Identifying Patterns of Temporal and Geo-Referenced Content Generated by Mobile Social Users[1]

Marcelo G. Malcher, Markus Endler, Karin Breitman

Departamento de Informática – PUC-Rio
Rua Marquês de São Vicente, 225 – 22453-900 – Rio de Janeiro - Brazil

{marcelom,endler,karin}@inf.puc-rio.br

Abstract. *Users of mobile social networks continuously generate different kinds of data and content that gets stored in Cloud services. It may vary from raw data acquired from the mobile device's sensors, like GPS coordinates or accelerometer data, to text messages, photos, or videos by which users share presence information, exchange news, express their thoughts or convey their current mood or feelings. The main goal is to share this current information or status with all the contacts in their social network instantaneously. But isn't this large amount of continuously generated content/data a valuable source of global information? We believe that is possible to identify patterns in these data and detect and classify real-world collective situations that are actually happening in a city or region. Thus, in this paper we propose an analytical approach for detecting such situations that are composed of four steps: identification of temporal, spatial, and social correlation, and a final correlation step, that is dependent of the particular type of content/data. Inspired by social networks like Twitter, where is possible to post tweets and associate them to geographic coordinates, we designed the last correlation step as a semantic analysis of these tweets, looking for events such as music- sports- and political-motivated events happening at a certain location in a specific period of time. To evaluate the developed analysis framework, we developed a simulator to generate temporal geo-referenced content of a hypothetical crowd, as if it were generated by real mobile Social Network users. For the data generated by our simulator, we expect to execute our analysis framework to identify patterns characterizing these situations.*

1. Introduction

As happened with most WEB 2.0 services, social networks also migrated to the mobile world. With the proliferation of wireless networks and the evolution of mobile phones to high-end devices with specialized capabilities like GPS and accelerometer sensors and video camera, the use of social services by mobile users became a regular activity. This is confirmed by data presented in several reports, like (Informa, 2008), which

[1] Supported by CNPq 557.128/2009-9 and FAPERJ E-26/170028/2008 grants (Programa INC&T - Projeto: Instituto Brasileiro de Pesquisa em Ciência da Web)

mentions that about 50 million people already use the mobile phone for social networking. Another report (ABI Research, 2008). says that by 2013 there will be approximately 140 million mobile social users. An even more optimistic report states that the number of mobile social users will surpass 800 million by 2012 (eMarketer, 2008).

As users started to access social networking services *anytime* and *anywhere* from their mobile phones, it was possible to notice a change in their behavior: they are much more active in their social communities than non-mobile users, as they continuously generate different kinds of data and content that gets stored in social services. These data may vary from raw data acquired from the mobile device's sensors, like GPS coordinates or accelerometer data, to text messages, photos, or videos by which users share presence information, exchange news, express their thoughts or convey their current mood or feelings.

Observing the rapidly growing number of mobile social users, and the huge amount of data and content generated by them every day, we initiated our research based on the following question: isn't this large amount of continuously generated content/data a valuable source of global information? We believe it is, and that is possible to identify patterns in these data and content in order to detect and classify real-world collective situations that are actually happening in a city or large region.

For this, we propose an analytical approach for detecting such situations that is composed of four correlation steps: identification of temporal, spatial, and social correlation, and a final correlation step, that is dependent of the particular type of content/data. As our first case study, we were inspired by the social network Twitter, where it is possible to post status updates (tweets) associated with geographic coordinates. In this case, our final correlation step would be a semantic analysis of these status updates, where we specifically seek for sport or music events, as well as political-motivated gatherings/demonstrations that are happening at a certain location in a specific period of time.

In order to evaluate and guide the development of the proposed analysis framework, we realized the need of a controlled environment, where we can generate temporal geo-referenced content of a hypothetical crowd, as if it were generated by real mobile social users. We developed a simulator where we can define specific crowd situations that could happen at a certain time and location. With the data generated by our simulator, we intend to develop our analysis framework as a cloud service to identify patterns characterizing these situations.

