Geo-located community detection in Twitter with enhanced fast-greedy optimization of modularity: Case study for Typhoon Haiyan.

Article *in* International Journal of Geographical Information Science · December 2014

DOI: 10.1080/13658816.2014.964247

CITATIONS
52

READS
488

3 authors:

Mohamed Bakillah
Calgary University and Heidelberg University
72 PUBLICATIONS **1,024** CITATIONS

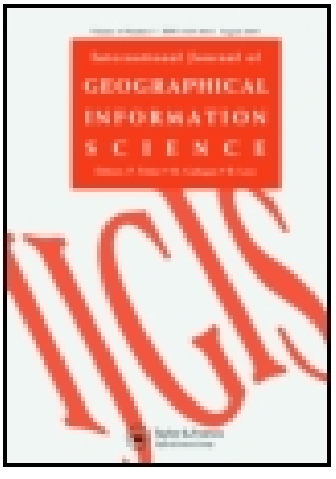
SEE PROFILE

Leah Li
New York University
1 PUBLICATION **52** CITATIONS

SEE PROFILE

Steve Liang
The University of Calgary
81 PUBLICATIONS **1,683** CITATIONS

SEE PROFILE



International Journal of Geographical Information Science

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tgis20

Geo-located community detection in Twitter with enhanced fast-greedy optimization of modularity: the case study of typhoon Haiyan

Mohamed Bakillah[ab], Ren-Yu Li[a] & Steve H.L. Liang[a]

[a] Department of Geomatics Engineering, GeoSensorWeb Lab, University of Calgary, Calgary, Alberta, Canada

[b] GIScience Research Group, Institute of Geography, University of Heidelberg, Heidelberg, Baden-Württemberg, Germany

Published online: 24 Oct 2014.

To cite this article: Mohamed Bakillah, Ren-Yu Li & Steve H.L. Liang (2014): Geo-located community detection in Twitter with enhanced fast-greedy optimization of modularity: the case study of typhoon Haiyan, International Journal of Geographical Information Science, DOI: 10.1080/13658816.2014.964247

To link to this article: http://dx.doi.org/10.1080/13658816.2014.964247

International Journal of Geographical Information Science, 2014
http://dx.doi.org/10.1080/13658816.2014.964247

Taylor & Francis
Taylor & Francis Group

Geo-located community detection in Twitter with enhanced fast-greedy optimization of modularity: the case study of typhoon Haiyan

Mohamed Bakillah[a,b]*, Ren-Yu Li[a] and Steve H.L. Liang[a]

[a]*Department of Geomatics Engineering, GeoSensorWeb Lab, University of Calgary, Calgary, Alberta, Canada;* [b]*GIScience Research Group, Institute of Geography, University of Heidelberg, Heidelberg, Baden-Württemberg, Germany*

(*Received 23 December 2013; final version received 3 September 2014*)

As they increase in popularity, social media are regarded as important sources of information on geographical phenomena. Studies have also shown that people rely on social media to communicate during disasters and emergency situation, and that the exchanged messages can be used to get an insight into the situation. Spatial data mining techniques are one way to extract relevant information from social media. In this article, our aim is to contribute to this field by investigating how graph clustering can be applied to support the detection of geo-located communities in Twitter in disaster situations. For this purpose, we have enhanced the fast-greedy optimization of modularity (FGM) clustering algorithm with semantic similarity so that it can deal with the complex social graphs extracted from Twitter. Then, we have coupled the enhanced FGM with the varied density-based spatial clustering of applications with noise spatial clustering algorithm to obtain spatial clusters at different temporal snapshots. The method was experimented with a case study on typhoon Haiyan in the Philippines, and Twitter's different interaction modes were compared to create the graph of users and to detect communities. The experiments show that communities that are relevant to identify areas where disaster-related incidents were reported can be extracted, and that the enhanced algorithm outperforms the generic one in this task.

Keywords: fast-greedy optimization of modularity; geo-located communities; social media; spatial clustering; Twitter

1. Background and problem statement

Social media attract more users than ever. They are increasingly seen as a source of information on different types of geographical phenomena. A well-known example is the prediction of the time and place of infectious disease outbreaks from the analysis of Twitter messages whose content reveals the occurrence of a particular disease (Culotta 2010). Another representative example is the use of social media for the early detection and monitoring of natural disasters, including earthquakes and typhoons (Sakaki *et al.* 2010, Crooks *et al.* 2013). Vieweg *et al.* (2010) explains that geo-located Twitter messages enable the derivation of 'situational' information (about damages, casualties, etc.) that helps to organize the emergency response in case of disaster. Twitter is even regarded as a distributed sensor system to monitor geographical phenomena (Crooks *et al.* 2013).

The discovery of groups of social network users has several interesting applications in geography (Mooney and Corcoran 2013). For example, the online communities that are formed in response to a natural disaster are a source of information for emergency

*Corresponding author. Email: mohamed.bakillah@geog.uni-heidelberg.de

management personnel who need to understand the situation at the local level. In particular, emergency management personnel must have access to detailed geographic information and know the needs of people in affected areas, whether people at risk have followed evacuation orders, what are the damages to local infrastructure, etc. (Bakillah *et al.* 2007). Studies demonstrate that population affected by disasters use online social connections and communities to seek and disseminate information (Hagar 2009). Shklovski *et al.* (2008) explains that finding communities in social media can help to access relevant information for emergency management at a fine level of detail. A few investigations have been conducted to extract geographic information from social media, especially for geospatial event detection (Sakaki *et al.* 2010, Crooks *et al.* 2013, Walther and Kaisser 2013). In addition, research on the detection of geo-located communities in social networks has been conducted. For example, Expert *et al.* (2011) proposed a method for uncovering space-independent communities in spatial networks. Their research focuses on developing an algorithm that eliminates the effect of space to reveal 'hidden' structural similarities between nodes. Also, Walsh and Pozdnoukhov (2011) explored the spatial structure and dynamics of urban communities through the analysis of traffic on a mobile telephone network. Meanwhile, research on the detection of geo-located communities in social networks is still quite scarce, and the objective of our work is more specific, as it focuses especially on community detection in Twitter's network of users.

While no agreement exists on the definition of communities in social networks, in this article, we refer to communities as 'groups of vertices that are more densely connected to each other than to the rest of the network', as per Papadopoulos *et al.*'s explanation (2012, p. 516) and Murata's definition (2010). In social media, communities can be implicit: they are not necessarily formed in an intentional manner and for a specific purpose or interest. However, they can be discovered by analysing the relationships between members of the network. Therefore, these implicit communities are distinct from 'virtual communities', where 'people with common interests, goals, or practices interact to share information and knowledge, and engage in social interactions' (Chiu *et al.* 2006, p. 1880).

There is a large body of research on community detection in networks (Murata 2010, Wang *et al.* 2010, Bakillah and Liang 2012, Papadopoulos *et al.* 2012, Zhang *et al.* 2012). Typically, the network is represented as a graph $G = (V, E)$, where V and E are sets of vertices and edges respectively, and some graph clustering method is applied. Meanwhile, Twitter's community structure is complex and noisy: some of the shared content and user-created connections are not helpful for detecting meaningful communities; instead, they make the graph structure difficult to analyse and the discovery of communities challenging. It is therefore critical to identify the types of relations between users that are useful to detect communities that will be meaningful for the intended purpose (i.e. obtaining insight into disaster situation). Besides, Twitter messages are associated with a temporal component (the submission time), and geo-location information, available in the form of coordinates. To use Twitter to get an insight into an emergency situation, it is critical that communities be located in space and time. However, most community detections methods in social media or large networks do no incorporate space and time.

The objective of this article is to propose a method for the detection of geo-located communities in Twitter. The method integrates a semantically enhanced version of the fast-greedy optimization of modularity (FGM) graph clustering algorithm (Clauset *et al.* 2004) and a spatial density-based clustering algorithm, varied density-based spatial clustering of applications with noise (VDBSCAN), which detects spatial clusters at

different temporal snapshots. The method was used in a case study on the typhoon Haiyan, which devastated the Philippines in 2013. Using Twitter's different interaction modes, different ways to establish relations between its users were tested to examine which of these relations are more useful to get an insight into the disaster situation. The case study shows that the combination of explicit links between Twitter users and common topics being shared in tweets results in more meaningful communities than if we use common topics only, demonstrating that community detection is more useful than simple text mining of tweets.

