Expert Systems With Applications 78 (2017) 319–333

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Sensing the city with Instagram: Clustering geolocated data for outlier detection

Daniel Rodríguez Domínguez*, Rebeca P. Díaz Redondo, Ana Fernández Vilas, Mohamed Ben Khalifa

Information & Computing Lab., AtlantTIC Research Center, School of Telecommunications Engineering, University of Vigo, 36310, Spain

ARTICLE INFO

Article history:
Received 3 June 2016
Revised 8 February 2017
Accepted 9 February 2017
Available online 14 February 2017

Keywords:
Data mining
Location-based social network
Crowd detection
Instagram
Density-based clustering

ABSTRACT

Early detection of unusual events in urban areas is a priority for city management departments, which usually deploy specific complex video-based infrastructures typically monitored by human staff. However, and with the emergence and quick popularity of Location-based social networks (LBSNs), detecting abnormally high or low number of citizens in a specific area at a specific time could be done by an expert system that automatically analyzes the public geo-tagged posts. Our approach focuses exclusively on the location information linked to these posts. By applying a density-based clustering algorithm, we obtain the pulse of the city (24 h–7 days) in a first training phase, which enables the detection of outliers (unexpected behaviors) on-the-fly in an ulterior test or monitoring phase. This solution entails that no specific infrastructure is needed since the citizens are the ones who buy, maintain, carry the mobile devices and freely disclose their location by proactively sharing posts. Besides, location analysis is lighter than video analysis and can be automatically done. Our approach was validated using a dataset of geo-tagged posts obtained from Instagram in New York City for almost six months with good results. Actually, not only all the already previously known events where detected, but also other unknown events where discovered during the experiment.

1. Introduction

Online Social Networks (OSNs) have experienced an uninterrupted growth in both number of users and activity: subscribers proactively share text messages, pictures, video... mainly from their mobile devices, which are mostly GPS-enabled. This information, the GPS location, is directly linked as meta-data to their posts and constitutes a data source so important that entailed the emergence of the Location-based Social Networks (LBSNs). The activity in this kind of OSNs is location-centered, i.e. users' experiences are shared at the right place and time they are happening (Forthsquare, Twitter, Instagram, etc.)

Having access to this vast and constantly produced data is specially interesting in the context of smart cities for a wide variety of applications (Scellato, Noulas, Lambiotte, & Mascolo, 2011; Zheng, Xie, & Ma, 2010), like spatial, temporal and social contexts; online relationships; trajectories and mobility patterns (Cheng, Caverlee, Lee, & Sui, 2011); friend-based location recommendations (Ye, Yin, & Lee, 2010); or crowd detection and analysis (Sprake & Rogers, 2014). Precisely in this field, crowd detection and monitoring, LBSNs constitute a good supplement/alternative to the most usual mechanisms based on costly video-surveillance infrastructure monitored by human staff. A system able to analyze the data gathered from LBSNs would automatically detect events or incidents by identifying unusual activity on these OSNs, without any human supervision. Besides, the infrastructure requirements are really low, since citizens are the ones in charge of their own mobile devices. Finally, due to the ubiquity of the LBSNs, the area under analysis may be easily change without any extra investment.

Our proposal focuses exclusively on the analysis of the GPS location information linked to publicly shared posts in LBSNs to detect crowds anomalies in urban areas. Our methodology works in two stages. The first one obtains a pattern of the behavior of citizens in the area under analysis: usual density of posts throughout the geographical area per time of the day and per weekday, i.e. the pulse of the city for 24 h–7 days. In our initial work (ben Khalifa, Redondo, Vilas, & Rodríguez, 2016) the model for the usual activity was taken from only one day, which was considered a typical one. However, in the methodology introduced in this paper, this mechanism has been improved and the pulse of the city is obtained from a much longer accumulative analysis for several

* Corresponding author.
E-mail addresses: daniel@det.uvigo.es (D.R. Domínguez), rebeca@det.uvigo.es (R.P. Díaz Redondo), avilas@det.uvigo.es (A.F. Vilas), mbk@det.uvigo.es (M.B. Khalifa).

http://dx.doi.org/10.1016/j.eswa.2017.02.018
0957-4174/

months, which avoid possible errors due to a potential wrong selection of the reference day. We gathered data from days that could be considered similar, the same weekday in our approach, whose activity is combined to obtain an averaged reference day.

In the second stage, the activity in LBSN in the area is automatically obtained and compared to the pattern in order to detect unexpected behaviors, differentiating between moderate and extreme outliers. This on-the-fly analysis can be automatically performed by an expert system without human supervision and by a quite simple comparison of thresholds. Contrarily to other approaches based on the same data sources (LBSNs) our proposal: (i) provides a pattern of the usual activity in LBSNs for the urban area under analysis; (ii) does not assume a specific number of crowds or a initial position of crowds in the area; and (ii) does not only focus on the detection of unusually big crowds, but on the detection of any kind of abnormal behavior (small crowds, absence or presence of crowds in areas where the normal behavior is the opposite). Finally, our approach was validated using a dataset of geo-tagged posts obtained from Instagram in New York City for almost six months with good results. Actually, not only all the already previously known events where detected, but also other unknown events where discovered during the experiment.

The remainder of this paper is organized as follows. Section 2 provides an overview of other proposals in the crowd and events detection field, especially those which are based on social media data. After summarizing the main techniques used in our approach (clustering and outlier detection) in Section 3, our methodology is detailed in Section 4. The criteria which lead us to use Instagram as data source are explained in Section 5, whereas the experiment is detailed in Section 6. The obtained results are discussed in Section 7 and, finally, Section 8 is devoted to conclusions and future work.

2. Related work

In the crowd analysis field, multiple crowd detection systems have been proposed for different applications, although those for smart cities clearly stand out. Some works are based in the use of video, like low quality infra-red videos, which are used in Nanda and Davis (2002) to detect pedestrians; visible light video, used in Reisman, Mano, Avidan, and Shashua (2004) to detect groups of people from moving vehicles; and sequences of images, like applying optical flow and density-based clustering to detect crowds, their movement and their evolution in Santoro, Pedro, Tan, and Moeslund (2010). Also based in sequences of images, Andrade, Blunsden, and Fisher (2006) proposes applying spectral clustering to find the optimal number of models to represent normal motion patterns, whereas similar techniques are applied in Hamid et al. (2005) and Zhang, Gatica-Perez, Bengio, and McCowan (2005) for unusual events detection. Focused as well in the rare events detection, in Xu, Denman, Fookes, and Sridharan (2016) a weakly supervised approach using Kullback–Leibler (KL) divergence is applied.

However, the analysis of data gathered from LBSNs have become an interesting option. Proactive LBSNs users can be seen as sensors, constantly providing high volume of data of different nature (text, images, temperature, location, etc.) from the same device (mobile devices). Humans as sensors constitute an alternative or supplement to the costly sensing systems which are traditionally deployed all around urban areas. Additionally, and due to the ubiquity of social media, these applications may be easily used in new areas, without the installation and maintenance costs of dedicated sensor networks. This advantages have led to the use of this new data source for crowd analysis as well. In Adedoyin-Olowe, Gaber, Dancausa, Stahl, and Gomes (2016), for instance, tweets are analyzed to extract newsworthy content in sports and politics. Other approaches try to detect natural disasters, like earthquakes (Sakaki, Okazaki, & Matsuo, 2010) or forest fires (De Longueville, Smith, & Luraschi, 2009) from Twitter content (text). TweetTracker (Kumar, Barbier, Abbasi, & Liu, 2011) is another example of a Twitter-based application that helps humanitarian assistance and disaster relief (HADR) organizations to gather information in about the situation to aid disaster relief efforts.

Apart from the analysis of the content itself (mainly text), the location information linked to the shared posts has brought to several approaches to detect crowds and events in urban areas. In Ferrari, Rosi, Mamei, and Zambonelli (2011), most visited locations are detected applying the EM-Algorithm to the location of tweets in intervals of 2 h. This popular places are associated to a ZIP code. Those ZIP codes are processed using Latent Dirichlet Allocation (LDA) to find patterns in the movements of the crowds and track events with a strong relation with the city. LDA is also applied in Chae et al. (2012), in this case to the text content of the tweets, in order to find popular topics. Then an abnormality estimation is calculated using Seasonal Trend Decomposition based on Loess smoothing (STL), in an iterative process which requires expert human supervision. Watanabe, Ochi, Okabe, and Onai (2011) also takes advantage of textual content, since it is first used to assign a location to non-geolocated tweets and, after obtaining the popular locations using Geohash, it is used again to decide if the place is popular due to an unusual event or not. Local events are also the focus in Walther and Kaisser (2013), with an approach which constructs clusters of tweets according to the number of tweets in a given area (density). Then, these clusters are scored according to different criteria: textual content, number of users, number of tweets, etc. In Ranneries et al. (2016) posts from both Twitter and Instagram are clustered according to their hashtags. After that, the density-based clustering algorithm DBSCAN is applied twice to these clusters in order to associate a single place to each cluster. A different clustering approach is presented in Lee and Sumiya (2010), where k-means is used to group the geolocated tweets and define Regions of Interest (RoI). Over these regions, the number of tweets is analyzed in order to detect outliers. The objective is to develop a geo-social event detection to monitor crowd behaviors and local events. The approach introduced in Lee (2012) tries to infer spatio-temporal information about the events mentioned in the shared tweets by applying text mining techniques (a density-based online clustering method), with the aim of detecting events in urban areas. In this approach, the location of a tweeted event is obtained by the text analysis, when the geo-tagged information linked to the tweets are not consistent with the text.

According to the main characteristics of these previous approaches (Table 1), our proposal stands out in the following three aspects. Firstly, we select Instagram as data source after a deep analysis of all the available options (Section 5). In short, it provides higher number of geo-tagged posts, better extraction conditions and have been practically unexplored in similar approaches, where Twitter is clearly the predominant option: only Ranneries et al. (2016) uses also Instagram data, whereas other less common options like Flickr and Youtube are used in Chae et al. (2012). Secondly, we previously establish a pattern of the urban area under analysis, i.e. we obtain the usual location of collectives of citizens all around the area for the different times and weekdays. For this analysis, based on a density-based clustering algorithm, neither the number of cluster nor a initial location of the crowds is needed in advance, as other approaches in the literature like in Lee and Sumiya (2010), where the use of K-means directly entails predefining the number of clusters or in Ferrari et al. (2011), where the clusters are linked to zip codes. In fact, our results only depends on the number of gathered posts and their location, no on any other variable. Finally, and while most approaches only focus on detecting crowded places, our proposal is able to detect four different kinds of outliers or unexpected behaviors: (i)

Table 1
Main features of other proposals.