The rest of the paper is organized as follows. In the next section, we present the definitions that fundament our work and in section 3 we present our simulator of temporal, social and geo-referenced content. Then, in section 4, we describe the model of our analytical approach and how it will execute to identify patterns and detect situations. Section 5 presents some similar works. Finally, section 6 discusses about the challenges we foresee, what is expected to be done in the future and it concludes this position paper.

2. Definitions

For a better understating of our ongoing research we present two important definitions that underlie it: our notion of contents, and what type of content we expect to use to detect real-world collective situations.

2.1. Content

We will consider any kind of content with the following mandatory properties:

- *Time*: the instant of time when the content was created. This information will help us to split the large amount of data and content in time windows where we expect to identify patterns.
- *Place*: the geographical coordinates of the place where the content was created. With these coordinates we expect to identify regions where some situation might be happening.
- *Social identification*: the identification of the mobile social user who created this content. With this identification, we expect to access the mobile social network and correlate it with contents generated by other mobile social users – friends, with same interests, belonging to the same communities, among others.

2.2. Collective situation

We consider as collective situation any situation identified by the sharing of content generated by a group of mobile social users that is relevant to others. Table 1 presents examples of these situations and the type of content used to detect them.

Table 1 - Real-world collective situations and the related content type used to detect them

Real-world collective situation	Content type
Any non-predictable/unplanned public event occurring somewhere in a city and whose occurrence might affect or interest other people, like traffic jams, political demonstrations, celebrations of sports team supporters, ad hoc cultural performances, etc.	Status updates (text messages) from people who are witnessing these events. *For instance, mobile users could update their status with phrases containing the expressions "traffic jam", "car accident", among others.*
Meeting of friends or a certain group of people interested in related topics.	Mobile social users' profiles. *When analyzing these profiles from users located in some region, we believe it is possible to identify similarities between them and detect if a meeting is currently happening.*
Analysis of network connectivity in some region.	Devices' context information. *While receiving the computational context information from mobile devices, we believe it is possible to check the network*

	connection from the devices located in some region and realize if this region has poor quality network connection.

3. Simulator

Our work was inspired by social networks like Twitter, where is possible to post tweets and associate them to geographic coordinates. Twitter itself offers access to its tweets through the Twitter Streaming API (Twitter, 2010). However, in order to guide the development of our analytical approach we noticed the need of a controlled environment, where we can define collective situations to happen in a certain period of time and location and receive content related to it. Thus, we developed a simulator to generate temporal geo-referenced content of a hypothetical crowd as if it were generated by real mobile social network users. For the data generated by the simulator, we expect to use our analysis approach to identify patterns characterizing these pre-defined situations.

Our simulator was developed as a framework in order to generate any kind of content, such as, text messages, devices' context information, among others. To do this, we defined the following concepts for our simulation: *environment*, *devices, attractors* and *launchers*.

3.1. Environment

The *environment* is where the simulation happens and is the starting point of it. It receives a bounding box – with the southwest and northeast geographic coordinates – and the time of execution in minutes. During the simulation, the *environment* is responsible for informing the *devices* that they are inside some *attractors' region of influence*.

3.2. Devices

The *devices* act as mobile social users moving inside the *environment's bounding box*. Each *device* executes as a different thread and has the following properties:

- *Unique identification*: the identification of the device in the simulated social network.
- *Speed range:* the minimum and maximum speed that the device is able to move.
- *Current speed:* the device's current constant speed (for a certain time span) which is constrained by the device's *Speed range*; Thus, current speed changes after each stop.
- *Location:* the device's current geographic coordinates.
- *Stop time:* the time in seconds that a device stands until it moves again.
- *Number of moves:* the number of moves the device makes until it stands.
- *Direction:* the device's direction considering the environment's bounding box limits. This direction changes at every move (after each stop).
- *Execution time:* the time in seconds that the device executes in the simulation.