The article is organized as follows: Section 2 summarizes the existing methods for community detection and related challenges. Section 3 describes the data set. Section 4 describes the method. Section 5 contains experimental results and visualization of resulting communities. The work and outcomes are summarized in Section 6.

2. Related work and challenges

2.1. Community detection

The problem of community detection consists in identifying groups of objects that are more densely connected to each other than to the rest of the network's objects (Papadopoulos *et al.* 2012). In practice, community detection is considered equivalent to graph clustering (Schaeffer 2007, Fortunato 2010). Several recent classifications of community detection/graph clustering approaches exist (Papadopoulos *et al.* 2012, Xie *et al.* 2013). Papadopoulos *et al.* (2012) identify five categories of methods: (1) cohesive subgraph discovery; (2) vertex clustering; (3) divisive clustering; (4) model-based clustering and (5) community quality optimization.

Cohesive subgraph discovery methods are based on the assumption that communities are characterized by well-defined structural properties, such as cliques (Palla *et al.* 2005) and cores (Batagelj and Zaveršnik 2011). They are usually computationally expensive and based on a very specific definition of community that may exclude more loosely organized communities.

Vertex clustering is inspired from data clustering and uses a distance or similarity between vertices to find those that are closer to each other, such as random-walk-based similarity (Pons and Latapy 2005). Distances can also be calculated in a vector space (Donath and Hoffman 1973, Yang and Liu 2008, De Meo *et al.* 2014). However, different authors disagree on which eigenvectors should be used, and the exact computation of eigenvectors for large-scale data is impossible (Fortunato 2010). Divisive clustering methods find components of the network that lie between communities and remove them until clusters are revealed. Divisive methods exist that focus on removing edges (Girvan and Newman 2002, Fortunato *et al.* 2004, Newman and Girvan 2004) or vertices (Vragovic and Louis 2006). Model-based methods are based on the assumption that communities are formed according to an underlying model, such as a statistical model (Hastings 2006) or a dynamic process (Arenas *et al.* 2006, Reichardt and Bornholdt 2006, Raghavan *et al*. 2007 , Šubelj and Bajec 2014). Community quality optimization aims at finding communities by optimizing some quality metric of clusters, such as conductance (Kannan *et al.* 2004), reachability (Chen *et al.* 2009), connectivity (Rhouma and Romdhane 2014) and modularity (Newman 2004b, Duch and Arenas 2005, Blondel *et al.* 2008). Modularity optimization has given rise to an important body of research (Papadopoulos *et al.* 2012). It is based on the assumption that a random graph does not have a community structure (Newman 2004a). The 'optimal' clustering result is obtained by maximizing the total edge weight difference between the actual graph structure and a random graph structure using a quality function Q. There are several modularity optimization methods to optimize Q, the earliest one being the

greedy optimization of modularity (GM) (Newman 2004a). The GM method assigns each vertex to a cluster and successively merges clusters into larger clusters to maximize the value of Q. Since GM involves many redundant operations, Clauset *et al.* (2004) proposed an enhanced version, called FGM.

Besides Papadopoulos *et al.*'s (2011) classification, authors also distinguish hierarchical clustering, which uncovers nested clusters through agglomerative (bottom-up) algorithms (Fernandez and Gomez 2008) or divisive (top-down) algorithms (Freeman 1977, Li *et al.* 2006). Hierarchical clustering does not require a preliminary knowledge of the cluster size and number of clusters; however, the clustering result may not be adequate when the hierarchical structure of the graph is not obvious, and within our context, we cannot assume that clusters will be hierarchically organized.

The FGM method considers all edges as equal. Consequently, since social networks like Twitter are dense and noisy graphs, results are likely to include communities that are meaningless to our purpose. It is therefore important to analyse the relationships between vertices to determine whether they are pertinent to the types of communities we aim to discover. One way of doing so with Twitter is to compare the clustering results obtained based on the different types of interactions (e.g. follow relations, mentions, tweets, etc.). This is the approach that we have explored in this study.

2.2. *Spatial clustering*

Spatial clustering is the process of grouping objects based on their spatial proximity (Kisilevich *et al.* 2010). Early research on spatial clustering has begun within specific application fields, such as epidemiology (e.g. Glass and Mantel 1969, Smith *et al.* 1976). The geographical analysis machine (GAM) was the first algorithm for automatic detection of clusters (Openshaw *et al.* 1987). It searches through a number of circles across the region of interest. Improvements to the GAM procedure have been proposed until recently (Conley *et al.* 2005, Charlton 2006). Algorithms like the GAM procedure assume that clusters have a regular shape, which is rarely the case. As a result, methods have been developed to detect irregularly shaped clusters, such as k-medoids (Kaufman and Rousueeuw 1990). In this approach, spatially distributed objects are initially divided into k random groups. Then, objects are iteratively exchanged between the initial groups until the quality of the clusters reaches equilibrium and stops improving. Other methods that allow spatial objects to be part of more than one cluster have been developed, such as the fuzzy C-means algorithm, where an object has variable degrees of membership to different clusters (Xu and Wunsch 2005). This method assumes that the total number of clusters is known in advance, whereas this is rarely the case. Another approach is hierarchical clustering, where clusters are formed in a bottom-up fashion: closest pairs of clusters are gradually agglomerated until a preselected number of clusters are generated (Zhang *et al.* 1996, Nanopoulos *et al.* 2001). Some early hierarchical methods, like the implementation of Ward's clustering algorithm, aggregate the discovered clusters until they are all included into a single cluster (Ward 1963, Carvalho *et al.* 2009). The analyst must then choose, *a posteriori*, the number of clusters to determine at which level the algorithm should stop the aggregation process. Among existing clustering approaches, density-based algorithms are more efficient at detecting clusters with varying density (Parimala *et al.* 2001). This is important in our context, since Twitter users reporting on similar incidents vary in number, and we need to be able to capture clusters despite variations in their density. DBSCAN is a well-known implementation of the density-based algorithm (Martin *et al.* 1996). DBSCAN starts by randomly selecting an object O from the data set. A range query centred on O is processed,

with preselected radius. If the density of objects within the radius reaches a preselected threshold, the objects are added to the cluster centred on *O*. This process is repeated until the size of the cluster stabilizes. A review of algorithms that were developed based on DBSCAN is presented in Sahoo (2013). One of the drawbacks of DBSCAN is its heavy reliance on pre-defined thresholds (radius, threshold density). With large data sets, it is difficult to choose an optimal value for these thresholds *a priori*. VDBSCAN is a more recent density-based algorithm that addresses this issue by optimizing the thresholds based on the characteristics of the data set and therefore does not require user input (Liu *et al.* 2007, Rasheduzzaman Chowdhury and Asikur Rahman 2010).

More recently, in addition to the spatial dimension, the temporal dimension has become important, and interest towards spatiotemporal clustering has increased in GIScience, due to the pervasiveness of various types of location-based devices, smartphones and monitoring systems that enable the production of important volumes of spatiotemporal data (Chang *et al.* 2008, Khan *et al.* 2009, Mountrakis and Gunson 2009, Mennis 2010, Nakaya and Yano 2010, Pei *et al.* 2010). These approaches can be used to discover moving clusters. Meanwhile, spatial clustering techniques, if not able to discover moving clusters, can be used to discover snapshot clusters during a given time window. In our context, the objective is not to discover moving clusters, as geo-located tweets are not used to identify the trajectory of a user. Therefore, while the communities that we aim to discover change in time, our aim is not to track the movement of a community. Therefore, a spatial clustering approach like VDBSCAN is appropriate to discover spatial communities at different time periods.

3. Case study

In order to assess the performance of the proposed method and to illustrate its usefulness, we use it to discover communities that were formed via Twitter in relation to the typhoon Haiyan. On 8 November 2013, typhoon Haiyan (Yolanda) hits the Philippines, causing major damage (e.g. approximately 80% of Tacloban, the capital of the Philippine province of Leyte, has been destroyed (BBC 2013a)) and resulting in more than 5000 casualties (BBC 2013c). Besides the direct damages, regions were still flooded in the aftermath of the disaster, creating an additional obstacle for first responders who were trying to reach affected people. Media reported that planes and motor vehicles that were sent to deliver aid were delayed from reaching affected zones because of the damages to infrastructures. In this context, emergency management personnel had difficulty assessing the extent of the damage and the urgent needs of people in affected areas (BBC 2013b).

We have collected a sample of Twitter feeds from the Twitter application programming interface in the days following the event and extracted from this sample the tweets that were relevant using the keyword Haiyan and hashtag #Haiyan, which resulted in 25,552 tweets. Among these tweets, 2630 had geographical coordinates (disclosing geolocation is voluntary in Twitter). These 2630 tweets were used for our experiments.