	Data source	Method	Ref. normal	Objective
(Ferrari et al., 2011)	Geo-located Tweets	EM-Algorithm & LDA	Multiple days ZIP Code	Movement patterns
(Chae et al., 2012)	Twitter Youtube & Flickr	LDA & STL	Usual topics & human support	Abnormal topics
(Watanabe et al., 2011)	Twitter	Geotag, GeoHash, & Terms	Input params	Popular places. Events
(Walther & Kaisser, 2013)	Twitter	Density & Text	Input params	Local events
(Ranneries et al., 2016)	Geo-located Tweets	Hashtags & DBSCAN	Input params	Local events
(Lee & Sumiya, 2010)	Geo-located Tweets	K-means & Num of tweets	Multiple days per RoI	Huge crowds
(Lee, 2012)	Twitter	Density & text	Input params	Events detection
Our proposal	Geo-located Instagram	DBSCAN Avg vs. DBSCAN 1 day	Multiple days. Location dependant	Oultiers in number or possition

unusually big crowds, (ii) unusually small crowds, (iii) crowds in areas that usually have much less activity, and (iv) the absence of crowds in areas that usually have higher activity. In the literature, all the approaches focus on detecting big crowds, none of them face cases (iii) an (iv) and only in Walther and Kaisser (2013) tries also to detect unusually small crowds. Additionally, and for each of these four cases, we define different grades of anomaly, according to the detected activity and the reference obtained for the area, time of the day and weekday.

Of course, our approach can also have some drawbacks, compared to the proposals in Table 1. On the one hand, we are not analyzing the content of the shared posts, i.e. we are not analyzing the text and/or the images in order to find extra information about location, which is done in Ranneries et al. (2016), Chae et al. (2012) and Walther and Kaisser (2013). For those posts that are not geo-located this analysis would enlarge the size of the dataset and, therefore, the number of data points used for the analysis, as it is done in Watanabe et al. (2011). On the other hand, our methodology analyses the gathered posts in small time intervals, but we are not performing a historic analysis of the evolution of the detected clusters at each interval, which is one of the goals in Ferrari et al. (2011).

3. Background on clustering and outlier detection

Clustering is "*the process of partitioning a set of data objects into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters*" (Han, Kamber, & Pei, 2011). There are multiple clustering methods, that are generally classified in the following four categories (Han et al., 2011):

1. Partitioning methods: Data points are organized in a given number k of groups, so there is at least an element in each group. The most well-known example is k-means.
2. Grid-based methods: The object space is quantized into a finite number of cells that form a grid structure on which the clustering is performed.
3. Hierarchical methods: The data set is decomposed into multiple levels, organizing the data into a tree of clusters. It is quite inflexible, since once a step is done it cannot be undone.
4. Density-based clustering methods: The notion of density is used. The general idea is to grow a given cluster while its density excess a given threshold. This way clusters are formed in dense areas while the points in sparse areas are considered as noise.

Clustering is the most appropriate technique for our purpose of detecting crowds: we intend to find groups of points, which represent people, which are similar in terms of their geographic location. More specifically, the chosen method needs to be able to discover clusters of arbitrary shapes, since people do not organize in regular shapes; to handle noise, as points in sparse regions should not be considered to belong in any cluster; and to work without knowing the number of clusters in advance, because the number of crowds can vary depending on several external variables. Taking this considerations into account, density-based algorithms present the best characteristics to our purposes. First, they can discover clusters with arbitrary shapes, while distance-based methods are limited to spherical-shaped clusters. Second, the number of clusters is not needed as a parameter of the algorithm. Finally, density-based methods consider sparse regions as noise. Therefore all the demanded characteristics are fulfilled.

A review of different approaches to density-based methods is performed in NafeesAhmed and Abdul Razak (2014). We mainly considered two of these algorithms: (i) DBSCAN (Ester, Kriegel, Sander, & Xu, 1996a) (Density-Based Spatial Clustering of Applications with Noise) finds core objects (those with dense neighbourhoods) and connect them and their neighbourhoods to form dense regions; and (ii) OPTICS (Ester, Kriegel, Sander, & Xu, 1996b) extends DBSCAN to produce a clustering order, that is, a linear list of all objects under analysis which represents the density-based clustering structure of the data. Both algorithms are similar, but the ordering given by OPTICS allows to obtain the clustering results for a wide set of parameters. Therefore, OPTICS will be more useful in case that testing different parameters over the same data set is needed.

Two parameters are necessary in DBSCAN to define the density measure and obtain the clusters: the radius of a circle around the data point (ϵ) and the minimum number of points that should be in this circle in order to be considered a cluster (*minPoints*). To completely understand the DBSCAN algorithm, some additional concepts need to be defined (see Fig. 1):

1. ϵ-neighbourhood of $x_i := N_\epsilon := \{x_j \in D | dist(x_i, x_j) \leq \epsilon\}$, where the function $dist(x_i, x_j)$ is the distance between x_i and x_j.
2. Core points: $x_i \in D$ is a core point if the $\epsilon - neighbourhood$ of x_i contains at least *minPoints* points.
3. Noise point: $x_i \in D$ is a noise point if the $\epsilon - neighbourhood$ of x_i contains less than *minPoints* points and none of its neighbours is a core point.
4. Directly density reachable: a point x_i is directly density reachable form point x_j if x_i is in the $\epsilon - neighbourhood$ of x_j and x_j is a core point.
5. Density-reachable: a point x_i is density reachable from point x_j if x_i is not in the $\epsilon - neighbourhood$ of x_j and x_j is not a core point but they are reachable through chains of directly density reachable points.
6. Density-connected: two points x_i and x_j are density connected if they are density-reachable from a core point. A cluster is a maximal set of density-connected points.

Taking all that definitions into account, the DBSCAN algorithm works as follows:

1. Given $D = (x_i)_{i=0}^n$ a set of data points, DBSCAN randomly selects any point $x_i \in D$, retrieves all points density-reachable from x_i and check if it is a core point and a new cluster must be created.

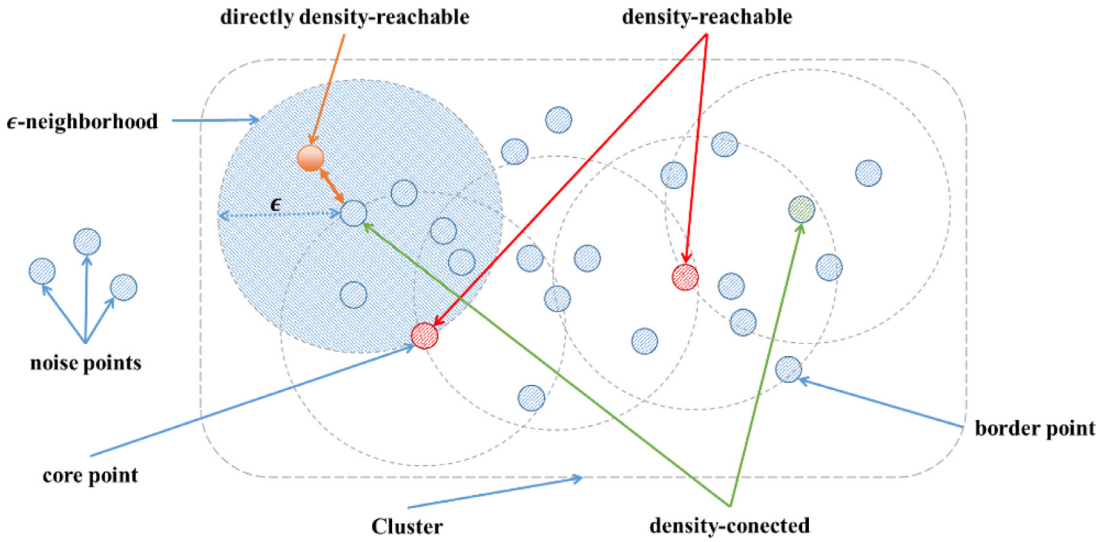


Fig. 1. DBSCAN definitions overview.

2. Iteratively points that do not belong to any cluster and are directly density reachable from the core points of the new cluster are added to it, until it can no longer be expanded.
3. To find the next cluster, an unvisited point is selected randomly and the process continues until all the points are visited.
4. Finally, those points that area not placed in any cluster are considered as noise and is not placed in any cluster.

After being able to detect groups of geographically close citizens with activity in social media (crowds) by using DBSCAN, the second step is defining under which conditions these crowds are considered outliers. According to Hawkins "*an outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism*" (Hawkins, 1980). Outlier techniques can be classified in parametric or non-parametric, depending on whether knowing the distribution of the variables is necessary or not; and in univariate or multivariate techniques, according to the number of variables used.

In our case, an outlier is a cluster whose number of points differs from the number of points of other clusters that were found in a similar location, day and hour. Therefore, we need a univariate technique, since we only consider the number of points as a variable, and non-parametric, since we do not know the distribution of points. There are two main options in these cases (Acuna & Rodriguez, 2004):

For the first method, let $\widetilde{x}$ be the mean and let s be standard deviation of a set of n observations. One observation is considered an outlier if it lies outside of the interval $(\widetilde{x} - ks, \widetilde{x} - ks)$, where k is usually taken as 2 or 3. This choice comes from the assumption of a normal distribution, in which 95% of the data will be inside the interval centred in the mean with a radius equals to two standard deviations, and the whole data in the interval with radius equals to three standard deviations. The main problems of this approach is that the data does not always follows a normal distribution and that the mean and standard deviation are highly sensitive to outliers.

The second option was introduced by Tukey (1977) when Boxplot representation was proposed. This method distinguishes two types of outliers: mild outliers and extreme outliers. An extreme outlier lies outside of the interval $(Q_1 - 3IQR, Q_3 + 3IQR)$, where $IQR = Q_3 - Q_1$ is the Interquartile Range. A mild outlier lies outside of the interval $(Q_1 - 1.5IQR, Q_3 + 1.5IQR)$. Again,the values 1.5 and 3 are chosen by comparison with the normal distribution. However, in this case the estimators are more robust to the presence of outliers, so this will be the better option to our purposes.

