4D$^+$SNN: A Spatio-temporal Density-based Clustering Approach with 4D Similarity

Ricardo Oliveira, Maribel Yasmina Santos
ALGORITMI Research Centre
University of Minho
Guimarães, Portugal
pg23779@alunos.uminho.pt, maribel@dsi.uminho.pt

João Moura-Pires
CENTRIA, Faculty of Science and Technology
New University of Lisbon
Lisbon, Portugal
jmp@fct.unl.pt

Abstract — Spatio-temporal clustering is a subfield of data mining that is increasingly gaining more scientific attention due to the advances of location-based or environmental devices that register position, time and, in some cases, other semantic attributes. This process pretends to group objects based in their spatial and temporal similarity helping to discover interesting patterns and correlations in large data sets. One of the main challenges of this area is the ability to integrate several dimensions in a general-purpose approach. In this paper, such general approach is proposed, based on an extension of the SNN (Shared Nearest Neighbor) algorithm. The 4D$^+$SNN algorithm allows the integration of space, time and one or more semantic attributes in the clustering process. This algorithm is able to deal with different data sets and different discovery purposes as the user has the ability to weight the importance of each dimension in the discovery process. The results obtained are very promising as show interesting findings on data and open the possibility of integration of several dimensions of analysis in the clustering process.

Keywords - clustering; density-based clustering; spatio-temporal data; distance function; spatio-temporal clustering

I. INTRODUCTION

Nowadays, recent developments in information systems and technologies support the collection and storage of large amounts of data by organizations. The analysis of such data collections to support the decision-making process represents a major challenge [1].

When dealing with spatial or spatio-temporal data, the complexity of the data analysis task increases. It is necessary to study the relationship of the objects or entities with the space and, also, the spatial relationships among those objects (like neighborhood, proximity, among others). Time indicates when an event was recorded and semantic attributes classify the objects and may change over time [2]. The analysis of these data is becoming more and more relevant due to the huge amounts of spatio-temporal data that is generated everyday by positioning and sensing technologies.

Several approaches have been proposed for clustering spatio-temporal data but, to the best of our knowledge, none can deal with geo-reference data in a way that space (2D), time (1D) and one (or more) semantic attribute(s) (1D) are considered in such a way that are simultaneously analyzed in the clustering process.

The 4D$^+$SNN algorithm, proposed in this paper for clustering of spatio-temporal data, is an extension of the SNN (Shared Nearest Neighbor) algorithm.

This paper is organized as follows. Section II summarizes related work on spatio-temporal clustering. Section III presents an analytical perspective of spatio-temporal data. Section IV describes the SNN algorithm and the proposed extension to handle 4D data in the clustering process. Section V presents the obtained results clustering two synthetic data sets and a real data set. Section VI concludes with a summary of the main findings and proposal of future work.

II. RELATED WORK

Clustering is a popular data-mining technique able to extract synthetic information from complex data sets [1]. The objective of clustering is to identify groups of categories or clusters that divide the analyzed data, identifying homogeneous groups of objects. This means that objects in the same group have to be as similar as possible and objects in different groups have to be as dissimilar as possible, ensuring high intra-cluster similarity and low inter-cluster similarity [3]. Clusters emerge naturally from the data under analysis using some distance function established to measure the similarity among objects. From the several categories of clustering [1][4], partition, hierarchical, density-based and grid-based, this paper is focused on density-based ones as they proved to be adequate in the analysis of spatio-temporal data [5][6].

Density-based algorithms can handle noise, outliers and can create clusters of different sizes and shapes. This type of algorithm generally needs as input parameter the radius of the neighborhood of a point and the minimum number of points in that neighborhood [4].

For the analysis of spatio-temporal data, using clustering algorithms, several approaches have been proposed. Birant and Kut proposed the ST-DBSCAN [7] based on the DBSCAN algorithm [8]. First, ST-DBSCAN filters the spatio-temporal data in order to identify the temporal neighbors and their corresponding spatial values.

Afterwards, the DBSCAN algorithm is applied to form the clusters. The authors use the Euclidean distance to measure the spatial distance between points and propose another equation, also based on the Euclidean distance, to measure the similarity of non-spatial values. They can handle temporal aspects as the algorithm first filter the data by retaining only the temporal neighbors and their corresponding values. The authors define that two objects are temporal neighbors if they are in consecutive time units such as consecutive days in the same year or in the same day in consecutive years. This algorithm requires more input parameters, compared with DBSCAN, adding more complexity to the input parameters tuning process. In this approach, the two dimensions (space and time) are not analyzed in an integrated way, requiring a previous temporal selection on the data for spatial analysis.

Other work [9] used the DBSCAN algorithm to perform spatial clustering of the data and the temporal clustering of the obtained spatial clusters. The developed approach is devised also for spatio-temporal events. This strategy was also followed by [10] to cluster trajectories. The authors combined the spatial and temporal dimensions in a similar way. First, spatial clustering is used to extract spatially similar trajectories and then temporal clustering is applied to the obtained clusters. This is a two-step clustering process in which the second step is influenced by the results obtained in the first step.