4. Methodology

4.1. Building social graphs in Twitter

The problem of building a social graph is the following: considering a set of vertices, how to create edges that connect vertices and how to assign a weight to these edges. Graphs can be built using Twitter's various interaction modes:

(1) *Graph based on mentions*: A mention is a Twitter update that contains '@username' anywhere in the body of the tweet. It is used to reference another user. To build the graph, we create an edge between any pair of users u_1 and u_2 where u_1 issued a mention $@u_2$. The weight assigned to the edge increases with the number of mentions between u_1 and u_2 (nb_mentions(u_1, u_2)), but is normalized according to the total number of mentions in the graph and the total number of users:

$$\text{Weight}_{u_1u_2} = \frac{\text{nb\_mentions}(u_1, u_2)}{\text{total\_nb\_mentions}} \times \text{total\_nb\_users} \quad (1)$$

(2) *Graph based on 'follow relations'*: A social graph is built by establishing an edge between users linked by a 'follow relation'. The weight assigned to the edge normalizes the importance of the follow relation according to the number of follow relations by user 1 and the average number of follow relations by user in the network:

$$\text{Weight}_{u_1u_2} = \frac{\text{nb\_follow\_rel}(u_1)}{\text{total\_nb\_follow\_rel}} \times \text{total\_nb\_users} \quad (2)$$

(3) *Graph based on shared URLs*: A social graph is built by creating an edge between any pair of users u_1 and u_2 who have shared the same URLs. The weight assigned to the edge increases with the number of URLs that u_1 and u_2 have shared (nb_URLs(u_1, u_2)), but is normalized according to the total number of shared URLs in the graph and the total number of users:

$$\text{Weight}_{u_1u_2} = \frac{\text{nb\_URLs}(u_1, u_2)}{\text{total\_nb\_sharedURLs}} \times \text{total\_nb\_users} \quad (3)$$

(4) *Graph based on similar Tweet content*: A social graph is built by using the words that are common to different users' tweets. The resulting graph represents shared interests between users, but not necessarily an explicit relation between them.

To build the graph based on similar tweet content, a pre-processing phase is required to extract relevant words. Initially, all elements that are not words are removed: URLs, hashtags, Twitter handlers, emoticons, etc. Then, textual features are extracted with speech recognition techniques: tweets are separated into N-grams (i.e. meaningful sequences of N words). Stop words are removed, and the remaining words are transformed into a basic, normalized form through lemmatization. For example, the lemmatization of 'rescuing' gives 'to rescue' (the infinitive), while 'damages' gives 'damage'. Finally, the WordNet English language terminological database is queried to assign each word a set of synonyms.

The resulting text elements are not all useful (meaningful) to establish relationships between users in the context of disaster response. Therefore, a manual classification process is performed to identify the text elements that are more relevant. First, a random sample of tweets was classified according to pre-defined categories of messages. For this purpose, we have adopted the disaster-related information categories proposed by Vieweg *et al.* (2010) in their paper on improving situational awareness following natural hazards events: (1) caution and advice; (2) casualties and damages; (3) donation of money, goods

Table 1. Categories of tweets.

Categories of tweets	Examples of corresponding text elements
Caution and advice	'to evacuate', 'flooded street', 'flooded', 'blocked'
Casualties and damages	'destroyed', 'killed', 'dead', 'damage'
Donation of money, goods and services	'shelter', 'volunteer', 'blood', 'food', 'water', 'money'
People missing, found or seen	'missing', 'missing relative', 'missing kid', 'looking for', 'help find', 'safe', 'to survive'
Information source	'picture', 'video', 'image', 'media'

and services; (4) people missing, found or seen and (5) information source. Then, the most meaningful text elements contained in these tweets were manually identified (Table 1).

Then, to classify the remaining text elements extracted from tweets, the ID3 implementation of the unpruned decision tree (see Bishop 2006) was used. The ID3 unpruned decision tree is a machine learning algorithm that relies on a set of training data (here, the manually classified text elements) to classify objects according to pre-defined categories (the five above-mentioned categories). To determine whether a text element belongs to a given category, a lexical and syntactic matcher that compares the meaning of strings, described in Bakillah *et al.* (2014), was employed. Of note is that because the classification process is based on pre-defined categories, it is subjective. Still, as of today, due to the very high heterogeneity of Twitter data, most recent approaches for extraction and classification of data extracted from Twitter are also based on pre-defined categories or preselected keywords, or assume that users are employing tags from a pre-defined set (e.g. Sakaki *et al.* 2010, Vieweg *et al.* 2010, Wang *et al.* 2010, Walther and Kaisser 2013). Methods to extract latent topics from text, such as those proposed in Phan *et al.* (2008), are interesting, but tend to be able to extract limited information, such as only the most obvious categories that resemble keywords (Chen *et al.* 2011). Due to the complexity of the experiments required to adapt these methods to the context investigated in this article, the improvement of the classification method was left for future work.

An undirected social graph is created by establishing edges between users whose tweets contain some common text elements. Edge weights are assigned according to the number of common text elements in tweets of users u_1 and u_2 and normalized according to the total number of text elements that were extracted from tweets:

$$\text{Weight}_{u_1u_2} = \frac{\text{nb\_common\_text\_el}(u_1, u_2)}{\text{total\_nb\_text\_el\_extracted}} \times \text{total\_nb\_users} \tag{4}$$

Still, some of the four types of interactions used to build the social graphs may not be helpful for identifying communities that are relevant to disaster response. Instead, they can make the data noisy and the extraction of meaningful communities difficult. For example, 'follow relations' may be established for numerous reasons that are not relevant to the disaster situation; users can share URLs that are not related to the disaster either. In Section 5 (experimentation), communities that are discovered by using these different graphs are compared to find which graph enables to reveal more meaningful communities.

4.2. Enhanced fast-greedy optimization of modularity

4.2.1. Fast-greedy optimization of modularity (FGM)

Modularity measures the quality of divisions of a network into communities. The idea of modularity is that a random graph is not expected to have a community structure. If the divisions are created based on an adequate model, the density of edges within communities should be high, while the density of connections between communities should be low. By maximizing the difference between the generated graph structure and a random graph structure (expressed with the quality function Q), one can obtain the optimal clustering results. The quality function is defined as follows (Clauset *et al.* 2004):

$$Q = \sum_{c=1}^{n} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right] \tag{5}$$

where n is the number of clusters, m is the total number of edges, l_c is the total number of edges joining the vertices of cluster c and d_c is the sum of the expected random graph degree of the vertices of c. By progressively merging communities that can achieve the largest quality change (ΔQ), we obtain the largest value of Q. As shown in Algorithm 1, FGM first assigns each vertex into one community (i.e. cluster) (lines 1–4). For each pair of communities that have at least one edge between them, it calculates the value of ΔQ and stores it into a sparse matrix Q (lines 9–10). Each row of the sparse matrix stores the values of ΔQ from one community to all other communities that connect to it. The max-heap H then stores the largest ΔQ value of each row of Q and the corresponding community id u, v (lines 13–15). After the initialization of Q and H, in the consecutive process, FGM iteratively pops out the object in H that has the highest ΔQ value and merges the corresponding communities until only one community is left (lines 17–29). It also progressively merges pairs of rows and columns in Q as the corresponding communities are merged. FGM successively merges two communities (i.e. merged and merging community) to form a bigger community, and it considers three scenarios when merging a pair of communities: (1) community k is connected to both merged and merging community, (2) k is connected to merged only and (3) k is connected to merging only. Meanwhile, the ΔQ values related to k, merged and merging community must be updated. When Q is updated, the maximum value for each row may be changed. Every time a pair of communities is merged, FGM sorts each row of Q to obtain the maximum value and updates H accordingly (Algorithm 2). H changes depending on the update of Q, which means that the order of merging relies on the updated values of ΔQ, which are determined by the connectivity of communities relative to the overall graph. Consequently, clustering results are very sensitive to the graph structure. In a social network like Twitter, the graph structure is usually noisy; FGM may fail to cluster the graph accurately. To address this problem, we introduce text similarity measure to the MergeCommunities (line 28), in which the UpdateMaxHeap method is called.

Algorithm 1: FGM.