4. Methodology

The methodology presented in this document consists in two main phases: (A) Training Phase and (B) Detection phase. The Training Phase objectives are (i) obtaining the location of the Reference Clusters for an average day (usual behavior of the city) and (ii) defining the expected size of the clusters that match each of these Reference Clusters. In other words, it obtains the models which represent general behavior of the citizens. This phase requires combining data from several days that can be considered similar and it is only performed once: before starting the Detection Phase. There are different criteria to decide that different days can be considered with similar behavior, we decided to aggregate for weekdays (Mondays, Tuesdays, etc.), but the methodology is independent of the aggregation rule. The Detection Phase objectives are (i) obtaining the clusters for each individual day and (ii) deciding which of these clusters have an unexpected number of points in comparison to the number of points expected for the Reference Cluster they match. This phase is performed on-the-fly as soon as the data is available. The methodology requires to group the points (shared posts) in temporal chunks (30 min, for instance) in such a way that each temporal interval is independently analyzed. Therefore, the Detection Phase can be performed as soon as the data of a temporal chunk is available.

In Fig. 2 an overview of the steps in both phases is shown, using the 00:00 interval (beginning at midnight, with the chosen length) as an example, as the procedure is the same for each interval. For the Training Phase, the model of the city are obtained of-the-record previous to the Detection Phase, analysing the data from multiple days in the Training Set, by applying the following procedure for each of the intervals in which we divided all the similar days:

1. Parameter estimation: Obtain the necessary parameters for DBSCAN, first for each day, and then the average of all the similar days.
2. Individual clustering: Use DBSCAN to obtain the clusters for each individual day and discard noise points.
3. Reference clustering: Use DBSCAN again to the data of all days together, adapting the parameters to the fact that there are more days, and so more points.

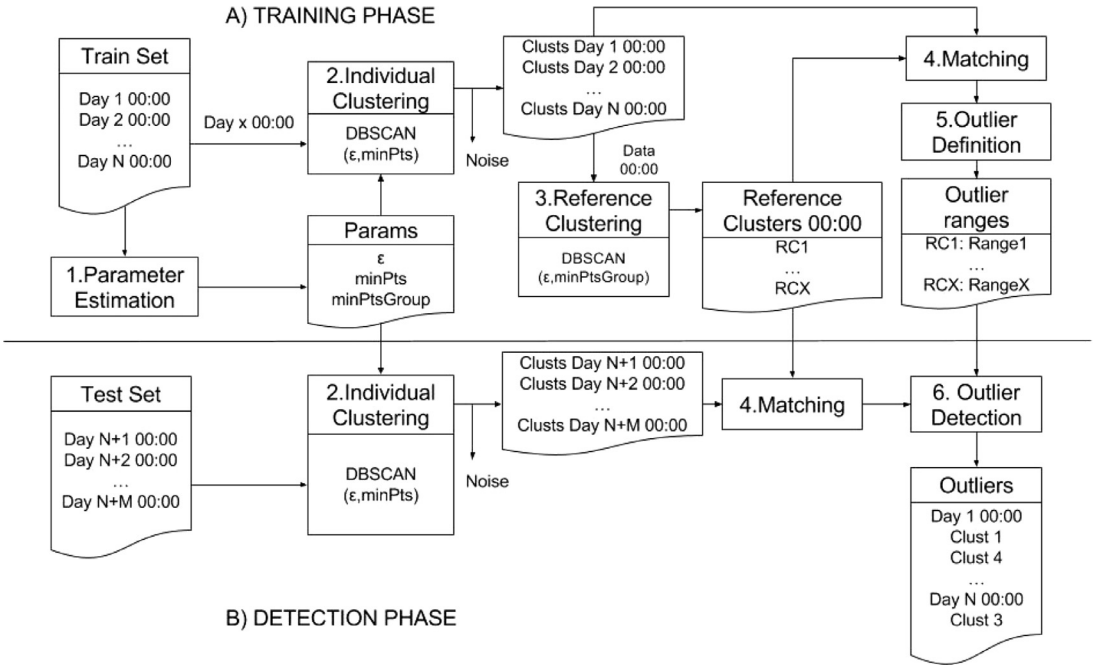


Fig. 2. Methodology overview.

4. Matching: Match Individual Clusters with Reference Clusters based on the distance between them.
5. Outlier definition: Define the outlier limits for each Reference Cluster based on the number of points that all the individual clusters that match it have.

For the Detection Phase, we use the parameters, Reference Clusters and Outlier Limits obtained during the Training Phase, as well as the data from the Test Set, which is obtained and analysed on-the-fly. This means that as soon as the data from a time interval is collected in real time, it can be instantly analysed to detect the outliers by performing the following steps:

1. Individual clustering: Use DBSCAN using the parameters estimated from the Training set.
2. Matching: Match Individual Clusters from the Test set with Reference Clusters obtained from the Training Phase.
3. Outlier detection: Compare the number of points in each Individual Cluster with the limits defined for the Reference Clusters they match.

In the remaining of this section we will give all the details related to each of the steps.

4.1. Parameter estimation and individual analysis

The individual analysis performed to each interval in each day follows a strategy similar to the presented in our previous work (ben Khalifa et al., 2016): we estimate the parameters, ϵ and *minPoints*, and apply DBSCAN. The main difference is in the estimation of those parameters, due to the fact that we are analysing multiple days with the objective of combining the results. Thus, after estimating the parameters for each individual day and interval, we calculate the mean parameters for each interval, combining the results from all the similar days. The whole estimation process is performed as follows (see Fig. 3):

We define ϵ as the mean distance from a point to the nearest neighbour, using the Haversine distance to take into account the fact that the points are in the surface of the Earth. In other words, for each point x_i in the data set we calculate the distance to all the other points and we keep only the minimum distance, $dist_i$. Then we obtain the average for all those minimum distances.

$$dist_i = min(dist(x_i, x_j)), \forall x_j \in D$$

$$\epsilon = \frac{1}{n}\sum_{i=0}^{n} dist_i$$

Once we obtained our estimation of ϵ, we obtain *minPoints* as the mean number of points that a point x_i has in its $\epsilon - neighbourhood$ (a circle with center in x_i and radius ϵ). So we define the function $isNeighbour(x_i, x_j)$ as the function which takes value 1 if the point x_j is in the $\epsilon - neighbourhood$ of x_i, 0 otherwise:

$$isNeighbour(x_i, x_j) = \begin{cases} 1 & \text{if } dist(x_i, x_j) \leq \epsilon \\ 0 & \text{if } dist(x_i, x_j) > \epsilon \end{cases}$$

Therefore, the number of points in the $\epsilon - neighbourhood$ of x_i will be:

$$numPoints_i = \sum_{\forall j \neq i} isNeighbour(x_i, x_j)$$

and we estimate *minPoints* as the ceiling of the average of *numPoints*, which should be always greater equals than 3:

$$minPoints = max\left(\left\lceil \frac{1}{n}\sum_{i=0}^{n} numPoints_i \right\rceil, 3\right)$$

Since we intend to combine several days to obtain a general model, the parameters used in those days should be the same. Therefore, we propose using the mean parameters for each interval. Using the mean parameters that we have estimated, we apply DBSCAN to obtain the individual clustering for the city on each day and half-hour interval. This way we can know the behavior of the city, combine the results from similar days to obtain reference clusters or compare the individual cluster with reference clusters that have already been obtained to detect outliers.

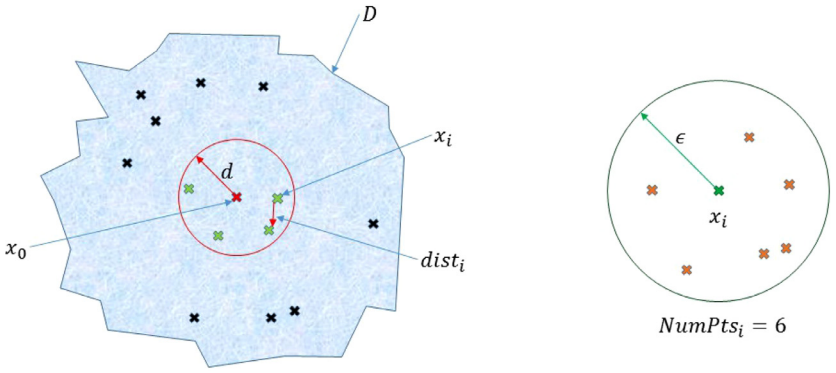


Fig. 3. Estimating DBSCAN parameters.

4.2. Reference clustering

Our next objective is combining the information obtained in the individual clustering in Reference Clusters that represent the behavior of an average day in the city for each one of the half-hour intervals (chunks) in which we divided the days. To this purpose, we remove from the data set the points considered as noise in the individual analysis (those which are in no cluster), and we group the remaining data according to the 30 min interval. The noise points must be removed in order to avoid that they can be combined in an extraordinarily dense cluster and lead to wrong reference clusters.

To obtain the reference clusters, we apply DBSCAN to these new interval-grouped data sets. However, as showed in Fig. 2, we change the *minPoints* parameter for *minPointsGroup*, which is defined as follows:

$$minPointsGroup = N * minPoints$$

where *minPoints* is the value used in the previous section N is the number of days in the training set. This new parameter is defined to reflect the fact that we have merged points from N days, so the minimum number of points in an usually crowded area should be, at least, N times the value we used for an individual day. This approach does not imply that we need a cluster in all the days, but that we have multiple clusters which are together dense enough.

However, this way of obtaining the reference clusters presents a drawback: an extremely dense individual cluster could create a reference cluster by itself if it has more than *minPointsGroup*. However, as the number of days combined is higher this situation will be more unlikely to occur. Furthermore, after obtaining the Reference Cluster we analyze the number of points in each of the clusters which belong to it, so we can detect a Reference Cluster formed by to few clusters. In the following subsection we will explain the outlier detection more in deep.

4.3. Matching and outlier definition and detection

Once we have the Reference Clusters which represent an average day, we need to stablish a method to compare the individual clusters obtained for a single day with the model of its equivalent average day. This requires two different steps: defining when an individual cluster fits a Reference Cluster and so they are comparable, and define the which are the limits in the number of points of a cluster to be considered normal.