Another algorithm that was also extended to analyze spatio-temporal data was the SNN algorithm [11]. Liu, Bi and Yang [12] presents the STSNN algorithm to cluster spatio-temporal data and they tested it with data about earthquakes. This algorithm needs a new input parameter (*Delta T*), which is combined with the three original input parameters of the SNN algorithm (*Eps*, *k*, *MinPts*), allowing the definition of the time window in which two spatio-temporal events are considered neighbors. The specified temporal window influences the clustering result.

To cluster moving objects, [13] created a new concept, the REMO-matrix, in which the motion of the objects is logged at regular intervals of time and then transformed into angles. After that, these angles are matched to the generic motion patterns proposed by the authors, like Constance, Concurrence and Trend-setter. Looking for these patterns in the movement of people or animals allow the identification of tracks, flocks or leadership patterns, respectively. To identify these patterns it is necessary to calculate the spatial proximity between moving point objects as many objects can move in a similar way but be far from each other not representing any kind of moving pattern.

Previous paragraphs synthetized several approaches already proposed to deal with spatio-temporal data. To the best of our knowledge, none of the existing approaches can handle time and space in an integrated way and can also incorporate one or more semantic attributes in this process. Next section briefly presents a real data set and discusses how one may consider different analytical perspectives of spatio-temporal data.

III. Analytical perspectives of Spatio-temporal Data

For the purpose of the work presented in this paper, we consider that spatio-temporal data includes at least spatial and temporal components and may include additional semantic components.

Let us consider a real data set integrating 35,941 fires occurred in Continental Portugal in 2011. This data set will be named as the fires data set in this paper. Each fire in this data set is described using 38 attributes that include the spatial coordinates; the type of fire; the locality, parish, municipality and district; the date and time of the fire alert; the burnt area; if it was a false alarm; and many other attributes. Table 1 shows an extract of this data set emphasizing where the fire took place (spatial coordinates), when the fire started, its type and the total burnt area (in hectares).

Table 1. Sample of 2011 Portugal fires data set.

Type	X	Y	Date	Hour	Burnt Area
Florestal	187786	519555	30/01/2011	17:40	1.51
Agrícola	194201	509450	31/01/2011	20:19	0.002
Florestal	183556	356452	01/02/2011	10:55	0.005
Queimada	273293	386444	01/02/2011	12:04	0.003
Florestal	197440	474255	02/02/2011	18:20	0.2
Florestal	178876	479812	03/02/2011	14:15	0.16
Florestal	185465	498113	03/02/2011	15:51	0.04
Florestal	181452	501020	03/02/2011	18:30	0.1

If we only consider the where (spatial coordinates) and when (time), we are dealing with events allowing us to verify what are the places with more incidence of fires and in what period of the year fires are more frequent. But we can also group fires into clusters taking into consideration the burnt area. Instead of the burnt area the user can use the type of fire, obtaining clusters that consider space, time and type. Moreover, in an analytical perspective in which several views or dimensions of the data can be integrated in the data analysis process, allowing a deeper understating of the phenomenon under analysis, the user may want to see where, when, the burnt area and the type of fire. Such analysis creates several analytical perspectives that provide different views of the data.

However, this inclusion in the clustering process of the burnt area or the type of fire cannot be undertaken in the same way. Both variables behave differently as the burnt area is a real number of a ratio measurement scale and the type of fire is a categorical attribute that has a finite number of values.

Next section introduces the proposed approach to deal with these different analytical perspectives.

IV. 4D+SNN: The Shared Nearest Neighbor Algorithm For 4D Data

The SNN algorithm is a density-based clustering algorithm proposed by [11]. It has the capability to identify clusters of different shapes, sizes and densities, as well as the capability to deal with noise.

SNN is based on the notion of similarity and defines this similarity between points by calculating the number of nearest neighbors that two points share. The nearest neighbors are calculated using a distance function and the density of a point is the number of points within a given radius. Points with high density are classified as core points and points with low density will become noise points [14]. This similarity definition between points allows the algorithm to deal with data sets of variable density, being able to identify clusters with those different densities [11].

This algorithm needs three input parameters: *k, Eps* and *MinPts*. *k* is the number of neighbors, *Eps* defines the threshold density and *MinPts* is the minimum density that a point has to have to be considered a core point [11].

The most important input parameter is *k* (neighborhood list size) because it strongly influences the granularity of the clusters. If *k* is too small, even a uniform cluster will be split into several clusters and because of that, the algorithm will have a tendency to find many small, but tight, clusters. On the contrary, if *k* is too high, the algorithm will find only a few large, well separated clusters [11].