Requires: social graph $G = (V, E)$, sparse matrix Q, max-heap H, cluster set C

1: **for all** $v \in V$ **do**
2: cid = AssignCluster (v)

```
3:    C ← cid
4:  end for
5:  for all u ∈ C do
6:    count = 0
7:    for all v ≠ u ∈ C do
8:        if u has connection to v then
9:            Q (u, count) = CalculateDeltaQ (u, v, G)
10:           count = count+1
11:       end if
12:   end for
13:   max Q, max_v = max(Q(u,:))
14:   heapObj = (u, max_v, maxQ)
15:     H(u) ← heapObj
16:     end for
17:     While size(H) > 0 do
18:         (pop_u, pop_max_v, pop_maxQ) = heappop(H)
19:         support_u = GetNumberOfConnection(pop_u)
20:         support_v = Get NumberOfConnection(pop_v)
21:         if support_u < support_v then
22:           merged_node = u
23:           merging_node = v
24:         else
25:           merged_node = v
26:           merging_node = u
27:         end if
28:           MergeCommunities (Q, H, merged_node, merging_node)
29:     end while
```

Algorithm 2: UpdateMaxHeap.

```
Requires: Q, H, target_u
1:   rowQ = RetrieveRow(Q, target_u)
2:   maxQ, max_v = max(rowQ)
3:   heapObj = (target_u, max_v, maxQ)
4:   H (target_u) ← heapObj
```

4.2.2. Enhanced FGM with similarity measure

To make FGM less sensitive to the graph structure, we enhance it by integrating graph-based and text-based (text similarity measure) models to balance the importance of shared content versus graph structure. The improved algorithm compares the similarity of text elements associated with communities when Q and H are updated, i.e. Algorithm 4 is executed while communities are merged. We consider that any pair of communities that have high text similarity and that are linked by more than one connection should be part of the same communities, and its merge priority should be increased. By setting the right condition for text similarity measure, we compensate the sensitivity of the original clustering algorithm. To integrate the text similarity measure into FGM, we need to determine when to merge two communities based on text similarity and when to merge them based on graph structure. To achieve this, we need a threshold T for text similarity to determine whether two communities are similar enough to increase the priority to merge them. The value given to T should depend on the graph structure; for instance, if

the community structure of the graph is weak, the value of T should be low, while if, overall, the values of text similarity are high, T should be high, and vice versa. In this study, we generate an extra matrix S like Q to record community similarities and initialize the values with the text similarity between communities (vertices) (Algorithm 3, line 10). To initialize each object of H, our model considers two scenarios: for each row, (1) if the largest value of the row of S is smaller than T, it sorts the row of Q and selects the object with largest ΔQ to update H; (2) if the largest value of the row of S is larger than T, it uses the ΔQ object with highest text similarity to update H (Algorithm 3, lines 14–22). Whenever our model progressively merges pairs of communities, it updates text resources for the new communities to allow determining new text similarities. Afterwards, it updates the ΔQ values for community k and merged and merging community, as well as S and the objects in H by running Algorithm 4. The cosine similarity measure is used to compute similarity between communities' set of terms:

$$\text{Cosine similarity} = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{k=1}^{l}(A_k \times B_k)}{\sqrt{\sum_{i=1}^{n}(A_i)^2 \times \sum_{j=1}^{m}(B_j)^2}} \tag{6}$$

A and B represent the frequency of a term in the set of terms associated with the first and second community, respectively. The similarity value ranges from −1, meaning dissimilarity, to 1, meaning exact similarity. Empirically, we found that using 0.2–0.3 as value of T generates better results in terms of recall (see experiments in Section 5.1).

Algorithm 3: Enhanced FGM.

Requires: social graph $G = (V, E)$, sparse matrix Q, max-heap H, cluster set C, similarity matrix S, threshold for similarity ε

```
1:  for all v ∈ V do
2:      cid = AssignCluster(v)
3:      C ← cid
4:  end for
5:  for all u ∈ C do
6:      count = 0
7:      for all v ≠ u ∈ C do
8:          if u has connection to v then
9:              Q (u, count) ← CalculateDeltaQ (u, v, G)
10:             S (u, count) ← CalculateTextSim(text_u, text_v)
11:             count = count +1
12:         end if
13:     end for
14:     max Q, max_v = max(Q(u,:))
15:     if maxQ < ε then
16:         heapObj = (u, max_v, maxQ)
17:     else
18:         max_S, max_v = max(S(u,:))
19:         most_similar_Q = Q9 u, max_v)
20:         heapObj = (u, max_v, most_similar_Q)
21:     end if
22:     H(u) ← heapObj
23: end for
24: While size(H) > 0 do
```

```
25:     (pop_u, pop_max_v, pop_maxQ) = heappop(H)
26:      support_u = GetNumberOfConnection(pop_u)
27:      support_v = Get NumberOfConnection(pop_v)
28:      if support_u < support_v then
29:         merged_node = u
30:         merging_node = v
31:      else
32:         merged_node = v
33:         merging_node = u
34:      end if
35:      MergeCommunities(Q, H, merged_node, merging_node)
36:  end while
```

Algorithm 4: Enhanced UpdateMaxHeap.

```
Requires: Q, H, target_u
 1: rowQ = RetrieveRow(Q, target_u)
 2: maxQ, max_v = max(rowQ)
 3: if maxQ < ε then
 4:       heapObj = (target_u, max_v, maxQ)
 5: else
 6:       max_S, max_v = max(S(u,:))
 7:       most_similar_Q = Q(target_u, max_v)
 8:       heapObj = (target_u, max_v, most_similar_Q)
 9: end if
10:  H(target_u) ← heapObj
```

4.2.3. *Spatial clustering at different temporal snapshots*

In order to extract spatial information on the event that is being discussed in Twitter, we now need to analyse the communities that were obtained with the enhanced FGM (we call them thematic communities) from a spatial point of view. Thematic communities are formed with Twitter users that discuss similar topics but that may be spatially and temporally distributed. We need to discover snapshot spatial clusters into them. To do this, we apply the VDBSCAN algorithm discussed in Section 2.2 to each thematic community for disjoint time periods. The computational complexity of the VDBSCAN is $O(n\log n)$.

Let D be the set of points corresponding to the geo-located tweets found in a given thematic community and that were issued during time period T. The objective of the VDBSCAN spatial clustering algorithm is to find spatial clusters based on regions with higher density. The spatial clustering algorithm is based on two parameters:

(1) r: the value of the radius that will be used to select members of a cluster and
(2) MinPts: minimal density to form a cluster.

The neighbour of p is defined as follows (Liu *et al.* 2007):

$$N(p) = \{q \in D | \text{dist}(p, q) \leq r\} \tag{7}$$

A cluster is generated based on the following properties, called 'density-reachable' and 'density-connected':

(1) A point q is directly density-reachable from a point p if

$$q \in N(p,r) \text{ and } |N(p,r)| \geq \text{MinPts} \tag{8}$$

(2) A point q is density-reachable from a point p if there exists a sequence of points $p_1, \ldots, p_n$ where $p_1 = p$ and $p_n = q$ such that p_{i+1} is directly reachable from p_i, for all i.
(3) A point q is density-connected to a point p if there is a point o such that p and q are density-reachable from o.

A cluster is a non-empty subset of D that satisfies the following properties:

(1) $\forall\ p,\ q,$ if $p \in C$ and q is density $-$ reachable from $p,$ then $q \in C$ (maximality).
(2) $\forall\ p,\ q \in C,\ p$ is density $-$ connected to q (connectivity).

In the VDBSCAN algorithm, the parameters r and MinPts are optimized automatically based on the variation in density of the data set; see details in Liu *et al.* (2007).

5. Case study and evaluation of enhanced FGM

5.1. *Algorithm's performance*

We first ran an experiment to test the algorithm's performance. We have simulated a social graph composed of 350 vertices associated with selected Twitter messages (from the data set) and 750 edges. We have set up 30 communities for this simulated graph (authoritative communities).

We have evaluated the clustering results generated by the original FGM and the enhanced FGM with the recall and precision quality measures. The recall is the fraction of the discovered communities that are part of the set of authoritative communities:

$$\text{recall} = \frac{|\text{relevant retrieved CoIs}|}{|\text{authoritative CoIs}|} \tag{9}$$

We consider that a cluster is relevant if its purity is larger than a given threshold. The purity is defined as follows (Cao *et al.* 2006):

$$\text{purity} = \frac{\sum_{i=1}^{k} \frac{|C_i^d|}{|C_i|}}{k} \times 100\% \tag{10}$$

In this equation, k is the number of communities, $|C_i^d|$ is the number of vertices that were correctly assigned to the community and $|C_i|$ is the number of vertices in community i.

