Universidade do Minho
Escola de Engenharia

João Ricardo Leite Mota Oliveira

Spatio-temporal SNN: Integrating Time and Space in the Clustering Process

outubro de 2013

Spatio-temporal SNN: Integrating Time and Space in the Clustering Process
João Ricardo Leite Mota Oliveira
UMinho | 2013

Universidade do Minho
Escola de Engenharia

João Ricardo Leite Mota Oliveira

Spatio-temporal SNN: Integrating Time and Space in the Clustering Process

Dissertação de Mestrado
Mestrado em Engenharia e Gestão de Sistemas de Informação

Trabalho realizado sob orientação da
Professora Doutora Maribel Yasmina Santos

outubro de 2013

Nome: João Ricardo Leite Mota Oliveira

Endereço eletrónico: jrlmoliveira@gmail.com

Título dissertação:

Spatio-temporal SNN: Integrating Time and Space in the Clustering Process.

Orientador: Professora Doutora Maribel Yasmina Santos

Ano de conclusão: 2013

Designação do Mestrado: Mestrado em Engenharia e Gestão de Sistemas de Informação

Universidade do Minho, ___ /___ /_____

Assinatura:_____

ACKNOWLEDGMENTS

Although this work is individual, many people directly or indirectly contributed for its development. Without their help, the quality of this work would surely be inferior. To all of them I would like to express my appreciation, but cannot help showing my special gratitude:

Firstly, to my advisor, Professor Maribel Yasmina Santos, for sharing her knowledge and experience, for her countless theoretical and technical contributions to this work, for the given motivation in difficult times and for trusting me to complete this project.

To Professor João Moura-Pires for his opinions and text reviews. To Guilherme and Fernando, project colleagues, for the brainstorming meetings, and for helping me to familiarize with the tool used in this project.

To Sérgio, Joana and Rui, "brothers in arms" in this little adventure that was the master's degree, for the constant exchange of ideas and the companionship in this lonely work.

To Adriano, Frederico and Gonçalo, long-time friends, for the social moments and relaxing times needed to distract the mind from work.

Finally, a very special thanks to my family, my father João Daniel and my mother Maria Salomé for the education and personal values passed on, making me the person I am today. For giving me all the necessary conditions so I could complete this degree, and for always believing in my abilities, even when I did not give them reason to. I would also like to thank my brother Tiago for all the moments of fun and joy throughout life.

This work was done with the support of the research project "GIAP - GeoInsight Analytics Platform (LISBOA-01-0202-FEDER- 024822)", funded by Comissão de Coordenação e Desenvolvimento Regional de Lisboa e Vale do Tejo (PORLisboa), included in Sistema de Incentivos à Investigação e Desenvolvimento Tecnológico (SI I&DT), through the research fellowship UMINHO/BI/304/2012.

ABSTRACT

Spatio-temporal clustering is a new subfield of data mining that is increasingly gaining scientific attention due to the technical advances of location-based or environmental devices that register position, time and, in some cases, other semantic attributes. This process intends to group objects based in their spatial and temporal similarity helping to discover interesting patterns and correlations in large datasets. One of the main challenges of this area is that there are different types of spatio-temporal data and there is no general approach to treat all these types. Another challenge still unresolved is the ability to integrate several dimensions in the clustering process with a general-purpose approach. Moreover, it was also possible to verify that few works address their implementations under the SNN (Shared Nearest Neighbour) algorithm, which gives the opportunity to propose an innovative extension of this particular algorithm.

This work intends to implement in the SNN clustering algorithm the ability to deal with spatio-temporal data allowing the integration of space, time and one or more semantic attributes in the clustering process. In this document, background knowledge about clustering, spatial clustering and spatio-temporal clustering are presented along with a summary of the main approaches followed to cluster spatio-temporal data with different clustering algorithms. Based on those approaches, and in the analysis of their advantages and disadvantages, the boundaries of this work are defined in order to incorporate the space, time and semantic attribute dimensions in the SNN algorithm and thus propose the $4D^{+}$SNN approach.

The results presented in this work are very promising as the approach proposed is able to identify interesting patterns on spatio-temporal data. Concretely, it can identify clusters taking into account simultaneously the spatial and temporal dimension and it also has good results when adding one or more semantic attributes.

Keywords: *Clustering; Density-based Clustering; Spatio-temporal Data; Distance Function; Spatio-temporal Clustering.*

RESUMO

O clustering espaço-temporal é uma nova área do data mining que está a ganhar crescente atenção por parte da comunidade científica devido aos avanços tecnológicos dos dispositivos de localização ou monitorização ambiental que registam posição, tempo e, em alguns casos, outros atributos semânticos. Este processo pretende agrupar objectos segundo as suas similaridades espaciais e temporais ajudando assim a descobrir padrões interessantes e correlações em grandes conjuntos de dados. Um dos grandes desafios nesta área é a existência de vários tipos de dados espaço-temporais e não existe uma abordagem geral para tratar todos estes tipos. Outro desafio ainda por resolver é a capacidade para integrar várias dimensões no processo de clustering com uma abordagem geral. Além disso, foi possível verificar que poucos trabalhos de investigação usam o algoritmo SNN *(Shared Nearest Neighbour)* nas suas implementações o que dá a oportunidade para propor uma extensão inovadora para este algoritmo em particular.

Este trabalho pretende implementar no algoritmo de clustering SNN a capacidade para lidar com dados espaço-temporais permitindo assim a integração do espaço, tempo e um ou mais atributos semânticos no processo de clustering. Neste documento, serão apresentados alguns conceitos sobre clustering, clustering espacial e clustering espaço-temporal assim como um resumo das principais abordagens usadas para fazer o clustering de dados espaço-temporais com algoritmos de clustering diferentes. Baseado nestas abordagens e na análise das suas vantagens e desvantagens, serão definidos os limites deste trabalho de modo a incorporar as dimensões espaço, tempo e atributo semântico no algoritmo SNN e, assim, propor a abordagem $4D^{+}$SNN.

Os resultados apresentados neste trabalho são bastante promissores pois a abordagem proposta é capaz de identificar padrões interessantes em dados espaço-temporais. Concretamente, consegue identificar clusters tendo em consideração simultaneamente as dimensões espaço e tempo e também obtém bons resultados quando adicionando um ou mais atributos semânticos.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
CLARA	Clustering Large Applications
CLARANS	Clustering Large Applications based on Randomized Search
CLIQUE	CLustering In QUEst
CURE	Clustering Using REpresentatives
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DENCLUE	DENsity-based CLUstEring
DSR	Design Science Research
GPS	Global Positioning System
OPTICS	Ordering Points to Identify the Clustering Structure
PAM	Partitioning Around Medoids
ROCK	RObust Clustering using linKs
SNN	Shared Nearest Neighbour
STING	Statistical Information Grid
STSNN	Spatio-temporal Shared Nearest Neighbour

1 - INTRODUCTION

This chapter introduces the area of this work and the motivation for undertaking it. It will be demonstrated why this work is important and presented the research question. After that, the objectives outlined for this project and the expected results for this work are presented. Then, some considerations about the methodology used in this work are reported. In the end of this chapter is presented the structure of this report.

1.1 - Framework and Motivation

Nowadays, great amounts of data are being collected by organizations. Besides the progress in this collection, the analysis of these data for decision support is still a great challenge, as these organizations cannot know beforehand what information is useful for their business or not (Han, Kamber, & Pei, 2012).

When dealing with spatio-temporal data, in addition to these challenges, we must consider the complexity of handling the space in which the events occurred and the moment in time in which they were verified. This analysis of movement patterns in spatio-temporal data is a relatively new area of research and is becoming gradually more important because large amounts of spatio-temporal data are being generated by devices like cell phones, GPS (Global Positioning System) and remote sensor devices (Tork, 2012).

These data introduce new challenges to data analytics and require new techniques for knowledge discovery and spatio-temporal clustering can be one of these new techniques. This new sub field of Data Mining is gaining high popularity especially in geographic information sciences due to the availability of cheap sensor devices which caused an exponential growth of geo-tagged data in the last years. The great challenge is not to get the right data but to analyse all the data we can get (Kisilevich, Mansmann, Nanni, & Rinzivillo, 2010). One key aspect is that it is possible to analyse all these data because, only in the last years, the necessary technological development was achieved that enabled the discovery of knowledge in vast amounts of data (Laube, Wolle, & Gudmundsson, 2007).

The analysis of spatio-temporal data is very complex because it involves time, geographical space, objects appearing and disappearing in space and multidimensional attributes that are in constant change over time (Andrienko et al., 2011).

The usual clustering approaches do not consider time and space. The ones that consider can only process spatio-temporal events but cannot treat moving objects because this is a complex type of data (Birant & Kut, 2007; Liu, Deng, Bi, & Yang, 2012). Therefore, the discovery of spatio-temporal clusters is a challenging issue in the knowledge discovery domain and is very important in many areas of science and technology such as meteorology, biology, sociology, transportation engineering, telecommunications, etc. (Dodge, Weibel, & Lautenschütz, 2008). These techniques are used essentially to get patterns of animal behaviour, human movement and traffic, surveillance, security, military and even sports movement (Laube et al., 2007).

From all the families of clustering algorithms, the density-based seems to be a good candidate as it will be demonstrated in the next chapters. From the algorithms in the density-based family, the Shared Nearest Neighbour (SNN) algorithm seems to be an appropriate choice for this project as it will be demonstrated by some studies presented in the literature review.

The SNN algorithm is a density-based clustering algorithm that can identify clusters of different sizes, shapes and densities. Moreover, it can identify noise objects in the data (Ertoz, Steinbach, & Kumar, 2002). In order to measure the similarity between data objects, a distance function is necessary. The most common distance function is the Euclidean distance among objects. The choice of this function is very important because it influences greatly the resulting clusters (Lin, Xie, Song, & Wu, 2009). Some authors add the time dimension to this distance function in order to cluster spatio-temporal data transforming a 2D vector into a 3D vector of $< x, y, t >$ with x and y representing the spatial coordinates and t the time in which the position was recorded (Tork, 2012).