The main steps of the SNN algorithm are presented next [11]:

1. **Construct the similarity matrix:** a similarity graph with data points as nodes and edges whose weights are the similarities between those data points;
2. **Reduce the similarity matrix:** keep only the *k* most similar neighbors, i.e., the *k* strongest links of the similarity graph;
3. **Create the SNN graph:** using the similarity matrix and applying a similarity threshold;
4. **Calculate the SNN density of each point:** using the *Eps* value to filter what are equal or superior;
5. **Find the core points:** filter points that have a density greater than *MinPts*;
6. **Form clusters:** if two core points are within a radius, *Eps,* they are placed in the same cluster;
7. **Discard all noise points:** all non-core points that are not within a radius, *Eps*, of a core point are considered noise and consequently, discarded;
8. **Assign all the other points to clusters:** non-noise and non-core points are assigned to the nearest core point.

For measuring the similarity of data points, a distance function is necessary and, because of the computational complexity, the choice of this distance function is very important. Moreover, the results of this function will greatly influence the clusters so an effective and efficient distance function to help the algorithm is needed [15].

Considering spatio-temporal data, namely for clustering events, 3D vectors of $< x, y, t >$ can be used to distinguish between the objects that are near each other in space and time. The distance function that is proposed in this work to identify the distance between two events $p_1(x_1, y_1, t_1)$ and $p_2(x_2, y_2, t_2)$ is presented in (1).

$$3D(p_1, p_2) = w_s * \frac{Ds(x_1,x_2,y_1,y_2)}{MaxS} + w_t * \frac{Dt(t_1,t_2)}{MaxT} \quad (1)$$

With this approach, the user can use any function (*Ds*, *Dt* and *Da*), considering the problem domain to calculate the differences (respectively, spatial, temporal and attribute) between points. For example, the user can select a specific function to calculate the spatial distances between points (e.g., Euclidean distance or geodesic distance), and for the temporal dimension, the user can take into account the cyclical behavior of time (days, years, season, etc.). w_s and w_t are used to assign a weight in the clustering process to each one of these components (spatial and temporal dimensions). By this way, the user can control the pretended results attending to the analytical context. As will be presented in the results section, these weights are powerful calibration instruments that the user can use to tune the clustering results.

MaxS and *MaxT* are used to normalize the spatial and the temporal dimensions. When looking for the *k*-nearest neighbors of a point, it is expected (and usually is what happens) that the neighbors are relatively close in space and time, which means that the spatial distance and the temporal distance to the neighbors present similar values. For this reason, it is not appropriate to set *MaxS* and *MaxT* to the maximum possible values in each domain, as this will penalize the dimension with higher amplitude in its distance values. Moreover, the existence of noise will highly influence those distances. To overcome this situation, the approach that is proposed makes *MaxS* and *MaxT* emerge from the data under analysis.

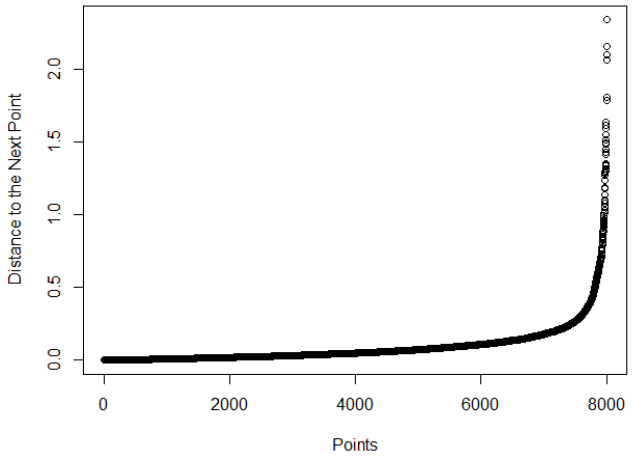

Figure 1. Sorted distances for the spatial dimension in t5.8k.

In a data-driven process, the approach used to identify these values start by identifying the bounding box for the spatial component. For each point present in the data set, the spatial distance between the point and the lowest left point of the bounding box is calculated. All these distances are sorted in an ascending order and the difference between two consecutives distances is calculated. Afterwards, these differences are sorted by ascending order. The results that are

obtained through this approach allow the identification of common distances between neighbors and the identification of those distances that are influenced by the presence of noise points in the data set (Figure 1). This process was inspired by the *k-sort graphs* proposed in [8].

The analysis of several data sets, either synthetic or real, allowed the identification of the distance value given by the 80% *decile* as an appropriate value for the *MaxS*. The distance present in this *decile* split the distances that are usually associated to neighbors' values and those that start to be influenced by noise points. When, in a spatially homogeneous data set, i.e., the points of the data set are spatially concentrated and present few outliers, the difference between the consecutive distance values in the sorted distance array is equal to 0, even in the 80% *decile*, the algorithm successively increments the 80% value by 1% until a difference different than 0 is found. Distance values shown in Figure 1 were obtained for the t5.8k data set, described in more detail in the following section. Figure 2 shows the distance values for the real data set used in this paper, the 2011 Portugal fires data set.

