TweeProfiles: Detection of spatio-temporal patterns on Twitter*

Tiago Cunha, Carlos Soares, and Eduarda Mendes Rodrigues

Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n
4200-465 Porto Portugal
{tiagodscunha, csoares, eduarda}@fe.up.pt

Abstract. Online social networks present themselves as valuable information sources about their users and their respective behaviours and interests. Many researchers in data mining have analysed these types of data, aiming to find interesting patterns. This paper addresses the problem of identifying and displaying tweet profiles by analysing multiple types of data: spatial, temporal, social and content. The data mining process that extracts the patterns is composed by the manipulation of the dissimilarity matrices for each type of data, which are fed to a clustering algorithm to obtain the desired patterns. This paper studies appropriate distance functions for the different types of data, the normalization and combination methods available for different dimensions and the existing clustering algorithms. The visualization platform is designed for a dynamic and intuitive usage, aimed at revealing the extracted profiles in an understandable and interactive manner. In order to accomplish this, various visualization patterns were studied and widgets were chosen to better represent the information. The use of the project is illustrated with data from the Portuguese twittosphere.

Keywords: Data Mining; Clustering; Spatio-temporal patterns; Visualization

1 Introduction

In recent years, social media services have adapted the paradigm of social networks and achieved a huge importance in social life and also in business strategies for companies, as they are regarded as a timely and cost-effective source of spatio-temporal and behavioural information [1]. The massive adhesion and the number of platforms that provide social interaction lead to a growth in the data stored within these services and its usage by researchers.

Twitter has proven to be a popular data source within social media due to the large number of active users and the easy access to their public API. As such, it has fuelled several studies [2–4]. Twitter data can probably be organized into

*An earlier version of this work was presented at the Encontro Nacional de Inteligência Artificial e Computacional (Brazilian AI meeting - ENIAC).

subgroups that represent profiles of tweets and, thus, of the users. These profiles can be useful for many tasks (marketing, political science, government, product development, etc.). However, given the amount of data as well as its complex nature (space, time, content and social), these patterns are not easily extracted using classical data mining strategies. Moreover, the ability to assign different importance to each data dimension may prove useful to control the data mining process and further reveal hidden patterns.

Therefore, our goal is the design and development of an automatic tool to extract and display the patterns mined. The contributions of this project are: the development of a Data Mining process that enables a weighted combination of multiple clusterings in various dimensions, and a web application that facilitates a flexible and interactive exploration of those patterns. This is indeed the greatest contribution of this paper: a weighted combination of multiple clusterings, each one obtained in various heterogeneous dimensions (i.e. time, space, content and social relations). The approach is applied to data from the Portuguese twittosphere to illustrate the type of patterns that can be obtained.

This paper is organized as follows: Section 2 contains the state of the art for the scientific fields of clustering algorithms and distance measures. In Section 3, the concepts and decisions for the data mining process are explained in detail, namely the distance measures used and the combination process. Section 4 presents the visualization tool developed to represent and analyse the patterns extracted with our methodology. In Section 5, we present some results obtained in the case study considered in this project. Finally, in Section 6 we list our conclusions and tasks for future work.

2 Related Work

Clustering is formally defined as the process of grouping a set of data objects into multiple groups, or clusters, so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters [5]. Similarity assessment is calculated through distance functions. Clustering is the logical choice for our project: clustering algorithms are able to extract patterns from unlabelled data, such as Twitter posts; and clustering provides groupings of similar objects, which can be regarded as representing profiles of tweets.

Clustering algorithms can be sorted into 4 different types: Partitioning, Hierarchical, Density-based and Grid-based [5]. Partitioning algorithms organize the objects to create partitions accordingly to a particular criterion. They are known for generating clusters of spherical shape by using distance-based techniques to group the objects. They generally use mean or medoid to represent cluster centres and have proven effective up to medium size sets [5]. Within this set of algorithms, the most well known are k-Means and k-medoids [5].

A Hierarchical clustering method works by grouping data objects into a hierarchy or a tree of clusters [5]. This method can either be agglomerative (if it starts with small clusters and recursively merge them to find a single final cluster) or divisive (all objects are in a single cluster and iteratively are divided

until each one has only one object or the objects are very similar). Usually, the results of hierarchical algorithms are represented by a dendrogram (i.e. tree diagram). The most representative algorithms within this class are BIRCH [6] and Chameleon [7].