In order to match an individual cluster to a Reference Cluster, we need to set a measure of distance between an individual cluster C_x and a Reference Cluster P_y. We define c_{x_i} as the point i in cluster C_x, and p_{y_j} as the point j in the Reference Cluster P_y. We obtain $dist_{x_iy}$ as the distance between the point c_{x_i} and the nearest point p_{y_j} (we do not need to know which point it is exactly, only the Reference Cluster).

$$dist_{x_{iy}} = min(dist(c_{xi}, p_{xj})), \forall p_{y_j} \in P_y$$

We define $dist_{xy}$, the distance between the cluster C_x and the Reference Cluster P_y, as follows

$$dist_{xy} = \frac{1}{n_{C_x}} \sum_{i=0}^{n_{C_x}} dist_{x_{iy}}$$

were n_{Cx} is the number of points in the cluster C_x.

We consider the cluster C_x to fit Reference Cluster P_y if it holds that

$$P_y = \arg\min(dist_{xy})/dist_{xy} \leq \epsilon$$

This definition of distance allows to decide which individual clusters match each Reference Cluster. Looking back to Fig. 2, we can check that matching is applied both in Training Phase, before the Outlier Definition, and in Detection Phase, in order to perform the final Outlier Detection.

In the case of the Training Phase, matching allows us to know which individual clusters led to each Reference Cluster. This way, we can analyse the distribution of the number of points of the individual clusters which form each Reference Clusters and, this way, we can set the limits of a normal behavior. Following the proposals from Tukey (1977) and Acuna and Rodriguez (2004), we use the following definitions for upper and lower outlier limits:

$$UpperOutlier(UO) = Q_3 + \delta * IQR$$

$$LowerOutlier(LO) = Q_1 - \delta * IQR$$

According to Acuna and Rodriguez (2004), and taking as reference a normal distribution, the value of δ is $\delta = 1.5$ for moderate outliers and $\delta = 3$ for extreme outliers. However, we need to take into consideration some special situations and decide how to proceed in those cases:

1. If no cluster fits a Reference Cluster for a given day, we consider that a cluster exists, but it has zero points.
2. Taken the previous point into account, a Reference Cluster will be considered as invalid if $Q_2 = 0$. This means that the area represented by the Reference Cluster is empty more than the 50% of the days. Clusters which are in this Reference Cluster are considered to belong to no Reference Cluster.
3. If two clusters belong to the same Reference Cluster, date and time, they will be considered as a unique cluster. This condition, along with the previous one, allows to have the same number of values to obtain the quartile for all the Reference Clusters.

4. All the clusters from a time interval that do not belong to a Reference Cluster are considered to belong to the same artificial Reference Cluster. This Reference Cluster will be tagged with the number 0, as it is the number used in DBSCAN to tag noise points. Since they are in an usually empty place, they will always be considered as upper outliers, so the Lower Outlier definition in those cases is not used. The Upper Outlier definition will be obtained as usual, with all the points belonging to the Reference Cluster 0.
5. In some cases the limits for Lower Outliers may be negative numbers. Since the number of points is always greater equals to zero, we redefine the Lower Moderate Outlier (LMO) and Lower Extreme Outlier (LEO) limits as follows:

$$LMO = min(Q_1 - 1.5 * IQR, minPoints)$$

$$LEO = min(Q_1 - 3 * IQR, 0)$$

This definition reflects that a cluster has an unusually low number of points when it is lower than the *minPoints* used in the individual clustering, and extremely low when there is no cluster where it should be.

All those issues considered, we define the classification criteria for the individual clusters which match a given Reference Cluster, as well as for those with no matching Reference Cluster. This way, in Detection Phase, after the Matching we can perform the Outlier Detection by comparing the number of points in an individual cluster with the criteria defined for its matching Reference Cluster.

5. Selecting the data source

In order to conduct our experiment we need to gather posts from a OSN which contain location information attached to them. Therefore, the choice of our data source depends on the facilities provided by the API of each OSN to extract and filter the posts and the final amount of publications that can be obtained in the area we are interested in.

Twitter is one of the most well known OSN and it has been used in studies of different fields in which data mining techniques are needed, including crowd and event detection (De Longueville et al., 2009; ben Khalifa et al., 2016; Kumar et al., 2011; Sakaki et al., 2010). The Twitter Search API provides the endpoints to recover tweets that were published in the previous two weeks, with the possibility of filtering according to several criteria, including location. On the other hand, Twitter Streaming API returns 1% of the tweets that match some search parameters in real time. Finally, Twitter Firehose provide access to the 100% of the tweets, but it is not a free-access API. Furthermore, according to Morstatter, Pfeffer, Liu, and Carley (2013), the geo-located tweets returned by the Streaming API cover up to the 90% of the geo-located tweets extracted from Firehose API. However, this study also reveals that the number of geo-located tweets is low, being only a 1.45% of the tweets obtained from Firehose API and a 3.17% of the tweets obtained from Streaming API.

Therefore, although all the facilities provided by the APIs and its popularity make Twitter a good candidate to be our data source, we considered other OSNs needed to be analysed in order to check if some of them provide a higher number of geo-tagged data that can be easily extracted through an Application Programming Interface (API). Foursquare and Instagram were our main options, since some studies have been conducted using data from this two platforms. In the case of Instagram, the behavior of the users, as well as the content of their posts has been analysed in Hu, Manikonda, Kambhampati et al. (2014) and Hochman and Manovich (2013). The latter also includes an analysis of different publication patterns taking into account the location and the time

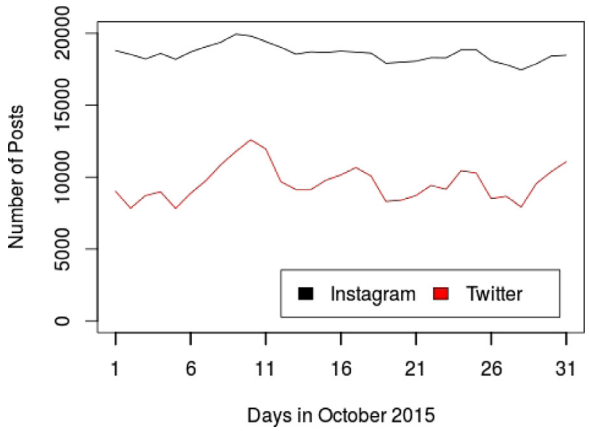


Fig. 4. Comparative of the number of posts in the area of NYC during October 2015.

of the posts. The studies conducted about Foursquare follow a similar line: analysing the emergent patterns among the users (Cramer, Rost, & Holmquist, 2011) and analysing the geographic mobility patterns (Noulas, Scellato, Mascolo, & Pontil, 2011).

Instagram is the most suitable OSN. First, in comparison to Foursquare, which is based on giving opinions about venues and so the location of the posts may be influenced by the location of the venues, Instagram is not biased in that way. For this reason, Instagram allows to recover the posts directly by defining an area and the time, while in Foursquare it is necessary to recover the venues and then analyse the opinions related with them. However Foursquare can be considered for future analysis. Second, due to its growth in the recent years (Duggan, Ellison, Lampe, Lenhart, & Madden, 2015), Instagram already has more monthly active users than Twitter (in January 2016), and a higher number of the posts contain information about location. In Fig. 4 we compare the number of posts extracted daily from both Twitter and Instagram in an circle with radius 5 km around Times Square (New York City). Although here we only show the comparison for the data extracted in October 2015 as an example, the number of posts is always higher in Instagram.

Instagram API provides equivalent functionalities to Twitter Search and Streaming, allowing to extract data published in the past or in real time respectively, but with and additional advantage: while Twitter Search API only searches against a sampling of recent Tweets published in the past 7 days, Instagram allows to recover data from any moment in the past, so it provides more flexibility to choose the dates for the studies. Another points to take into is the rate limitations: there is a maximum number of calls per time interval that can be performed to the APIs. Instagram "media/search" endpoint (search posts in a circle defined by center and radius) is only affected by the general limitation per user (500 calls/h for Sandbox mode, 5000 calls/h for Live mode), while Twitter search/tweets endpoint (general method for searching tweets) is limited both per user (180 calls/15 min) and per application (450 calls/15 min), and Twitter Streaming API limitations are not public, apart from the fact that only a live stream can be opened with the same credentials and it returns 1% of all the published tweets.

All of that taken into account, we decided to conduct our experiment using the data extracted from Instagram, since it contains a higher number of posts that Twitter does; data is always available to be extracted, so the data set can be expanded at any time or re-extracted if needed; and the rate limits, despite being smaller in Instagram for a single user, cause no problems since there is no limit for an application and parallel extraction can be performed using access tokens from multiple users. More

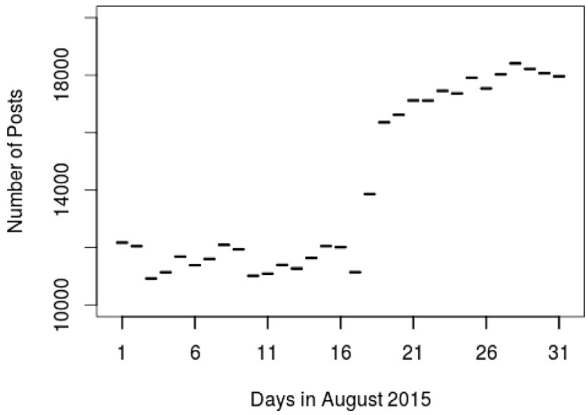


Fig. 5. The number of posts remains stable until 17th August, increases on 18th-19th August and from then on it keeps stable.

specifically, we gathered the data using the media/search endpoint from Instagram API. Each call to this endpoint returns the most recent posts, up to 20 results, in a circular area, so it requires as parameters the latitude and longitude where the center of the area is set; the radius of the circle, which can be up to 5 km; and the minimum and maximum date of the posts. The data is extracted using a script in R, which performs iterative calls to the API, obtaining the data in JSON format and converting it to a R data frame, which can be saved in R native format.