This all brings us to the research question: “How can we integrate the space and time dimensions in the clustering of spatio-temporal data using the SNN algorithm?” To answer this question, an extensive bibliography research was done to know what the main techniques that researchers are using in the spatio-temporal clustering area. With that knowledge, it was expected to implement a working prototype, following a defined approach, which can discover clusters in spatio-temporal datasets using the SNN algorithm.

This work is part of the project "Geo Insight Analytics Platform" leaded by Novabase Business Solutions, which intends to develop a platform that enables the analysis, correlation and visualisation of spatio-temporal data for the telecommunication business.

1.2 - Objectives and Expected Results

In order to answer the research question of this project: "How can we integrate the space and time dimensions in the clustering of spatio-temporal data using the SNN algorithm?" some objectives and expected results were defined.

Clustering spatio-temporal data implies the inclusion of space and time when looking for similarities. In order to be able to add these two dimensions to the clustering process, different distance functions that measure the similarity between objects need to be defined, implemented and tested. Besides the several distance functions, which allow the identification of different types of clusters, the granularity of space and time plays a major role in the clustering process. Along with the spatial and temporal dimension, it is necessary to understand how other dimensions can be added to the clustering process.

This work has as main goal the integration of space and time and one or more semantic attributes in the clustering process, allowing the temporal cataloguing of events and the verification of the clusters evolution across time.

For the accomplishment of this goal, several objectives were set:

- Identification of the several types of spatio-temporal data and their corresponding characteristics;
- Identification of the current approaches that cluster spatio-temporal data as well as their advantages and disadvantages;
- Conceptualization of an approach to cluster spatio-temporal data with or without semantic attributes;
- Implementation of a prototype that uses the SNN algorithm to cluster spatio-temporal data with or without semantic attributes following the proposed approach;
- Validation of the implemented prototype and verification of the quality of the obtained clusters.

The expected results are:

- An approach to cluster spatio-temporal data;
- A working prototype that can cluster spatio-temporal data;
- A sensibility analysis of the influence of each input parameter of the SNN algorithm in the proposed approach.

1.3 - Methodology Approach

So this project can run smoothly and without delays, several research methodologies were studied and Design Science Research (DSR) was chosen because it gives the principles, practices and procedures required to finish a work successfully in this area. This methodology is well known in the Information Systems area and it has been used with good results in several studies (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007).

Besides its applicability to the Information Systems area, DSR is appropriate to this work as it intends to build an artefact that solves a real problem helping in the development of the theory in this area (Hevner, March, Park, & Ram, 2004). Using this methodology, it is expected that the project runs more smoothly and faster than using an ad-hoc approach.

The DSR methodology involves six steps (Peffers et al., 2007): Problem Identification and Motivation, Definition of the Objectives for a Solution, Design and Development, Demonstration, Evaluation and, lastly, Communication.

The methodology is structured in a sequential order but the researcher does not have to do all steps in sequential way. Going back to a previous step may be needed to achieve better results at the end of this process. The methodology and all the connections between steps can be seen in Figure 1.

Applying this methodology to the project in hands, in the first step, it required a literature review in the spatio-temporal clustering area in order to acquire knowledge about the problem. It was also necessary to do a contextualization with the GeoInsight Analytics Platform project in order to understand what has already been done and what are the problems that need to be solved. In the next step, it was defined the objectives that the artefact should achieve according to the context of the problem.

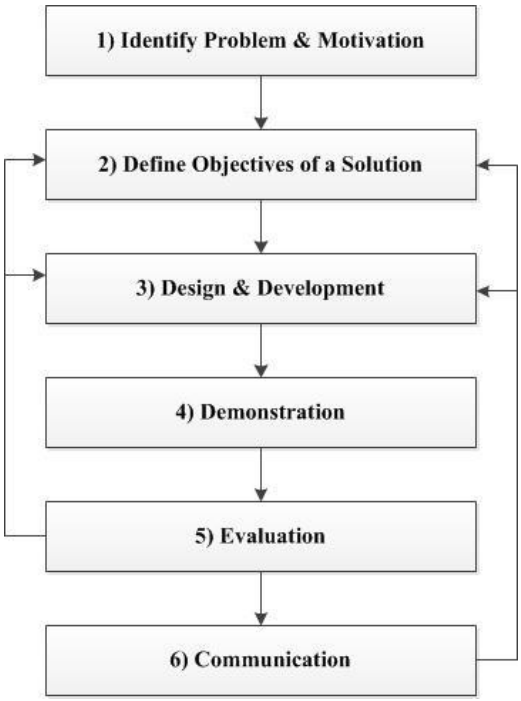

Figure 1 - Design Science Research Process.

Then, the development phase began and a prototype that can be used to cluster spatio-temporal data was implemented.

After that, a long series of tests to the prototype were made. These tests involved the use of synthetic datasets as well as real datasets. The results of these tests were analysed in order to perceive the efficacy and efficiency of the prototype. If the results were not the expected ones for this solution, the prototype went back to the design step in order to improve the results and achieve the optimal solution to the problem.

Lastly, all the process and its contribution were published as a dissertation of a master thesis and as a publication of a paper in an international conference.

1.4 - Structure of the Report

This work is organized as follows. In Chapter 2 the main concepts of this work are described as well as the SNN algorithm. The current approaches used to cluster spatio-temporal data will also be presented. Chapter 3 describes the proposed approach, $4D^{+}$SNN, the types of used spatio-temporal data and the heuristics proposed in this work. Chapter 4 presents the obtained results clustering two synthetic datasets and three real datasets. Finally, Chapter 5 concludes this document with a summary of the main findings and proposals of future work.

2 - CONCEPTUAL FRAMEWORK

In this chapter, several concepts related with the area of this work are presented. After that, several approaches already proposed to cluster spatio-temporal data are summarized pointing out their main advantages and disadvantages. In the end, it will be explained how the SNN algorithm works, mentioning its advantages when dealing with spatio-temporal data.

For a simpler understanding, all the process from classical (through spatial) to spatio-temporal clustering will be explained and divided in three sections.

2.1 - Clustering as a Data Mining Technique

In this section, the concepts of Data Mining and Clustering will be presented. After that, the types of clustering that exist in the literature will be presented and discussed which one is more adequate to the project in hands. Then, it will be shown the algorithm chosen for this project and why it is the most suitable.

2.1.1 - Data Mining

Data Mining is an area with great growth and expansion in the last decades because of the ever growing available data. This concept has many definitions and even different names that came along with the various authors that have been developing studies in this area. Sometimes called Knowledge Extraction, Data Archaeology, Information Harvesting or Data Dredging, the definitions have various shared characteristics that combined can give this final definition: application of methods and techniques in large databases to find tendencies or patterns with the objective of finding knowledge (M. F. Santos & Azevedo, 2005).

These patterns can be rules, affinities, correlations, trends or prediction models. The extraction and identification of useful information and consequently knowledge from large volumes of data are achieved with the usage of statistical, mathematical, artificial intelligence and machine-learning techniques (Turban, Shardam, & Delen, 2011).

Data Mining is one of the key steps of the Knowledge Discovery process (see Figure 2). After being selected, treated and processed in previous phases, the data are analysed using a Data Mining technique. The choice of the technique will be influenced by the type of intended results and, for some cases, more than one technique will be necessary to achieve the proposed objectives because the quality and type of the data influences the final results (M. Y. Santos & Ramos, 2009).

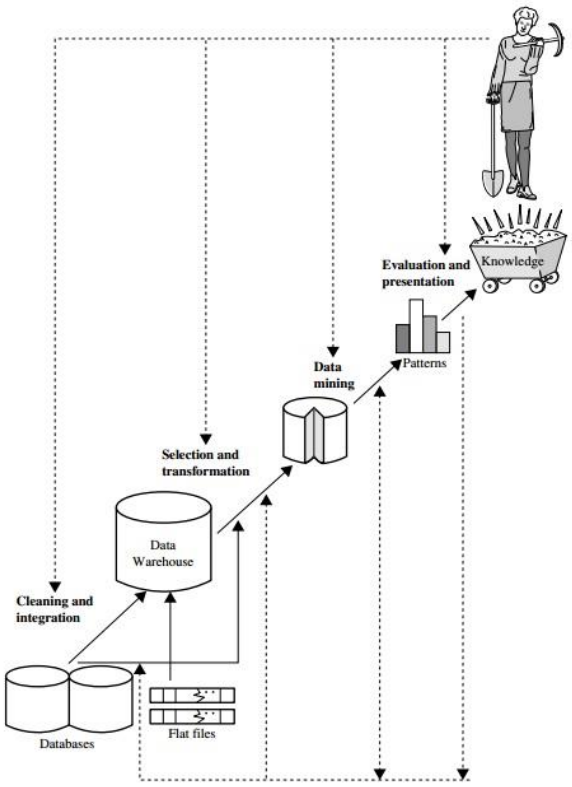

Figure 2 - Data Mining as a Step in the Knowledge Discovery Process (Han et al., 2012).

Data Mining can be categorized into a few types of tasks in accordance with the objectives of the work (Hand, Mannila, & Smyth, 2001):

- Exploratory Data Analysis – exploring the data without any clear idea of what to look for;
- Descriptive Modelling – describe all the data usually with density estimation or cluster analysis;
- Predictive Modelling – construction of models that predict the value of one variable knowing the values of other variables;
- Discovering Patterns and Rules – searching for combinations in the data that occur frequently and indicates a pattern;
- Retrieval by Content – finding similar patterns to a pattern of interest to the user.

2.1.2 - Clustering

The objective of clustering is to identify groups of categories or clusters that divide the analysed data, identifying homogeneous groups of objects. This means that the 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 low inter-cluster similarity and high intra-cluster similarity (Jain, Murty, & Flynn, 1999).

This technique is a non-supervised learning technique because the user does not have any influence in the definition of the clusters. Clusters emerge naturally from the data under analysis using some distance function used to measure the similarity among objects (M. Y. Santos & Ramos, 2009).

One good simple example on how we use clustering in our daily lives is when we do our laundry. In order to keep our clothes with their original colours, we group the white ones, the black ones, the coloured ones, etc. because they have that characteristic in common. This is very important since if we mix the clothes from different groups, they will be ruined in the wash. This task is generally very simple unless we have a white shirt with red stripes and then we do not know for sure in which group to put it. When using clustering techniques in decision making, the clusters are often much more dynamic (sometimes they keep changing everyday) causing the decisions concerning to which cluster an object belongs much more difficult (Berson & Smith, 1997).