- *List of related devices' identifications:* the list of devices' identifications related to the device itself. This list identifies relationships (links) in our simulated social network.
- *Content generation delay range:* this delay range defines the minimum and maximum time (in seconds) that the device waits until it generates a content about an attractor.
- *List of contents:* this list defines the contents about attractors the device will generate. A content about an attractor is only inserted in this list if: (i) the device is inside the attractor's region of influence; (ii) the device's interests match attractor's type; and (iii) the result of a calculation considering the device's content generation probability and the attractor's influence factor. Then, the content is associated with a *content generation delay,* the time in seconds the device waits until generates the content. This *content generation delay* is set randomly when the content is inserted in the list and respects the limits established by the device's content generation delay range.
- *Content generation probability:* this value defines the chance of a device to generate content about an attractor.
- *Interests:* the device's interests which will be compared with attractors' types.

The *current speed* and the *direction* are changed randomly during each *device's* execution. All other properties, with the exception of the *unique identification* and the *list of contents*, are defined randomly during *device's* instantiation and remain always the same values.

3.3. Attractors

The *attractors* are the hypothetical real-world situations that happen in the simulation and may influence the *devices'* behavior. Like *devices*, each *attractor* executes as a different thread and has the following properties:

- *Unique identification:* the identification of the attractor.
- *Execution time:* the time (in seconds) that the attractor executes in the simulation.
- *Type:* the type of the attractor.
- *Influence factor:* this value defines the attractor's power to influence devices to generate content about it.
- *Region of influence:* a bounding box with the southwest and northeast geographic coordinates respecting the environment's bounding box.

With the exception of the *unique identification*, all these properties are defined randomly in the moment the attractor is instantiated.

3.4. Launchers

The launchers are responsible for the instantiation and the launch of *devices* and *attractors* in the *environment*. For this, it is needed to inform them the maximum number of *devices* and *attractors* and the *frequency* to launch them.

3.6. Case Study

In our first case study, we instantiated our simulator framework to generate content as text messages regarding sports, music and politics. We defined a fixed content-tree structure related to these types, as shown in Figure 1.

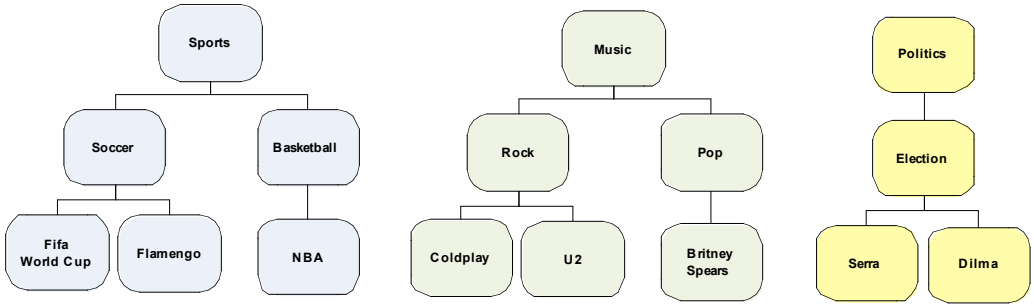

Figure 1 – Topic content-tree structure in our instantiated simulator

Thus, during the execution of our simulation, the *devices* generated content as text messages like *"Fifa World Cup"*, *"U2"*, *"Election"*, among others. These text messages were associated with the *device's identification* and *current location* and the *environment's current time*. The output of our simulation was a large amount of generated content and the list of *attractors* launched during the execution of it.

4. Analytical Approach

We defined our analytical approach as a composition of four steps: temporal, spatial and social correlation steps and a final one, dependent of the particular type of content. The idea is to receive a large amount of content and, after the processing of all steps, to check if the content volume is sufficient to identify a pattern and identifies a collective situation. It is important to notice that a following step will only occur if there is a minimum amount of content to be analyzed. Figure 2 shows the process and the steps involved while analyzing content to identify patterns and detect situations.