6. Experiment

In order to test our methodology, we conducted a particularized experiment for modelling an predefined area and detecting crowds, as well as uncommon behaviors. The idea is to analyse the citizen's habits according to the time and the geographical area. For a given area, we obtain the usual location of the crowds for each half-hour interval in each day of the week by combining the data from several days, as well as the usual size range of the crowds in each of this specific locations. Then, once we have modelled an average whole week in 30 min intervals, we test the model against two different days: a day that can be considered as normal and another day in which citizens are expected to have an uncommon behavior due to a special event. For this experiment we selected New York City and analysed the impact that Storm Jonas (Saturday 23rd January 2016) had on the behavior of the citizens. We compare both the crowds detected the day of the storm (special day) and a week before (Saturday 16th January 2016, normal day) with the averaged model of the city obtained by the combination of the data from previous Saturdays.

6.1. Data description

We extracted geo-tagged data using Instagram API, setting the center of the area in Times Square (40.756667 N, 73.986389 W) and using the maximum radius allowed (5 km). The period of extraction goes from the 1st June 2015, to match the dates with the data that we had previously extracted using Twitter Streaming API, to 24rd January 2016, the first Sunday after Jonas Storm. This dataset contains data from 34 weeks (238 days) and a total of 4,557,165 posts, including both normal days and special days, due to festivities like Christmas or natural phenomena like the weekend when Storm Jonas hit the United States, which can be used to test the outlier detection, since this dates are supposed to have an uncommon behavior (higher densities in Christmas, lower during the Storm).

Table 2
Post distribution per day.

Day	Total	Daily avg
Mon	506,479	23,021.77
Tue	495,620	22,528.18
Wed	489,545	22,252.05
Thu	490,591	22,299.59
Fri	508,247	23,102.14
Sat	503,879	22,903.59
Sun	503,851	22,902.32
	3,498,212	22,715.66

Table 3
Post distribution per month.

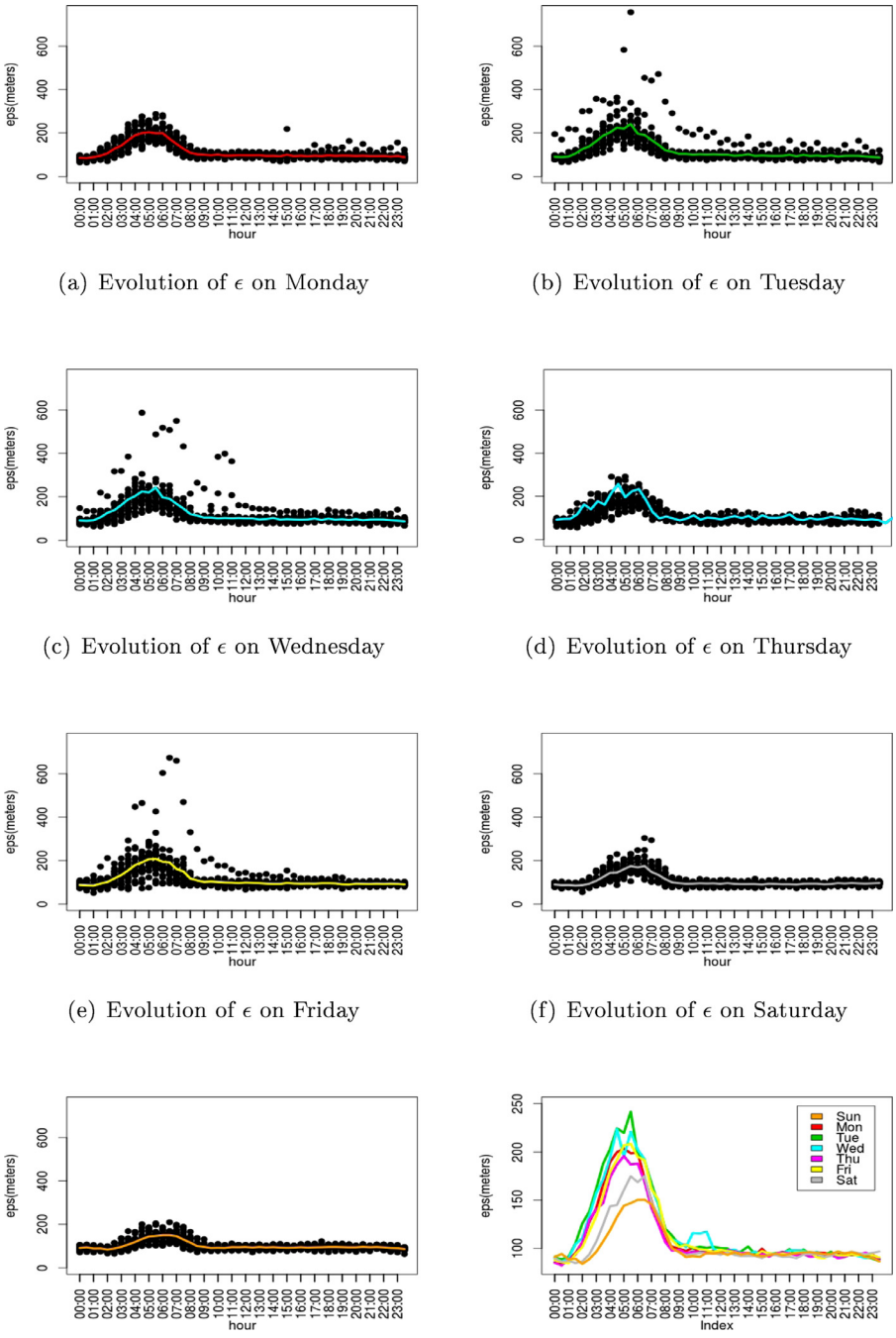
Month	Total	Daily avg
2015-08	162,602	20,325.25
2015-09	688,405	22,946.83
2015-10	720,607	23,245.39
2015-11	666,256	22,208.53
2015-12	697,927	22,513.77
2016-01	562,415	23,433.96
	3,498,212	22,715.66

After revising the number of posts extracted per day (Fig. 5) we notice an abrupt increasing on the 18th August 2015. Before that date the number of posts moves around 11,000, while after the 19th August it is always higher than 16,000. Since the number of posts affect to the estimations of the DBSCAN parameters and to the DBSCAN algorithm itself, we decided to discard the data previous to 24th August and work only with the data after this date, as it keeps the number of points stable and we want to have whole weeks. This leaves us with a data set containing 22 weeks (154 days) and 3,498,212 posts with the distributions per day of the week and per month showed in Tables 2 and 3 respectively. Note that we only keep last week in August, and the last week in January is missing.

Therefore, from this 22 weeks we use the first 20 as the training set, and the last two as a test set. This data needs to be prepared before starting with the crowd detection and analysis. Apart from removing incomplete or duplicated entries, we divide the data set into chunks of 30 min since the behavior of the city is not expected to be the same during the whole day and we want to detect more instantaneous changes. On the other hand, we group those chunks belonging to the same day of the week and 30 min interval (for instance, all the chunks that belong to different Sundays at 11:00 a.m). This is because we assume that the behavior of the city is similar every week and thus we can find a model for each of this groups of data by analysing them together as we explained in the Section 4. With the training set we obtain the average location and size of the crowds in the area for each day of the week at each half hour interval. With the test set we check the validity of the model against two different weeks: the first is considered completely normal, while the second will be influenced by Jonas Storm as we approach to the weekend.

7. Results

We applied our method to the data extracted divided in 30 min interval, obtaining for each step the results for individual days and then averaging them to obtain a model of each interval in each day of the week. First, we obtain the parameters for DBSCAN (ϵ and *minPoints*), both for each individual day and the average for each day of the week. Then, using the average parameters, we obtain the clusters for each individual day, and combine them according to the day of the week, interval and location, to obtain the Reference Clusters which represent the zones which are usually crowded. Finally, for each of those Reference Clusters, we obtain



(a) Evolution of ϵ on Monday

(b) Evolution of ϵ on Tuesday

(c) Evolution of ϵ on Wednesday

(d) Evolution of ϵ on Thursday

(e) Evolution of ϵ on Friday

(f) Evolution of ϵ on Saturday

(g) Evolution of ϵ on Sunday

(h) Comparison of ϵ on each day

Fig. 6. Evolution of ϵ parameter on each day of the week.

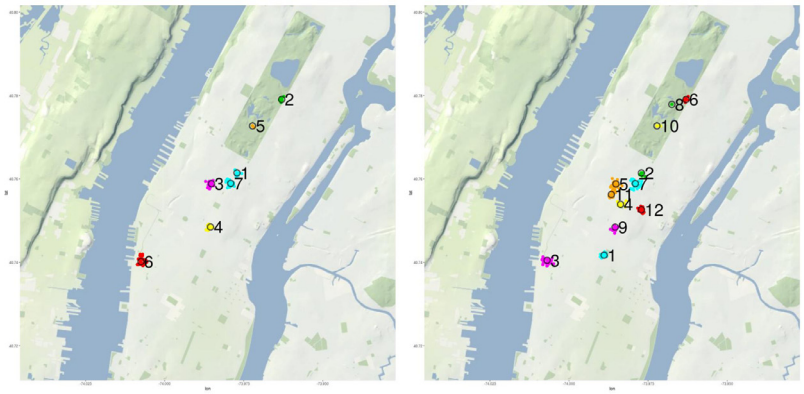
a set of limits which bound the normal values for the number of points in the area. Using these limits we can detect when the behavior of the people in a zone is abnormal.

We will test the model for New York City comparing it with different days. First we use the Saturday in which Storm Jonas hit New York City (23rd January 2016), which can be considered as an unusual day, and the previous Saturday (16th January 2016), which can considered as normal. The expected result for this analysis is that during the Storm some of the usually crowded areas are empty since people may decide to stay at home instead of going out. Additionally we will show the individual clustering for two of the days used to build the model, since they help to demonstrate some special cases that need to be taken into consideration.

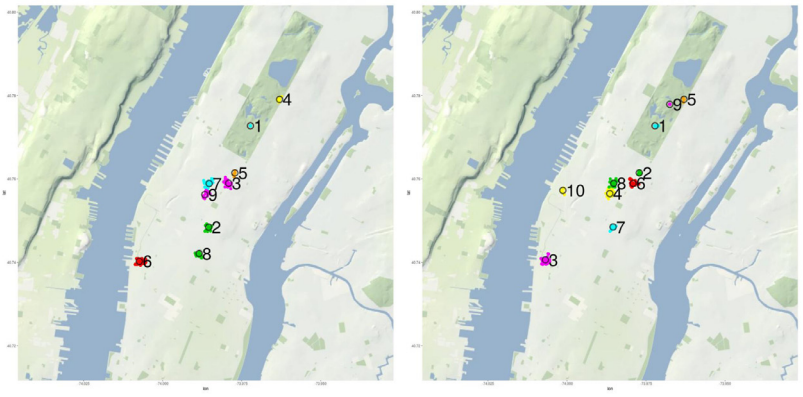
7.1. Model of the city

In order to be able to decide if any given day presents an uncommon behavior at a given obtained the Reference Clusters for every interval in each day of the week. Our purpose is to model the average behavior of the citizens during a whole week, so any new given day can be compared with its correspondent model to detect anomalies.