The important aspect of this technique is the notion of distance, i.e., how it will be decided whether an object or set of objects is similar to other object or set of objects. This "distance" is not mandatorily a real geographical distance, but a measurement of similarity. This measure can be obtained directly from the objects in study or through vectors of characteristics describing each object (Hand et al., 2001). To compute these distances, a "distance function" is defined to measure if an object is near or far from another (Rinzivillo et al., 2008).

It should be highlighted that clustering is not a standalone technique that gives immediate results. The interpretation of the clusters by the user is an essential part of the clustering task. Only that way, the results have some meaning and value and with that, knowledge (Rinzivillo et al., 2008). As it is an unsupervised technique, the user does not have any real intervention and because of that, sometimes it is difficult to interpret the final result.

However, there are some strategies that can be adopted to overcome this problem. Three of the more popular strategies are (Berry & Linoff, 2000):

- Building decision trees that have the clustering result as the target variable;
- Graphical views which are used to verify how the clustering is affected by changes in the input parameters;
- Checking the differences in the distribution of the variables from group to group, analysing one variable at a time.

2.1.2.1 - Clustering Types

Before explaining some of the approaches the authors use to cluster spatial and spatio-temporal data, it is important to describe the clustering approaches. The clustering technique, according to the majority of the authors, is divided in four main categories (Halkidi, Batistakis, & Vazirgiannis, 2001; Han et al., 2012):

- Partition Clustering;
- Hierarchical Clustering;
- Density-based Clustering;
- Grid-based Clustering.

In the first category (Partition), the dataset is decomposed into a set of clusters. The number of clusters is determined by a *k* number (parameter given by the user) that optimises a certain criterion function. Each cluster must contain at least one object and each object belongs to exactly one or none group, i.e., the same object cannot be in two groups at the same time.

Most partitioning methods are distance-based and can be divided in two groups: Centroid-based or Representative Object-based techniques. The first one defines the centroid of a cluster as the mean value of the points within the cluster. One of the algorithms used for this technique is the *k-means* and it is one of the most used clustering algorithms in literature. The second technique derives from the first one. The way the cluster is characterised is defined from a measure (for example, average distance) between a point and the point that defines the cluster. Some of the algorithms used in this technique are: k-Medoids, PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications) and CLARANS (Clustering Large Applications based on Randomized Search) (Han et al., 2012).

The hierarchical clustering technique groups the data objects into hierarchies or "trees" of clusters. This technique has two different approaches: divisive algorithms and agglomerative algorithms. In divisive, the algorithm starts by considering that all the objects are in one group and, after that, it starts to divide this group in two or more and, also, divide these groups if necessary. This iterative process stops when the maximum number of clusters is reached or the metrics indicate that the set of clusters is the best possible solution. The second strategy is the opposite of the first one. It starts by considering that each object is a group and then integrates clusters to form new clusters. The stopping criteria are the same of the division algorithm (Berry & Linoff, 2000).

The most cited algorithm that uses divisive techniques is CURE (Clustering Using REpresentatives) whilst some algorithms that use agglomerative techniques are BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), Chamaleon and ROCK (RObust Clustering using linKs) (Halkidi et al., 2001).

Unlike partitioning and hierarchical methods, density-based algorithms identify clusters independently of their shape. Typically, they classify dense regions as clusters and classify regions with low density of objects as noise. To achieve that the algorithms usually look for objects that are near to each other. Some density-based algorithms are SNN (Shared Nearest Neighbour), DBSCAN (Density-Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points to Identify the Clustering Structure) and DENCLUE (DENsity-based CLUstEring) (Han et al., 2012).

The last technique mentioned is the space-driven Grid-based clustering. In this, the space is divided into a finite number of cells creating a grid structure. After that, all the operations for clustering are done in each cell. Some of the algorithms used in this technique are STING (Statistical Information Grid), WaveCluster and CLIQUE (CLustering In QUEst) (Halkidi et al., 2001).

After analysing the several types of clustering and looking at their main approaches, it is necessary to look at the context of this work, clustering spatio-temporal data, and select the suitable strategy for the analysis of this type of data.

Partition clustering algorithms are applicable mainly to numerical datasets and they cannot handle noise and outliers. Other disadvantage of this approach is that it only discovers

clusters with convex shape. This type of algorithms needs, as an input parameter given by the user, the number of clusters (Halkidi et al., 2001).

Hierarchical clustering algorithms are, with the exception of BIRCH, are worse than the other types of clustering algorithm in terms of complexity which makes them very slow for large datasets. Instead, BIRCH is faster than the usual hierarchical algorithms but is order sensitive, i.e., with the same input data it may generate different results for different data entry orders. Also, BIRCH does not perform well with clusters of different sizes and shapes (Halkidi et al., 2001).

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 neighbourhood of a point and the minimum number of points in that neighbourhood. This can be a problem because these parameters are very difficult to determine and the algorithms are very sensitive to them (Halkidi et al., 2001).

Finally, some Grid-based algorithms can detect clusters with arbitrary shape but these techniques do not perform well when clustering high dimensional data. Another problem with this type of algorithm is the ratio efficiency/quality, i.e., in order to have clusters with quality, the simplicity and efficiency of the algorithm has to be compromised (Han et al., 2012).

So, it appears that, for the purpose of this work, the density-based clustering algorithms are the more appropriate technique because (Auria, Nanni, & Pedreschi, 2006; Birant & Kut, 2007; Halkidi et al., 2001; Manso, Times, Oliveira, Alvares, & Bogorny, 2010; Rinzivillo et al., 2008):

- Previous knowledge of the dataset is not required (it does not need the number of clusters as input parameter);
- They can discover clusters with arbitrary shapes such as linear, concave, oval, etc. unlike the classical *k-means* and hierarchical methods;
- They can process very large databases;
- They can efficiently separate noise in the dataset;
- They have the ability to discover an arbitrary number of clusters to better fit the data under analysis.

One problem with this type of algorithms is that they require a set of input parameters like the radius of the neighbourhood or the number of neighbours. This problem practically occurs with all types of algorithms and some studies have already been carried out to define heuristics that estimate the values of that input parameters (Ester, Kriegel, Sander, & Xu, 1996; Silva, Moura-Pires, & Santos, 2012).

From the family of the density-based clustering algorithms, the SNN seems to be an appropriate choice for this work because, and unlike the other algorithms in this family, it has the ability to identify clusters of different sizes, shapes and densities. Besides that, in some studies, SNN revealed better results than other density algorithms (Ertoz et al., 2002; Liu et al., 2012; A. Moreira, Santos, & Carneiro, 2005). Although few publications are available about this algorithm, this work tries to contribute to a better understanding of it and to propose extensions that enable it to cluster complex data, as spatio-temporal data.

2.1.2.2 - Shared Nearest Neighbour Algorithm

The SNN algorithm is a density-based clustering algorithm proposed by Ertoz et al. (2002). It has the ability to identify clusters of different (convex and non-convex) shapes, sizes and densities, as well as the ability to deal with noise.

SNN is based on the notion of similarity and defines this similarity between points by calculating the number of nearest neighbours that two points share. The nearest neighbours 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 (A. Moreira et al., 2005). This similarity definition between points allows the algorithm to deal with datasets of variable density, being able to identify clusters of different densities (Ertoz et al., 2002).

This algorithm needs three input parameters: *k, Eps* and *MinPts*. *k* is the number of neighbours, *Eps* defines the threshold density and *MinPts* is the minimum density that a point has to have to be considered a core point (Ertoz et al., 2002).

The most important input parameter is *k* (neighbourhood list size) because it strongly influences the granularity of the clusters. If this parameter 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. The input parameter *MinPts* should be a fraction of the number of

neighbours, *k* (Ertoz et al., 2002). The importance of these input parameters can be seen in Figure 3 where the *k* value was set to different values (8 and 12) and the results are very different, ranging from a large set of very small clusters to a small set of very large clusters.

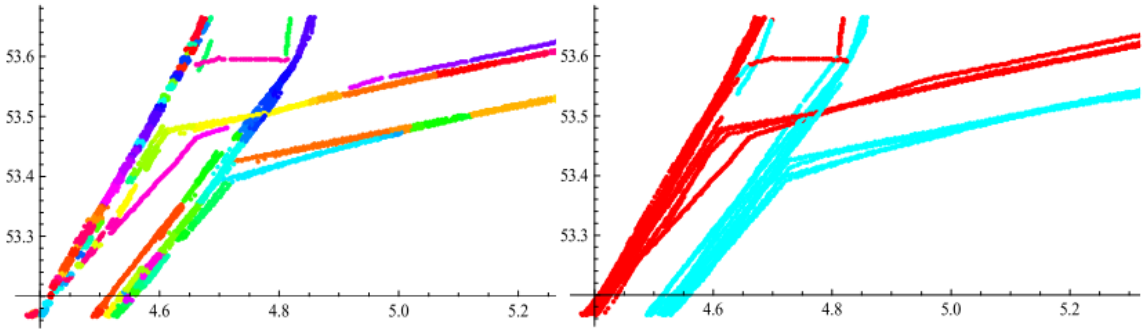

Figure 3 - Clustering Results with Different Values of *k* (M. Y. Santos, Silva, Moura-Pires, & Wachowicz, 2012).

The main steps of the SNN algorithm are presented next (Ertoz et al., 2002):

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:** only keep the *k* most similar neighbours, 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 which are equal or superior;
5. **Find the core points:** filter points that have 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 necessary (Lin et al., 2009).

For example, in spatial data, we could use the Euclidean distance to measure the distance between points. Adding the temporal dimension, 3D vectors of $< x, y, t >$ can be used to distinguish between the objects that are near each other in space and time. As can be seen in Equation 1, this distance function can be adapted to the application domain.

Equation 1: $dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (t_2 - t_1)^2}$

2.2 - Spatial Data

Nowadays, there is an increasing amount of data about the mobility of people, animals, objects, etc. because tracking systems became more advanced. The number of applications using this type of data are increasing in many areas like (Gonçalves, 2012):

- Search for patterns in human mobility;
- Leisure;
- Optimization of vehicles trajectories (boats, airplanes, etc.);
- Implementation of services based on localization;
- Meteorological services;
- Touristic services;
- Mobile computing.

