event detection approach with the baseline approach on both synthetically generated data and real world data collected from Twitter. We show experimentally that the proposed approach can effectively detect events of different temporal and spatial scales. On the one hand, we believe that the modeling of the relationship and interaction between temporal and spatial scales and the detection of multiscale events provide new insights into the task of event detection with social media data. On the other hand, the proposed framework can be further generalized to other application domains that involve multiscale or multiresolution data analysis.
2 Spatiotemporal detection of events

In this paper, we define an “event” in social media as follows.

**Definition.** Events in social media are real world happenings that are reflected by data that are concentrated either in both time and space, or in at least one of the two dimensions.

Events defined as above are usually of different temporal and spatial scales, namely, they span different intervals in time and space. In addition, there exist data that do not contain any information about ongoing events. In the case of Twitter, such examples can be tweets that are like: “At work”, or “It feels great to be home...”. When non-informative tweets constitute a large part of the input data, the event-relevant tweets could however be buried in noise. It becomes very difficult in this case to identify the information of interest. In this paper, we focus on the Twitter data streams and consider the following objective.

**Objective.** Consider a Twitter data stream that contains temporal, spatial and text information. Our goal is to design event detection approaches that (i) are able to identify events that appear at multiple spatiotemporal scales, namely, events that affect or take place in different temporal and spatial intervals, and (ii) are robust against the ambiguous and noisy information present in the data.

In this paper, we cast event detection as a graph-based clustering problem, where the vertices of the graph represent the tweets, and the edges reflect their similarities. The goal is to group similar tweets into the same cluster such that they correspond to a real world event. The clustering algorithm utilizes a similarity measure between tweets that takes into account the temporal, spatial, and textual features of a tweet. Intuitively, two tweets that are generated by users that are participating in the same event should share a number of common terms and be closely located in time and/or space. In this paper, we compare two different ways of measuring similarity between tweets, the first a baseline approach based on spatiotemporal constraints and the second a novel wavelet-based scheme. Then, in order to effectively handle the noisy information, we study the spatiotemporal distribution of the noise in the Twitter data, especially using a homogeneous Poisson process as a statistical model in our analysis. This is helpful to analyze the behavior of the baseline and the proposed event detection algorithms.

3 Local event detection via spatiotemporal constraints

Events defined as in the previous section can have different localization behavior in time and space. When the events are localized in both dimensions, event detection can be effectively implemented by imposing spatiotemporal constraints on the data. In this section, we first describe a baseline approach
for detecting events that are localized both in time and space, which serves as a first step towards the understanding of multiscale event detection presented later. We formulate a clustering problem, where we wish to group together the tweets that correspond to the same real world event. The similarity measure between different tweets is thus important. In our baseline event detection approach, we measure the similarity between every pair of tweets \( t_i \) and \( t_j \) as:

\[
S_1(t_i, t_j) = \begin{cases} 
    s_{tf-idf}(t_i, t_j) & \text{if } t(t_i, t_j) \leq T_t \text{ and } d(t_i, t_j) \leq T_d, \\
    0 & \text{otherwise.}
\end{cases}
\]  

(1)

where \( t(t_i, t_j) \) and \( d(t_i, t_j) \) are the temporal difference in minutes and the spatial distance in meters, respectively, between \( t_i \) and \( t_j \). The thresholds \( T_t \) and \( T_d \) enforce the locality of the events and impose strict spatiotemporal constraints. Under such constraints, two tweets \( t_i \) and \( t_j \) that have a reasonably high text similarity tend to refer to the same event in real world. The function \( s_{tf-idf}(t_i, t_j) \) represents the text similarity of \( t_i \) and \( t_j \) in terms of the cosine angle between the vector representations of the two tweets using the term frequency-inverse document frequency (tf-idf) weighting scheme (Manning et al., 2008).

Given \( S_1(t_i, t_j) \) as the pairwise similarity between tweets, we can create an undirected and weighted graph with adjacency matrix \( W_1 \):

\[
W_1(i, j) = \begin{cases} 
    S_1(t_i, t_j) & \text{if } i \neq j, \\
    0 & \text{if } i = j,
\end{cases}
\]  

(2)

where the vertices represent tweets and the edges (along with the associated weights) are defined by \( S_1(t_i, t_j) \). By partitioning the vertices of the graph into disjoint clusters, each cluster is then expected to contain tweets that are likely to correspond to the same event. Furthermore, due to the constraints introduced in Eq. (1), these events are localized in both time and space. In this paper, we perform graph-based clustering using the Louvain method (Blondel et al., 2008). This is a greedy optimization method that first find small communities in a local way by maximizing the modularity function (Newman, 2006), before repeating the same procedure by considering the communities found in the previous step as vertices in a new graph, until a maximum of modularity is attained\(^2\). The Louvain method is suitable for our purpose of event detection because of the following advantages: (i) Unlike most of the clustering methods, it does not require prior knowledge about the number of clusters; This is important because we usually do not know the number of events a priori. (ii) Unlike the popular approach based on normalized graph cut (such as spectral clustering (von Luxburg, 2007)), it does not necessarily favor a balanced clustering; This enables the detection of small-scale clusters together with some relatively larger ones. (iii) It is also computationally very efficient when applied to large scale networks. Specifically, the complexity of

\(^2\) Since we are interested in local clusters, we apply the non-recursive version of the Louvain method which stops after the first iteration.
Fig. 2 (a) Events of similar scales and are localized in both time and space. (b) Events of different scales and are not necessarily localized in both time and space.

The greedy implementation in Blondel et al (2008) is empirically observed to be close to $O(n \log n)$ where $n$ is the number of the vertices in the graph.

The graph-based clustering approach described above outputs a set of clusters that correspond to events localized in both time and space. This can be illustrated by Fig. 2(a), where each cluster corresponds to a particular time-space “cube”. After clustering, we apply simple post-processing steps to identify those clusters that are likely to correspond to meaningful events in real world. For example, we consider that a meaningful event should be observed by a sufficient number of users with sufficient information reflected on Twitter. Therefore, we consider a cluster as a local event if and only if the number of tweets and distinct Twitter users within the cluster are above certain thresholds (see Sect. 7 for the implementation details of these post-processing steps).

The algorithm for local event detection is summarized in Algorithm 1.

Obviously, the choices for the values of thresholds $T_t$ and $T_d$ in Eq. (1) are critical in LED. Without prior information we may choose them such that they correspond to the expected temporal and spatial spans of events to be discovered. By setting $T_t$ and $T_d$ appropriately, the algorithm would then be efficient at detecting events that are of similar scales and that are sufficiently concentrated in both time and space. For events of different scales, however, setting the thresholds too low might break down some event clusters while setting them too high would generally lead to a higher amount of noisy information in other clusters

One may think of applying LED with small values for $T_t$ and $T_d$ before grouping similar clusters together using a second clustering step. In fact, the second and further iterations of the Louvain method already offers such a grouping. Alternatively, a hierarchical clustering algorithm can be applied to the clusters obtained by LED. However, such further grouping process does not usually lead to a clear interpretation in terms of the spatiotemporal scales of the resulting event clusters, and it is often difficult to decide when to stop the recursive process and output the eventual clusters.
Algorithm 1: Local Event Detection via Locality Constraints (LED)

1: Input:
   \( T \): a set of tweets with temporal, spatial, and text information
   \( T_t \): temporal threshold
   \( T_d \): spatial threshold
2: Compute the pairwise similarities \( S_1(t_i, t_j) \) between tweets in \( T \) using Eq. (1), and the adjacency matrix \( W_1 \) using Eq. (2).
3: Apply the non-recursive Louvain method to \( W_1 \), and retain the meaningful clusters \( \{c_i\}_{i=1}^m \) after post-processing steps.
4: Output:
   \( \{c_i\}_{i=1}^m \): clusters that correspond to events that are localized in both time and space.

4 Multiscale event detection using wavelets

In this section, we propose a novel algorithm for multiscale event detection. Specifically, we first introduce a new model of the relationship and interaction between the temporal and spatial scales. We then propose a wavelet-based scheme for computing the pairwise multiscale similarities between tweets.