Density-based clustering algorithms follow the strategy of modelling clusters as dense regions in the data space, separated by sparse regions [5]. The most well known algorithms are DBSCAN [8] and DENCLUE (DENsity-based CLUstEring) [9].

DBSCAN finds core objects (i.e. points with dense neighbourhood) and iteratively connects them to the neighbours if these are in the core object's ε--neighbourhood. The ε -neighbourhood is defined through a user defined parameter: the radius ε and states that a point is in the core object's ε--neighbourhood if it is within the pre-defined radius. Therefore, for two points p and q, we can say that p is directly density reachable from q if it is in the ε--neighbourhood of q. Another user input is *MinPts* that determines if a point is a core object. If within the ε-neighbourhood there are at least *MinPts* points, then we are in the presence of a core object. The algorithm takes in account the two previous concepts and iteratively connects core objects to its ε-neighbourhood until all objects are processed.

Grid-based algorithms use a space-driven approach instead of a data-driven approach as in the previous algorithms [5]. They partition the space into cells of a multi-resolution grid data structure. This ensures a fast processing time independent from the size of the data set, although it is affected by the resolution of the grid. One notable examples of this clustering algorithms is STING (STatistical INformation Grid) [10].

Clustering algorithms need distance functions in order to calculate dissimilarities between objects and to group these objects by similarity. The objective function aims for high intra-cluster similarity and low inter-cluster similarity [5]. The distance functions chosen for clustering depend on the data types and the representation spaces. Therefore, one must divide the different distance functions accordingly to the dimensional types we use. In this paper we consider the following dimensions: spatial, temporal, content and social. In the spatial dimension, data is defined by latitude and longitude, which are numeric values extracted from the tweets. Therefore, similarity functions between numeric values must be explored. The 4 most important distances of this type in a euclidean space are the Euclidean Distance, the Manhattan Distance, the Minkowski Distance, the Mahalanobis Distance and the Chebychev Distance [5]. For the specific case of latitude and longitude coordinates there is a better suited distance measure that considers the earth's shape: the haversine distance.

As far as the temporal dimension goes, contrary to the spatial dimension, which is mapped in R^2, time is represented in R, which facilitates the difference calculation. For each pair of tweets, the timestamp values are used to compute the temporal distance. However, any of the previous distance functions for euclidean space is applicable to the temporal dimension. We note that this distance function is limited to explore the time difference between posts, whereas other

patterns could be mined, such as for instance seasonality of events. In this way, clusters could reflect events in the same weekday, for instance, and therefore extract events than happen regularly.

Considering the connections between users, it is possible to assume the existence of a social graph in Twitter. Therefore, the social distance can be simplified to a distance between nodes within a graph. Two distance measures for graphs are the Geodesic Distance and SimRank [5]. Network Similarity [11] and a pseudo-distance measure [12] can also be used.

In order to calculate the similarity between two texts, one must explore Text Mining distance functions. The Cosine similarity distance [5] (as the most commonly used) and a variation denominated Tanimono distance. Lastly, a variation of Jaccard similarity complemented with Dice's coefficient [13] can also be used. In order to apply a distance function on text, document representations must be specified in a previous stage. The main idea is to create a document-term matrix and extract the vectors to compute their dissimilarity. The main document representation techniques are TF [14] and TF-IDF [15].

3 Clustering on multiple dimensions

We present in this section our methodology to tackle this problem, including the distance functions used, the clustering algorithm chosen and the strategy proposed for combining dimensions.

Although many other distance functions than those presented here provide a more elaborate and possibly more powerful approach, they tend to be more difficult to interpret. This decision is transversal to all the distance functions chosen, since our approach must provide a visual platform to navigate through the patterns. It is therefore essential to use intuitive metrics that are easily represented and interpreted.

We consider that each tweet is formally defined as t_i, where i is the index identifier on the tweet data collection. The distance functions between two tweets t_i and t_j are defined as $dist^X(t_i, t_j)$, where X is the dimension on which the function maps the values. X can take the values Sp, T, C, So which are related respectively to the spatial, temporal, content and social dimensions.