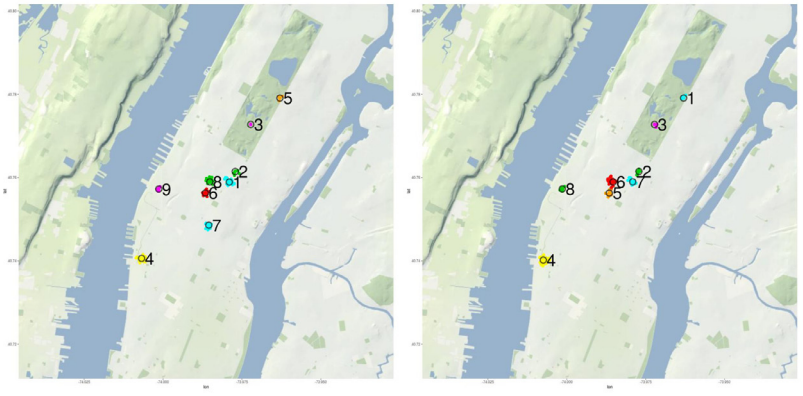
First we need to check whether the assumption that days which are the same day of the week have a similar behavior. To this purpose, we check the estimation of the ϵ parameter used in DBSCAN. In Fig. 6 we can see the evolution of mean ϵ during every day of the week, along with the values estimated for each individual day.



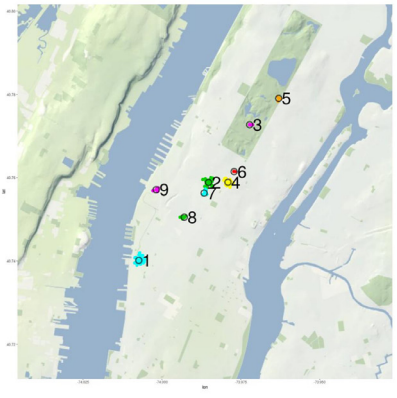
(a) Monday (b) Tuesday



(c) Wednesday (d) Thursday

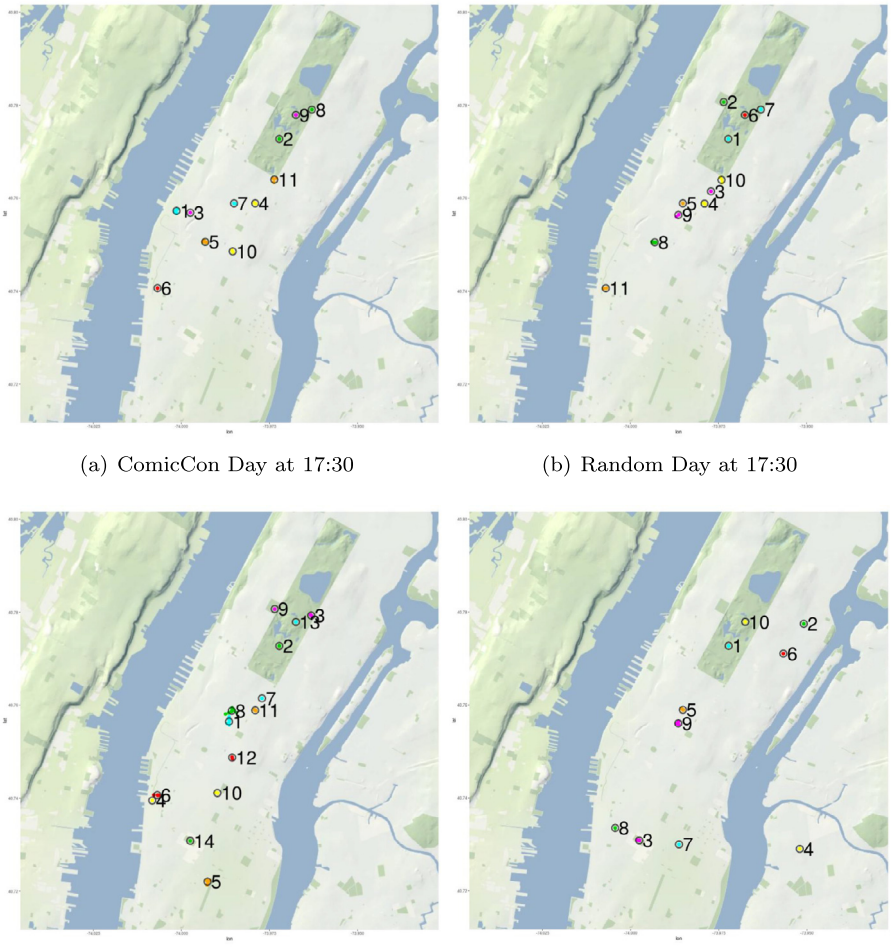


(e) Friday (f) Saturday



(g) Sunday

Fig. 7. Reference clusters for each day of the week.



(a) ComicCon Day at 17:30 (b) Random Day at 17:30

(c) Saturday before Jonas Storm at 17:30 (d) Jonas Storm Day at 17:30

Fig. 8. Individual clustering for different days.

The first thing that we observe is that all the days of the week follow a similar pattern: the parameter ϵ is higher during the hours when people are usually sleeping, growing from 1 a.m. to 6 a.m and decreasing from 6 a.m to 9 a.m, when it reaches a value which keeps stable during the rest of the day. Therefore, during the night the average distance to the nearest neighbour represented by ϵ is higher, since there are a lower number of posts due to the fact that people is normally sleeping.

This fact is also reflected in the differences between days at this early hours. In Fig. 6(h) we can see that ϵ is lower on Saturday and Sunday at these early hours. Furthermore, the increase of the value starts later, about 2 or 3 a.m., representing the fact that these are the days in which people usually go out at night. This facts lead to a higher number of posts with the consequent lower value of ϵ.

On the other hand, these early hours can be also harder to model and predict accurately, due to the higher dispersion in the values of ϵ in those intervals. This reflects the fact that, while during the day the citizens usually follow a more recognizable pattern due to their daily tasks, their behavior at night can be more changeable, since they are not doing any mandatory task. An extreme case of this higher dispersion can be observed in Fig. 6(b), (c) and (e). The estimated parameter for an individual day is extremely higher than the other values, due to a smaller number of posts extracted on this day. However, while in the early hours the differences are high, in the second half of the day the estimated values are closer to the average.

Using the mean parameters for DBSCAN, we obtain the Reference Clusters for each interval in each day of the week. In Fig. 7 we compare the Reference Clusters obtained at 14:00. Note that numbers were assigned randomly for each day, so we will use numbers from Friday as reference. First thing we can note is that the general appearance is similar for all the days: the centre of the map, around Times Square, has always some cluster. Clusters are also expected to appear in the area of Central Park (clusters 3 and 5) and at the park in Von Furstenberg Sundeck (cluster 4).

On the other hand, we can appreciate some differences: First, at weekends there are no clusters near the Empire State Building (cluster 7). On Tuesday and Wednesday we can see clusters near the Madison Square Garden (cluster 1 on Tuesday 7(b) and 8 on Wednesday 7(c)). Finally, from Thursday to Sunday there is a cluster on the left of the map (clusters 10, 9, 8 and 9, respectively). However, we found out that this clusters appear due to a high concentration of people in the area during the four days when the ComicCon was celebrated in the Jacob K. Javits Center (8th–10th October 2015). This kind of situations are more exceptional as we combine more days to obtain the model. Furthermore, even when this occurs, the extremely dense individual cluster generated by the event is always identified as an outlier, while smaller clusters in the area from other days are considered as normal.

7.2. Testing the model

To test our model, we take a normal day and an unusual day corresponding to the same day of the week. This way we can check them against the same Reference Clusters and see the differences. The selected day of the week is Saturday, and the

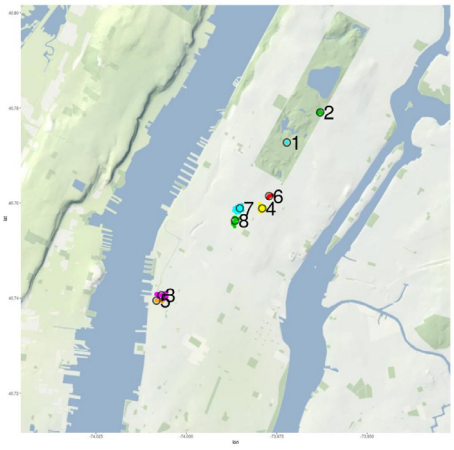


Fig. 9. Reference clusters at 17:30.

Table 4
Reference Clusters at 17:30.

ID	Q1	Q2	Q3	UMO	UEO	LMO	LEO
0	6.00	7.00	9.00	13.50	18.00	–	–
1	18.75	21.50	25.00	34.38	43.75	9.38	0
2	6.00	7.00	10.25	16.62	23.00	6	0
3	6.00	8.00	12.75	22.88	33.00	6	0
4	8.00	10.00	19.25	36.12	53.00	6	0
5	0.00	6.00	10.00	25.00	40.00	6	0
6	0.00	8.50	11.50	28.75	46.00	6	0
7	0.00	11.50	14.25	35.62	57.00	6	0
8	0.00	5.00	11.25	28.12	45.00	6	0

Table 5
Reference Clusters at 14:00.

ID	Q1	Q2	Q3	UMO	UEO	LMO	LEO
0	5.00	6.00	8.00	12.50	17.00	–	–
1	5.00	6.00	8.50	13.75	19.00	5	0
2	6.75	8.50	11.00	17.38	23.75	5	0
3	15.75	22.00	25.75	40.75	55.75	5	0
4	10.75	13.50	15.50	22.62	29.75	5	0
5	0.00	2.50	10.50	26.25	42.00	5	0
6	7.00	13.00	17.50	33.25	49.00	5	0
7	3.75	8.00	18.00	39.38	60.75	5	0
8	0.00	0.00	7.00	17.50	28.00	5	0

Table 6
Clusters which fit the Reference Cluster 8 (Comic Con area).