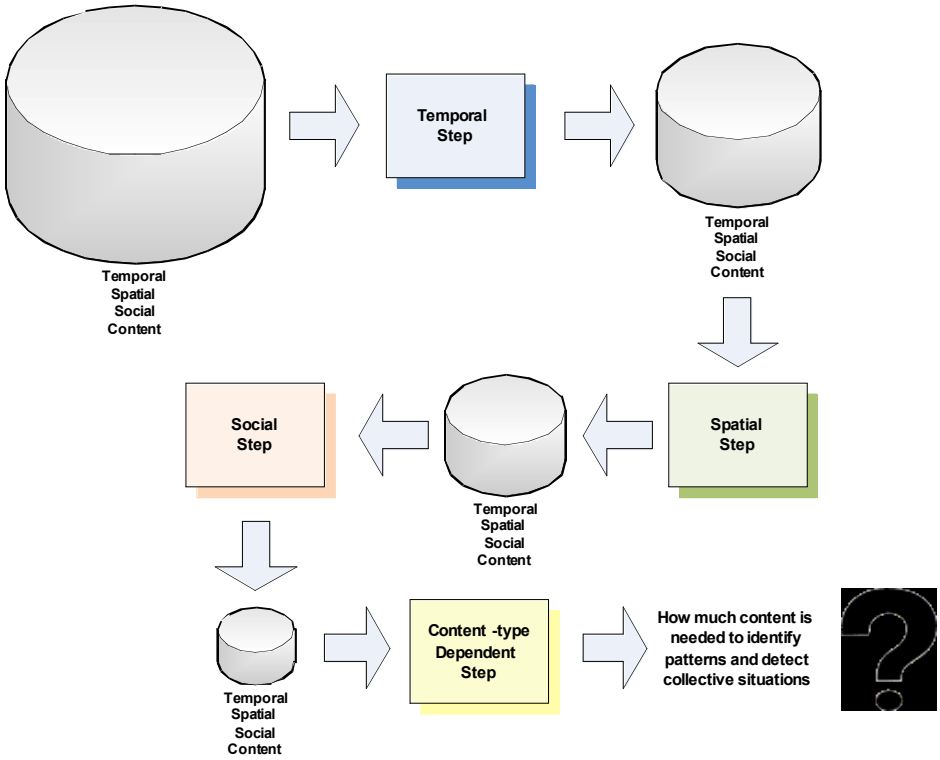

Figure 2 - Analytical steps for searching correlated content

As we expect to deal with a significant large amount of content, we are developing this analysis using the Map/Reduce programming model (Dean and Ghemawat, 2008). This programming model aims the processing and generation of large amount of data, and basically involves two phases: the mapping and the reduction. In the first one, a master node splits the input data and distributes these parts to the *mappers*. Each *mapper* processes its data and returns the answer to the master node. After receiving answers from all *mappers*, the master node starts the reduction phase where the answers are combined and sent to the *reducers*. Each *reducer* processes these combinations and returns a new answer to the master node. These new answers solve the original problem.

4.1. Temporal

The temporal correlation step splits the content by its temporal property. It receives the contents to be analyzed, the start and end time and the duration of the time window. The map phase will split the content in these time windows and the reduction phase will count the number of contents in each time window, as shown in the following algorithm.

```
Map phase:

for each content in Contents do
  timeWindows = getTimeWindow(content.time)
  output.collect(timeWindow, content)
end

Reduction phase (timeWindow):

int soma = 0
for each content in Contents do
  soma++
end
ouput.collect(timeWindow, soma)
```

Only the contents belonging to the time windows which have a greater volume of content than the minimum required follow to the next step.

4.2. Spatial

The spatial correlation step splits the content by its location property. It receives the contents to be analyzed, the bounding box and the minimum bounding box to analyze. Differently from the temporal correlation step, the spatial correlation step executes many rounds of the map-reduction phases in recursion, i.e. for stepwise smaller bounding boxes until it analyzes the minimum bounding box. At each step, it splits the input bounding box into nine nested regions, as shown in Figure 3.