In order to calculate the spatial distance between two points, these must be mapped in space. The data received from Twitter for the spatial dimension consists of the latitude and longitude of each tweet. For each pair of tweets t_i and t_j, the distance function uses the latitudes ϕ_{t_i} and ϕ_{t_j} and longitudes λ_{t_i} and λ_{t_j} to determine the distance. The value R is the earth's radius:

$$dist^{Sp}(t_i, t_j) = 2Rsin^{-1}\left(\left[sin^2\left(\frac{\phi_{t_i}-\phi_{t_j}}{2}\right) + cos\phi_{t_i}cos\phi_{t_j}sin^2\left(\frac{\lambda_{t_i}-\lambda_{t_j}}{2}\right)\right]^{0.5}\right)$$

The temporal distance was simply calculated as the difference of timestamps, in seconds. The conversion to seconds is necessary due to the use of a generic

clustering algorithm, which is, thus, unable to process timestamped differences. For each pair of tweets t_i and t_j, the timestamp values Δ_i and Δ_j are used to compute the distance:

$$dist^T(t_i, t_j) = |\Delta_i - \Delta_j| \tag{1}$$

To calculate text similarity, the choice fell on TFIDF for vector representation and cosine similarity to determine the dissimilarity between two texts. TFIDF was chosen to reduce the importance of frequently used words and therefore give preference to discriminative ones. On the other hand, the cosine similarity is a traditional yet powerful metric also used in many papers [13, 16].

For a tweet t_i we define the tweet text as α_i and calculate its TFIDF representation in a document matrix D. The following equation elucidates on this transformation, where TF represents a term frequency and IDF represents Inverse Document Frequency:

$$TFIDF(\alpha_i, D) = TF(\alpha_i, D) * IDF(\alpha_i) \tag{2}$$

The cosine function takes as input two TFIDF representations, β_i and β_j, for two tweets, t_i and t_j, and returns a similarity value between 0 and 1, which we adopt for the distance between texts:

$$dist^C(t_i, t_j) = 1 - \frac{\beta_i.\beta_j}{||\beta_i||\beta_j||} \tag{3}$$

Finally for social distance, the geodesic distance between the users who make the posts in the graph defined by the follow relations was chosen. This distance measure introduces a problem that was not present in the previous measures: the possibility of infinite distances. These occur when two nodes are not connected (neither directly nor through other nodes). This case must be treated before combining distances for all dimensions. We replace the infinite value for the total number of nodes plus one. This ensures that the distance between two users that are not connected is always greater than the distance between any two connected users.

All these distance measures were applied to all pairs of tweets and stored in dissimilarity matrices for further processing. The dissimilarity matrices are formally defined as D^X, where X is the dimension. X can take the values *Sp*, *T*, *C*, *So* which are related respectively to the spatial, temporal, content and social dimensions. Within each dissimilarity matrix D^X, the distance value in row i and column j is defined as $d^X_{(i,j)}$. These matrices are hereby defined as single dimensional distance matrices.

The combination of distances refers to the weight system applied over the all four dimensions to obtain a new multi-dimensional dissimilarity matrix. The weights determine the importance of the dimensions, given as percentage values, while the overall result shows the importance that should be assigned to each dimension. The multi-dimensional dissimilarity matrix will contain the sum of all dissimilarities multiplied by the weight value for the corresponding dimension,

always ensuring that this sum is equal to 100%. More formally, for each weight value w_{Sp}, w_T, w_C, w_{So}, we have that $w_{Sp}, w_T, w_C, w_{So} \in \{0, 0.25, 0.5, 0.75, 1\} \mid w_{Sp} + w_T + w_C + w_{So} = 1$.

However, in order to obtain relevant results, one must normalize the single dimensional distance matrices, since the scales for each dimension are different. The purpose is to use a similar scale for all dimensions to ensure that the importance of each dimension is determined by the weights and not its scale. Therefore, a min-max normalization [5] was applied to all single dimensional distance dissimilarity matrices D^X with the goal of switching from a scale $[min_{D^X}, max_{D^X}]$ into a new scale $[newmin_{D^X}, newmax_{D^X}]$ which is similar in all dimensions. The normalizing equation is applied to all distance values $d^X_{(i,j)}$ in order to obtain the normalized values $d'^X_{(i,j)}$.

In our problem, the minimum value must always be zero since this is the lowest possible value for distance. Therefore, the normalization formula can be simplified, since both min_{D^X} and $newmin_{D^X}$ equals zero:

$$d'^X_{(i,j)} = \frac{d^X_{(i,j)}}{max_{D^X}} * newmax_{D^X} \tag{4}$$

The application of this formula results in scales that lie between 0 and $newmax_{D^X}$. This is also important for defining the clustering algorithm parameters which directly depend on the scale of each dissimilarity matrix.