Day	Points	clustType
2015-09-19	8	N
2015-10-10	89	UEP
2015-10-17	9	N
2015-10-24	7	N
2015-10-31	16	N
2016-01-09	7	N

specific days are 16th January 2016 (a week before Jonas Storm) and 23rd January 2016 (Jonas Storm), both included in our test set. In Fig. 8 we show the clusters detected for this two days in the interval that starts at 17:30. This Figure also includes the clusters for two days included in the training set, which show the influence of unusual days in the obtained model: 10th October 2015 (Comic Con) and 28th November 2015 (a randomly chosen day). Note that the cluster numbers and colors are assigned randomly in each individual clustering, so there is no relationship between clusters with the same number in different images or between clusters with the same colors, both in the same or different images.

Table 7
Clusters (C) on 16th January 2016 matched with Reference Clusters (RC) at 17:30.

RC	C	Points	Type
0	9	6	PO
0	13	7	PO
0	14	7	PO
0	12	7	PO
0	5	7	PO
0	10	7	PO
1	2	24	N
2	3	6	N
3	6	7	N
4	11	12	N
5	4	9	N
6	7	13	N
7	8	13	N
8	1	9	N

Table 8
Clusters (C) on 23th January 2016 matched with Reference Clusters (RC) at 17:30.

RC	C	Points	Type
0	8	7	PO
0	4	6	PO
0	7	14	PO
0	3	9	PO
0	10	6	PO
0	2	9	PO
0	6	7	PO
1	1	41	UMO
2	–	0	LEO
3	–	0	LEO
4	–	0	LEO
5	–	0	LEO
6	–	0	LEO
7	5	7	N
8	9	8	N

Analysing the map we can see some interesting facts:

1. In Fig. 8(a) we can see two clusters (1 and 3) which are not in any other day. Those clusters are around the location of the Jacob K. Javits Convention Center, where the ComicCon is celebrated.
2. Some areas are crowded the three first days, or even on the day of the Storm. For example, taking as reference the numbers in the Fig. 8(a), clusters 2, 4, 6, 8 and 9 are shared with the other days.
3. The day of the Storm (Fig. 8(d)) the distribution of the clusters is completely different, as there are much less clusters in the center of the map.

The Reference Clusters of the area at 17:30 are showed in Fig. 9, evidencing that the Reference Clusters appear in the areas in which there are clusters in multiple days. In this case, unlike what we have seen in Fig. 7(f), the clusters from the Comic Con day are not dense enough to create a Reference Cluster by themselves.

The next step is to stablish which individual clusters match the Reference Cluster, using the distance formula we defined in Section 4.3. Once we have related clusters to Reference Clusters we obtain the limits for the outlier detection, following the methodology and the considerations explained in 4.3.

In Table 4 we show the values of the quartiles and the limits for Upper Moderate Outlier (UMO), Upper Extreme Outlier (UEO), Lower Moderate Outlier (LMO) and Lower Extreme Outlier (LEO) estimated each Reference Cluster for the interval at 17:30. The IDs are the same used in Fig. 9, with the additional ID = 0, which rep-

Table 9
Type of cluster count comparison between 16th and 23rd January 2016.

Hour	PreJonas						Jonas					
	LEO	LMO	N	PO	UMO	UEP	LEO	LMO	N	PO	UMO	UEP
00:00	2	0	3	6	0	0	1	1	3	5	0	0
00:30	1	0	6	5	0	0	0	0	8	5	0	0
01:00	2	0	5	4	0	0	2	0	5	6	0	0
01:30	1	0	6	2	1	0	2	0	5	5	1	0
02:00	1	0	6	3	0	0	1	0	5	10	0	0
02:30	1	0	7	9	0	0	0	0	8	11	0	0
03:00	2	0	5	7	0	1	1	0	5	10	1	0
03:30	0	0	9	9	1	2	1	0	6	17	1	2
04:00	1	1	3	13	0	1	1	0	6	6	0	0
04:30	1	0	4	11	0	1	2	0	5	8	1	1
05:00	0	0	7	6	1	0	1	0	6	11	0	0
05:30	1	0	3	4	0	1	0	0	6	2	1	0
06:00	1	0	5	11	0	1	0	0	8	12	0	0
06:30	0	0	5	4	1	0	0	0	5	5	1	0
07:00	2	1	1	1	1	0	1	0	5	5	0	0
07:30	4	0	1	2	0	1	1	0	4	2	0	0
08:00	3	0	2	1	1	0	0	0	6	4	0	0
08:30	3	0	4	1	1	0	2	0	6	3	1	0
09:00	2	0	3	1	0	0	2	0	3	4	0	0
09:30	2	0	5	3	0	0	3	0	4	2	0	0
10:00	1	0	6	3	0	0	3	0	4	4	1	0
10:30	1	0	6	5	0	0	3	0	4	5	0	0
11:00	0	0	10	7	1	0	6	0	3	8	0	0
11:30	3	0	5	8	1	0	5	0	3	11	1	0
12:00	2	0	8	9	0	0	3	0	7	9	0	1
12:30	2	0	8	10	0	0	6	0	4	10	0	1
13:00	1	2	6	6	0	0	6	0	3	6	0	1
13:30	1	0	8	7	0	0	5	0	4	9	1	0
14:00	1	0	8	6	0	0	4	0	4	8	0	0
14:30	0	0	7	5	0	0	4	0	2	10	1	0
15:00	0	0	8	10	0	0	3	0	3	9	0	1
15:30	0	0	6	7	0	0	4	0	2	6	0	0
16:00	1	0	6	7	1	1	4	0	3	11	1	0
16:30	0	0	8	9	1	0	3	1	3	11	1	0
17:00	1	0	7	9	0	0	4	0	3	9	1	1
17:30	0	0	8	6	0	0	5	0	2	6	2	0
18:00	0	0	8	5	0	0	3	0	4	9	1	1
18:30	0	1	7	5	1	0	4	0	3	6	1	0
19:00	1	0	7	5	0	0	4	0	2	5	1	0
19:30	1	0	6	3	0	0	4	0	3	5	0	0
20:00	2	0	5	4	0	0	4	0	2	5	2	0
20:30	2	0	5	5	0	0	3	0	3	8	1	1
21:00	0	0	6	6	0	0	2	0	4	8	0	0
21:30	0	0	7	5	0	0	4	0	3	8	0	0
22:00	0	0	6	8	0	0	3	0	2	6	1	1
22:30	1	1	5	4	0	0	3	0	3	4	1	0
23:00	2	0	5	5	0	0	3	0	2	7	2	0
23:30	1	1	5	6	0	0	4	0	3	7	1	0

resents the clusters which are in no Reference Cluster. Note that for the Reference Cluster 0 only the Upper Extreme Outlier is meaningful, since the clusters will always be considered as outliers.

In Table 5 we show the same data for the Reference Clusters at 14:00, and we can see that $Q2 = 0$ for the Reference Cluster 8, which indicates that this Reference Cluster was created in spite of the place being empty more than the half of the days. Therefore, this Reference Cluster is probably formed due to a high concentration of posts in the area for an individual day. Table 6 confirms our suspicions, since on 10th October 2015 the are many more points than on the other days. An analysis of the content of the posts extracted on this day reveals that the Comic Con took place in the Jacob K. Javits Convention Center, right in the location where the cluster is. However, note that the cluster found on this day is considered an Extreme Upper Outlier, even when the Reference Cluster was formed due to its influence.

In Tables 7 and 8 we show the comparison of 16th January and 23rd January with the Reference Clusters at 17:30. For each cluster we show the Reference Cluster related to it, the number of points, and the type of cluster according to the limits defined in Table 4. We define 6 types of cluster according to its position and number of points, represented with the following letters: N (normal), PO (position outlier, those clusters which are in no Reference Cluster and have an small number of points), UMO (Upper Moderate Outlier), UEO (Upper Extreme Outlier), LMO (Lower Moderate Outlier) and LEO (Lower Extreme Outlier). From those table we can see that, while all the clusters on the 16th fit the predictions, four of the places that are usually crowded (Reference Clusters 2–6) are empty on the day of the storm (clusters with negative cluster number). Furthermore, cluster 1 (Reference cluster 1, at Central Park) show an unexpectedly high number of points during the storm. Analysing the textual information included in the data we discovered that many people went there to try to enjoy the snow at Central Park. However, this textual analysis and its automation is not the purpose of our experiment, and it has been done by hand in this specific case.

For a more general view of the differences between both days, in Table 9 we show the number of clusters of each type in both days for each 30 min interval in which we divided the data. We can see that during the night the behavior of both days is similar,

but at 9:00 a.m. the number of Lower Extreme Outliers grows for the day of the storm, being always higher than in the normal day, in which we only have one or two of these type of outliers. We also see more Upper Outliers in these hours. In general, the behavior of the city during the day is similar to the described for the interval at 17:30 and the model was able to detect that in a special day Instagram users vary their habits, representing the expected behavior of the citizens.

8. Conclusions and further work

The growing number of citizens proactively using mobile devices to share facts and opinions in Social Media and LBSNs can be considered as sensors, providing additional information to the data retrieved from the existing infrastructure sensors that are already installed in smart cities. The information extracted from these users on-the-move provides real-time geo-located readings which can be used for multiple purposes. Our approach applies this information to model the usual activity of the citizens by using the public geo-tagged posts from LBSNs to, later on, detect unexpected behaviors.

Consequently, the proposed methodology consists of two phases. In the Training Phase, we analyze the distribution of people in the urban area under study in chunks of 30 min for several months to finally combine the intervals corresponding at both the same day of the week and time interval. The objective is to obtain patterns of the usual posting activity of the citizens for each interval and weekday. Each pattern characterizes the activity of the area by means of a set of clusters representing the usual number of shared posts. Therefore, and contrarily to other proposals in the literature, we are obtaining a global characterization of the usual activity in LBSNs of citizens in the selected urban area. By using data from a continuous period of time (several months) to model the usual behaviour we have improved our early work (ben Khalifa et al., 2016) by removing the problem of selecting a day that can accurately represent the general behavior of the city. Additionally, and since we are using DBSCAN, the division of the region in different areas does not need to be performed in advance, but are obtained directly by the clustering algorithm. In other words, the algorithm only needs the points all over the area, and it will identify regions with any arbitrary shape which have a high concentration of points.