4.1 Relationship model between temporal and spatial scales

The fundamental question in designing approaches towards multiscale event detection resides in properly handling events that are of different scales and do not have simultaneous temporal and spatial localization. An illustration is shown in Fig. 2(b), where three events are represented by rectangular cuboids that span different time and space intervals. Two of them are only concentrated in one dimension but spread in the other one. In such cases, we need to compute a similarity score \( S_2(t_i, t_j) \) between pairs of tweets \( t_i \) and \( t_j \) that carefully considers the temporal and spatial scales of different events. We shall relax the strict constraints in both temporal and spatial dimensions as defined in Eq. (1), so that \( S_2(t_i, t_j) \) is computed at appropriate scales that actually correspond to the span of the underlying events. To this end, we propose in this paper to model the relationship and interaction between the temporal and spatial scales as follows.

Scale relationship model. When two tweets \( t_i \) and \( t_j \) share common terms and are close in space, we could tolerate a coarser temporal resolution in computing \( S_2(t_i, t_j) \). Vice versa, when they are close in time, we could tolerate a coarser spatial resolution.

Our scale relationship model essentially says that, for two tweets \( t_i \) and \( t_j \) to be considered similar, they should be similar at a fine resolution in at least one of the temporal or spatial dimensions, but not necessarily in both simultaneously. It thus represents a tradeoff between time and space in the detection of events of different spatiotemporal scales. This matches the observation that real world events often happen within a small geographical area but could span longer time intervals (such as a protest at a certain location in a city), or they take
Fig. 3 Two time series of the number of occurrences of a certain term (computed using the temporal resolution $\Delta t$) within two different geographical cells. These geographical cells are defined by discretizing the geographical area using the spatial resolution $\Delta d$.

place only within short time intervals but could spread a larger geographical area (such as a brief power outage across different areas of a city). Therefore, based on the proposed model, we can relax the strict constraints defined in Eq. (1) in event detection.

In order to do so, however, we do not compare two tweets $t_i$ and $t_j$ with large temporal or spatial distances by simply choosing higher thresholds $T_t$ and $T_d$, since this would suffer from text ambiguity generally present in the Twitter data stream (the same word having different meanings depending on context). We do not either incorporate directly the exact temporal and spatial distances between them into the computation of the similarity metric $S_2(t_i, t_j)$, since this might lead to domination of one scale to the other. These limitations motivate us to propose a more detailed analysis model, that is, instead of considering the temporal and spatial information of each tweet as a whole, we now analyze spatiotemporal patterns of the terms (or keywords) contained in each tweet. More specifically, to compare two tweets $t_i$ and $t_j$, we propose to look at the similarity between the time series of the number of occurrences of the common terms shared by them (the occurrence is evaluated in terms of in how many tweets these terms appear). On the one hand, this enables us to study the interaction between the temporal and spatial scales when computing the similarity between keyword time series. On the other hand, this does not affect the clustering-based event detection framework, as similarities between tweets would eventually be computed based on similarities between time series of the common terms shared by them.

We build the time series of keywords as follows. We start with initial temporal resolution $\Delta t$ and spatial resolution $\Delta d$. Next, for each term shared by $t_i$ and $t_j$, we compute using the temporal resolution $\Delta t$ two time series of its number of occurrences, that are based on data corresponding to the two geographical cells to which $t_i$ and $t_j$ belong. These geographical cells are defined by discretizing the geographical area using the spatial resolution $\Delta d$. The keyword time series are illustrated in Fig. 3.
4.2 Wavelet-based similarity computation

We now propose to use a wavelet-based method to measure similarities between time series of keywords. Similarity between time series are often measured by the correlation of their coefficients under the wavelet transform (Daubechies, 1992), which is a well-developed tool in signal processing that leads to a multisiresolution representation of the signals. In this paper, we consider the discrete wavelet transform (DWT) using the Haar wavelet, since it provides a natural way to handle different temporal scales as required in our approach. Specifically, due to the properties of the Haar wavelet, the approximation coefficients of DWT at different levels naturally correspond to aggregating the time series from fine scales (starting with the initial temporal resolution) into coarse scales, each time by a factor of two. Therefore, to evaluate the similarity of the time series at a certain temporal scale, we only need to measure the correlation between a specific set of the DWT coefficients at the corresponding level (see Fig. 4 for an illustration).

Our key idea is then to evaluate the similarity between the two time series shown in Fig. 3 at a properly chosen temporal scale, which is in turn determined by the spatial distance between the two geographical cells. More specifically, we introduce a number of predefined spatial scales for the spatial distance. Then, if the spatial scale is coarse, which means that \(t_i\) and \(t_j\) are distant, then we require the time series to be compared at a finer temporal scale (the finest temporal scale being the initial temporal resolution); Alternatively, if the spatial scale is fine, which means that \(t_i\) and \(t_j\) are close, then the time series could be compared at a coarser temporal scale. Given the number of spatial scales specified by the parameter \(n_{\text{scale}}\), we define \(n_{\text{scale}}\) distance ranges using logarithmical equispacing between the minimum and maximum distances between two distinct geographical cells (measured based on the center of the cells), which correspond to these spatial scales.

According to the scale relationship model, the temporal scale \(S_t\) is then selected inversely according to the spatial scale:

\[
S_t = n_{\text{scale}} + 1 - S_s. \tag{3}
\]

For instance, if we choose to have \(n_{\text{scale}} = 4\) spatial scales \(S_s \in \{1, 2, 3, 4\}\), 1 being the coarsest and 4 the finest, then we would have \(S_t = 4, 3, 2, 1\), respectively, that represent from the finest to the coarsest temporal scale. This in turn means that we compute the DWT at levels from 1 to 4, respectively. This procedure is illustrated in Fig. 4.

We can now define a new similarity metric between two tweets \(t_i\) and \(t_j\) as follows:

\[
S_2(t_i, t_j) = s_{\text{f-idf}}(t_i, t_j) \times s_{\text{st}}(t_i, t_j), \tag{4}
\]

When two tweets come from the same geographical cell, they would share the same time series for any common term. In this case, the correlation of DWT coefficients would always be 1 regardless of the level at which we compute the transform (or the temporal scale). This special case can be interpreted as only keeping the spatial constraint in LED but relaxing the temporal constraint.
Fig. 4 An illustration of wavelet-based similarity computation for time series of length 16 (zero padding can be applied when needed if the length of time series is not a power of 2). The dots indicate the DWT coefficients at the corresponding level that are used for computing correlation, while the dots highlighted by the rounded rectangle correspond to approximation coefficients at each level. The approximation coefficients at DWT level $k$ correspond to a time series generated by aggregating every $2^k$ entries in the original time series (up to a constant).

where $s_{\text{tf-idf}}(t_i, t_j)$ is the text similarity of $t_i$ and $t_j$ defined as in Eq. (1). For each term shared by $t_i$ and $t_j$, we can compute a similarity of the corresponding time series; $s_{st}(t_i, t_j)$ is then defined as the maximum such similarity among all the terms shared by $t_i$ and $t_j$. The reasons why we choose the maximum similarity are as follows. First, social media platforms that are ideal for event detection usually contain short textual data where two pieces of text, if corresponding to the same event, would share only a few but informative common terms, such as hashtags in Twitter or tags in Youtube or Flickr. Second, in Twitter specifically, although many tweets may share the same popular term, it is less often that there would be a high similarity between the two keyword time series in terms of their spatiotemporal patterns, especially at fine temporal scales, after a term-filtering procedure which we propose in the next section that removes the “noisy” terms that generally spread in time and space. We thus consider high similarity between time series as a strong indicator that $t_i$ and $t_j$ may be related to the same event. Taking the maximum instead of the average similarity helps us preserve such information and promote a higher recall metric (retrieval of positive links between tweets) that we favor.