The multi-dimensional dissimilarity matrix of all four dimensions D^{4D} is the sum of each single dimensional distance matrices $(D^{Sp}, D^T, D^C, D^{So})$ multiplied by the respective weight value $(w_{Sp}, w_T, w_C, w_{So})$:

$$D^{4D} = w_{Sp}D^{Sp} + w_T D^T + w_C D^C + w_{So} D^{So} \tag{5}$$

The clustering algorithm was then applied to the multi-dimensional dissimilarity matrix, in order to obtain the clusters to display in the visualization tool. The clustering algorithm chosen was DBSCAN due to various reasons:

- Density based algorithms are able to detect arbitrarily shaped clusters [17].
- It is not necessary to specify the number of clusters to calculate. This is important in our project because one of the goals is to find patterns although there is no guarantee that the data contain clusters. The enforcement of the number of clusters is unsuitable for some real world applications [17].
- Microblog messages contain noise and using a density based approach, this noise is considered as an outlier and filtered from the results [17].

4 TweeProfiles Tool

The visualization tool is responsible for showing the information mined in the most intuitive and simple way possible, while enabling interaction. To achieve such goals, we represent the 4 dimensions as:

- Spatial: map. The clusters are represented as red circles while the tweets as dots. Each dot is colored depending on the cluster it belongs. Click events are associated to both tweets and clusters to display further information;
- Temporal: timeline. It containing bars with the start and end date for each cluster and with a click event associated to each cluster;
- Content: wordcloud. It displays the words in a cluster with sizes proportional to their frequencies;
- Social: minimum spanning tree. It contains the most important users in a cluster (There is a limitation of 10 users to be displayed in the tool) and a click event to view the user's profile in Twitter;

For the map, the Google Maps JavaScript API[1] was the choice, due to the completeness of information displayed, intuitive interactivity, ability to integrate data from various sources and JavaScript libraries and the typically fast response time to load and refresh the webpage.

D3[2] was used to display the tweets, plotted on top of the map. This representation was overlapped on the map using a Google Maps class named Overlay. An Overlay containing the tweets is anchored to specific latitudes and longitudes so when the map is changed, the points change accordingly. Each point is also associated with an on-click event listener to show a tooltip with the related data to the point specified. If the points are not clustered, they are not presented in the visualization tool to avoid displaying unnecessary information.

The last objects to be displayed on the map are the cluster circles. These circles were generated using Google Maps Polygons and are also inserted in an Overlay class. The event associated with clicking on this circle is the display of a new division in the webpage. This division is used to present a summary of cluster information and also a D3-based interactive social graph representing the most relevant users in the cluster and their connections.

Besides the map representation, the timeline is also used to present all the clusters. The JavaScript library used was the Timeglider JQuery plugin.[3] This widget requires only the input of a JSON file with a specific syntax and it does all the mapping of clusters in horizontal bars. It guarantees that the bars displayed do not overlap and therefore all clusters are accessible through mouse events.

When a single bar is clicked on, it shows a tooltip with a summary of information for the specific cluster. On the other hand, by simply hovering the mouse on the bar, the cluster summary division appears, in the same manner as when the clusters circles in the map are clicked on.

The final widget we implemented is the social graph through a D3 interactive graph example. This implementation was chosen due to the ability to drag a node in the graph, which triggers a transformation that re-arranges all the nodes and edges. Additionally, the names of the users represented by the nodes and the edges weight are available in tooltips.

[1] https://developers.google.com/maps/documentation/javascript/reference
[2] http://d3js.org/
[3] http://timeglider.com/widget/index.php

The previous widgets are responsible for representing the information in various dimensions. Interaction with them enables the analysis of different patterns. To navigate through the subsets and to change the weights of the dimensions, we used slidebars. We used the JavaScript implementation of DHTMLX slidebar.[4] This widget enables the definition of minimum, maximum and step values. For each dimension, since they are represented as weight percentages, the minimum value of the sidebar is set to 0 and maximum to 100. The step value is 25, since this was the step value used for the computation of the combined dissimilarities.

5 Results

The data extraction was done using a tool that retrieves data from online social networks, namely SocialBus.[5] By using this platform and filtering the data for this project's purposes, 119,558 tweets were retrieved with spatial, temporal and content attributes. This dataset contains data from May 2012 to February 2013, with tweets written in several languages (mostly Portuguese and English) and published from various countries (although the majority of tweets are from Portugal and Brazil).