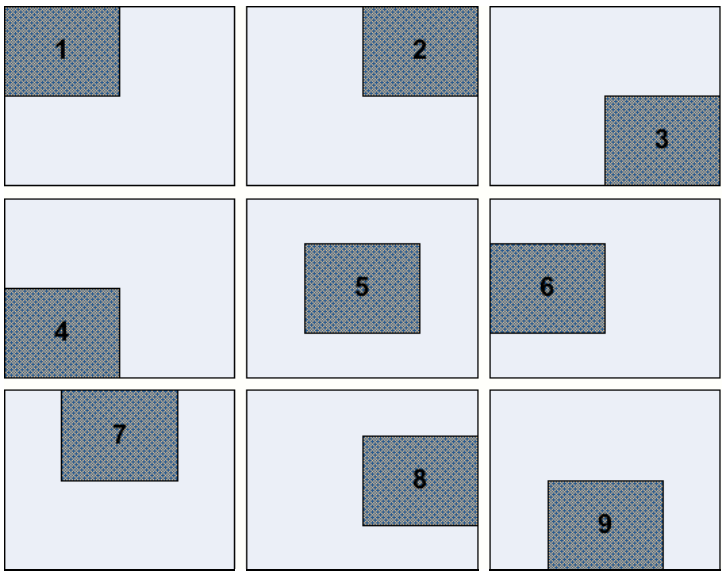

Figure 3 - The analysis of different regions inside a bounding box

For each region, it computes the corresponding volume of content. Only the content belonging to regions which have a content volume that is larger that a minimum volume will proceed to the next round. This recursive refinement is repeated until it is computed the volume of content belonging to the minimum bounding box. The round's algorithm is presented below.

```
Map phase:

for each content in Contents do
  boundingBoxes = getBoundingBoxes(content.Location)
  for each boundingBox in boundingBoxes
    output.collect(boundingBox, content)
  end
end

Reduction phase (boundingBox):

int soma = 0
for each content in Contents do
  soma++
end
ouput.collect(boundingBox, soma)
```

Only the contents belonging to bounding box with a (pre-defined) minimum size, and which have a larger volume of content than the minimum required proceed to the next step.

4.3. Social

The social correlation step is an optional step that analyzes the contents' social properties. It receives the contents to be analyzed and the list of mobile social users and the list of social interests. With the first list, the algorithm verifies if the mobile social users who generate the contents are related to each member of the list. With the list of social interests, the algorithm verifies if social interests from contents' creators belong to this list. The algorithm is shown below.

```
Map phase (mobSocialUsers, interests):

for each content in Contents do
  //Users
  for each user in mobSocialUsers do
    if user.isRelatedTo(content.User)
      output.collect(user, content)
  end
  //Interests
  for each interest in interests do
    if mobSocialUser.isInterestedIn(interest)
      output.collect(interest, content)
  end
end

Reduction phase (mobSocialUser):

int soma = 0
for each content in Contents do
  soma++
end
ouput.collect(mobSocialUser, soma)

Reduction phase (interest):

int soma = 0
for each content in Contents do
  soma++
end
ouput.collect(interest, soma)
```

Only the contents belonging to mobile social users/interests which have a greater volume of content than the minimum required follow to the next step.

4.4. Content-type dependent (Semantic)

The final correlation step is dependent on the content type, and here the pattern identification is performed. In our first case study we used text messages as contents, and so, we defined this step as a semantic correlation step. It receives the contents to be analyzed and the types of situations to be detected, as showed in the following algorithm.

```
Map phase (situations):

for each content in Contents do
  //Situations
  for each situation in situations do
    if isSemanticRelated(situation, content.Message)
      output.collect(situation, content)
  end
end

Reduction phase (situation):

int soma = 0
for each content in Contents do
  soma++
end
ouput.collect(situation, soma)
```

Only the contents belonging to situations which have a greater volume of content than the minimum required are considered for pattern identification.