In Eq. (4), we consider the overall similarity between two tweets as a product of their text similarity ($s_{\text{tf-idf}}(t_i, t_j)$) and the similarity of spatiotemporal patterns of the terms shared by them ($s_{st}(t_i, t_j)$). This leads to an interesting comparison between LED and MED: Both approaches only consider the text similarity $s_{\text{tf-idf}}(t_i, t_j)$ that is meaningful in event detection; However, while the former relies on fixed temporal and spatial constraints on $t_i$ and $t_j$, the latter looks at similar spatiotemporal patterns of the common terms, thus offers more flexibility for events of different scales. Finally, we can use our new similarity metric to construct an undirected and weighted graph $W_2$:

$$W_2(i, j) = \begin{cases} S_2(t_i, t_j) & \text{if } i \neq j, \\ 0 & \text{if } i = j, \end{cases}$$

(5)
Based on this similarity graph, we can again apply the Louvain method to detect event clusters. The complete algorithm for the proposed multiscale event detection approach is summarized in Algorithm 2.

**Algorithm 2 Multiscale Event Detection using Wavelets (MED)**

1: **Input:**
   - $T$: a set of tweets with temporal, spatial, and text information
   - $\Delta t$: initial temporal resolution
   - $\Delta d$: initial spatial resolution
   - $n_{\text{scale}}$: number of spatial scales
2: For every pair of tweets $t_i$ and $t_j$ in $T$, extract the common terms $\{w_i\}_{k=1}^{n_{\text{scale}}}$.
3: For each $w_i$, compute using $\Delta t$ the time series of its number of occurrences, that are based on data corresponding to the two geographical cells (defined using $\Delta d$) to which $t_i$ and $t_j$ belong.
4: Determine using Eq. (3) the temporal scale $S_t$ using the spatial scale $S_s$ to which the distance between the two geographical cells corresponds.
5: Apply DWT to the two time series, and compute the similarity between them as the correlation between a specific set of DWT coefficients at the level corresponding to $S_s$.
6: Compute $s_{st}(t_i, t_j)$ as the maximum time series similarity among $\{w_i\}_{k=1}^{n_{\text{scale}}}$. Compute $s_{st}(t_i, t_j)$ using Eq. (4), and the adjacency matrix $W_2$ using Eq. (5).
7: Apply the non-recursive Louvain method to $W_2$, and retain the meaningful clusters $\{c_i\}_{i=1}^{n_{\text{scale}}}$ after post-processing steps.
8: **Output:**
   - $\{c_i\}_{i=1}^{n_{\text{scale}}}$: clusters that correspond to events of different temporal and spatial scales.

There are a number of parameters in our multiscale event detection approach. First, the initial resolution parameters $\Delta t$ and $\Delta d$ are used for constructing keyword time series; Compared to $T_t$ and $T_d$ in LED, they do not have to adapt to the “true” scales of various events, thanks to the scale relationship model and the scale adjustment afterwards using the wavelet-based scheme. In practice, we can simply choose them to be relatively small, for example, as the expected minimum temporal and spatial intervals a desired event may span (specific example choices are presented in Sect. 7). Second, the number of spatial scales $n_{\text{scale}}$ can be considered as a choice in the design of the algorithm. Intuitively, an $n_{\text{scale}}$ too small would not take full advantage of the spatiotemporal scale relationship model, while $n_{\text{scale}}$ being too large might lead to unnecessary increase in computational cost. The choice of this parameter is also influenced by the resolution parameters $\Delta t$ and $\Delta d$. One the one hand, $\Delta d$ determines the number of geographical cells $l_d$ along one dimension hence the spatial variability in the data. This implicitly controls the maximum $n_{\text{scale}}$ such that the resulting distance scales are meaningful. On the other hand, given a certain time span of data, the temporal resolution $\Delta t$ would determine the length of keyword time series $l_t$, which in turn determines the maximum (meaningful) level of DWT computation using a Haar wavelet and hence the maximum temporal scale. Because of the relationship in Eq. (3), the maximum spatial scale is thus determined accordingly. Based on these two observations, we therefore suggest considering $\lceil \min(\log_2 l_d, \log_2 l_t) \rceil$ as an upper bound for $n_{\text{scale}}$, where $\lceil \cdot \rceil$ denotes the ceiling of a number. In our
experiments, we choose $n_{\text{scale}} = 4$ to ensure a certain level of spatial variability while respecting this upper bound.

5 Spatiotemporal analysis of noise in Twitter

One challenge in designing event detection algorithms for Twitter data is that we often need to deal with a large amount of “noise” tweets that do not provide any information regarding real world events. Examples can be tweets such as “Could really use a drink” or “Nachos for lunch”, or discussions between Twitter users about personal matters. We consider these tweets as noise and event detection algorithms should be able to discard them and not allow them to influence the event detection result. In the literature, several works (such as Sakaki et al (2010)) have employed keyword filtering techniques in order to tackle this problem and derived a working set of tweets that contain information relevant to the types of events they wish to detect. Since we do not focus in this paper on specific event types, but rather on events that take place in specific locations and time intervals, we analyze in this section the spatiotemporal structure of the noise, namely, the event-irrelevant tweets in the data. This analysis will allow us to define a novel term-filtering procedure, and to evaluate empirically the performance of the event detection algorithms in this paper using simulated noisy data under different space-time parameters.

5.1 Spatial distribution of noise in Twitter data

In order to get an intuition about the relevant spatial statistics models that can be useful for analyzing the spatial distribution of the noise, we focus on a set of geo-located tweets collected from a specific day (22-01-2012) in New York City. In this dataset, four of the top-ten frequent terms are: nyc contained in 335 tweets (183 of which are located in middle and lower Manhattan), love contained in 674 tweets (145 of which are located in middle and lower Manhattan), lol contained in 1080 tweets (110 of which are located in middle and lower Manhattan), and night contained in 355 tweets (97 of which are located in middle and lower Manhattan). These terms, albeit being among the most frequent ones in the daily collection of tweets, do not appear to be relevant to a specific event of interest. In Fig. 5 we illustrate the locations of the tweets (in middle and lower Manhattan) that contain these frequent terms. One can observe that the tweets have a slight, but not strong spatial concentration and appear to be almost randomly distributed within the Manhattan area. Based on these spatial plots we seek the appropriate spatial statistics tools to model these distributions.

In the spatial statistics literature (Cressie and Wikle, 2011), the lack of spatial structure is commonly assessed using the concept of Complete Spatial Randomness (CSR). CSR considers that the points on a map (locations
of tweets in our context) follow a homogeneous Poisson point process. This implies that the numbers of tweets in non-overlapping areas in the map are independent and follow a Poisson distribution with some intensity parameter $\lambda$. More precisely, if we denote the number of tweets within an area $A$ as $N(A)$, CSR asserts that $N(A)$ follows a Poisson distribution with mean $\lambda \cdot V(A)$, where $V(A)$ denotes the size of the area $A$. Intuitively, the CSR property asserts that points are “randomly” scattered in an area and are not concentrated in specific locations.

We consider the task of assessing the levels of noise in Twitter data (with respect to the target event detection task) by testing the CSR property for tweets that contain common terms\(^5\). In particular, we initially select a term (say the most frequent term in a collection of tweets) and then we test whether the locations of the tweets that contain this term have the CSR property. In case a term has the CSR property (i.e., the locations of the tweets that contain this term follow a Poisson point process distribution), the edges in the twitter similarity graph that are based on these terms can be considered as noise and may result in the identification of clusters that are not related to events of interest.

\(^5\) The direct usage of the CSR tests for the whole input tweet stream would not be particularly informative since both of our algorithms construct a similarity graph between tweets where the edge weights (i.e., the similarities between tweets) are based on the terms that two tweets have in common. In this case, noise or event-irrelevant tweets would affect the construction of the graph only when two “noise” tweets have a term in common (i.e., resulting in the formation of an edge that connects event-irrelevant tweets in the tweet similarity graph).
In order to evaluate the CSR property we have employed Ripley’s $K$-function (Cressie and Wikle, 2011), which is a commonly used measure for assessing the proximity of a spatial distribution to a homogeneous Poisson point process. The sample-based estimate of Ripley’s $K$-function is defined as

$$ \hat{K}(s) = V(A) \sum_{i \neq j} N(d_{ij} < s) / n^2 $$

for a given distance value $s$, where $d_{ij}$ denotes the Euclidean distance between two sample points $i$ and $j$ (two tweets in our context) in the space, $N(d_{ij} < s)$ counts the number of sample pairs that has a distance smaller than $s$, $n$ is the total number of points, and $V(A)$ is the size of the area $A$. It is known that, when a spatial Poisson process is homogeneous, the values of the $K$-function are approximately equal to $\pi s^2$. Thus, the proximity of $\hat{K}(s)$ to $\pi s^2$ can be employed for evaluating how similar our data distribution is to a homogeneous Poisson process. In this paper, we use the standardized $K$-function:

$$ \hat{L}(s) = \sqrt{\frac{\hat{K}(s)}{\pi}} - s $$

and the proximity to a homogeneous Poisson process is measured by the proximity of the values of $\hat{L}(s)$ to 0.