To evaluate the social dimension, it was necessary to extract more data from Twitter in order to build a social graph of users, since the only information available at this point were the usernames of each tweet author. This data was retrieved from the Twitter RESTful API and the graph was built, defining nodes as the users and links as the following relationship. We retrieved only 9,794 edges for our 9,362 users. This result is justified by the fact that most users are not directly connected to any user in our database and are, therefore, excluded from the final graph. This lead to the existence of only 932 connected nodes in the graph.

At this point, we applied the methodology explained in Section 3. However, due to the size of the data, we had to split the original dataset into smaller ones. This decision, together with the weighting methodology led to several hundred clusterings. As we cannot display all results in this paper, we will just present results of some illustrative combinations. The first result we discuss illustrate the effect of setting different values for the dimensions weights on the results. Figure 1 shows a clustering with 100% importance set to the spatial dimension. It creates 3 clusters: Europe (blue points), America (orange points) and Africa (yellow points). It is visible in the timeline that all clusters occupy the entire time span (from June until October 2012).

When we set 100 % importance to the temporal dimensions for the same data (Figure 2), we obtain 2 clusters: one with tweets between June 3 and June 6 2012 (blue points) and another cluster between August 16 and October 13 2012 (orange points). In the spatial dimension, it is visible that there is no clear separation among clusters and that the clusters are, in fact, overlapped in this dimension, as could be expected given the weights selected.

[4] http://dhtmlx.com/docs/products/dhtmlxSlider/

[5] http://reaction.fe.up.pt/socialbus/

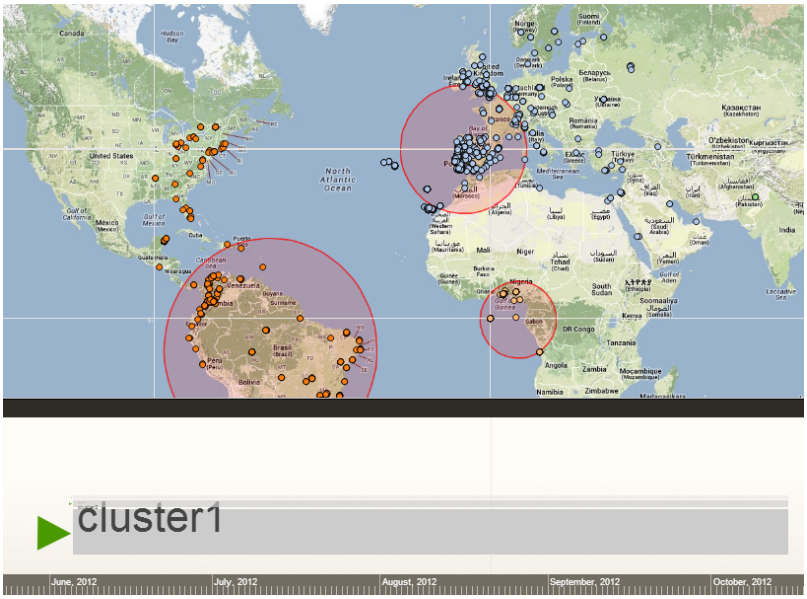

Fig. 1. Example clustering with 100% importance assigned to the spatial dimension.

However, when we assign 50% importance to both the spatial and temporal dimensions (Figure 3), the results are a combination from both the previously presented clusterings. Now, we have 4 clusters: Europe between June 4 and June 6 (blue points), America in the same time period (orange points), Europe between August 16 and October 13 (yellow points) and America in the same time period (green points).

If we were to assign weights to the other dimensions, the outcome would reflect also the same behaviour as in these examples. The difference is that it is harder to evaluate them in the platform, due to the large amount of information displayed and the fact that the content and social dimensions are harder to represent and interpret in clustering.

The following results represent a couple of interesting patterns. Only two examples are presented due to lack of space. Figures 4 and 5 exemplify some patterns retrieved using our Data Mining approach and represented on our visualization tool. Figure 4 shows different users that practise cycling and used the same sports application to publish their performance on Twitter. The positions of each user are visible, as well as the dates of the event. Figure 5 shows a cluster of posts by users who attended the same sports event, namely a football match between Portugal and Northern Ireland on the 16 October 2012.