One good example of how a complex problem can be solved with the use of spatial data is the famous case of John Snow (1855) about the cholera epidemic that emerged in London in the XIX century. More than 500 people died in 10 days and the population got scared. Most of them abandoned the city shortly after, which caused an interruption in the city's social and economic life.

At that time, nothing about the cause of cholera was known and the general population thought that it had spread through the air, or in a more superstitious way, that it was caused by the vapours of the places where people that died of cholera were buried, two centuries before.

John Snow did not believe that and he was convinced that cholera spread through contaminated water so he registered in a map of London, the number of people that died of cholera and grouped them by houses (example of clustering) and marked the water pumps available to the population. The result can be observed in Figure 4.

Figure 4 - Map of London by John Snow (1855).

The blue circles are water pumps and the black rectangles are the number of people who died from cholera (the bigger the rectangle, the greater the number). As can be seen, there is a water pump that seems like the epicentre of the epidemic (Broad Street). With this, Snow went to those places and asked where they got the water they used in their daily life and, obviously, all the water was from that particular pump. This case is one of the first documented cases using analysis of spatial data.

Spatial data differs from classical data because unlike the latter, it has coordinate values that refer to a specific reference system. An example of this type of data can be seen in Figure 5.

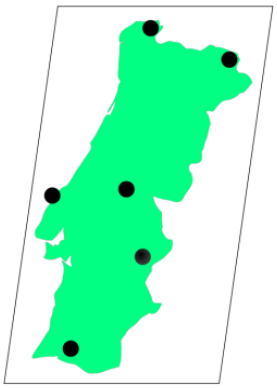

Figure 5 - Example of Spatial Data.

Spatial data can be represented using three different abstractions of space (Bivand, Pebesma, & Gómez-Rubio, 2008):

- Point: refers to a single point location such as a GPS reading;
- Line: contains an ordered set of points that when connected create a straight line;

- Polygon: represents an area, outlined by one or more enclosing lines, which may contain holes in it.

The spatial location used in these abstractions is generally related to a position on the Earth's surface and does not have any temporal information associated with it (Tork, 2012). In spatial data represented by points, sets of vectors are studied that have information about the same characteristics of a given phenomenon in *n* different locations of a limited spatial domain. An example using points to represent spatial data can be seen in Figure 6.

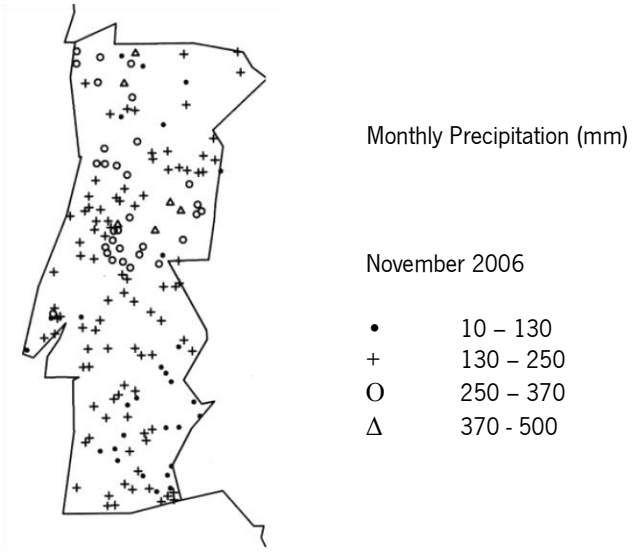

Figure 6 - Monthly Precipitation in November 2006 (Carvalho & Natário, 2008).

The phenomenon in study is the monthly precipitation in Portugal, measured in millimetres, registered in 158 stations of the national meteorological network during November of 2006. Analysing Figure 6 it is possible to verify that although it was a month of strong rains, its distribution was not uniform. With this type of data, a model can be found for the geographical distribution of precipitation in Portugal that can give some estimation for locals that do not have meteorological stations (Carvalho & Natário, 2008).

Polygons or areas are used to refer sets of vectors which have data about the same characteristics of a given phenomenon in *n* sub-regions of a limited spatial domain. This domain is partitioned in various regions, regular or irregular, who have their border well defined using lines (Carvalho & Natário, 2008). In particular, these regions can be created in two different ways, a single categorical variable, such as administrative regions, or a series of attributes like the altitude of the given points as we can see in Figure 7.

Figure 7 - Map of Altitudes of the Iberian Peninsula (source: www.maps.com).

These two kinds of representation usually use colour stains to classify the valour of the variables in study. The interest in this kind of maps is increasing over the decades because they can give a summarized idea and be easily interpreted, according to the regional distribution (Carvalho & Natário, 2008).

2.2.1 - Spatial Data Mining

Looking at the information that is collected in the organizations every day, some is about addresses, postal codes, geographic coordinates, or simply information stored in a map. The associated spatial component of this kind of data makes it more difficult to analyse because there must be a verification of the spatial relations between the items (such as topological, direction or distance information). The "classic" Data Mining techniques cannot process this kind of data unless there are some modifications that allow them to embody the spatial component. This problem has motivated the appearance of some spatial Data Mining algorithms that allow this kind of systems to deal with spatial data (Maimon & Rokach, 2010).

Spatial Data Mining refers to the extraction of knowledge from spatial data so that it is possible to retrieve information and value from it, as well as, discover spatial relationships and relations between both spatial and non-spatial data (Han et al., 2012).

2.2.2 - Spatial Clustering

The spatial clustering process is very similar to the "classical" one. It has the same objective (create groups of objects) but the difference is that, in addition to the similarity aggrupation, in spatial clustering, the position of the object is also important. For example, two objects could be very similar but if they are far apart, they will not be in the same cluster if the distance function only uses the geographical position as the similarity measure. So, spatial clustering can be used to combine spatial and non-spatial attributes of the objects.

This technique can be used, for example, to fight crime. Many police agencies are using the benefits of this technology to identify crime hot spots in order to take preventive strategies like intensive patrolling in the detected problematic areas (Maimon & Rokach, 2010).

2.3 - Spatio-temporal Data

Spatio-temporal data refers to a set of objects with their position registered at different time periods (Rosswog & Ghose, 2008). This kind of data has three components: space, time and object. Associated with these components, there are three basic questions that can be answered: space (where), time (when) and object (what) (Andrienko et al., 2011). Unlike spatial data, this type is more complex because the time attribute can be involved in many different ways. Along with the spatial dimension, this type of data has another dimension to classify: the temporal dimension (Kisilevich et al., 2010).

Different forms of spatio-temporal data types are available. They all share the usage of the two dimensions (space and time) but they differ in the amount of information and the way that information is related between dimensions. For point-wise objects, the main classes of spatio-temporal data types are (Kisilevich et al., 2010):

- Events;
- Geo-referenced variables;
- Moving data item or moving objects;
- Geo-referenced time series;
- Trajectories.

In spatio-temporal events, there is no relation between the items of the dataset and there is no identification for each data item (or it is not relevant to the study). The spatial and temporal information of the items are both static and, because of that, no movement or any kind of evolution is verified. Each event is usually recorded with the location and the corresponding timestamp. One example of this kind of data is the register of earthquakes by sensors (Kisilevich et al., 2010). A sample of such type of data is presented in Table 1. Using earthquakes as example for the dataset in Table 1, it is possible to see six earthquakes registered (two in each time instant) in six different places.

Table 1 - Sample Dataset of Spatio-temporal Events.

Latitude	Longitude	Time
X1	Y1	1
X2	Y2	1
X3	Y3	2
X4	Y4	2
X5	Y5	3
X6	Y6	3

The difference between spatio-temporal events and geo-referenced data items is that the latter, in addition to the spatial and temporal dimension, has an associated non-spatial value. Figure 8 presents an example of a dataset of this type of spatio-temporal data. Each point of the dataset in the spatial domain has a timestamp and a semantic attribute value associated.

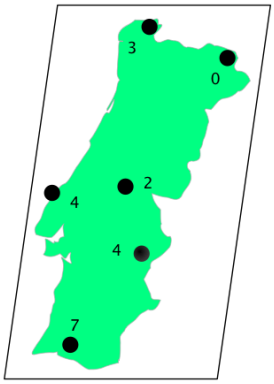

Figure 8 - Example of Geo-referenced Variables.

The next type is the moving data item or moving object. In such datasets, the items are moving and, therefore, the spatial location of the object is also time-changing. The data generated by moving objects is normally of this kind *(id, x, y, t),* where *id* represents the item identifier and *x* and *y* are related to the geographical position, usually longitude/latitude based, of

the object at that time slice *(t)* (Manso et al., 2010). Usually, datasets with this kind of spatio-temporal data only have the last known location and no trace of the past locations is kept such as real-time traffic monitoring. Figure 9 presents an example of a dataset of this kind where the colour of the point is the identification parameter.

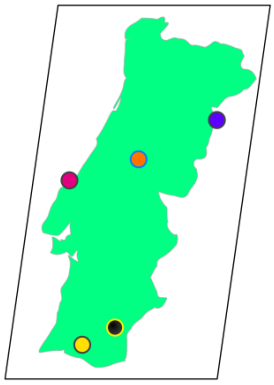

Figure 9 - Example of Moving Objects.

Along with this capability to describe the movement behaviour of the objects, this kind of data can give some other information about the object. The addition of attribute data, which can be static (same value regardless of the position and time, e.g., object type) or dynamic (attribute changes over time, e.g., physical properties such as velocity), is often used in real applications (Mcardle, Tahir, & Bertolotto, 2012).

The two other classes of spatio-temporal data types, geo-referenced time series and trajectories are variations of, respectively, geo-referenced variables and moving objects.

When it is possible to store the attribute variation of an object across time, we have a geo-referenced time series. In this type of data, the spatial dimension of the object stays the same across time whereas the semantic attribute evolves (Kisilevich et al., 2010). This type of data is usually seen in meteorological stations sensors (e.g., temperature or precipitation readings). For example, Table 2 shows the measured temperatures in several cities in three different time instants.

Table 2 - Sample Dataset of Geo-referenced Time Series.