We now assess the spatial distribution of the sets of tweets shown in Fig. 5 (tweets containing the terms “nyc”, “love”, “lol” and “night”). Specifically, we illustrate in Fig. 6 the values of their standardized $K$-function for different values of $s$ (distances) up to 4km, depicted in the black lines. Moreover, we simulate (2000 times) a homogeneous Poisson process and compute the maximum and minimum values for $\hat{L}(s)$, depicted in the blue and red dashed lines, respectively. We can observe that, the values of $\hat{L}(s)$ obtained using the locations of these tweets are close to, and in several cases within the ranges of, the values of $\hat{L}(s)$ obtained from the simulated homogeneous Poisson processes. This indicates that these tweets are slightly more concentrated in space than what a homogeneous Poisson process would produce (possibly due to the differences in the concentration of twitter users in different areas in middle and lower Manhattan), but their spatial distribution is still close to a homogeneous Poisson process.

To further explain what we mean by “still close to a homogeneous Poisson process”, let us consider what appears to be one of the most extreme differences between the spatial distribution of tweets and a homogeneous Poisson process in Fig. 6, which is the value $\hat{L}(s) = 0.19$ that is achieved for a distance value $s = 1$km for the term “nyc”. Based on the number of tweets that contain the term “nyc” on 22-01-2012 (in middle and lower Manhattan), a homogeneous Poisson process would require an intensity parameter $\lambda = 7.93$ per square kilometer to generate the same number of tweets. This would mean that on average, the number of tweets per square kilometer that contain the term “nyc” should be 7.93. In our case, the value of $\hat{L}(s) = 0.19$ for $s = 1$km means that, for small distances, the actual concentration of tweets is slightly higher, with an intensity parameter $\lambda = 11.21$ per square kilometer. This shows that, even in this worst case, the spatial distribution of tweets is still not far from a homogeneous Poisson process.
In order to evaluate whether our observation for the four specific terms holds for a larger tweet collection, we analyze all the geo-located tweets from the New York area for the duration between 01-11-2011 and 01-04-2013. Specifically, for each day, we have retrieved the top-ten frequent terms, and for each frequent term we have computed the sample-based estimates of $\hat{L}(s)$ for $s$ from 0.1km to 1.2km, again focusing on the middle and lower Manhattan area. To avoid cases where the number of samples is low, we have computed the values of $\hat{L}(s)$ only when the number of tweets in middle and lower Manhattan is larger than 100. The results are presented in the boxplot of Fig. 7, which illustrates the mean, the variance and the range of the values of $\hat{L}(s)$ (around 5000 values in total, ten for each of the 500 days), for different values of $s$. As we can see, the boxplot in Fig. 7 illustrates that the most frequent terms in our Twitter data do not have a strong spatial pattern and follow a distribution
that is close to a homogeneous Poisson process, only exhibiting slightly higher tweet concentrations for small distances.

5.2 Temporal distribution of noise in Twitter data

In order to analyze the temporal pattern of the noise in Twitter data, we have assessed whether the distribution of the timestamps of event-irrelevant tweets is close to a uniform distribution. A uniform distribution of the timestamps can serve as a strong indication that these tweets are not relevant to an event that takes place in a confined time interval. In order to test this hypothesis, we have collected the timestamps of the top-ten frequent terms of each day between 01-11-2011 and 01-04-2013. We focus our analysis on a 6-hour interval between 11am and 5pm. For this time interval we tested whether the timestamps of tweets that contain a specific frequent term follow a uniform distribution, using the Chi-squared goodness of fit test. Interestingly, we could reject the null hypothesis that the timestamps are uniformly distributed at a 5% confidence level only in 27% of the cases. This result suggests that a large number of frequent terms in our data does not have a strong temporal pattern.

In summary, the spatiotemporal analysis of the distribution of the noise in Twitter data presented in this section allows us to (i) conduct synthetic experiments with simulated noisy data that help us understand the behavior of the event detection algorithms under different space-time parameters, and (ii) consider a term-filtering mechanism that removes tweets that contain the terms with low values for $\hat{L}(s)$. We will describe both aspects in more details in the next section.
6 Synthetic experiments

In order to better understand the behavior of the event detection algorithms LED and MED, and the potential influence of the noise in the data, we present in this section experimental results based on synthetic data. Specifically, we generate artificial documents that are considered as “tweets” posted at different time instants and diverse spatial locations. By creating some artificial “events” in this setting, we are able to evaluate quantitatively the performances of the proposed methods under different choices for the parameter values. In what follows, we first explain the experimental setup, and then present the event detection results.

6.1 Experimental setup

We work with a spatial area of 10 by 10, which are defined by bottom left and top right coordinates (0, 0) and (10, 10) respectively in a 2-D Euclidean space, and a temporal interval of (0, 32) on the real line. We then define events that span different spatial areas and temporal intervals in diverse experimental settings. First, for each event, we choose a number between 3 and 10 uniformly at random as the number of tweets related to that event. These event-relevant tweets are uniformly distributed in the spatial area and temporal interval spanned by that event. We also generate, based on the spatiotemporal analysis presented in Sect. 5, event-irrelevant tweets, namely, noise, which follows a 2-D Poisson point process in the whole spatial area and are distributed uniformly in the whole temporal interval. Next, the content of each tweet is generated as follows. We take geo-located tweets from New York collected on a random day (in this case 21-01-2012) as a reference, and choose 59 terms as event-relevant terms (referred to as signal terms) and consider all the other terms that appear in the tweets on that day as noise (referred to as noise terms). We select the number of terms in each event-relevant tweet uniformly at random between 5 and 10. In particular, in each event-relevant tweet, one term is selected uniformly at random from the 59 signal terms, and the rest are randomly chosen from the noise terms with probabilities that depend on their numbers of occurrences in the actual daily tweets. We also create event-irrelevant tweets, and the number of terms in each event-irrelevant tweet is selected uniformly at random between 3 and 10. The terms in each event-irrelevant tweet are only chosen from the noise terms. We present event detection results in the following scenarios.

6.2 Event detection results in synthetic data

6.2.1 Events concentrated in both time and space without noise

In a first scenario, we consider 20 events, each of which is concentrated in a 2 by 2 spatial area and a temporal interval of 2. The spatial and temporal loca-
tions are chosen uniformly at random in the whole spatial area and temporal interval. We only consider event-relevant tweets, where the goal is to detect the 20 clusters that correspond to the events by clustering the tweets into different subsets. For MED, we focus on terms that appear in at least 3 tweets. We choose $n_{\text{scale}} = 4$ unless its upper bound $\lceil \min(\log_2 l_d, \log_2 l_t) \rceil$ goes below 4 due to the increase of resolution parameters. In our experiments, we take the same value for the four parameters in the two methods, namely $T_t$ and $T_d$ in LED and $\Delta t$ and $\Delta d$ in MED, and evaluate the clustering performance in terms of Normalized Mutual Information (NMI) and F-measure (Manning et al, 2008). The F-measure is computed using a choice of $\beta = 2$ meaning that it is slightly in favor of recall, as we consider that it is more important to ensure that tweets related to the same event are grouped into one cluster. The results obtained by averaging 10 test runs are shown in Fig. 8. As we can see, in terms of both evaluation criteria, the performance of LED with small values of the thresholds $T_t$ and $T_d$ is not satisfactory as it is not able to capture the links between all the tweets within the same event. However, the performance increases noticeably as the temporal and spatial thresholds are chosen to be close to or larger than the “true” scales of the events (2 in this case for both time and space). When the thresholds get too large, the performance drops slightly, as the chance of grouping two different events together in one cluster increases. Compared to LED, MED achieves much better performance even when the resolution parameters $\Delta t$ and $\Delta d$ are small. The reason is that, even at very fine initial resolutions, the wavelet-based representation in MED is able to aggregate the time series appropriately such that the similarity of the time series is actually computed at a coarser scale. This suggests that MED is better at capturing the links between tweets corresponding to the same event, even with suboptimal choices for the value of the parameters. Therefore, the performance of MED is much less sensitive to parameter selection than that of LED.