The precision is the fraction of discovered communities that are part of the authoritative clusters versus the total number of discovered communities:

$$\text{precision} = \frac{|\text{relevant retrieved CoIs}|}{|\text{discovered CoIs}|} \tag{11}$$

First, the results obtained with the graphs that were formed based on different types of edges and their combinations (as detailed in Section 4.1) were compared (Table 2). Two combinations of edge types are reported: (1) graph based on mentions/follow relations and similar tweet content, where edges were established between users linked by both mention or follow relations and similar tweet content and (2) graph based on mentions/follow relations and shared URLs. In this way, it is tested whether the combination of an explicit relation (mentions or follow relations) with an implicit relation (similarity of shared information) improves the meaningfulness of the results. The results are reported for three values of purity threshold (0.3, 0.4 and 0.5) and a similarity threshold of 0.3 (since this value generated better results, see Tables 3 and 4).

The worst results are obtained with the graph based on shared URLs, confirming that for the current data set, the shared URLs were generally not related to the disaster situation and not reliable to detect useful communities. Optimal recall and precision are obtained when combining explicit relations with filtered tweets' content. Of note is that for establishing the authoritative communities, only the content of tweets was considered, but not the structure of explicit relations. Therefore, the fact that combining explicit relations with filtered tweets' content results in higher recall and precision than filtered tweets' content alone suggests that in this specific case, the users were more likely to share information about the disaster with the people they had already interacted with previously. This indicates that the explicit structure of the social graph is useful to propagate information about disaster response.

In the second part of the performance evaluation, the algorithm was compared with the generic FGM with different purity and similarity thresholds, with the graph being built by combining explicit relations with filtered tweets' content (Tables 3 and 4).

The enhanced FGM uncovers more relevant communities and diminishes the number of irrelevant communities compared to the generic FGM, assuming the similarity threshold is appropriately determined. By setting the purity threshold to 0.4 instead of 0.5, more relevant communities are uncovered by both the generic and the enhanced FGM. This indicates that enhanced FGM does not change the nature of FGM; it simply reorders the merging priorities. In Figure 1, we can also see that the enhanced FGM (blue line) generally enables the discovery of more relevant users in each community.

In terms of processing time, the enhanced FGM's time complexity is $O(n^2 d \log n)$, which, according to Papadopoulos *et al.* (2012), makes it efficient for scales up to 10^6 nodes. The efficiency of the algorithm thus compares to methods that include vertex clustering (e.g. Pons and Latapy 2005) and other modularity optimization methods (e.g. Arenas *et al.* 2007).

5.2. *Spatial visualization of communities*

In the second part of the experiments, a spatial analysis of the communities discovered at different time is conducted. The objective was to discover if by re-dividing thematic communities into smaller communities through spatial clustering at different times, we would discover meaningful sub-clusters. We have selected four categories of communities that were created from the 2630 tweets: users sharing 'warning', 'damage', 'help' and 'casualties and wounded' tweets, which were the main categories of community

Table 2. Recall and precision of community discovery results with various graphs (similarity threshold = 0.3).

Type of graph	Purity threshold = 0.3	Purity threshold = 0.4	Purity threshold = 0.5
Graph based on mentions	Recall = 36%, precision = 16%	Recall = 33%, precision = 15%	Recall = 30%, precision = 12%
Graph based on follow relations	Recall = 30%, precision = 14%	Recall = 28%, precision = 11%	Recall = 28%, precision = 11%
Graph based on shared URLs	Recall = 12%, precision = 6%	Recall = 10%, precision = 5%	Recall = 8%, precision = 4%
Graph based on similar tweet content	Recall = 79%, precision = 76%	Recall = 76%, precision = 74%	Recall = 72%, precision = 72%
Graph based on mentions/ follow relations and similar tweet content	Recall = 90%, precision = 82%	Recall = 88%, precision = 80%	Recall = 80%, precision = 75%
Graph based on mentions/ follow relations and shared URLs	Recall = 12%, precision = 8%	Recall = 10%, precision = 5%	Recall = 8%, precision = 4%

Table 3. Recall and precision, purity threshold = 0.4.

Algorithm tested	Recall (%)	Precision (%)
Generic FGM	66	52
Enhanced FGM, similarity threshold, $T = 0.1$	66	45
Enhanced FGM, $T = 0.2$	72	58
Enhanced FGM, $T = 0.3$	88	80
Enhanced FGM, $T = 0.4$	70	65

Table 4. Recall and precision, purity threshold = 0.5.

Algorithm tested	Recall (%)	Precision (%)
Generic FGM	60	45
Enhanced FGM, similarity threshold, $T = 0.1$	58	38
Enhanced FGM, $T = 0.2$	70	55
Enhanced FGM, $T = 0.3$	80	75
Enhanced FGM, $T = 0.4$	64	59

discovered. We have run the spatial clustering at regular 12 hour time intervals, i.e. considering only the tweets that were issued from 0 to 12 hours, 12 to 24 hours, etc. To estimate the meaningfulness of the geo-located communities, we have identified the main text elements in each spatial cluster to detect the main theme discussed by members of the discovered community (e.g. 'flooded streets' as part of 'warning' category). Then, we have evaluated the purity value (Equation (10)), which measures the number of correctly assigned users given the identified theme. In Figures 2 and 3, we see that the purity of spatial clusters varies widely in time, from 0.12 to nearly 0.6.

When analysing the peaks where purity is high, we discover that tweets within clusters are more consistent and report on similar incidents. The variations in purity are more obvious with respect to 'warning' and 'damage' tweets, possibly because these tweets are

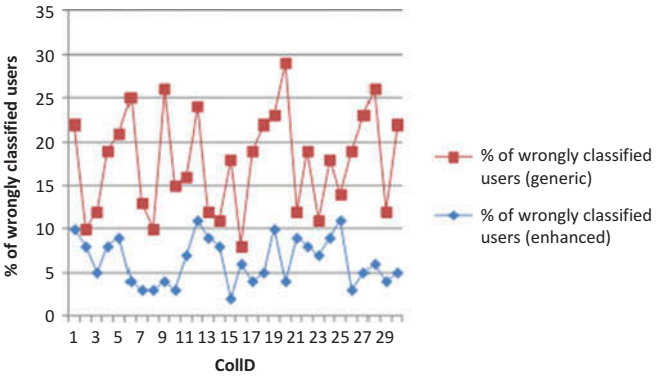


Figure 1. Comparison of the percentage of wrongly classified users.

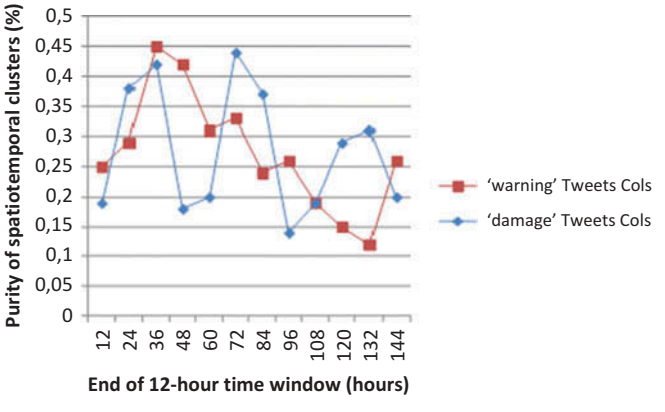


Figure 2. Purity of 'warning 'and 'damage'geo-located communities.

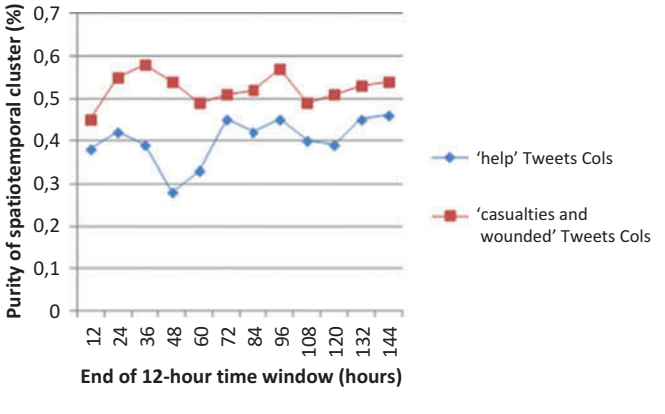


Figure 3. Purity of 'help' and 'casualties and wounded' geo-located communities.

most clearly related to spatiotemporal incidents than requests for 'help' and reports of 'casualties'. For example, in Figure 2, the peak located at 36 hours contains several clusters containing tweets that relate to a destroyed harbour. This is visible in Figure 4, where several communities (in red) located near destroyed areas were found.