Latitude	Longitude	Time	Value
X1	Y1	1	3º
X2	Y2	1	0º
X3	Y3	1	2º
...	...	...	...
X1	Y1	2	13º
X2	Y2	2	12º
X3	Y3	2	15º
...	...	...	...
X1	Y1	3	3º
X2	Y2	3	5º
X3	Y3	3	10º
...	...	...	...

How this type of data can be represented in a geographical way is presented in Figure 10. For each time period (1,2,3), a different image and different values reflect the values presented in Table 2.

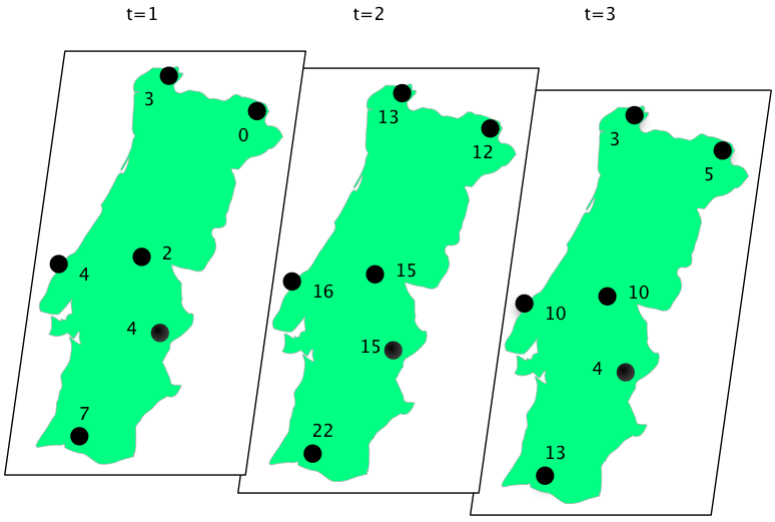

Figure 10 - Example of Geo-referenced Time Series.

Generally, moving objects can be described as trajectories as long as the whole history of the item is stored and available for analysis, i.e., the sequence of spatial positions together with the respective timestamps (Mcardle et al., 2012). One of the main objectives of trajectory clustering is to identify objects with similar movement behaviour, e.g., objects following similar paths in different time periods, objects moving constantly together, etc. (Kisilevich et al., 2010).

A sample of a dataset of trajectories can be seen in Table 3, with the geographical representation presented in Figure 11.

Table 3 - Sample Dataset of Trajectories.

ID	Latitude	Longitude	Time
Black	X1	Y1	1
Black	X2	Y2	2
Black	X3	Y3	3
Blue	X4	Y4	1
Blue	X5	Y5	2
Blue	X6	Y6	3
Orange	X7	Y7	1
Orange	X8	Y8	2
Orange	X9	Y9	3
Pink	X10	Y10	1
Pink	X11	Y11	2
Pink	X10	Y10	3
Yellow	X12	Y12	1
Yellow	X13	Y13	2
Yellow	X14	Y14	3
White	X15	Y15	3

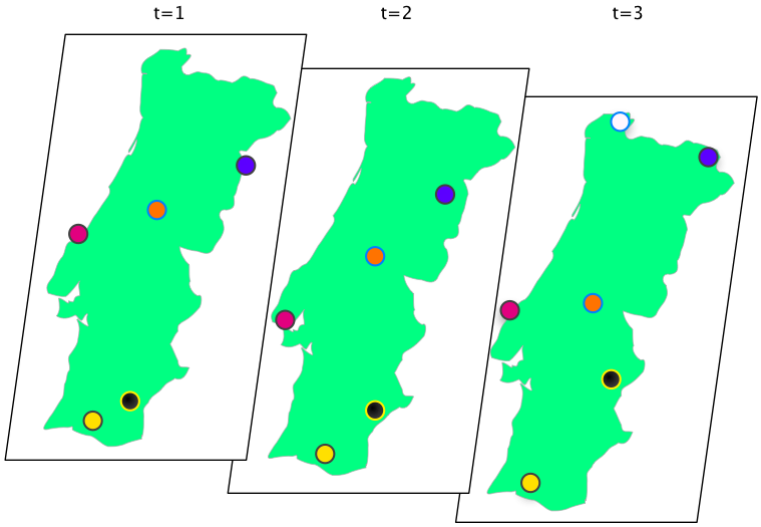

Figure 11 - Example of Trajectories.

As can be seen, the only thing that enabled us to identify the moving pattern of the object was its identification parameter, specifically the colour in this example. Without it, these items would be spatio-temporal events with no relation with each other.

In this work, the simpler type of object in spatio-temporal data was studied, point-wise objects. There are other types of spatio-temporal data that are spatially more complex such as lines or areas. As the focus of this work is point-wise objects, the other types were not studied in more detail.

2.3.1 - Spatio-temporal Clustering

The spatio-temporal aspect of the objects involved in the Data Mining process adds more complexity to it. Along with the spatial relations between objects, both metric (like distance) and non-metric (like shape, direction), arises a new relation, the temporal one (like before or after) that needs to be considered in the Data Mining techniques.

This area of study is relatively new and the development of novel algorithms and techniques for the successful analysis of large spatio-temporal datasets is necessary (Rashid & Hossain, 2012).

As a consequence of the emergence of large amounts of spatio-temporal data, aroused the need to analyse that data to discover new knowledge. This brings more complexity to the clustering algorithms because they have to consider both spatial and temporal aspects of the objects in study in order to discover useful knowledge. Even so, the main idea in clustering remains the same, which is to check the characteristics of the objects and to verify which ones are similar and which are not. Spatio-temporal clustering appeared as a new research area in spatio-temporal Data Mining that is growing especially in geographic information sciences, medical imaging and weather forecasting (Birant & Kut, 2007; Kisilevich et al., 2010).

This area will be the focus of the work developed in this thesis and more about the techniques and algorithms that are being used will be presented below.

Various authors have already studied this type of problem and some of them selected the density-based clustering algorithm family but used those algorithms in different ways.

Birant & Kut (2007) created the ST-DBSCAN based on the DBSCAN algorithm (Ester et al., 1996). First, they filtered the spatio-temporal data in order to get the temporal neighbours and their corresponding spatial values and then, applied the algorithm to create the clusters. The authors use the Euclidean distance to measure the spatial distances between points *(Eps1)* and create another equation, based on the Euclidean distance, to measure the similarity of non-spatial values *(Eps2)*. The data they used to test was composed of locals and temperatures and was in the following format: $A(x_1, y_1, t_1, t_2)$, where x_1 and y_1 are the coordenates of the object, in longitude and latitude, and t_1 and t_2 are, respectively, day time temperature and night time temperature recorded for that position. With another point in the same format (e.g. $B(x_2, y_2, t_3, t_4)$), *Eps1* and *Eps2* are calculated with the following formulas:

Equation 2: $Eps1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Equation 3: $Eps2 = \sqrt{(t_1 - t_3)^2 + (t_2 - t_4)^2}$

This implementation can handle temporal aspects as the algorithm first filter the data by retaining only the temporal neighbours and their corresponding values. The authors define that two objects are temporal neighbours 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 (from two to four), adding more complexity to the algorithm tuning process. With this approach the two dimensions (space and time) are not analysed in an integrated way, requiring the application of rules to preselect the data.

Other study (Pöelitz, Andrienko, & Andrienko, 2010) used the DBSCAN algorithm to perform, first, spatial clustering of the data and, then, the temporal clustering of the obtained spatial clusters. The developed approach is also devised for spatio-temporal events. This strategy was followed by Mcardle et al. (2012) to cluster trajectories. They combined both techniques (spatial and temporal) in a very similar way. First, the authors used spatial clustering to extract spatially similar trajectories and then temporal clustering to those clusters was applied. This approach was tested with a dataset containing a relatively low number of records (120 trajectories).

The analysis of the state-of-the-art undertaken allowed the identification of one work that used the SNN algorithm to cluster spatio-temporal data, which is the work of Liu et al. (2012). The authors extended the SNN algorithm and created the STSNN (Spatio-temporal Shared Nearest Neighbour), clustering spatio-temporal events about earthquakes. This algorithm needs a new input parameter (ΔT), which is added to 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 neighbours.

For the calculation of the spatio-temporal neighbours, the authors use an abstraction of a cylinder as can be seen in Figure 12.

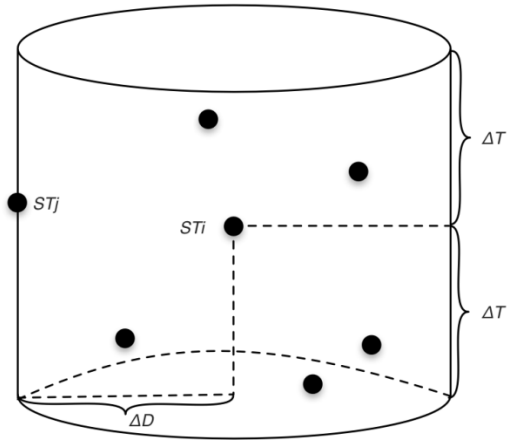

Figure 12 - Spatio-temporal Neighbourhood (Liu et al., 2012).

In the centre of the cylinder is the spatio-temporal event to which the neighbours need to be identified. The height of the cylinder is given by ΔT and the radius is defined by the windowed distance between the spatio-temporal event and its k^{th} closest spatio-temporal neighbour.

This algorithm has the following main steps:

- Identify the *k* spatio-temporal neighbours for each spatio-temporal event;
- For each spatio-temporal event, search the spatio-temporal shared nearest neighbours;
- Calculate the spatio-temporal density of each spatio-temporal event and detect the core ones;
- Expand the clusters by selecting the core events and employing, for each one, a recursive strategy to add all events which are spatio-temporal reachable and spatio-temporal connected;
- Identify noise events.

The authors tested their algorithm with different input parameters and concluded that the value of *MinPts* is dependent and can be a percentage of *k*. This percentage, according to the several tests done by the authors, should be around *0,5k*.

The authors compared the results of the STSNN with the ST-DBSCAN and concluded that the latter cannot find two adjacent clusters with different densities at the same time whereas the STSNN can.