6.2.2 Events concentrated in only one dimension without noise

We now consider events that are not necessarily concentrated in both time and space but only in one of the two dimensions. Specifically, we consider 20 events, where 10 of them are concentrated in a temporal interval of length between 1 to 2 but spread in a spatial area with a size from 8 by 8 to 16 by 16. The other 10 events are concentrated in a spatial area with a size from 1 by 1 to 2 by 2 but spread in a temporal interval of length between 8 to 16. We still consider a noise-free scenario as in the previous experiment. The clustering results are shown in Fig. 9. We see that, while MED can handle the scale changes in this scenario with a performance that remains comparable to that in the previous experiment, the performance of LED drops significantly. Specifically, due to the lack of a single temporal and spatial scale for all the events, LED only performs reasonably well when the threshold values for $T_t$
and \(T_d\) are large enough to cover the scales of all the events. This experiment highlights the advantage of MED in handling events of different scales and in the absence of simultaneous temporal and spatial localization.

### 6.2.3 Events concentrated in both time and space with noise

We now move to noisy scenarios where we also consider event-irrelevant tweets in addition to event-relevant tweets. Specifically, we generate event-irrelevant tweets that follow a 2-D Poisson point process with an intensity parameter \(\lambda = 10\) within the whole spatial area of 10 by 10. This generates around 1000 noise tweets in addition to the tweets that correspond to 20 events generated as in Sect. 6.2.1. The goal is to detect the events by applying clustering to all the tweets in the dataset. To measure the clustering quality, we define the groundtruth to be a combination of 20 event clusters and noise clusters where each noise tweet is considered as a single cluster. The reason for this setting is that we wish to group tweets that correspond to the same event, and at the same time we want to ensure that the noise tweets remain as separated as possible. Based on the analysis in Sect. 5, for MED, we propose to evaluate
the values of the standardized $K$-function $\hat{L}(s)$ for all the terms that appear in at least 3 tweets for $s$ chosen to be 0.5, 1, 1.5 and 2, and only consider terms that have an average $\hat{L}(s)$ value no smaller than 1 as valid terms for generating keyword time series. The clustering results are shown in Fig. 10. In the noisy scenario, we see that the \textit{NMI} and \textit{F-measure} curves show different trends. Specifically, with small values for the threshold or resolution parameters, the number of links between tweets created by both methods is small, and most of the noise clusters remain well-separated. When the parameter values increase, noise tweets starting forming more links to event-relevant tweets as well as to themselves, which penalizes the clustering. Therefore, we see that the \textit{NMI} curves show an almost monotonically decreasing trend as the parameter values increase. In contrast, the \textit{F-measure} is a weighted combination of precision and recall, which penalizes both false positives and false negatives. Therefore, for both methods, we see that the \textit{F-measure} curves initially increase as the parameter values increase (where the number of false negatives generally decreases), and decrease as these parameters become large (where the number of false positives increases).

We now compare the performance of \textit{LED} and \textit{MED} in the same experiment. For \textit{NMI}, we see that the performance of \textit{LED} drops significantly when the thresholds exceed the “true” scales of the events, as large thresholds in \textit{LED} tend to increase the number of event-relevant and noise tweets that are linked to each others. In comparison, the performance of \textit{MED} is relatively more stable, which is partly due to the term-filtering procedure employed. Similarly, we see that \textit{MED} outperforms \textit{LED} for a large range of parameter values in terms of the \textit{F-measure}. In addition, the performance of \textit{MED} is again more stable in the sense that it peaks at a wider range of parameter values, while \textit{LED} only performs well when the threshold values are chosen at the “true” event scales.

6.2.4 Events concentrated in only one dimension with noise

Finally, we show in Fig. 11 the experimental results in a noisy scenario where the events are concentrated either in time or space as defined in Sect. 6.2.2. While the \textit{NMI} curves are similar to those in Fig. 10, the \textit{F-measure} curves show that the performance of both methods drops significantly in this challenging scenario. Still, \textit{MED} outperforms \textit{LED} in terms of both peak performance and stability.

6.2.5 Influence of parameter settings

We now take a closer look at the parameter settings for the synthetic experiments. Especially, we investigate how the length of the temporal interval, the size of the spatial area, and the number of signal terms in each event-relevant tweet, influence the performance of both algorithms in terms of the \textit{F-measure} in the scenario of Sect. 6.2.4, that is, the performance curve in Fig. 11(b)).
First, given a fixed parameter for the Poisson point process and fixed spatial area of 10 by 10, the total number of noise tweets remains the same. In this case, we observe that the performance of both algorithms has improved when the temporal interval increases from 32 to 128 (Fig. 12(a)), due to decreased noise density in the temporal dimension hence a higher signal-to-noise-ratio. Such a gain is more dramatic for LED especially at large parameter values, in which case the performance of this approach is more sensitive to the density of the noisy information.

Second, given a fixed temporal interval of 32, as the spatial area increases from 10 by 10 to 16 by 16, the total number of noise tweets increases quadratically. In this case, we see from Fig. 12(b) that the performance of both algorithms decreases mainly because of that, as the total number of noise tweets increases, generally more links are formed between noise tweets.

Finally, we have investigated the influence of the number of signal terms in each event-relevant tweet on the performance of the algorithms. Specifically, we increase the number of signal terms from 1 to 3 in each event-relevant tweet and repeat the same experiments. We have observed performance gain...
Fig. 12 Clustering performance in terms of F-measure on events concentrated in only one dimension with noise. (a) The same setting as Sect. 6.2.4. but with a temporal interval of 128. (b) The same setting as Sect. 6.2.4. but with a spatial area of 16 by 16. (c) The same setting as Sect. 6.2.4. but with 3 signal terms in each event-relevant tweet.

In Fig. 12(c) for both algorithms which matches the intuition that a higher signal-to-noise-ratio generally leads to better performance.

In summary, the synthetic experiments suggest that LED is efficient at detecting events that are concentrated in both time and space, provided that these events are of similar scales and that the correct temporal and spatial thresholds are chosen in the algorithm. In comparison, although we employed a term-filtering procedure in MED in the noisy scenarios, the results on synthetic data generally suggest that MED is better than LED at detecting events of different scales and in the absence of simultaneous temporal and spatial localization. MED is also less sensitive to parameter selection and leads to more robust and stable event detection performance.

7 Real world experiments

We now test the performance of LED and MED in real world event detection tasks. We focus in this section on the comparison between these two event detection methods, since (i) such a comparison would highlight the difference between LED and MED in detecting real world events of various temporal and spatial scales, and (ii) to the best of our knowledge, there is no other multiscale method in the literature that is dedicated to event detection. We first describe the data and some implementation details, and then present the event detection results. Finally, we discuss about the scalability of the proposed algorithm.

7.1 Data description

We have collected geotagged public tweets in the New York area, which corresponds to a geographical bounding box with bottom left GPS coordinates pair (40.4957, -74.2557) and top right coordinates pair (40.9176, -73.6895), from November 2011 to March 2013. The streams of public tweets are retrieved using Twitter’s official Streaming API with the “locations” request
parameter\textsuperscript{7}. After the initial retrieval, we filter out those tweets that have no geotags or have geotags outside the predefined bounding box. This results in 16449769 geotagged tweets in total. As a pre-processing step, we remove those tweets that contain a clear location indicator, such as the ones corresponding to Foursquare check-ins, which we do not consider as events of interest.

7.2 Implementation details

We implement both event detection algorithms LED and MED on a daily basis, that is, we aim at detecting events from each day. The \textit{tf-idf} weighting scheme in the vector space model is implemented using the Text to Matrix Generator (TMG) MATLAB toolbox (Zeimpekis and Gallopoulos, 2006), where we also remove a list of stop words provided by the toolbox (with an additional one “http”), and set the minimum and maximum length of a valid term to be 3 and 30.