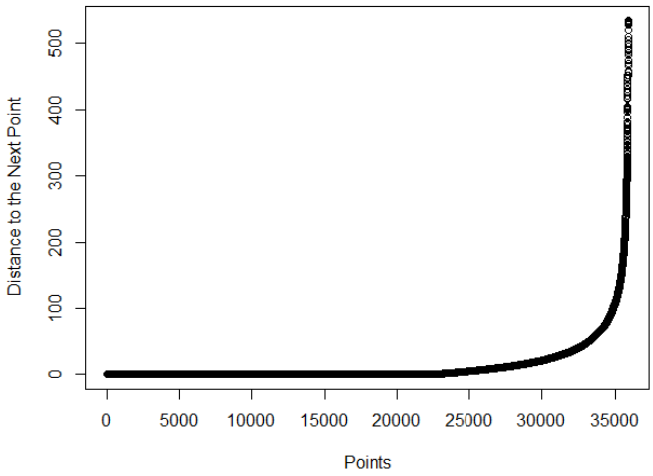

Figure 2. Sorted distances for the spatial dimension for the fires data set.

For the temporal dimension, the identification of *MaxT* follows a similar process like the one just described for *MaxS*. The only difference is that the temporal distance between each point and the minimum time instant present in the data set is calculated.

For adding more dimensions to the clustering process, in this case a semantic attribute ($< x, y, t, a >$) with continuous values, the distance function needs to be extended (2). The new dimension also needs to be normalized and for that the *MaxA* value is calculated following the approach of *MaxT*. A weighting factor (w_a) is also added allowing the user to control the type of patterns that can be obtained.

Finally, in the last dimension (attribute), the user can employ a function that suits the attribute domain or even use a function that deals with two or more semantic attributes combined in one.

$$4D(p_1, p_2) = w_s * \frac{Ds(x_1,x_2,y_1,y_2)}{MaxS} + w_t * \frac{Dt(t_1,t_2)}{MaxT} + w_a * \frac{Da(a_1,a_2)}{MaxA} \quad (2)$$

This extension of the SNN algorithm allows the simultaneously analysis of 4 or more dimensions. Following this process, new dimensions can be integrated either adding new attributes to the distance function or combining several non-spatial attributes in one semantic one.

V. Results

This section presents the results obtained from clustering two synthetic data sets that were modified to incorporate the time dimension. This was needed as synthetic spatio-temporal data sets are hard to find. Afterwards, a real data set, the fires data set, is also used.

For the purpose of this work and for all the results presented next, the spatial distance chosen was the Euclidean distance (3). (4) was used for time and (5) for the semantic attribute

$$Ds(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3)$$

$$Dt(p_1, p_2) = |t_1 - t_2| \quad (4)$$

$$Da(p_1, p_2) = |a_1 - a_2| \quad (5)$$

In order to have a starting point in the search for the appropriate values for the SNN input parameters, it was used an approach proposed by Moreira, Santos and Moura-Pires [16]. With the values given by this approach, it was performed the first test and according to the results, the SNN parameters were adjusted to the final parameters used in results presented next.

A. Synthetic Data Sets

The data set named t5.8k integrates 8000 points which spatial distribution is shown in Figure 3.

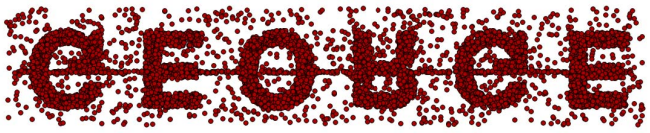

Figure 3. Spatial Distribution of t5.8k. Chameleon data sets [17].

In this data set, two different modifications (named t5.8k.a and t5.8k.b in this paper) were made for adding the temporal dimension. First (t5.8k.a), the data set was vertically split in six days. For each day, the several points were randomly distributed along the day.

Using the 4D$^+$SNN implementation, and for t5.8k.a, Figure 4 shows the 6 resulting clusters using the same weight for space and time, 50%. Noise points (black points) were identified in the boundary of each time transition. This result shows the adequacy of the distance function defined to measure the similarity of the objects and confirm the correctness of the heuristic applied to identify the variables

used in the normalization of each dimension, namely *MaxS* and *MaxT*.

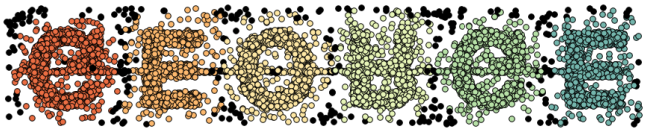

Figure 4. Result of Spatio-temporal Clustering Using 50%-50% Weights (SNN Parameters, k = 50, *Eps* = 20, *MinPts* = 35) for t5.8k.a.

The second transformation (t5.8k.b) was to assign the first four letters to one day and the other two to the following day being the points randomly distributed in each day.

For t5.8k.b, and in order to be possible the identification of the temporal distribution artificially introduced in the dataset, a weight of 20% for space (w_s) and 80% for time (w_t) identifies the expected result (Figure 5). Inverting the importance of each dimension in the clustering process, increasing the weight for space to 75% and assigning 25% for time, given more importance to where points are located and not when they were verified, allows the identification of a similar result like the presented in Figure 4, only with small differences in the identified noise points (Figure 6).