There are other works that use density-based clustering algorithms in other ways like Rinzivillo et al. (2008) who used the OPTICS algorithm (Ankerst, Breunig, Kriegel, & Sander, 1999) to cluster progressively. In this work, the authors created multiple distance functions and

different input parameters and employed them in a progressive way in accordance to the analysis objective. This approach requires that the analyst or domain expert progressively applies different distance functions to gain understanding of the data in a stepwise manner.

To cluster moving objects, Laube, Kreveld, & Imfeld (2005) created a new concept, the REMO-matrix (Figure 13). In this analysis matrix, the motion of the object is recorded 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: Constance, Concurrence and Trend-setter.

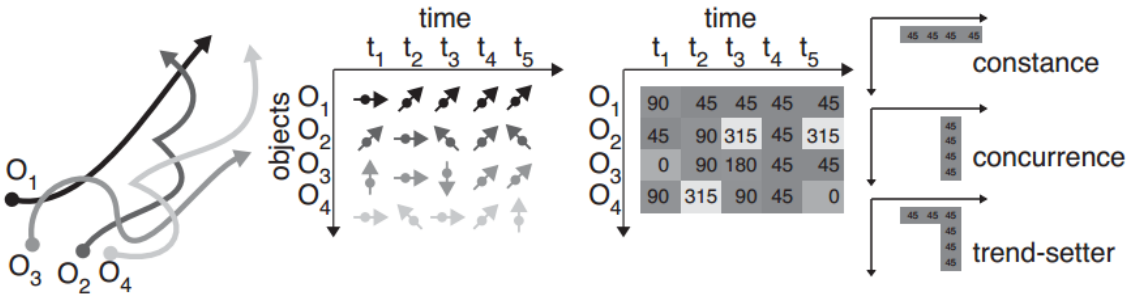

Figure 13 - The REMO Process (Laube et al., 2005).

Applying these patterns to the movement of people or animals allows 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 between the objects.

To be possible the identification of moving patterns it is necessary to understand the types of patterns that may exist in real world datasets. Several moving patterns have been described in the literature. The work of Dodge et al. (2008) proposes a taxonomy for the classification of the movement patterns suggesting that those patterns should be applicable to all known types of movement in the human, animal and objects domain (Figure 14).

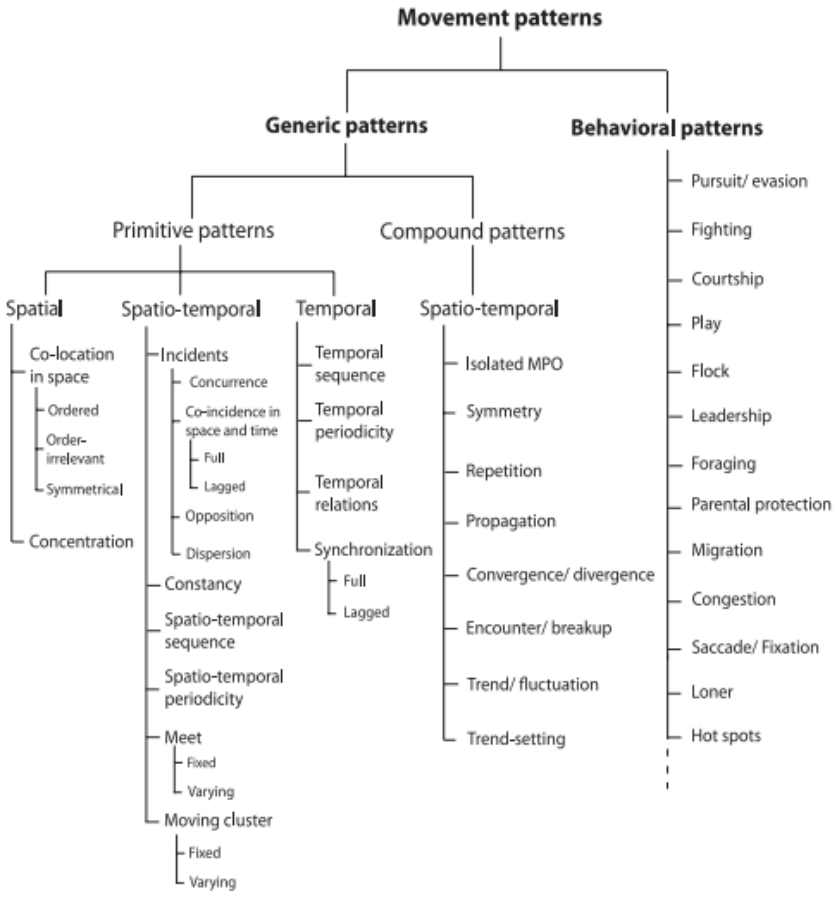

Figure 14 - Classification of Movement Patterns (Dodge et al., 2008).

Generic patterns are simpler than behavioural patterns because the latter include movement patterns that can only be found in certain types of moving objects (e.g. a certain animal species).

The generic patterns are divided in two types: Primitive and Compound Patterns. The primitive patterns are the most basic form of movement where only a single parameter varies whilst the compound patterns are more complex because they involve inter-objects relations.

For the authors (Dodge et al., 2008), this kind of taxonomies is indispensable for the development of movement pattern recognition algorithms that are required to be effective, efficient and as generic as possible.

3 - 4D+SNN: AN APPROACH TO CLUSTER SPATIO-TEMPORAL DATA

The 4D+SNN intends to be an approach to cluster spatio-temporal data using the SNN density-based clustering algorithm thence the SNN initials in the approach name. 4D represents 4 Dimensions and the + symbol in the name means that the approach can treat data with more than 4 dimensions. In other words, the approach in this work will have the ability to cluster datasets with more than 4 dimensions allowing the integration of space, time and one or more semantic attributes in the clustering process. For this to happen, this algorithm has to allow the simultaneous analysis of 4 or more dimensions ensuring that new dimensions can be integrated either adding new attributes to the distance function used by the SNN algorithm or combining several non-spatial attributes in a single semantic one. This algorithm will also be able to deal with different datasets as well as different discovery purposes as the user will have the ability to define the importance of each dimension in the clustering process. This will be done using weights for each dimension in the distance function used by SNN. How these weights work will be explained in detail later in this chapter. Another parameter introduced in the distance function is the normalization parameter that each dimension used in the distance function has to have in order to adapt different scales and units to similar ones so one dimension is not penalised over other.

In order to test the several approaches defined along the project, it was necessary an implementation of the SNN algorithm. Since there are some implementations already available and free to use there was no need to create one tool that could cluster data with SNN from the beginning. From the studied implementations, it was chosen the one implemented by Antunes (2012). The objective of this implementation[1] was to improve the processing time of the SNN algorithm. In order to accomplish that, it was detected that the calculation of the neighbours list was the most inefficient step in the SNN algorithm. To improve the time of that step the author based his implementation in a concept of division of the space in a grid. One positive aspect of

[1] Available at: http://ubicomp.algoritmi.uminho.pt/projects/f-snn

this tool is that it is very easy to modify and to create new distance functions, as well as parameters to use in that distance function, to employ in the clustering process which was important in the several undertaken tests. This implementation was done using Java programming language.

The created procedures to find the normalization parameters presented in this chapter were developed using the R software. R[2] is an open source suite for data manipulation, calculation and graphical display that uses its own programming language (named R) and it is available for various platforms.

The SNN implementation used in this work does not have a graphical option to see the clustering results so other software programs were used in order to view the results in their spatial context. To present the results that appear in the next chapter, two software programs (QuantumGIS and R) were used. QuantumGIS[3] is an open source Geographic Information System that allows the user to view and analyse geospatial information on various platforms (Windows, Linux, MacOS, etc.) and it was used to generate the 2D figures presented in the results chapter. R is a tool presented previously and it was used to create the 3D figures that will also appear in the results chapter.

This chapter continues with the presentation of the types of spatio-temporal data that the 4D+SNN will treat. Then, the distance functions used in this approach to cluster three, four or more dimensions datasets will be presented. In the end of this chapter, the proposed approach to calculate the normalization parameters used by the distance functions will be presented as well as the previous approaches before reaching the final solution.

3.1 - Types of Used Spatio-temporal Data

For the purpose of this work, three types of spatio-temporal data are considered. The first type is associated with the analysis of events, other with the analysis of geo-referenced variables and the last with geo-referenced time series. First, the approach starts with the analysis of events, incorporating simultaneously space and time in the clustering process and, afterwards,

[2] R Official Site: http://www.r-project.org/

[3] QuantumGIS Official Site: http://www.qgis.org/

the approach is extended to deal with geo-referenced data. The last test in this work involves a dataset with geo-referenced time series.

As already mentioned in the previous chapter, spatio-temporal events do not present any relation between the several items in the dataset and there is no identification for each data item (or it is not relevant for the data analysis process). In geo-referenced variables, and besides the spatial and temporal dimensions, a non-spatial attribute is used to characterize the analysed data. Geo-referenced time series are an evolution of geo-referenced variables since they permit the study of the evolution of a semantic attribute in a determined position along the time.

Considering a real dataset integrating 35,941 fires occurred in Continental Portugal in 2011. This dataset will be named Fires 2011 throughout this work. Each fire in this dataset 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 4 shows an extract of this dataset emphasizing where the fire took place (spatial coordinates), when the fire started, its type (forest fire, slash-and-burn, etc.) and the total burnt area (in hectares).

Table 4 - Example of a Geo-referenced Dataset.

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 it is only considered the where (spatial coordinates) and when (time) dimensions, then the clustering process deals with events allowing the user to verify what are the places with more incidence of fires and in what period of the year fires are more frequent. However, treating these data as a geo-referenced variable allows the user to verify, not only where most of the fires are verified and when, but 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 for 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, the 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.

It is now appropriate to clarify the several types of attributes and scales that can be associated with the non-spatial attributes as they influence how the distance function needs to be set to measure the similarity between objects. The two typical types of attributes are discrete and continuous. A discrete value has a finite number of occurrences or possible values, either numeric or categorical (like counts or classes). A continuous value has an infinite number of possible real values (like weights).

Besides types, scales are also relevant. When dealing with qualitative data, the scale can be nominal, where different values are just different names – as colours, types or codes, or ordinal, where values reflect an ordering – as close or far. For quantitative data, the scale can be associated to intervals, where a unit of measurement exists – like temperature scales, or a ratio, where the ratios are meaningful (Steinbach, Ertöz, & Kumar, 2004).