The methodology proposed here assumes static data. Additionally, it has some scalability problems due to the high computational costs in calculating

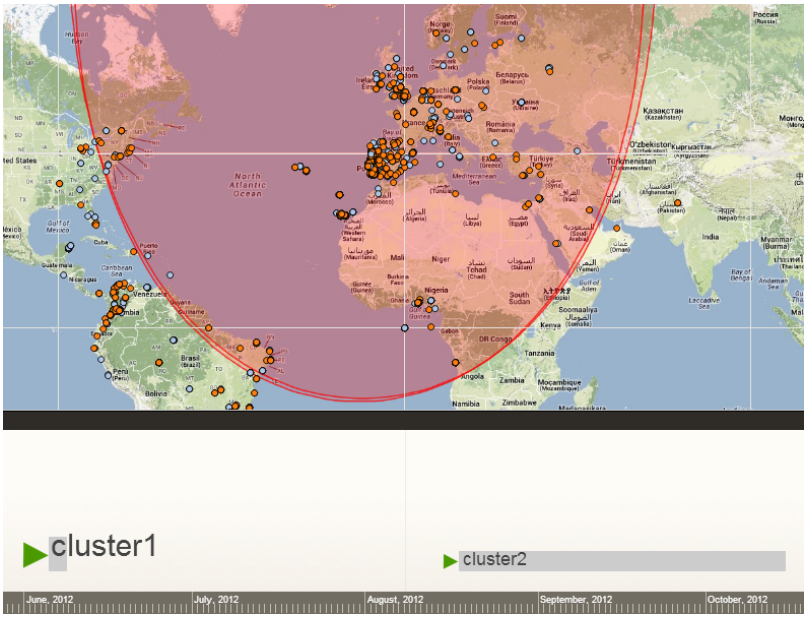

Fig. 2. Example clustering with 100% importance assigned to the temporal dimension.

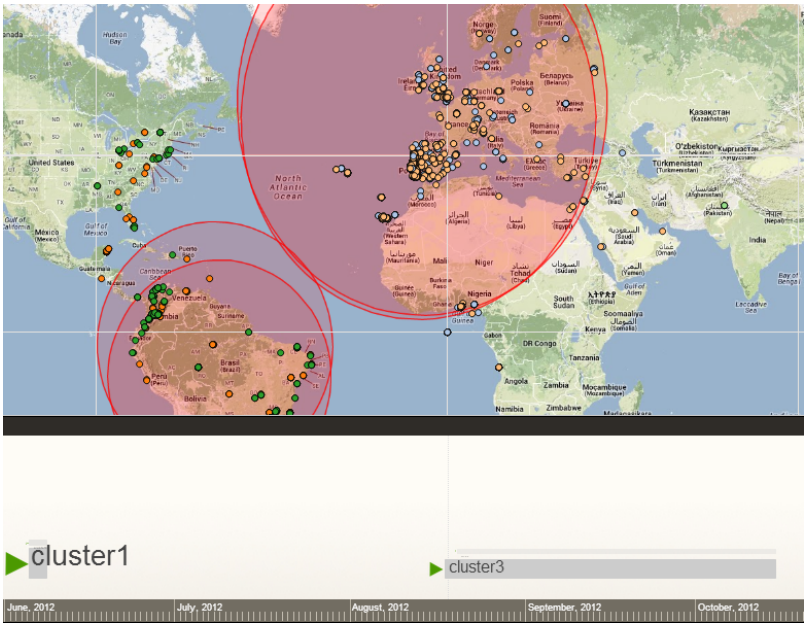

Fig. 3. Example clustering with 50% importance assigned to both the spatial and temporal dimension.