For LED, we use a temporal threshold of $T_t = 30$ minutes and spatial threshold of $T_d \approx 100$ meters (difference of 0.001 in latitude or 0.0015 in longitude) in Eq. (1) for the detection of local event clusters. For MED, we focus on terms that appear in at least 5 tweets. We evaluate the values of the standardized $K$-function $\hat{L}(s)$ for all these terms with $s$ chosen to be 0.2, 0.4, 0.6, 0.8 and 1, and only consider those that have an average $\hat{L}(s)$ value no smaller than 0.5 as valid terms for generating keyword time series. The initial temporal and spatial resolutions in MED are set to $\Delta t = 30$ minutes and $\Delta d = 100$ meters, and the number of spatial scales is set to $n_{\text{scales}} = 4$. Once the clusters are obtained by both methods, we perform simple post-processing steps that (i) remove clusters that contain less than 3 tweets or less than 3 distinct users, so that each event would contain sufficient information from sufficient number of observes, and (ii) remove clusters in which more than 50\% of the tweets come from a single user, so that the information source is sufficiently diverse, and finally (iii) remove clusters that correspond to job advertisements and traffic alerts posted by bots. While there is no general rule for such post-processing, we found these steps practical to remove clusters that are not meaningful and correspond to noisy information.

7.3 Event detection results

We now analyze the clustering results for both LED and MED algorithms. First of all, the clusters detected by LED do correspond to meaningful real world events of interest. For example, Table 1 shows some example local clusters obtained that correspond to several protests during the Occupy Wall Street (OWS) movement\textsuperscript{8} in New York City. To understand better the behavior of LED, we take 2011-11-17 as an example date, when many OWS

\textsuperscript{7} https://dev.twitter.com/streaming/overview/request-parameters#locations
\textsuperscript{8} http://en.wikipedia.org/wiki/Occupy_Wall_Street
protests took place. We first show in Fig. 13 all the 41 local event clusters detected on this date in middle and lower Manhattan, where different clusters are shown in different colors. Detailed information about the top 20 clusters are further shown in Table 2, where the six columns correspond to cluster id, median timestamp (GMT+0) of all the tweets in the cluster, minimal time interval (in seconds) that covers 80% of the tweets, mean latitude and longitude of the tweets, and (up to 10) top terms contained in the cluster. As we can see in Fig. 13 and in the third column of Table 2, all the clusters are highly localized in both time and space. In addition, due to the strict temporal and spatial constraints used by LED (see Eq. (1)), for the same event we get separate clusters, which correspond to different timestamps (such as clusters 2 and 5 that talk about protests at Zuccotti Park) or different locations (such as clusters 3 and 13 that talk about protests at Union Square). Ideally, we would like some of these separated clusters to be grouped together if they are related to the same real world event.

We now present the event detection results on data from the same date using MED. Table 3 summarizes the top 10 clusters detected by MED, four of which are visualized on the map in Fig. 14. From Fig. 14 and the third column of Table 3, we see that MED is able to detect events that spread in much larger spatial areas or longer time intervals than LED. Specifically, we see in Fig. 14(a) and Fig. 14(b) two clusters related to OWS protests at Zuccotti Park (cluster 1), and Union Square and Foley Square (cluster 2), respectively, both of which span rather long time intervals. Moreover, although most of the tweets in the two clusters are mainly posted from locations where the protests took place, there also exist tweets in the clusters that mention

### Table 1: Example local clusters detected by LED that correspond to protests in the OWS movement.

| Date       | Event                                                                 | Example Tweets                                                                                   | Tweet IDs            |
|------------|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------|
| 2011-11-15 | At about 5am, NYPD began to clear Zuccotti Park.                       | Lines of NYPD circulating inside park. Stand here, don’t stand there etc. outside perimeter lined by hot police. OWS | 1365938908950043905  |
| 2011-11-17 | More than 30,000 demonstrated in and around Zuccoti Park, Union Square, Foley Square, the Brooklyn Bridge, and other locations through the city. | Mostly media, police right now in Zuccotti Park. We need more numbers. Get down here. OWS vs 2017. Occupy wall street is occupying union square. As long as I can get home on the subway later chant on. March stretches from Brooklyn Bridge all the way back to Foley Square. Thousands lined up down Centre Street on way to bridge mil7. OWS. | 13726864529752929  |
| 2012-01-01 | New York police arrested 68 Occupy Wall Street protesters after they moved back in Zuccotti Park where the movement began last year. | Arrests happening now in Zuccotti Park now. OWS. | 153136159826050353  |
| 2012-01-03 | Approximately 200 Occupy protesters perform a flash mob at the main concourse of New York’s Grand Central Terminal. | @Occupy_forums protest in Grand Central @NewYork @NYC | 1543337390235379776  |
| 2012-03-17 | Occupy Wall Street demonstrators attempted to reoccupy Zuccotti Park to mark the movement’s six month anniversary. | Haven't seen Zuccotti like this in months. Some irritation by protesters but police seem tense today. Two MET. OWS. | 1801082983598550560  |
the same events but have been posted at quite distant locations. In Fig. 14(c) and Fig. 14(d), we see two clusters corresponding to the Raise Cache tech event (cluster 5) and the Mastercard free lunch promotion event (cluster 9), respectively, both of which are more concentrated in time but spread in space (with a few outliers in the latter case). Although there exists certain amount of noise tweets in the detected clusters, these examples demonstrate that MED is able to detect events that concentrate only in time or space, many of which are of different scales. In comparison, LED is not able to detect such event clusters. Specifically, LED produced many separated clusters for OWS protests, two separated clusters with some missing tweets for the Raise Cache tech event, and missed completely the Mastercard promotion event due to the lack of a group of tweets that are concentrated in both time and space.

Finally, we notice that even in the results obtained by MED there sometimes exists more than one cluster about the same event, for example, in Table 3 there are two clusters detected for both the OWS protests (clusters 1 and 2) and the Katy Perry concert (clusters 3 and 4). First, the protests at Zuccotti
Table 2: Detailed information about the top 20 local event clusters detected by LED on 2011-11-17. The six columns from the left to the right correspond to cluster id, median timestamp (GMT+0) of all the tweets in the cluster, minimal time interval (in seconds) that covers 80% of the tweets, mean latitude and longitude of the tweets, and (up to 10) top terms contained in the cluster.

| Cluster Id | Median Timestamp (GMT+0) | Minimal Time Interval | Mean Latitude and Longitude | Top Terms |
|------------|--------------------------|-----------------------|-----------------------------|-----------|
| 1  | 17-Nov-2011 21:21:56 | 2639 | 40.7145 74.0032 | nyc, occupy, join, occupy wall street, occupy, occupy wall street, building |
| 2  | 17-Nov-2011 22:12:23 | 3714 | 40.7093 74.0111 | st, occupy, occupy wall street, occupy wall street, occupy wall street, building |
| 3  | 17-Nov-2011 22:44:03 | 3426 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 4  | 17-Nov-2011 23:15:59 | 2490 | 40.7434 73.9937 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 5  | 17-Nov-2011 23:16:18 | 2494 | 40.7093 74.0111 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 6  | 17-Nov-2011 23:16:19 | 2494 | 40.7093 74.0111 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 7  | 17-Nov-2011 23:49:31 | 3224 | 40.7504 73.9923 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 8  | 17-Nov-2011 23:49:31 | 3224 | 40.7504 73.9923 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 9  | 17-Nov-2011 23:49:31 | 3224 | 40.7504 73.9923 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 10 | 17-Nov-2011 23:49:31 | 3224 | 40.7504 73.9923 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |

Table 3: Detailed information about the top 10 event clusters detected by MED on 2011-11-17. The six columns from the left to the right correspond to cluster id, median timestamp (GMT+0) of all the tweets in the cluster, minimal time interval (in seconds) that covers 80% of the tweets, mean latitude and longitude of the tweets, and (up to 10) top terms contained in the cluster.

| Cluster Id | Median Timestamp (GMT+0) | Minimal Time Interval | Mean Latitude and Longitude | Top Terms |
|------------|--------------------------|-----------------------|-----------------------------|-----------|
| 1  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 2  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 3  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 4  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 5  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 6  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 7  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 8  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 9  | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |
| 10 | 17-Nov-2011 23:25:35 | 2052 | 40.7561 74.0932 | occupy, occupy wall street, occupy wall street, occupy wall street, occupy wall street, building |

Park took place from the morning to noon, while the protests at Union Square and Foley Square happened in the afternoon after 3pm. Although there indeed exist semantic links between tweets that correspond to these two events, the rather different locations and timestamps lead to separate clusters. Second, for the Katy Perry concert, the two clusters highly overlap in both time and space, and the tweets in one cluster have quite strong links to those in the other one. In this case, clusters have been separated mainly because of the strong patterns present in the texts: While in cluster 3 the concert is described mostly using a single term “katy Perry”, in cluster 4 we see two separate terms “katy” and “perry”.
Fig. 14 Example event clusters detected by MED on 2011-11-17: (a) OWS protests at Zuccotti Park (cluster 1). (b) OWS protests at Union Square and Foley Square (cluster 2). (c) Raise Cache (cluster 5). (d) Mastercard free lunch promotion (cluster 9).