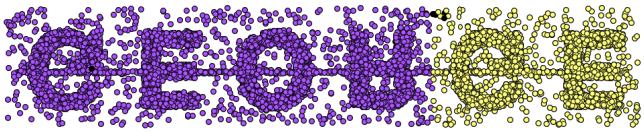

Figure 5. Result of Spatio-temporal Clustering Using 20%-80% Weights (SNN Parameters, k = 40, *Eps* = 16, *MinPts* = 24) for t5.8k.b.

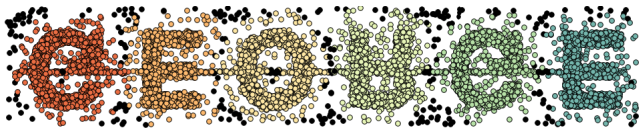

Figure 6. Result of Spatio-temporal Clustering Using 75%-25% Weights (SNN Parameters, k = 40, *Eps* = 16, *MinPts* = 24) for t5.8k.b.

The second data set is t4.8k, integrating also 8000 points. This data set was divided in 3 days, joining in the same day different spatial clusters. Using a similar weight for space and time produces a clustering result that reflects the division performed in the data set (Figure 7). Increasing the weight of space (75%) over time (25%), the 4D$^+$SNN algorithm is able to find the 6 expected clusters (Figure 8).

The results obtained so far produced appropriate results when clustering spatio-temporal events. The modification of the synthetic data sets allowed the verification of the sensibility of the proposed approach to space and time, when those dimensions are analyzed in an integrated way. Next subsection presents the results obtained when clustering the fires data set, considering events with or without semantic attributes.

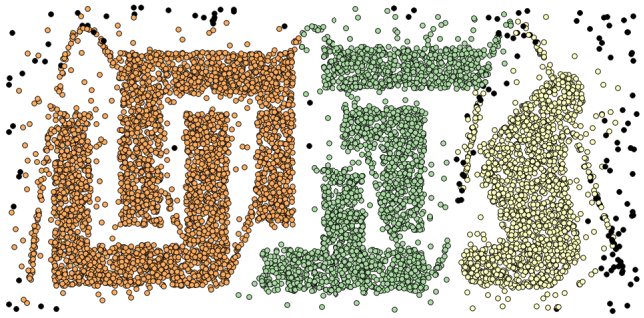

Figure 7. Result of Spatio-temporal Clustering Using 50%-50% Weights (SNN Parameters, k = 40, *Eps* = 16, *MinPts* = 24).

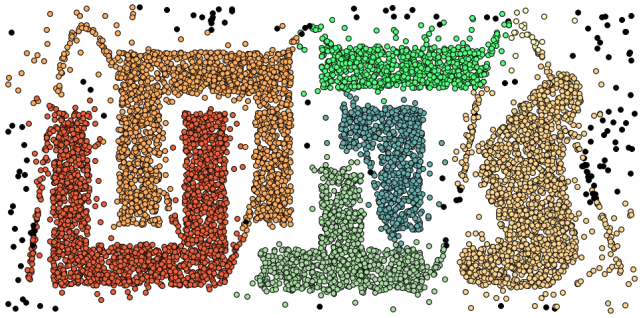

Figure 8. Result of Spatio-temporal Clustering Using 75%-25% Weights (SNN Parameters, k = 40, *Eps* = 16, *MinPts* = 24).

B. Fires Data Set

The spatial distribution of Portugal fires in 2011 is presented in Figure 9.

Figure 9. Spatial Distribution of the Fires Data Set.

The fires data set was clustered considering events with the spatial and temporal dimension. The results shown in Figure 10 were obtained assigning the same weight to space and time, allowing the identification of clusters that combine the same spatial region with different periods of time or different regions in the same time. The ticks at the temporal scale in the 3D graphics mean the beginning of that season.

Eight clusters were detected. Six of them in the northern and center part of the country (1,2,3,4,5 and 8), one in the south (6) and one big cluster (7) with the fires that occurred in the summer in the northern and center part of the country.

With the presented 3D visualization it is possible to verify when fires occur and in what regions they prevail. The North and South parts of the country present different seasoning behavior, with the North having fires almost all year long.