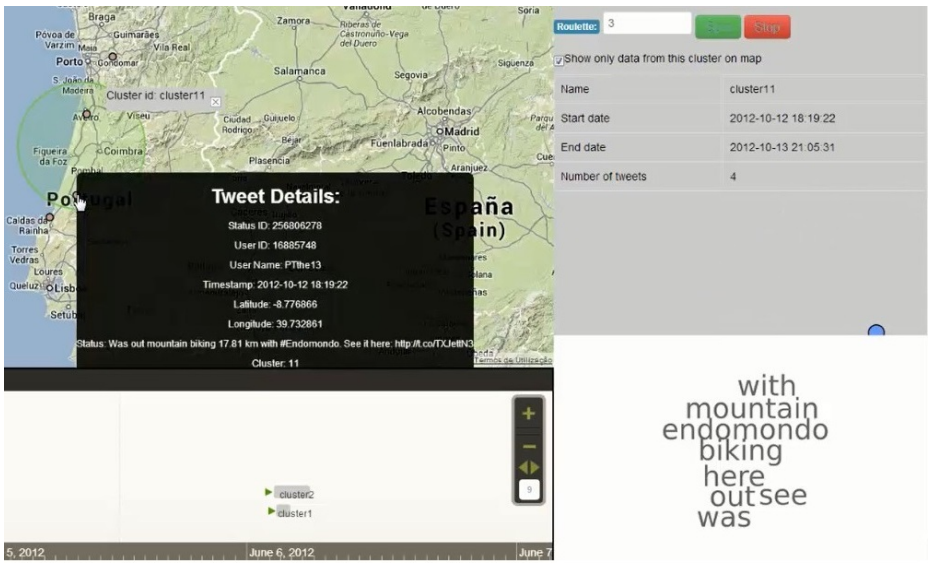

Fig. 4. Pattern presenting users using the same smartphone application for sports practising.

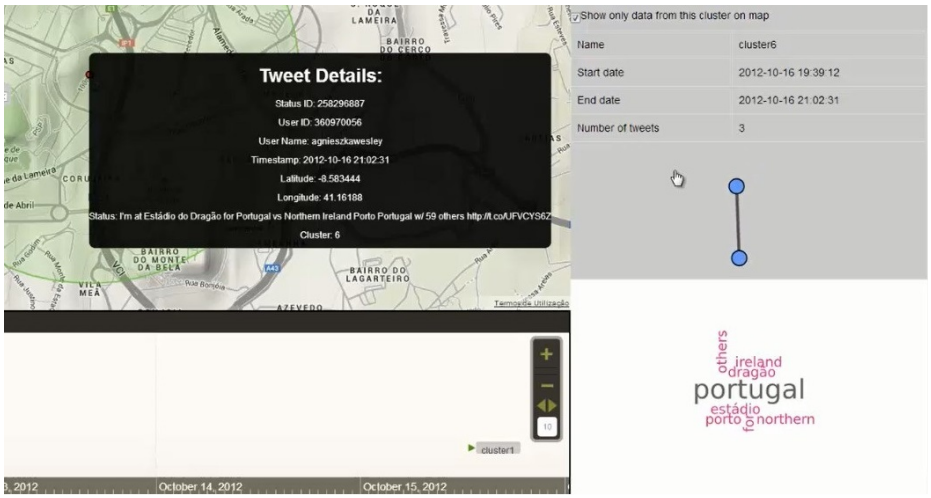

Fig. 5. Pattern presenting users in a sports event: football match between Portugal and Northern Ireland at Dragão Stadium.

and manipulating the matrices. One solution to these issues is the use of stream clustering methods [18].

This work has several shortcomings, such as computational costs, scalability, granularity of control of the weights and orthogonal treatment of different dimensions. However, it represents a simple solution to a complex problem. Most importantly, some of these shortcomings present interesting opportunities for scientific development. In particular, the simultaneous analysis of heterogeneous data, which was dealt with in a simple way in this project, is recognized as one of the most important issues in the Big Data community [19]. Furthermore, the ability to control the weight of each of those dimensions, which is important from the end user's perspective and that was also solved in a simple way in this project, is also a challenge from the computational, algorithmic and user interaction perspectives.

6 Conclusion

The goals of this paper were to develop a data mining approach for combining different types of information and also to apply our approach to Twitter data in several dimensions. The purpose and main contribution of this work was to use clustering to identify patterns across dimensions of data of very different nature (e.g., space and content), enabling the user to control the relative importance of each one of them in the process.

To accomplish these goals, a data mining process was developed with different stages: data preparation, dissimilarity matrices computation, normalization and combination and lastly, clustering using those matrices. A different distance function was chosen for each dimension (Haversine for space, Time Interval for time, cosine similarity on a TFIDF representation of the content and geodesic distance for social). A min-max normalization function was applied to all matrices. Given the computational cost of the process, the matrices were combined using a set of pre-defined weights, representing different levels of importance of each dimension. Clusterings were obtained by running DBSCAN on these matrices.

The visualization tool represents those patterns. Namely, a map, a timeline and a graph were implemented using various JavaScript libraries to create interactive widgets. These widgets enabled the simultaneous representation of the same information in different dimensions and to interact with them for a deeper and more flexible exploration of the results presented.