7.4 Scalability

The computational complexity of both LED and MED mainly depend on (i) the construction of a similarity graph, and (ii) the graph-based clustering process. As we mentioned before, the Louvain method used in the clustering process is empirically observed to be able to scale to large scale graphs. Therefore, we mainly discuss the computational cost of constructing similarity graphs in the two algorithms.

For both LED and MED, the construction of a similarity graph can be performed efficiently because the similarities need to be computed only for pairs of tweets that have common terms. Thus, the computational complexity of the similarity graph construction, using an appropriate index structure (such as an inverted index), can be $O(n \times \text{avg}\_\text{connectivity})$, where $n$ is the total number of tweets and \text{avg}\_\text{connectivity} denotes, given a tweet $t$, the average number of tweets in the dataset with non-zero similarity with $t$. In our real world experiment, \text{avg}\_\text{connectivity} corresponds to only 2% of the total number of tweets.

In addition, \text{avg}\_\text{connectivity} can be further reduced by the term-filtering procedure that is employed in MED for noise-filtering. Since term-filtering is applied to the most popular (frequent) terms, this can substantially affect
avg_connectivity. In our experiment, for example, after the filtering procedure avg_connectivity is further reduced by more than 40% compared to LED. Moreover, the filtering procedure potentially represents a tradeoff between the performance of the algorithm and its computational complexity. A more aggressive filtering can largely attenuate the influence of noisy information and at the same time reduce the computational cost. However, it might also filter out terms that are related to some relatively small-scaled events.

For MED, we need to compute the spatiotemporal similarity of time series for the valid terms (after term-filtering) shared by every pair of tweets. However, since the spatiotemporal similarity is defined between time series that come from different geographical cells, we only need to evaluate, for each valid term, the pairwise similarity between time series from different cells, instead of comparing every pair of different tweets containing that term. This keeps the number of DWT computations needed relatively low due to the small number of geographical cells.

Practically, for the daily Twitter stream with geotag in the middle and lower Manhattan area of New York City that we have considered in the experiment (~8000 geotagged tweets with 36000 terms in total), it takes only a few seconds to finish the construction of the similarity graph in LED. For the implementation of MED, it takes roughly 5 minutes for our MATLAB code to create the similarity graph on a lab server with average computing power or 8 minutes on a mid-2009 MacBook Pro (both single core process), where the main computational cost is due to the DWT computations. While we consider this computation time reasonable given the benefits of the algorithm, we certainly hope to further improve the scalability of our algorithm in future work.

8 Related work

Social media data have become pervasive due to the fast development of online social networks since the last decade. This has given rise to a series of interesting research problems such as event detection based on user-generated content (Sayyadi et al., 2009; Becker et al., 2009; Aggarwal and Subbian, 2012). As an example, Chen and Roy (2009) and Papadopoulos et al (2011) have proposed to detect social events using tagged photos in Flickr. A more popular platform is Twitter, which has attracted a significant amount of interest due to the rich user-generated text data that can be used for event detection (Atefeh and Khreich, 2013). Early works in the field have focused on more specific types of events, such as news (Sankaranarayanan et al., 2009) and earthquakes (Sakaki et al., 2010), while recent approaches detect various types of events (Petrovic et al., 2010; Marcus et al., 2011; Becker et al., 2011; Ozdikis et al., 2012; Li et al., 2012a; Parikh and Karlapalem, 2013; Berlingerio et al., 2013). Although the specific techniques presented in the state-of-the-art event detection approaches may vary from a technical point of view, many of them rely on the detection of certain behaviors in the Twitter stream such as the burstiness of certain
keywords, which indicates the emergence of particular events. In particular, several works use wavelets, which is a well-developed tool in signal processing, for event detection based on keyword burstiness patterns (Weng and Lee, 2011; Cordeiro, 2012).

Recently, there has been an increasing amount of interest in exploring both the temporal and spatial dimensions to better capture the meaningful information and reduce noise in the data from social media platforms. In Rattenbury et al (2007), the authors have proposed to analyze for event extraction the semantics of tags associated with the Flickr photos, by taking into account multiple temporal and spatial resolutions. In Chen and Roy (2009), the authors have proposed to cluster Flickr photos based on both the temporal and the spatial distributions of the photo tags using wavelets. In Becker et al (2010), the authors have considered combining text, temporal and spatial features in order to build an appropriate tweet similarity measure. In Lappas et al (2012), the authors have proposed two approaches to detect burstiness of keywords in both temporal and spatial dimensions simultaneously. In Sugitani et al (2013), the authors have proposed a hierarchical clustering procedure for event detection in Twitter, where both temporal and spatial constraints have been imposed to measure the similarities of tweets. They have also proposed to examine co-occurrences of keywords that present specific spatiotemporal patterns. Other examples include Lee et al (2011); Li et al (2012b); Thom et al (2012); Walther and Kaisser (2013) and Zaharieva et al (2013), where the authors have proposed spatiotemporal clustering methods for anomaly and event detection in Twitter and Flickr, respectively. These approaches are certainly inspirational to the idea proposed in the present paper; However, most of them do not explicitly handle multiple spatiotemporal scales in event detection.

Finally, there are a few approaches in the literature that have studied the influence of different resolutions for temporal and spatial analysis in event detection. For example, in Cooper et al (2005) and Rattenbury et al (2007), the authors have proposed to use a scale-space analysis of the data (Witkin, 1983). The common objective in these approaches is to select the most appropriate scale for event extraction and detection. More generally, multiscale or multiresolution clustering algorithms has been of interest in the machine learning, pattern recognition, and physics (Ronhovde et al, 2011, 2012) communities since the last decade. The approaches that take advantage of the properties of the wavelet transform to enable a multiresolution interpretation in the clustering process, such as the works in Sheikholeslami et al (2000) and Tremblay and Borgan (2012), are of particular interest. Although these approaches are not originally proposed for event detection in social media platforms, they have inspired us to consider wavelets in our framework. While they output multiple sets of clustering solutions at different resolutions, our approach however uses wavelets to choose the appropriate temporal and spatial resolutions for constructing a single data similarity graph.

In summary, although there exist many approaches that take into account the temporal and spatial dimensions of the social media data for event detection, they generally do not explicitly handle different scales in data analysis.
In contrast, our framework explicitly handles multiple spatiotemporal scales, which we believe is essential for building an efficient and generic event detection approach. Different scales in the temporal and spatial dimensions have been treated separately in most of the state-of-the-art analyses, but the relationship and interaction between these scales have been largely overlooked in the literature. To the best of our knowledge, our approach is the first attempt that is based on an explicit modeling of the relationship between different temporal and spatial resolutions. Finally, we present a statistical analysis of the temporal and spatial distributions of noisy information in the Twitter data, which we believe is the first of its kind. We believe our perspective contributes to the research in the field of social media analytics and provides new insights into the design of novel clustering and event detection algorithms.