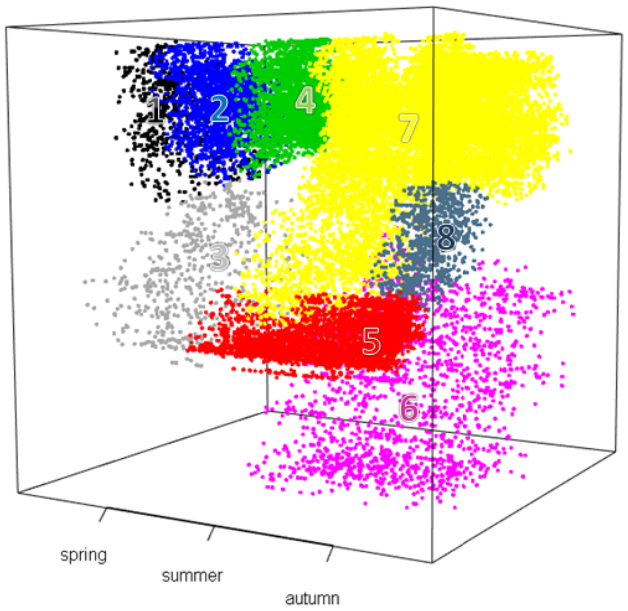

Figure 10. Clustering Events of the Fires Data set Using 50%-50% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200).

Looking at the obtained results, the eight clusters confirm the advantages of the SNN algorithm when used in spatial data, as it was able to identify very different clusters, either in shape, size and density. A total of 5013 noise points were also identified along all country. Those points were not shown in the figure for the sake of clarity.

In the next experiment a semantic attribute (the burnt area) was added. 14 clusters were identified (Figure 11). Each cluster is represented by a different color and by the number of the cluster. Clusters 1 and 2 share the same spatial region and the same time window but each one integrates fires with different burnt areas. Cluster 1 has an average burnt area of 0.05 hectares while Cluster 2 an average of 1.58. Clusters 3 and 4 are located in the same region (around the center of Portugal) and have similar burnt areas (average of 0.03 and 0.02, respectively) but were verified in different time windows. Some of the clusters previously identified (Figure 10) are now separated in different segments attending to the burnt area. Figure 12 presents a graphic with two scales, the left one (the black lines) shows the maximum and minimum burnt area of each cluster and the right scale (the colored bars) shows how many points each cluster has. The colors in the bars are the same as in Figure 11.

The obtained results show that there is only one cluster, number 7, which presents high amplitude in the values of the burnt area, ranging from 0.40 to 13.10 hectares. This cluster, the yellow one in Figure 11, is present in a wide temporal window that includes fires from February to December. Comparing with clusters number 5 (cyan), 8 (gray) or 9 (pink), cluster number 7 presents a different density of points, a smaller density, reason why this cluster emerged as a separated one.

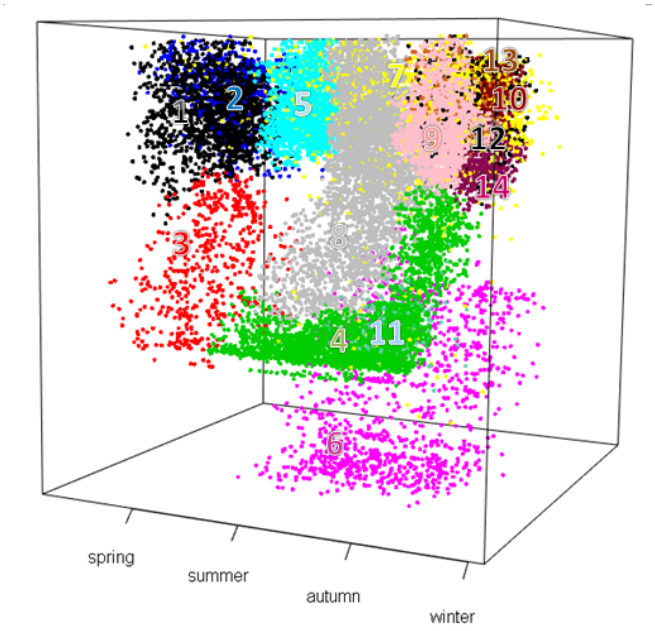

Figure 11. Clustering with a Semantic Attribute (burnt area) Using the Same Weights for the 3 Dimensions (SNN Parameters, k = 230, Eps = 45, MinPts = 200).

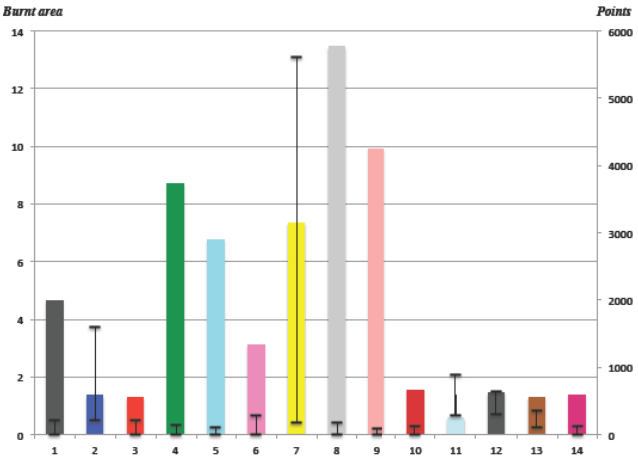

Figure 12. Number of Points, Maximum and Minimum per Cluster (Same Weights for the 3 Dimensions).