After presenting the types of spatio-temporal data that the 4D+SNN will treat, it will be presented the chosen distance functions used in 4D+SNN that were chosen in order to efficiently treat these types.

3.2 - Distance Function for Clustering Spatio-temporal Data

As seen previously, an effective and efficient distance function is necessary to have appropriate results. Besides influencing the results (as this function will measure the similarity of data points), the complexity of this function will have an impact in the computational time of the clustering process. 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 Equation 4.

Equation 4: $3DDistance(p_1, p_2) = w_s * \frac{Ds(x_1,x_2,y_1,y_2)}{MaxS} + w_t * \frac{Dt(t_1,t_2)}{MaxT}$

With this approach, the user can use any function (Ds and Dt), considering the problem domain to calculate the distances (respectively, spatial and temporal) 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 behaviour of time (days, years, season, etc.).

In this function w_s and w_t are used to assign a weight in the clustering process to each one of these components (spatial and temporal respectively). This way, the user can control the intended results attending to the analytical context. As will be presented in the results chapter, these weights are powerful calibration instruments that the user can employ to tune the clustering results. These weights can be 0 as minimum and 1 as maximum and the sum of them must be 1. So, instead of doing spatio-temporal clustering, the user can with this approach do only spatial or temporal clustering of the data giving all the weight to one of the dimensions.

MaxS and *MaxT* are used to normalize the spatial and temporal dimensions. When looking for the *k*-nearest neighbours of a point, it is expected (and usually is what happens) that the neighbours are relatively close in space and time, which means that the spatial distance, temporal distance and attribute distance to the neighbours 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 penalise the dimension with higher amplitude in its distance values. Moreover, the existence of noise will highly influence those distances. These variables are needed and very important in order to adapt different scales and units to similar ones so one dimension is not penalised over other. With these variables, it is expected that the range of values that the three dimensions have are somewhat similar between them.

For adding more dimensions to the clustering process, in this case a semantic attribute (using a 4D vector of $< x, y, t, a >$) with continuous values, the distance function needs to be extended (Equation 5). The new dimension also needs to be normalized and for that the *MaxA* value has to be calculated. A weighting factor (w_a) is also added allowing the user to control the type of patterns that can be obtained.

Equation 5: $4DDistance(p_1, p_2) = w_s * \frac{Ds(x_1,x_2,y_1,y_2)}{MaxS} + w_t * \frac{Dt(t_1,t_2)}{MaxT} + w_a * \frac{Da(a_1,a_2)}{MaxA}$

This extension of the SNN algorithm allows the simultaneous analysis of 4 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.

MaxA is used in this equation to normalize the semantic attribute dimension. As *MaxS* and *MaxT*, it is not appropriate to set this value to the maximum possible value in order to penalise this dimension over the others.

Da is similar to functions Ds and Dt. The user can employ a function that suits the attribute domain or even employ a function that deals with two or more semantic attributes combined in one. If the user does not want to combine two or more semantic attributes in one because it is not possible to do it, it is still possible to add two or more attributes. All the user has to do is add, to each attribute, a weight, a distance function to calculate the attribute similarity and a *MaxA* (Equation 6).

Equation 6: $4DDistance(p_1, p_2) = w_s * \frac{Ds(x_1,x_2,y_1,y_2)}{MaxS} + w_t * \frac{Dt(t_1,t_2)}{MaxT} + \sum_{i=1}^{n} w_{a_i} * \frac{Da_i(a_{i_1},a_{i_2})}{MaxA_i}$

Equations 5 and 6 allow the simultaneous 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.

The normalization parameters used in the distance functions presented before are essential in the clustering process so that a dimension is not penalised because it has higher amplitude of values in its domain. So, it was necessary to find what the optimal values for these parameters were.

This was not a trivial problem to solve therefore the process to find the heuristic to calculate the normalization parameters will be presented in the next subsection. Besides showing the final heuristic, the possible solutions that were proposed before the final solution will be presented.

From the beginning, it was decided that if the heuristic needs to compare distances between points, it could not compare all the points with every other points because with huge datasets (the ones usually used in this area) the processing time would be intolerable. So, one of the requirements to the heuristic was that it was not very time consuming. The other was that *MaxS*, *MaxT* and *MaxA* should emerge from the data under analysis. Only by that, can it be assured that the solution has a correct value for these parameters to all datasets.

3.3 - Identification of Normalization Parameters

In this section, the different approaches to calculate the normalization parameters will be reported. It was decided that first it would only be studied how to calculate the parameters *MaxS* and *MaxT* because with that two parameters it would be possible to cluster spatio-temporal events (one of the objectives of this work) and see some results. Only after achieving good results with the heuristic, the next step that involved calculating the *MaxA* would follow.

The first proposed approach tries to calculate the normalization parameters using the density of the dataset. The second approach tries to divide a dataset in *k* (the SNN input parameter) intervals in order to understand which will be the maximum distance between a point and the further point in the neighbour list of the SNN. The third approach is based in an approach proposed by the creators of the DBSCAN algorithm to find the zone of the dataset that differentiates the outliers from the other points. The fourth approach (final) is based in the two previous approaches (2 and 3). It finds the normalization parameters using a specific point in the distance differences list (these approaches will be presented in more detail in this chapter).

3.3.1 - Approach 1: Density

The first approach to solve this problem was through the density of points of the dataset. As the algorithm used in the clustering process is based in the density of points, it seemed a logical option to try to search the normalization parameters the same way. The first step was to try to calculate the *MaxS* since it seemed more complicated to get than *MaxT* because *MaxS* involves two dimensions unlike *MaxT* that only has one dimension.

In this approach, firstly was created a grid with the size of the dataset (using the minimum and maximum coordinates of the dataset). Then, the grid was overlapped with the spatial distribution of the points in the dataset. For each cell of the grid, the number of points of the dataset that were in that cell was calculated. After calculating the area of the cell, some variables were calculated: the density of each cell, the average density of the entire grid and the average density of the cells that had points (removing empty cells). Figure 15 presents a graphic with the various cells overlapped with a dataset. The dataset used in the figure was extracted from Twitter by Faustino (2012) and represents tweets made in the United States of America on

several days of December of 2011, of January 2012 and of February 2012. The scale in the right of the figure represents the number of points of each cell, blue means cells with high number of points, and consequently high density, and darker pink means cells with no points.

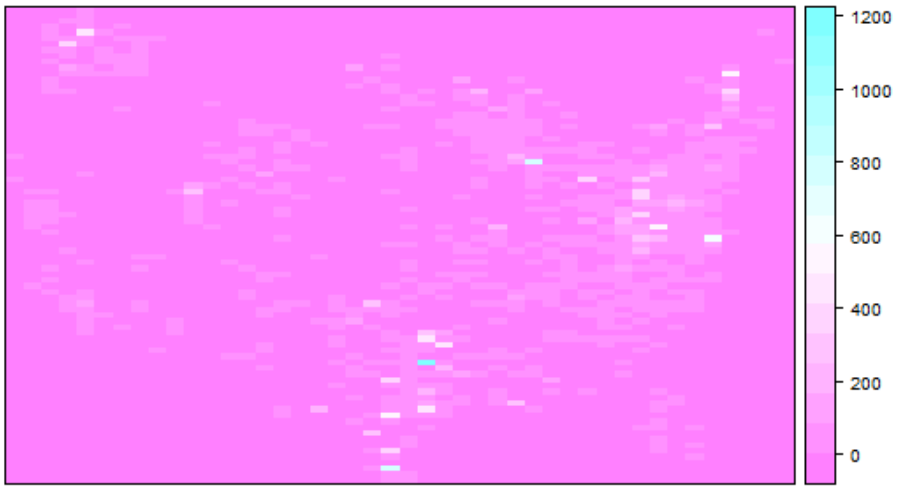

Figure 15 - Dataset with an Overlapped Grid.

The problem with this approach was that it was not possible to get any value that could be used as the normalization parameter *MaxS*. Another issue was to understand which value should be used as the size of the cell (matrix granularity) because that would influence greatly the results of the density of each cell and, consequently, influence the average density of the cells that had points.

3.3.2 - Approach 2: *k* Interval Dataset

The next hypothesis proposed was to try to understand at which distance the k^{th} neighbour would be. As the clustering algorithm (SNN) used in this work is based in the notion of neighbourhood, it seemed logical to use as the normalization parameter the maximum distance between a point and its k^{th} neighbour.

The objective of this approach was to divide the dataset in *k* sized intervals and thus understand which the maximum distance in a dataset between a point and its k^{th} neighbour was. *k* is the SNN input parameter (number of neighbours) that would need to be known to use this approach.

In order to do that, first a bounding box for the spatial component was created. Then, the spatial differences between every point in the dataset and the lowest left point of the

bounding box are calculated. After that, this distances list is sorted by ascending order and the differences between consecutive positions are calculated. Then, this differences list is sorted by ascending order.

After these calculations, and knowing the SNN parameter *k* that will be used in the clustering process, it divided *k* by the total number of points in the dataset *(n)*. Then, the list of the sorted differences between points is divided by the *k/n* value in a similar way to the quartiles method but instead of using 25% as the quartile it is used *k/n* %. Therefore, the approach divides the sorted data into k/n equal parts so that each part represents 1/(k/n) of the population.

With the complete differences list, the maximum difference between consecutive positions was calculated and that value was used as the normalization parameter. This difference was chosen to be the *MaxS* parameter because it would be the maximum distance that a point would be to its k^{th} neighbour.

Figure 16 presents a sample of the list generated. This sample was taken from the analysis of t5.8k dataset (it will be presented in more detail in the results chapter). It has 8009 points and using a value of 20 as *k*, the division of *k/n* is 0.002497191. So, that value was used as a percentage to divide the list of differences (0.2497191%).

0.0000000%	0.2497191%	0.4994381%	0.7491572%	0.9988763%	1.2485953%
269.9676	271.6426	272.9683	273.6074	274.2214	274.7840
1.4983144%	1.7480335%	1.9977525%	2.2474716%	2.4971907%	2.7469097%
275.5847	276.0011	276.4582	276.9156	277.3423	277.7331
2.9966288%	3.2463479%	3.4960669%	3.7457860%	3.9955051%	4.2452241%
278.1960	278.6256	278.9174	279.2582	279.5541	279.9748

Figure 16 - Sample with the First Elements of the Differences List Divided by *k* Intervals.