In comparison, the 'damage' communities obtained at 96 hours (when purity is at the lowest level) are more scattered, and only two (and more sparse) clusters were discovered (Figure 5).

Interestingly, the study suggests that discovering geo-located communities and assessing their purity level can help to identify and locate incidents occurring during emergency situations. However, as shown in Section 5.2, using explicit relations between users of social networks may not return relevant communities, as several of these relations are not related to the given event. Instead, the relations are useful in complement of relations established based on filtered and classified tweets' content. In addition, the discovered communities must not be considered as a simple set of users. The structure of the discovered communities is also important, as it can be leveraged to disseminate

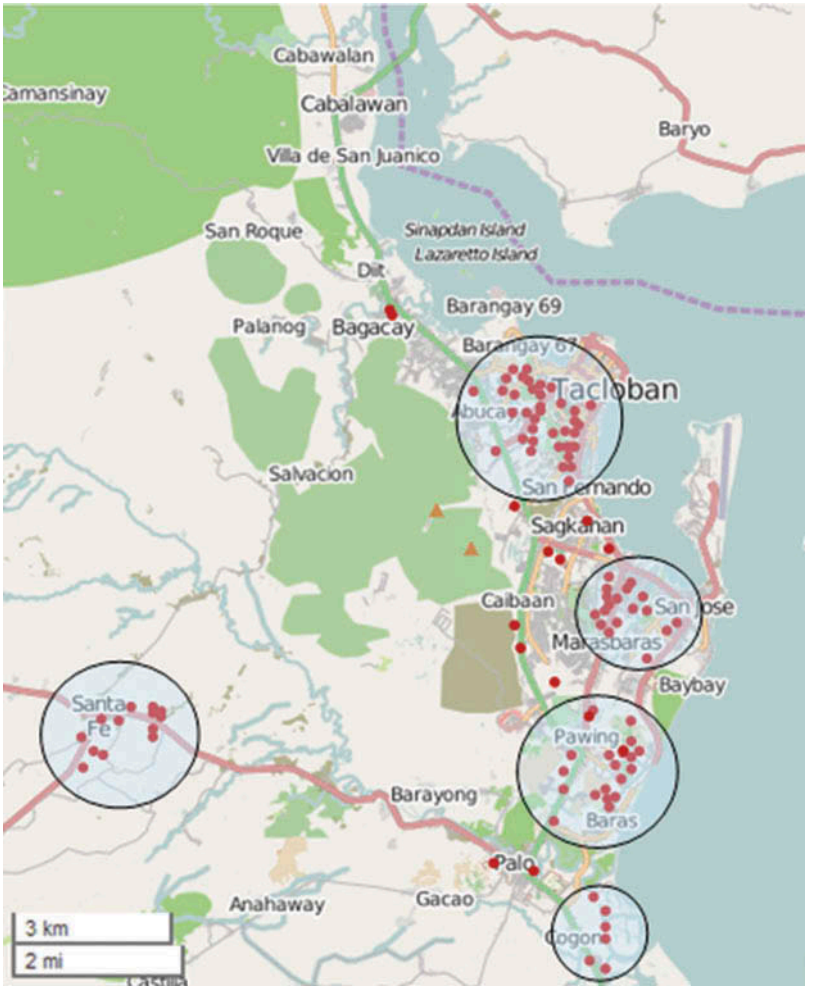


Figure 4. Visualization of 'damage' communities discovered at 36 hours.

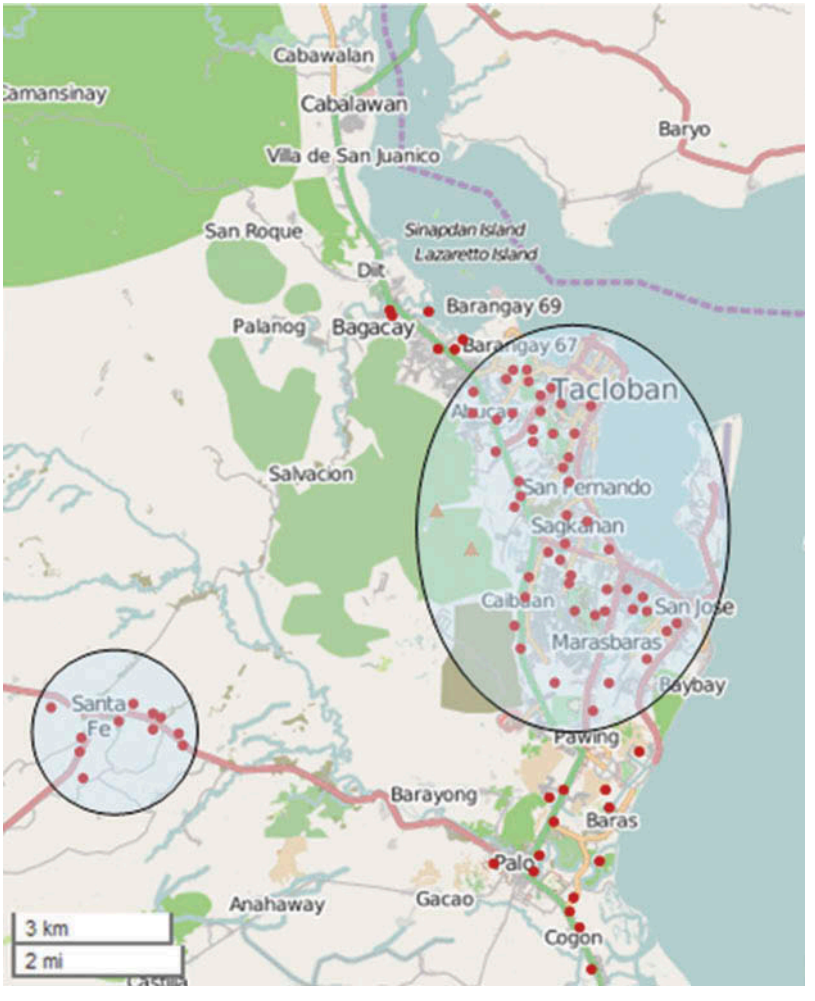


Figure 5. Visualization of 'damage' communities discovered at 96 hours.

information during disaster response or to organize relief operations. A strongly connected community is more likely to share information and act coherently. As an illustration, we have calculated the index of connectivity of communities, which depends on the number of edges connecting all members of the community (Rhouma and Romdhane 2014), and we have obtained variations from approximately 4 to 21 for all communities formed. A negative index would mean that the members of the community are more strongly connected to users that are not part of the community, whereas the higher the index, the more strongly connected are the members of the community. The density of links inside the communities suggests that the discovered communities are strongly connected.

6. Discussion and conclusions

Social media like Twitter are an emerging source of information to monitor geographical phenomena. More particularly, studies have demonstrated that people use social media in

disaster situations to report on various incidents related to damage, infrastructure, assistance, missing people, etc. Meanwhile, research on methods to extract relevant information from social media feeds is still at an early stage. Challenges are numerous, given the high volume of data, its heterogeneity and noisiness. The study that we have presented in this article contributes to this research field, by investigating how the combination of different clustering algorithms can be adapted to discover geo-located communities in Twitter data, more particularly in the case of a natural disaster. More specifically, we have enhanced the FGM algorithm to enable the discovery of communities in complex Twitter graphs, where different types of relations can be established. Then, we have coupled the enhanced FGM with a spatial clustering algorithm to detect geo-located communities within the discovered thematic communities. The Twitter data were extracted from tweets issued in relation to typhoon Haiyan, which occurred in November 2013. The algorithm was able to uncover communities exchanging information about damages, request for assistance, casualties of wounded people or warning. Of note is that due to the noisiness of the data, categories of tweets had to be predefined using a sample of tweets. Therefore, a pre-processing phase specific to the application is required for the method to be useful. The experiment have demonstrated that for this data set, the enhanced FGM is able to discover more relevant communities than the generic version, which shows the need to adapt data mining techniques to the particularities of social media. The experiments also demonstrated that more relevant communities are obtained when the explicit relations between users are taken into account, in addition to the content of exchanged messages. This is particularly important, as it shows that pure text mining of messages would not give as good results and that the structure of the network plays an important role. Further research remains to be done to enhance the method. For now, the method for setting the appropriate threshold is to evaluate the recall of the algorithm with a set of authoritative communities, i.e. the algorithm needs to be trained. While it was done manually in this article, an automatic optimization method to set the threshold could be developed. In-depth research and experiments must be conducted to develop and test such method and will be pursued in future work.