Using the same geo-referenced data, but changing the weights of each dimension in order to increase the relevance of the semantic attribute in the clustering process, with the aim of identifying clusters more aligned in terms of the burnt area, Figure 13 presents the 21 clusters obtained. The perspective in this 3D graphic is slightly different from Figure 11 because it is very difficult to see all the clusters formed in this result. So, in this view, it can be seen the temporal separation of the clusters and the separation between the north/center zone of Portugal and the south.

Although challenging due to the 21 colors used to plot the clusters, it is possible to verify that previous huge

clusters (Figure 11) were broken into smaller ones that optimize the similarity in terms of the burnt area (for example, clusters 6 and 10 were a single cluster when clustering with the same weight for every dimension). In this case the weighting factors applied were 20% for space, 20% for time and 60% for the semantic attribute.

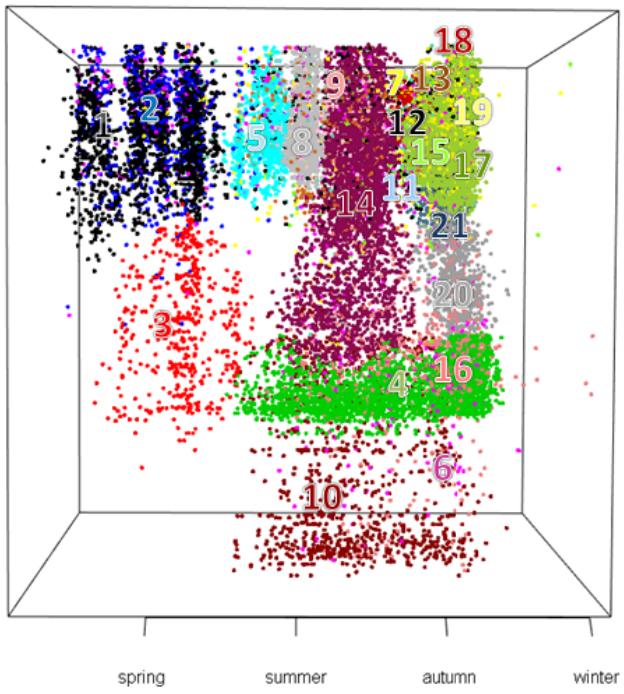

Figure 13. Clustering with a Semantic Attribute (burnt area) Using a w_s=20%,w_t=20%,w_a=60% (SNN Parameters, k = 230, *Eps* = 45, *MinPts* = 200).

Figure 14 presents a graphic (similar to Figure 12) with the number of incidents for each cluster and the maximum and minimum value of the burnt area for each cluster. As it can be seen, the amplitude of the burnt area value for each cluster is smaller than when clustering with the same weight for every dimension. Several clusters (4, 5, 8, 14, 17, 18, 20 and 21) have a difference so small (because they aggregate false alarms and small fires, maximum 0.14 ha) between the maximum and minimum values of the burnt area that is very difficult to see them in Figure 14.

The weighting factors in (1) affect the clustering results. In general, setting more weight to one dimension makes the clusters more concentrated in regard to that dimension. For instance, setting more weight to the spatial dimension, tends to make the objects inside the same cluster spatially closer. Instead, giving more weight to the temporal dimension will produce clusters that are closer temporarily. Such effects were observed in the experiments using the datasets t5.8k.a and t5.8k.b. The weighting factors in (2), that include a semantic attribute, also affect the clustering results. In the experiments with the fires dataset, with a weight w_a = 33% the 14 clusters present an average range for the burnt area of 1.59 and with w_a = 60% the 21 clusters present an average range for the burnt area of 0.87. The expected impact on the number of clusters and their (temporal, spatial and semantic) densities depend on the used weights and on the dataset itself.

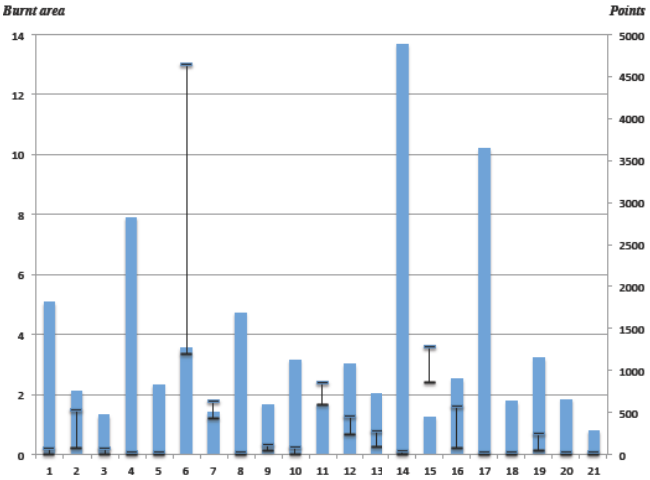

Figure 14. Number of Points, Maximum and Minimum per Cluster (w_s=20%,w_t=20%,w_a=60%).