In the Detection Phase we compare the current activity in the social media stream on-the-fly with the Reference Cluster that is located in the same area on the same day of the week and at the same time interval. Therefore, the outliers detection is not performed equally for every cluster, but locally. This means that, instead of comparing the number of points of all the clusters in a wide area, a cluster is only compared with the nearest Reference Cluster (if there is one which is near enough to be considered as comparable). This way, a more detailed view of the area can be obtained, as well as a more precise detection method. This allows our methodology to increase the detection capabilities, which are clearly wider than the ones offered by other approaches in the literature. Instead of uniquely addressing the problem of big crowds detection, our proposal is able to also detect unusual small crowds, crowds in areas with low activity and absence of crowds in areas with high activity.

The methodology was tested using Instagram, since this LBSN offers a high number of geo-tagged posts and suitable mechanisms to gather this data. The results were promising, detecting all the events that were previously knows and even others that were unknown by the time of the analysis and discovered as consequence of this, like the Commic Con Day in NYC. Anyway, and depending on the analysis ultimate goal, other considerations could be done. On the one hand, a different temporal interval could be selected, instead of the 30 min used in our experiment. However, it is worthy to note that longer intervals would deal with more general view of the city life, whereas shorter intervals would provide more accurate results. On another hand, the criterion used to decide which days behave in a similar way could be discussed. Opposite to our aggregation according to the day of the week, the data set could had been divided in working days and weekends, simplifying the model. Additional considerations can be added to those divisions, like seasonal changes. Again, the final choice would mainly depend on the application, as well as on the available data set.

The main advantage of using LBSNs as data source to sense the city and detect anomalies is clearly the absence of specific and costly infrastructure, like the usual sensor and video-surveillance networks. Thus, our proposal can be applied in any urban area without a previous investment. However, the social media stream analysis is not intended to replace other more sophisticated video-based analysis. On the contrary, the social media stream analysis, like the one we propose, could be consider as a early filter to detect potential anomalies in specific areas. This filter should trigger other more complex systems to study in detail only those video frames of interest that can offer additional information about what is happening in the area, like the one proposed in Xu et al. (2016). With the aim of increasing the number of geo-tagged posts for the analysis, other approaches that perform text mining techniques to the content of the posts (text) could be applied in order to infer the location of those messages that do not include the GPS information. The work in Lee (2012) faces this issue and extract spatio-temporal information to enrich the datasets. Our approach offers a horizontal solution not focused to a specific thematic field (accidents, sport events, street performance, etc.). However, changing the objective to detect unexpected activity in social media related to a specific field of interest, other approaches might be runned in parallel, like the one described in Adedoyin-Olowe et al. (2016), where the analysis of short posts focuses on both sports and political events detection.

Apart from adding other approaches to provide further information about what is behind an unusual behavior in the city (video and/or text analysis), another interesting research line is monitoring the evolution of the detected crowds in the urban area. The final objective would be trying to predict potential merges of small crowds as well as potential split-ups of big crowds. Even more, a space-temporal analysis could give information of the trajectories and movements in short-term.

Acknowledgements

This work is funded by: the European Regional Development Fund (ERDF) and the Galician Regional Government under agreement for funding the Atlantic Research Center for Information and Communication Technologies (AtlantTIC) and the Spanish Ministry of Economy and Competitiveness under the National Science Program (TEC2014-54335-C4-3-R).

References

Acuna, E., & Rodriguez, C. (2004). A meta analysis study of outlier detection methods in classification. *Technical paper*. Department of Mathematics, University of Puerto Rico at Mayaguez.

Adedoyin-Olowe, M., Gaber, M. M., Dancausa, C. M., Stahl, F., & Gomes, J. B. (2016). A rule dynamics approach to event detection in twitter with its application to sports and politics. *Expert Systems with Applications, 55*, 351–360.

Andrade, E. L., Blunsden, S., & Fisher, R. B. (2006). Modelling crowd scenes for event detection. In *Pattern recognition, 2006. ICPR 2006. 18th international conference on: 1* (pp. 175–178). IEEE.

Chae, J., Thom, D., Bosch, H., Jang, Y., Maciejewski, R., Ebert, D. S., & Ertl, T. (2012). Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *Visual analytics science and technology (VAST), 2012 IEEE conference on* (pp. 143–152). IEEE.

Cheng, Z., Caverlee, J., Lee, K., & Sui, D. Z. (2011). Exploring millions of footprints in location sharing services.. In *ICWSM: 2011* (pp. 81–88).

Cramer, H., Rost, M., & Holmquist, L. E. (2011). Performing a check-in: Emerging practices, norms and'conflicts' in location-sharing using foursquare. In *Proceedings of the 13th international conference on human computer interaction with mobile devices and services* (pp. 57–66). ACM.

De Longueville, B., Smith, R. S., & Luraschi, G. (2009). Omg, from here, i can see the flames! A use case of mining location based social networks to acquire spatio-temporal data on forest fires. In *Proceedings of the 2009 international workshop on location based social networks* (pp. 73–80). ACM.

Duggan, M., Ellison, N. B., Lampe, C., Lenhart, A., & Madden, M. (2015). Social media update 2014. *Pew Research Center, 19*.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996a). A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd: 96* (pp. 226–231).

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996b). A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd: 96* (pp. 226–231).

Ferrari, L., Rosi, A., Mamei, M., & Zambonelli, F. (2011). Extracting urban patterns from location-based social networks. In *Proceedings of the 3rd ACM sigspatial international workshop on location-based social networks* (pp. 9–16). ACM.

Hamid, R., Johnson, A., Batta, S., Bobick, A., Isbell, C., & Coleman, G. (2005). Detection and explanation of anomalous activities: Representing activities as bags of event n-grams. In *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on: 1* (pp. 1031–1038). IEEE.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques*. Elsevier.

Hawkins, D. M. (1980). *Identification of outliers*: 11. Springer.

Hochman, N., & Manovich, L. (2013). Zooming into an instagram city: Reading the local through social media. *First Monday, 18*(7).

Hu, Y., Manikonda, L., Kambhampati, S., et al. (2014). What we instagram: A first analysis of instagram photo content and user types.. In *Icwsm* (pp. 595–598).

ben Khalifa, M., Redondo, R. P. D., Vilas, A. F., & Rodríguez, S. S. (2016). Identifying urban crowds using geo-located social media data: A twitter experiment in new york city. *Journal of Intelligent Information Systems*, 1–22.

Kumar, S., Barbier, G., Abbasi, M. A., & Liu, H. (2011). Tweettracker: An analysis tool for humanitarian and disaster relief.. In *Fifth international AAAI conference on weblogs and social media* (pp. 661–662).

Lee, C.-H. (2012). Mining spatio-temporal information on microblogging streams using a density-based online clustering method. *Expert Systems with Applications, 39*(10), 9623–9641.

Lee, R., & Sumiya, K. (2010). Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *Proceedings of the 2nd ACM sigspatial international workshop on location based social networks* (pp. 1–10). ACM.

Morstatter, F., Pfeffer, J., Liu, H., & Carley, K. M. (2013). Is the sample good enough? Comparing data from Twitter's streaming api with Twitter's firehose. arXiv preprint arXiv:1306.5204.

NafeesAhmed, K., & Abdul Razak, T. (2014). A comparative study of different density based spatial clustering algorithms. *International Journal of Computer Applications, 99*(8), 18–25.

Nanda, H., & Davis, L. (2002). Probabilistic template based pedestrian detection in infrared videos. In *IEEE intelligent vehicle symposium: 1* (pp. 15–20).

Noulas, A., Scellato, S., Mascolo, C., & Pontil, M. (2011). An empirical study of geographic user activity patterns in foursquare.. *ICwSM, 11*, 70–573.

Ranneries, S. B., Kalør, M. E., Nielsen, S. A., Dalgaard, L. N., Christensen, L. D., & Kanhabua, N. (2016). Wisdom of the local crowd: Detecting local events using social media data. In *Proceedings of the 8th ACM conference on web science* (pp. 352–354). ACM.

Reisman, P., Mano, O., Avidan, S., & Shashua, A. (2004). Crowd detection in video sequences. In *Intelligent vehicles symposium, 2004 IEEE* (pp. 66–71). IEEE.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake shakes twitter users: Real–time event detection by social sensors. In *Proceedings of the 19th international conference on world wide web* (pp. 851–860). ACM.

Santoro, F., Pedro, S., Tan, Z.-H., & Moeslund, T. B. (2010). Crowd analysis by using optical flow and density based clustering. In *Signal processing conference, 2010 18th European* (pp. 269–273). IEEE.

Scellato, S., Noulas, A., Lambiotte, R., & Mascolo, C. (2011). Socio-spatial properties of online location-based social networks.. In *ICWSM: Vol. 11* (pp. 329–336).

Sprake, J., & Rogers, P. (2014). Crowds, citizens and sensors: Process and practice for mobilising learning. *Personal and Ubiquitous Computing, 18*(3), 753–764.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, Mass.

Walther, M., & Kaisser, M. (2013). Geo-spatial event detection in the twitter stream. In *European conference on information retrieval* (pp. 356–367). Springer.

Watanabe, K., Ochi, M., Okabe, M., & Onai, R. (2011). Jasmine: A real-time local-event detection system based on geolocation information propagated to microblogs. In *Proceedings of the 20th ACM international conference on information and knowledge management* (pp. 2541–2544). ACM.

Xu, J., Denman, S., Fookes, C., & Sridharan, S. (2016). Detecting rare events using Kullback–Leibler divergence: A weakly supervised approach. *Expert Systems with Applications, 54*, 13–28.

Ye, M., Yin, P., & Lee, W.-C. (2010). Location recommendation for location-based social networks. In *Proceedings of the 18th sigspatial international conference on advances in geographic information systems* (pp. 458–461). ACM.

Zhang, D., Gatica-Perez, D., Bengio, S., & McCowan, I. (2005). Semi-supervised adapted HMMs for unusual event detection. In *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on: 1* (pp. 611–618). IEEE.

Zheng, Y., Xie, X., & Ma, W.-Y. (2010). Geolife: A collaborative social networking service among user, location and trajectory.. *IEEE Data Engineering Bulletin, 33*(2), 32–39.