It was observed that by harvesting data from Twitter, a broad overview of some incidents can be obtained; however, due to the heterogeneity of messages, it is difficult to obtain further details on these incidents. At the moment, discovered communities are useful as pointers to areas where some category of incident has been reported, or as a way to use existing relations between users to mobilize people and resources and disseminate relevant information in a disaster situation. Further research is required to investigate how lexical analysis techniques can be deployed to extract more detailed information from communities. Also, further statistical methods could be deployed, in addition to the clustering model, to analyse the spatiotemporal structure of tweets' topical content in more detail. For example, a combination of the latent Dirichlet allocation topic model and spatial kernel density was used to analyse events in streaming text (Pozdnoukhov and Kaiser 2011). This approach has allowed extracting topics of discussion through data-driven techniques only. It suggests that it is possible to classify messages without relying on pre-defined categories, although the experiments conducted by the authors do not show that various categories were uncovered. This could potentially enable to avoid the problem of subjective classification mentioned in Section 4.1. However, further experiments would be required to determine whether the uncovered categories would be as rich and diverse as the pre-defined categories. This issue will therefore be addressed in future work.

References

Arenas, A., Díaz-Guilera, A., and Pérez-Vicente, C.J., 2006. Synchronization reveals topological scales in complex networks. *Physical Review Letters*, 96 (11), 114102. doi:10.1103/PhysRevLett.96.114102

Arenas, A., *et al*., 2007. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9 (6), 176. doi:10.1088/1367-2630/9/6/176

Bakillah, M. and Liang, S.H.L., 2012. Discovering sensor services with social network analysis and expanded SQWRL. *In*: S. Di Martino, A. Peron, and T. Tezuka, eds. *Proceedings of the 11th international conference on web and wireless geographical information systems (W2GIS)*, April 2012. Naples: Springer Verlag, 221–238.

Bakillah, M., *et al*., 2007. Mapping between dynamic ontologies in support of geospatial data integration for disaster management. *In*: J. Li, S. Zlatanova, and A.G. Fabbri, eds. *Geomatics solutions for disaster management*. Berlin: Springer, 201–224.

Bakillah, M., *et al*., 2014. A dynamic and context-aware semantic mediation service for discovering and fusion of heterogeneous sensor data. *Journal of Spatial Information Science*, 6, 155–185.

Batagelj, V. and Zaveršnik, M., 2011. Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 5 (2), 129–145. doi:10.1007/s11634-010-0079-y

Bishop, C.M., 2006. *Pattern recognition and machine learning*. New York: Springer-Verlag.

Blondel, V.D., *et al*., 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008, P10008. doi:10.1088/1742-5468/2008/10/P10008

British Broadcasting Corporation (BBC), 2013a. *Philippines Typhoon Haiyan: 'Storm surge crushed all of these buildings'* [online]. Available from: http://www.bbc.com/news/world-asia-24890795 [Accessed 6 June 2014].

British Broadcasting Corporation (BBC), 2013b. *Typhoon Haiyan: thousands feared dead in Philippines* [online]. Available from: http://www.bbc.com/news/world-asia-24887337 [Accessed 6 June 2014].

British Broadcasting Corporation (BBC), 2013c. *Typhoon Haiyan death toll rises over 5,000* [online]. Available from: http://www.bbc.com/news/world-asia-25051606 [Accessed 6 June 2014].

Cao, F., *et al*., 2006. Density-based clustering over an evolving data stream with noise. *In*: J. Ghosh, *et al*., eds. *Proceedings of the 6th SIAM international conference on data mining*, 20–22 April Bethesda, MD. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 328–339.

Carvalho, A.X.Y., *et al*., 2009. Spatial hierarchical clustering. *Revista Brasileira de Biometria*, 27 (3), 411–442.

Chang, W., Zeng, D., and Chen, H., 2008. A stack-based prospective spatio-temporal data analysis approach. *Decision Support Systems*, 45, 697–713. doi:10.1016/j.dss.2007.12.008

Charlton, M.E., 2006. A mark 1 geographical analysis machine for the automated analysis of point data sets: twenty years on. *In*: P.F. Fisher, ed. *Classics from IJGIS: twenty years of the international journal of geographical information science and systems*. Boca Raton, FL: CRC Press, 35–40.

Chen, J., Zaiane, O.R., and Goebel, R., 2009. Local community identification in social networks. *In*: *International conference on advances in social networks analysis and mining (ASONAM)*, 20–22 July 2009. Athens: IEEE, 237–242.

Chen, M., Jin, X., and Shen, D., 2011. Short text classification improved by learning multi-granularity topics. *In*: T. Walsh, ed. *Proceedings of the 22nd international joint conference on artificial intelligence (IJCAI)*, July 2011. Barcelona: AAAI Press, 1778–1781.

Chiu, C., Hsu, M., and Wang, E., 2006. Understanding knowledge sharing in virtual communities: an integration of social capital and social cognitive theories. *Decision Support Systems*, 42, 1872–1888. doi:10.1016/j.dss.2006.04.001

Clauset, A., Newman, M.E.J., and Moore, C., 2004. Finding community structure in very large networks. *Physical Review E*, 70, 6. doi:10.1103/PhysRevE.70.066111

Conley, J., Gahegan, M., and Macgill, J., 2005. A genetic approach to detecting clusters in point data sets. *Geographical Analysis*, 37 (3), 286–314. doi:10.1111/j.1538-4632.2005.00617.x

Crooks, A., *et al*., 2013. #Earthquake: twitter as a distributed sensor system. *Transactions in GIS*, 17 (1), 124–147. doi:10.1111/j.1467-9671.2012.01359.x

Culotta, A., 2010. Towards detecting influenza epidemics by analyzing Twitter messages. *In*: *KDD workshop on social media analytics*. New York: ACM, 115–122.

De Meo, P., *et al*., 2014. Mixing local and global information for community detection in large networks. *Journal of Computer and System Sciences*, 80 (1), 72–87. doi:10.1016/j.jcss.2013.03.012

Donath, W. and Hoffman, A., 1973. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17 (5), 420–425. doi:10.1147/rd.175.0420

Duch, J. and Arenas, A., 2005. Community detection in complex networks using extremal optimization. *Physical Review E*, 72, 27104. doi:10.1103/PhysRevE.72.027104

Expert, P., *et al*., 2011. Uncovering space-independent communities in spatial networks. *Proceedings of the National Academy of Sciences*, 108 (19), 7663–7668. doi:10.1073/pnas.1018962108

Fernandez, A. and Gomez, S., 2008. Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms. *Journal of Classification*, 25, 43–65. doi:10.1007/s00357-008-9004-x

Fortunato, S., 2010. Community detection in graphs. *Physics Reports*, 486, 75–174. doi:10.1016/j.physrep.2009.11.002

Fortunato, S., Latora, V., and Marchiori, M., 2004. Method to find community structures based on information centrality. *Physical Review E*, 70, 56104. doi:10.1103/PhysRevE.70.056104

Freeman, L.C., 1977. A set of measures of centrality based on betweenness. *Sociometry*, 40, 35–41. doi:10.2307/3033543

Girvan, M. and Newman, M.E.J., 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99 (12), 7821–7826. doi:10.1073/pnas.122653799

Glass, A.G. and Mantel, N., 1969. Lack of time-space clustering of childhood leukemia in Los Angeles County, 1960–1964. *Cancer Research*, 29 (11), 1995–2001.

Hagar, C., 2009. The information and social needs of Cumbrian farmers during the UK 2001 foot and mouth disease outbreak and the role of information and communication technologies. *In*: M. Döring and B. Nerlich, eds. *The socio-cultural impact of foot and mouth disease in the UK in 2001: experiences and analyses*. Manchester University Press.

Hastings, M.B., 2006. Community detection as an inference problem. *Physical Review E*, 74, 035102. doi:10.1103/PhysRevE.74.035102

Kannan, R., Vempala, S., and Vetta, A., 2004. On clusterings: good, bad and spectral. *Journal of the Acm*, 51 (3), 497–515. doi:10.1145/990308.990313

Kaufman, L. and Rousueeuw, P., 1990. Finding groups in data: an introduction to cluster analysis. *In*: *Wiley series in probability and mathematical statistics. Applied probability and statistics*. Vol. 1. New York: John Wiley and Sons. doi:10.1002/9780470316801

Khan, G., *et al*., 2009. Application and integration of lattice data analysis, network K-functions, and geographic information system software to study ice-related crashes. *Transportation Research Record: Journal of the Transportation Research Board*, 2136, 67–76. doi:10.3141/2136-08

Kisilevich, S., *et al*., 2010. Spatio-temporal clustering: a survey. *In*: O. Maimon and L. Rokach, eds. *Data mining and knowledge discovery handbook*. Berlin: Springer.