5. Related Work

Several researchers have highlighted the large-scale social, economic and political implications of small-scale interactions between mobile users. One example is the chains of text messages exchanged by Filipino citizens, causing the resign of President Estrada in 2001. (Katz and Aakhus, 2002) observed that the mobile phone can work as a tool to spur and coordinate the action of masses for political change. (Rheingold, 2002) coined the term '*smartmob*' to refer to the mobile- mediated large-scale mobilizations of people with a common goal. These works testify the use of mobile devices in collective situations. Our ongoing work attempts to automatic identify these collective situations while they are happening.

Other works were more focused in the reasoning of social data in order to identify events that already happened. (Zhao et al., 2007) attempts to detect events on social stream data, e.g. blogs, forums, and emails. It showed that social text stream data contains rich semantics that can be exploited to produce better results than existing state-of-the-art event detection approaches. (Sayyadi et al., 2009) proposed a new algorithm for event detection using the co-occurrence of keywords on historical social data. However, none of the proposed mechanisms are capable of instantly detecting multi-participation spontaneous events and do not consider associating them with geographical locations. The proposed analytical approach is even more generic as we consider events as a just one pattern to be identified.

In (Della Valle et al, 2009) a research about the application of stream reasoning was realized. While the authors suggest that this is a multi-disciplinary and complicated field of research, they consider that applying it to mobile social networking will have high-impact in the area.

6. Future Work and Conclusions

In this position paper we presented our research of an analytical approach to identify patterns and detect real-world collective situations. We will continue the development

of the cloud service responsible for executing the analytical approach. To do so, we intend to use the Apache Hadoop, a Java framework for the storage and process of large amount of data in clusters, for the Map/Reduce programming model (Hadoop, 2010). With the cloud service ready to use, we will execute it to analyze the content generated by our simulator in order to identify patterns characterizing the situations simulated by the attractors. As this is a controlled environment, we will be able to testify the accuracy of our analytical approach. After this, we expect to analyze content from social networks – e.g. Twitter, Facebook, Foursquare – to identify real-world collective situations.

During the evaluation of our analytical approach, we intend to make experimentations while changing the order of the correlation steps. We also intend to verify if the slightly change in parameters such as time window's duration or bounding box's minimum size will make relevant differences to the output situations.

After realizing those experiments, we expect to have a satisfactory level of accuracy in the identification of patterns and detection of real-world collective situations. Then, we intend to make the cloud service available for regular users and to provide a subscription mode for users who want to be notified if a specific collective situation is detected.

References

ABI Research, 2008. Mobile Social Networking. http://www.abiresearch.com/research/1001875-Mobile+Social+Networking. Last access: June, 2010.

Della Valle, E. et al, 2009. Research Chapters in the Area of Stream Reasoning. In: In Proceedings of the 1st International Workshop on Stream Reasoning SR2009, Heraklion - Crete, Greece, June 2009 , Vol. 466CEUR-WS.org , June (2009) , p. 1-9.

Dean, J. and Ghemawat, S. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (Jan. 2008), 107-113.

eMarketer, 2008. Mobile Social Networking Climbs. http://www.emarketer.com/. Last access: June, 2010.

Hadoop, 2010. Apache Hadoop Project. http://hadoop.apache.org/. Last Access: February 2010.

Informa, 2008. Mobile Social Networking growth accelerates: Revenues could reach US$52 billion by 2012; High growth is multi-dimensional, many new opportunities emerging. http://it.tmcnet.com/news/2008/02/11/3262827.htm. Last access: June, 2010.

Katz, J.E., Aakhus, M. (2002) Perpetual contact. Mobile communication, private talk, public performance. Cambridge University Press.

Rheingold, H. (2002) Smarmobs. The next social revolution. Basic books.

Sayaadi, H. et al., 2009. Event Detection and Tracking in Social Streams. In Proceedings of the 3rd International ICWSM Conference (2009).

Twitter, 2010. Streaming API Documentation. http://dev.twitter.com/pages/streaming_api. Last access: June, 2010.

Zhao, Q., et al., 2007. Temporal and information flow based event detection from social text streams. In Proceedings of the 22nd National Conference on Artificial intelligence (Vancouver, British Columbia, Canada, July 22 - 26, 2007).