9 Conclusion

In this paper, we have proposed a novel approach towards multiscale event detection in social media. Especially, we have shown that it is important to understand and model the relationship between the temporal and spatial scales, so that events of different scales can be separated simultaneously and in a meaningful way. Furthermore, we have presented statistical modeling and analysis about the spatiotemporal distributions of noisy information in the Twitter stream, which not only helps us define a novel term-filtering procedure for the proposed approach, but also provides new insights into the understanding of the influence of noise in the design of event detection algorithms. Future directions include (i) further investigation of the possibility of extending and generalizing the proposed scale relationship model to handle temporal and spatial scales simultaneously for multiscale event detection, (ii) more appropriate and accurate statistical models for analyzing noisy information present in social media data, and (iii) improvement on the scalability of the proposed algorithms.

References

Aggarwal CC, Subbian K (2012) Event Detection in Social Streams. In: SIAM International Conference on Data Mining (SDM), Anaheim, CA
Atefeh F, Khreich W (2013) A Survey of Techniques for Event Detection in Twitter. Computational Intelligence
Becker H, Naaman M, Gravano L (2009) Event Identification in Social Media. In: ACM SIGMOD Workshop on the Web and Databases (WebDB), Providence, RI
Becker H, Naaman M, Gravano L (2010) Learning Similarity Metrics for Event Identification in Social Media. In: The Third ACM International Conference on Web Search and Data Mining (WSDM), New York City, NY
Becker H, Naaman M, Gravano L (2011) Beyond Trending Topics: Real-World Event Identification on Twitter. In: The Fifth International AAAI Conference on Weblogs and Social Media (ICWSM), Barcelona
Berlingerio M, Calabrese F, Lorenzo GD, Dong X, Gkoufas Y, Mavroeidis D (2013) SaferCity: a System for Detecting and Analyzing Incidents from Social Media. In: IEEE International Conference on Data Mining (ICDM), Dallas, TX
Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment (10):P10,008 (12pp)
Chen L, Roy A (2009) Event Detection from Flickr Data through Wavelet-Based Spatial Analysis. In: The 18th ACM Conference on Information and Knowledge Management (CIKM), Hong Kong
Cooper M, Foote J, Girgensohn A, Wilcox L (2005) Temporal Event Clustering for Digital Photo Collections. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 1(3):269–288
Cordeiro M (2012) Twitter Event Detection: Combining Wavelet Analysis and Topic Inference Summarization. In: Doctoral Symposium on Informatics Engineering, Porto
Cressie N, Wikle CK (2011) Statistics for Spatio-Temporal Data (Wiley Series in Probability and Statistics). Wiley
Daubechies I (1992) Ten Lectures on Wavelets. SIAM
Lappas T, Vieira MR, Gunopulos D, Tsotras VJ (2012) On the Spatiotemporal Burstiness of Terms. In: The 38th International Conference on Very Large Databases, Istanbul
Lee CH, Yang HC, Chien TF, Wen WS (2011) A Novel Approach for Event Detection by Mining Spatio-temporal Information on Microblogs. In: International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Kaohsiung
Li C, Sun A, Datta A (2012a) Twevent: Segment-based Event Detection from Tweets. In: The 21st ACM International Conference on Information and Knowledge Management (CIKM), Maui, HI
Li R, Lei KH, Khadiwala R, Chang KCC (2012b) TEDAS: A Twitter-based Event Detection and Analysis System. In: The 28th IEEE International Conference on Data Engineering (ICDE), Washington, DC
von Luxburg U (2007) A Tutorial on Spectral Clustering. Statistics and Computing 17(4):395–416
Manning CD, Raghavan P, Schütze H (2008) Introduction to Information Retrieval. Cambridge University Press
Marcus A, Bernstein MS, Badar O, Karger DR, Madden S, Miller RC (2011) Twitinfo: Aggregating and Visualizing Microblogs for Event Exploration. In: ACM CHI Conference on Human Factors in Computing Systems, Vancouver
Newman MEJ (2006) Modularity and community structure in networks. Proceedings of the National Academy of Sciences of the USA 103(23):8577–8582
Ozdikis O, Senkul P, Oguztuzun H (2012) Semantic Expansion of Hashtags for Enhanced Event Detection in Twitter. In: The First International Workshop
on Online Social Systems (WOSS), Istanbul
Papadopoulos S, Zigkolis C, Kompatsiaris Y, Vakali A (2011) Cluster-Based
Landmark and Event Detection for Tagged Photo Collections. IEEE Multi-
Media 18(1):52–63
Parikh R, Karlapalem K (2013) ET: Events from Tweets. In: The 22nd Inter-
national Conference on World Wide Web (WWW), Rio de Janeiro
Petrovic S, Osborne M, Lavrenko V (2010) Streaming First Story Detection
with application to Twitter. In: The 11th Annual Conference of the North
American Chapter of the Association for Computational Linguistics, Los
Angeles, CA
Rattenbury T, Good N, Naaman M (2007) Towards Automatic Extraction of
Event and Place Semantics from Flickr Tags. In: ACM SIGIR Conference
on Research and Development on Information Retrieval, Amsterdam
Reuter T, Papadopoulos S, Petkos G, Mezaris V, Kompatsiaris Y, Cimiano P,
de Vries C, Geva S (2013) Social Event Detection at MediaEval 2013: Chal-
lenges, Datasets, and Evaluation. In: MediaEval Benchmarking Initiative
for Multimedia Evaluation (MediaEval) 2013 Workshop, Barcelona
Ronhovde P, Chakrabarty S, Hu D, Sahu M, Sahu KK, Kelton KF, Mauro
NA, Nussinov Z (2011) Detecting hidden spatial and spatio-temporal struc-
tures in glasses and complex physical systems by multiresolution network
clustering. The European Physical Journal E 34:105
Ronhovde P, Chakrabarty S, Hu D, Sahu M, Sahu KK, Kelton KF, Mauro
NA, Nussinov Z (2012) Detection of hidden structures for arbitrary scales
in complex physical systems. Scientific Reports 2:329
Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake Shakes Twitter Users:
Real-time Event Detection by Social Sensors. In: The 19th International
Conference on World Wide Web (WWW), Raleigh, NC
Sankaranarayanan J, Samet H, Teitler BE, Lieberman MD, Sperling J (2009)
TwitterStand: News in Tweets. In: The 17th ACM SIGSPATIAL Interna-
tional Conference on Advances in Geographic Information Systems, Seattle,
WA
Sayyadi H, Hurst M, Maykov A (2009) Event Detection and Tracking in Social
Streams. In: The Third International AAAI Conference on Weblogs and
Social Media (ICWSM), San Jose, CA
Sheikholeslami G, Chatterjee S, Zhang A (2000) WaveCluster: A Multi-
Resolution Clustering Approach for Very Large Spatial Databases. The In-
ternational Journal on Very Large Data Bases 8(3-4):289–304
Sugitani T, Shirakawa M, Hara T, Nishio S (2013) Detecting Local Events by
Analyzing Spatiotemporal Locality of Tweets. In: The 27th International
Conference on Advanced Information Networking and Applications Work-
shops (WAINA), Barcelona
Thom D, Bosch H, Koch S, Woerner M, Ertl T (2012) Spatiotemporal
Anomaly Detection through Visual Analysis of Geolocated Twitter Mes-
sages. In: 2012 IEEE Pacific Visualization Symposium (PacificVis), Songdo
Tremblay N, Borgnat P (2012) Multiscale Community Mining in Networks
Using Spectral Graph Wavelets. arXiv:12120689
Walther M, Kaisser M (2013) Geo-spatial Event Detection in the Twitter Stream. In: The 35th European Conference on Information Retrieval (ECIR), Moscow

Weng J, Lee BS (2011) Event Detection in Twitter. In: The Fifth International AAAI Conference on Weblogs and Social Media (ICWSM), Barcelona

Witkin A (1983) Scale Space Filtering. In: International Joint Conference on Artificial Intelligence (IJCAI), Karlsruhe

Zaharieva M, Zeppelzauer M, Breiteneder C (2013) Automated Social Event Detection in Large Photo Collections. In: ACM International Conference on Multimedia Retrieval, Dallas, TX

Zeimpekis D, Gallopoulos E (2006) TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections. In “Grouping Multidimensional Data: Recent Advances in Clustering”, J Kogan, C Nicholas and M Teboulle, eds pp 187–210