Li, X., Hu, W., and Hu, W., 2006. A coarse-to-fine strategy for vehicle motion trajectory clustering. *In*: *International conference on pattern recognition (ICPR)*, August 2006. Hong Kong: IEEE, 591–594.

Liu, P., Zhou, D., and Wu, N., 2007. Varied density based spatial clustering of application with noise. *In*: *Proceedings of IEEE conference ICSSSM*. New York: IEEE, 528–531.

Martin, E., *et al*., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *In*: *Proceedings of KDD*. Palo Alto, CA: AAAI, 226–231.

Mennis, J., 2010. Multidimensional map algebra: design and implementation of a spatio-temporal GIS processing language. *Transactions in GIS*, 14 (1), 1–21. doi:10.1111/j.1467-9671.2009.01179.x

Mooney, P. and Corcoran, P., 2013. Understanding the roles of communities in volunteered geographic information projects. *In*: J.M. Krisp, ed. *Progress in location-based services*. Berlin: Springer, 357–371.

Mountrakis, G. and Gunson, K., 2009. Multi-scale spatiotemporal analyses of moose–vehicle collisions: a case study in northern Vermont. *Ijgis*, 23, 1389–1412.

Murata, T., 2010. Detecting communities in social networks. *In*: B. Furht, ed. *Handbook of social network technologies and applications*. New York: Springer, 269–280.
Nakaya, T. and Yano, K., 2010. Visualising crime clusters in a space-time cube: an exploratory data-analysis approach using space-time kernel density estimation and scan statistics. *Transactions in GIS*, 14 (3), 223–239. doi:10.1111/j.1467-9671.2010.01194.x
Nanopoulos, A., Theodoridis, Y., and Manolopoulos, Y., 2001. C2P: clustering based on closest pairs. *In*: P.M.G. Apers, *et al*., eds. *Proceedings of VLDB*. San Francisco, CA: Morgan Kaufmann, 331–340.
Newman, M.E.J., 2004a. Detecting community structure in networks. *The European Physical Journal B – Condensed Matter*, 38, 321–330. doi:10.1140/epjb/e2004-00124-y
Newman, M.E.J., 2004b. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, 66133. doi:10.1103/PhysRevE.69.066133
Newman, M.E.J. and Girvan, M., 2004. Finding and evaluating community structure in networks. *Physical Review E*, 69 (2), 26113. doi:10.1103/PhysRevE.69.026113
Openshaw, S., *et al*., 1987. A mark 1 geographical analysis machine for the automated analysis of point data sets. *Ijgis*, 1 (4), 335–358.
Palla, G., *et al*., 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435 (7043), 814–818. doi:10.1038/nature03607
Papadopoulos, S., *et al*., 2011. Cluster-based landmark and event detection for tagged photo collections. *IEEE Multimedia*, 18 (1), 52–63. doi:10.1109/MMUL.2010.68
Papadopoulos, S., *et al*., 2012. Community detection in social media performance and application considerations. *Data Mining and Knowledge Discovery*, 24, 515–554. doi:10.1007/s10618-011-0224-z
Parimala, M., Lopez, D., and Senthilkumar, N.C., 2001. A survey on density based clustering algorithms for mining large spatial databases. *International Journal of Advanced Science and Technology*, 31 (1), 59–66.
Pei, T., *et al*., 2010. Windowed nearest neighbor method for mining spatio-temporal clusters in the presence of noise. *Ijgis*, 24, 925–948.
Phan, X.-H., Nguyen, L.-M., and Horiguchi, S., 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. *In*: *Proceedings of 17th WWW*, April 2008. New York: ACM, 91–100.
Pons, P. and Latapy, M., 2005. Computing communities in large networks using random walks. *In*: P. Yolum, *et al*., eds. *Computer and information sciences*. Berlin: Springer, 284–293.
Pozdnoukhov, A. and Kaiser, C., 2011. Space-time dynamics of topics in streaming text. *In*: *Proceedings of the 3rd ACM SIGSPATIAL international workshop on location-based social networks*. New York: ACM, 1–8.
Raghavan, U.N., Albert, R., and Kumara, S., 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76, 036106.
Rasheduzzaman Chowdhury, A.K.M. and Asikur Rahman, M.D., 2010. An efficient method for subjectively choosing parameter k automatically in VDBSCAN. *In*: V. Mahadevan and Z. Jianhong, eds. *Proceedings of Computer and Automation Engineering (ICCAE) 2010*, 26–28 February Singapore. Singapore: IEEE, 38–41.
Reichardt, J. and Bornholdt, S., 2006. Statistical mechanics of community detection. *Physical Review A*, 74, 016110.
Rhouma, D. and Romdhane, L.B., 2014. An efficient algorithm for community mining with overlap in social networks. *Expert Systems with Applications*, 41 (9), 4309–4321.
Sahoo, A.K., 2013. ADCA: advanced density based clustering algorithm for spatial database system. *International Journal of Computer Science and Mobile Computing*, 2 (7), 41–47.
Sakaki, T., Okazaki, M., and Matsuo, Y., 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. *In*: *Proceedings of the 19th international conference on WWW*, Raleigh, NC. New York: ACM, 851–860.
Schaeffer, S.E., 2007. Graph clustering. *Computer Science Review*, 1 (1), 27–64. doi:10.1016/j.cosrev.2007.05.001
Shklovski, I., Palen, L., and Sutton, J., 2008. Finding community through information and communication technology during disaster events. *In*: *CSCW'08*, 8–12 November 2008 San Diego, CA. New York: ACM, 127–136.

Smith, P.G., *et al*., 1976. Epidemiology of childhood leukaemia in greater London: a search for evidence of transmission assuming a possibly long latent period. *British Journal of Cancer*, 33 (1), 1–8. doi:10.1038/bjc.1976.1

Šubelj, L. and Bajec, M., 2014. Group detection in complex networks: an algorithm and comparison of the state of the art. *Physica A: Statistical Mechanics and Its Applications*, 397, 144–156. doi:10.1016/j.physa.2013.12.003

Vieweg, S., *et al*., 2010. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. *In*: *Proceedings of the 28th international conference on human factors in computing systems*, Atlanta, USA. New York: ACM, 1079–1088.

Vragovic, I. and Louis, E., 2006. Network community structure and loop coefficient method. *Physical Review E*, 74, 016105. doi:10.1103/PhysRevE.74.016105

Walsh, F. and Pozdnoukhov, A., 2011. Spatial structure and dynamics of urban communities. *In*: *Proceedings of the 2011 workshop on pervasive urban applications*, 12–15 June San Francisco, CA. Unpublished.

Walther, M. and Kaisser, M., 2013. Geo-spatial event detection in the twitter stream. *In*: P. Serdyukov, *et al*., eds. *Advances in information retrieval*. Berlin: Springer, 356–367.

Wang, X., *et al*., 2010. Discovering overlapping groups in social media. *In*: *IEEE international conference on data mining*, 13–17 December 2010. Sydney: IEEE, 569–578.

Ward, J.H.J., 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58, 236–244. doi:10.1080/01621459.1963.10500845

Xie, J., Kelley, S., and Szymanski, B., 2013. Overlapping community detection in networks: the state of the art and comparative study. *ACM Computing Surveys*, 45 (4), 1–35. doi:10.1145/2501654.2501657

Xu, R. and Wunsch II, D., 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16 (3), 645–678. doi:10.1109/TNN.2005.845141

Yang, B. and Liu, J., 2008. Discovering global network communities based on local centralities. *ACM Transactions on the Web (TWEB)*, 2 (1), 1–32. doi:10.1145/1326561.1326570

Zhang, T., Ramakrishnan, R., and Livny, M., 1996. BIRCH: an efficient data clustering method for very large databases. *In*: J. Widom, ed. *Proceedings of ACM SIGMOD*. New York: ACM, 103–114.

Zhang, Y., Wu, Y., and Yang, Q., 2012. Community discovery in Twitter based on user interests. *Journal of Computational Information Systems*, 8 (3), 991–1000.