VI. Conclusions and Future Work

The proposed approach, the 4D$^+$SNN, had interesting and promising results, with spatial and temporal data or with more dimensions, because both types of data were effectively clustered identifying relevant patterns. This approach has some advantages in relation to the ones studied in Section II. Specifically, the ST-DBSCAN [7] and the STSNN [12] were the two principal approaches to the field of study of this work. In comparison with these two approaches, the 4D$^+$SNN can cluster spatial and temporal dimensions in an integrated way because it considers all dimensions simultaneously in the distance function, imposing no restrictions to the clusters that can be found. Other advantage of this approach is that it does not add more input parameters to the algorithm, which facilitates the user experience (the ST-DBSCAN adds two more input parameters and the STSNN needs that the user knows what is the time window to an object be considered in the neighborhood of another object). The input parameters that this approach needs are the weighting factors that the user can tune in order to improve the results but this factors have a specific range of possible values (0-100) so it should not be difficult for the user to understand how these factors work, if he wants to give more importance to a specific dimension all he has to do is to increase the weight associated to that dimension. This is possible because this approach can find the normalization parameters needed to make the dimensions equivalent, i.e., stop using measures and scales. The main advantage of this approach is the capacity to cluster objects with spatial, temporal and semantic dimensions which, to the best of our knowledge, no other approach can do.

The results achieved with both synthetic and real data sets are very promising as spatio-temporal objects were effectively clustered identifying relevant patterns. In order to continue the work carried so far, as future work we plan to

introduce a discrete attribute in the clustering process, opening new possibilities in the analysis of non-spatial attributes. We also plan to further investigate our heuristic to identify the normalization factors, now depending of the 80% *decile,* through an approach that uses the properties of the data set to discover the normalization parameters. Moreover, the current implementation will be complemented with a graphical user interface that will allow any user to take advantage of it. This implementation will be available in the web as open-source code.

ACKNOWLEDGMENT

This work was partly funded by FEDER funds through the Operational Competitiveness Program (COMPETE), by FCT with the project: FCOMP-01-0124-FEDER-022674 and by *Novabase Business Solutions* with a co-funded QREN project (24822).

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*, 3rd Ed. Morgan Kaufmann, 2012.

[2] G. Andrienko, N. Andrienko, P. Bak, D. Keim, S. Kisilevich, and S. Wrobel, "A conceptual framework and taxonomy of techniques for analyzing movement," *Journal of Visual Languages & Computing*, vol. 22, no. 3, pp. 213–232, Jun. 2011.

[3] A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, 1999.

[4] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information*, pp. 107–145, 2001.

[5] A. Moreira, M. Y. Santos, M. Wachowicz, and D. Orellana, "The Impact of Data Quality in the Context of Pedestrian Movement Analysis," *Lecture Notes in Geoinformation and Cartography*, vol. 0, pp. 61–78, 2010.

[6] M. Y. Santos, J. P. Silva, J. Moura-pires, and M. Wachowicz, "Automated Traffic Route Identification through the Shared Nearest Neighbour Algorithm," in *Bridging the Geographic Information Sciences, International 15th AGILE'2012 Conference*, 2012, pp. 231–248.

[7] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208–221, Jan. 2007.

[8] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[9] C. Pöelitz, G. Andrienko, and N. Andrienko, "Finding arbitrary shaped clusters with related extents in space and time," in *EuroVAST 2010: International Symposium on Visual Analytics Science and Technology*, 2010, pp. 19–25.

[10] G. Mcardle, A. Tahir, and M. Bertolotto, "Spatio-Temporal Clustering of Movement Data: An Application to Trajectories Generated by Human-Computer Interaction," in *XXII Congress of the International Society for Photogrammetry and Remote Sensing*, 2012, no. September, pp. 147–152.

[11] L. Ertoz, M. Steinbach, and V. Kumar, "Finding Clusters of Different Sizes, Shapes, and Densities in Noisy ,High Dimensional Data," in *2nd SIAM International Conference on Data Mining*, 2002.

[12] Q. Liu, M. Deng, J. Bi, and W. Yang, "A novel method for discovering spatio- temporal clusters of different sizes, shapes, and densities in the presence of noise," *International Journal of Digital Earth*, no. December, pp. 1–20, Feb. 2012.

[13] P. Laube, M. van Kreveld, and S. Imfeld, "Finding REMO—detecting relative motion patterns in geospatial lifelines," in *11th International Symposium on Spatial Data Handling*, 2005, pp. 201–214.

[14] A. Moreira, M. Y. Santos, and S. Carneiro, "Density-based clustering algorithms–DBSCAN and SNN," 2005.

[15] F. Lin, K. Xie, G. Song, and T. Wu, "A Novel Spatio-temporal Clustering Approach by Process Similarity," in *6th International Conference on Fuzzy Systems and Knowledge Discovery*, 2009.

[16] G. Moreira, M. Y. Santos, and J. Moura-pires, "SNN Input Parameters: how are they related?," to be presented at *Crowd and Cloud Computing Workshop at International Conference on Parallel and Distributed Systems 2013*, 2013.

[17] G. Karypis, E. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.