The first task for future work is to study better ways for represent the social and content dimensions, in order to ease the interpretation of results. Another task is to adapt this methodology for stream clustering while enabling a direct connection to Twitter's Streaming API. We can also change the dimensions and research whether this is indeed a generic methodology that accomplishes multidimensional clustering. Finally, since the proposed method for clustering in several dimensions is generic, one must also evaluate clustering with other dimensions, as for instance images.

Acknowledgments

This work was partially supported by projects REACTION (Retrieval, Extraction and Aggregation Computing Technology for Integrating and Organizing News - UTA-Est/MAI/0006/2009) and POPSTAR (Public Opinion and Sentiment Tracking, Analysis, and Research - PTDC/CPJ-CPO/116888/2010); "NORTE-07-0124-FEDER-000059" and "NORTE-07-0124-FEDER-000057" funded by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF) and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT); and a research grant assigned by the Doctoral Program in Informatics Engineering at the Faculdade de Engenharia da Universidade do Porto; and Sapo Labs / UP from Portugal Telecom.

References

1. C.-H. Lee, H.-C. Yang, T.-F. Chien, and W.-S. Wen, "A Novel Approach for Event Detection by Mining Spatio-temporal Information on Microblogs," *2011 International Conference on Advances in Social Networks Analysis and Mining*, pp. 254–259, Jul. 2011.
2. M. Bosnjak and E. Oliveira, "TwitterEcho: a distributed focused crawler to support open research with twitter data," *Proc. of the Intl. Workshop on Social Media Applications in News and Entertainment (SMANE 2012), at the ACM 2012 International World Wide Web Conference*, 2012.
3. S. Golder, "Tweet , Tweet , Retweet : Conversational Aspects of Retweeting on Twitter," *HICSS '10 Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, pp. 1–10, 2010.
4. F. Abel, Q. Gao, G. Houben, and K. Tao, "Analyzing Temporal Dynamics in Twitter Profiles for Personalized Recommendations in the Social Web," *In Proceedings of ACM WebSci '11, 3rd International Conference on Web Science*, 2011.
5. J. Han, M. Kamber, and J. Pei, *Data Mining : Concepts and Techniques*, 3rd ed., M. Kamber, Ed. Massachussets: Elsevier Science & Technology, 2006.
6. R. R. T. Zhang and M. Livny, "BIRCH: an efficient data clustering method for very large databases," *In Procedings SIGMOD '96 Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, vol. 1, pp. 103–114, 1996.
7. G. Karypis, E. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.
8. M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.
9. A. Hinneburg and D. Keim, "An efficient approach to clustering in large multimedia databases with noise," *In Proceedings of 4th International Conference in Knowledge Discovery and Data Mining (KDD 98)*, pp. 58–65, 1998.
10. W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," *Proceedings of the 23rd International Conference on Very Large Data Bases*, pp. 186–195, 1997.

11. C. G. Akcora, B. Carminati, and E. Ferrari, "Network and profile based measures for user similarities on social networks," *2011 IEEE International Conference on Information Reuse & Integration*, pp. 292–298, Aug. 2011.
12. A. Dekker, "Conceptual Distance in Social Network Analysis," *Journal of Social Structure*, vol. 6, 2005.
13. H. Ryu, M. Lease, and N. Woodward, "Finding and exploring memes in social media," *Proceedings of the 23rd ACM conference on Hypertext and social media - HT '12*, p. 295, 2012.
14. C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.
15. A. Lopes, R. Pinho, F. Paulovich, and R. Minghim, "Visual text mining using association rules," *Computers & Graphics*, vol. 31, no. 3, pp. 316–326, Jun. 2007.
16. A. Rangrej, S. Kulkarni, and A. V. Tendulkar, "Comparative study of clustering techniques for short text documents," *Proceedings of the 20th international conference companion on World wide web - WWW '11*, p. 111, 2011.
17. C.-H. Lee, "Mining spatio-temporal information on microblogging streams using a density-based online clustering method," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9623–9641, Aug. 2012.
18. A. Mahdiraji, "Clustering data stream: A survey of algorithms," *International Journal of Knowledge-based and Intelligent Engineering Systems*, vol. 13, no. 2, pp. 39–44, 2009.
19. F. Provost and T. Fawcett, "Data Science and its Relationship to Big Data and Data-Driven Decision Making," *Big Data*, vol. 1, no. 1, pp. 51–59, Feb. 2013.