Algorithm 1 presents the heuristic described previously in a more concise and structured way.

Algorithm 1 Calculate *k* intervals

requires: dataset, *k*, lowest left point of the bounding box

```
1:  function calculate k intervals
2:      for all points do
3:          calculate distance between the point and the lowest left point of
        the bounding box
4:      end for
5:      sort the distances by ascending order
6:      for all distances do
7:          calculate the differences between consecutive positions
8:      end for
9:      sort differences
10:     divide k by the number of points in the dataset
11:     divide the sorted differences list in k/n equal parts
12:     for all parts do
13:         calculate maximum distance between consecutive parts
14:     end for
15: end function
```

The *MaxT* parameter was calculated in a similar way. The only difference is in the beginning of the process when, instead of calculating the spatial difference between all points and a point in the bounding box of the dataset, the temporal distance between all points and the minimum time instant present in the dataset was calculated.

Using the *k* value of the SNN parameters (the size of the neighbours list), it was expected that this heuristic could find, with some expected error, where the further point in the *k* list of the SNN algorithm would be and use that distance (either spatial or temporal) as the normalization parameter. With this value, some tests were conducted with the synthetic datasets, which will be presented later in this document, and the clustering results were interesting since it was possible to discover the patterns that were supposed to find.

This heuristic had one major problem that was when *k* was not a common divisor of *n*. For example, using the same scenario of the t5.8k dataset, if there were 8009 records in a dataset and *k* was 20, the *k/n* would give 0.249719... % and, because of that, dividing the list would never reach the 100% leaving records outside of the calculation as can be seen in Figure 17. This was not acceptable as this could influence the normalization parameter value and, as such, this heuristic was abandoned. Another problem of this heuristic was that *k* would have to be known before doing the clustering process. This would prove to be an obstacle since the search for the optimal SNN parameters was not a trivial task as will be explained later.

97.3904358%	97.6401548%	97.8898739%	98.1395930%	98.3893120%	98.6390311%
532.3714	533.1904	534.0803	535.1731	535.9807	536.6471
98.8887502%	99.1384692%	99.3881883%	99.6379074%	99.8876264%	
537.4509	538.2364	539.0942	542.7804	552.5926	

Figure 17 - Sample with the Last Elements of the Differences List Divided by *k* Intervals.

3.3.3 - Approach 3: Valleys

The third proposed approach was inspired by a heuristic similar to the one used by Ester, Kriegel, Sander, & Xu (1996) in which they used a valley function to understand the distribution of the density of the points in a dataset (Figure 18).

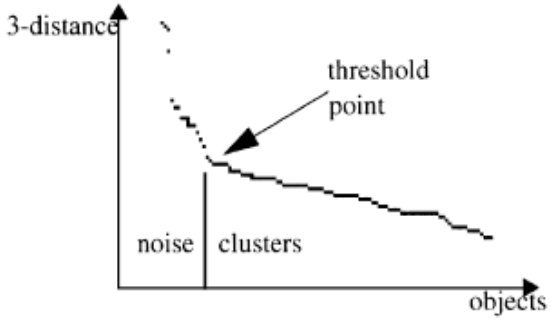

Figure 18 - Sorted 3-dist Graph (Sander, Ester, Kriegel, & Xu, 1998).

The objective was to find breaks in the *k-dist* function since these breaks would mean that the following points from these breaks were only outliers that are far away from the rest of the dataset. So, first it calculates the distances between consecutive points as the previous heuristic (using the lowest left point of the bounding box). After that, this distances list is sorted by ascending order and the differences between consecutive positions are calculated. After calculating the differences between consecutive points and saving them in a list by ascending order, the differences between consecutive positions of this list are calculated as well as the average of differences until that point. If that difference is greater than the average multiplied by a determined factor (initialised at 5), the jump between these points would be considered a break (valley). After that, the average value is initialised at 0 and the procedure is repeated for the rest of the array. If no break was found then the whole procedure is repeated but with a lower factor (factor - 1). With this heuristic, the objective was to calculate the frontier point between zones with high density and zones with low density of points.

Algorithm 2 presents the heuristic described previously in a more concise and structured way.

Algorithm 2 Calculate valleys

requires: dataset, factor, lowest left point of the bounding box

```
1:    function calculate valleys
2:        for all points do
3:            calculate distance between the point and the lowest left point of
          the bounding box
4:        end for
5:        sort the distances by ascending order
6:        for all distances do
7:            calculate the differences between consecutive positions
8:        end for
9:        sort differences
10:       for all differences do
11:           calculate the difference between consecutive differences
12:           if difference > average difference * factor then
13:               found valley
14:       end for
15:   end function
```

One problem detected in this heuristic was that in datasets with many records that had the same distances differences (for example, many distances between points of 1), the differences average would be very low, and when it found the next difference (continuing the example, a difference of 2) it would consider a break (because the difference would be greater than the average multiplied by the factor) even when using factors greater than 5. This problem happens when the range of values the dimension has is small, which leads to having many differences with the same result that consequently leads to the problem mentioned earlier. This was problematic because it would be very difficult to understand what breaks were detected because they were points that divide high density zones from low density zones from the breaks incorrectly detected.

```
> valleysT
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
 [20]  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38
 [39]  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57
 [58]  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76
 [77]  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95
 [96]  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114
[115] 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 134
[134] 135 136 137 138 139 141 142 143 144 145 146 147 149 151 152 153 155 157 158
[153] 159 162 164 165 166 168 169 170 173 174 178 180 181 191 194 204 240 266 306
[172] 380 492
```

Figure 19 - Result of the Valleys Approach for the Temporal Dimension for t5.8k.a Dataset.

As can be seen in Figure 19 (this dataset is a variation of the t5.8k mentioned in the previous heuristic and it will be presented in more detail in the results chapter), using a factor of 5, the heuristic found 172 breaks in this dataset. Only with a graphical help the user could choose which the best value to use as normalization parameter was. With this, the process automaticity would be lost and the responsibility of choosing the normalization parameters would fall in the user.

3.3.4 - Approach 4: Deciles

The final approach used to identify the normalization parameters was a mixture of the last two approaches (*k* interval and valleys approach). It starts by identifying the bounding box for the spatial component. For each point present in the dataset, 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 consecutive 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 neighbours and the identification of those distances that are influenced by the presence of noise points in the dataset (Figure 20). This process was inspired by the *k-sort graphs* proposed by Ester et al. (1996).

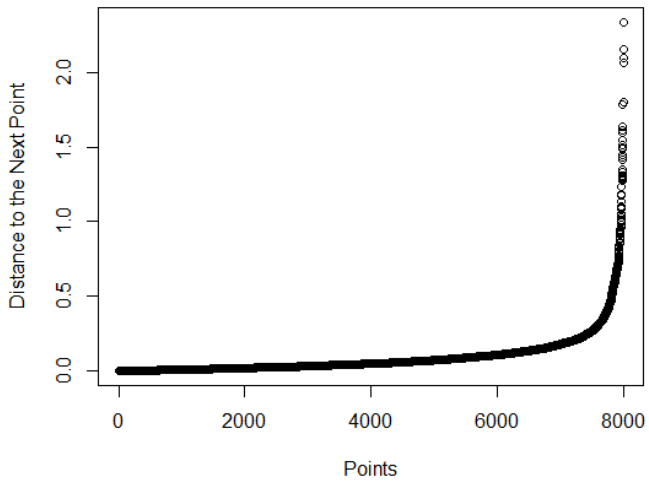

Figure 20 - Sorted Distances in t5.8k.

The analysis of several datasets, either synthetic or real, allowed the identification of the distance value given by the 80% *decile* as an appropriate value for the *MaxS* variable. The distance present in this *decile* split the distances that are usually associated to neighbours values and those that start to be influenced by noise points. When in a spatially homogeneous dataset,

i.e., the points of the dataset 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. Algorithm 3 presents the heuristic previously described in a more concise and structured way.

Algorithm 3 Calculate Spatial Deciles

requires: dataset, lowest left point of the bounding box

```
function calculate spatial deciles
    for all points do
        calculate distance between the point and the lowest left point of the bounding box
    end for
    sort the distances by ascending order
    for all distances do
        calculate the differences between consecutive positions
    end for
    sort differences
    calculate deciles of the differences list
    choose 80% decile value
end function
```

Figure 20 presents the differences between distances in a synthetic dataset and Figure 21 presents these differences but in a real dataset (both synthetic dataset t5.8k and real dataset Fires 2011 will be discussed later).

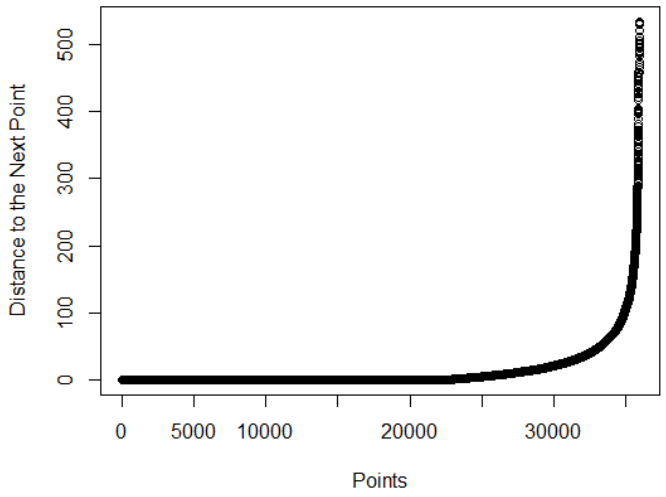

Figure 21 - Sorted Distances in Fires 2011 Dataset.

As can be seen, the figures present a similar graph which indicts that the 80% *decile* seems to be a good option as it is a point in the dataset where the distances between points start

to grow a lot because it is the zone where the outliers of the dataset appear. Therefore, this value separates the outliers from the rest of the points of the dataset.

For the temporal and semantic attribute dimensions, the identification of *MaxT* and *MaxA* follow 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 value present in the dataset is calculated. The same applies to the calculation of *MaxA* where the distance between each point attribute is calculated to the minimum attribute value present in the dataset.

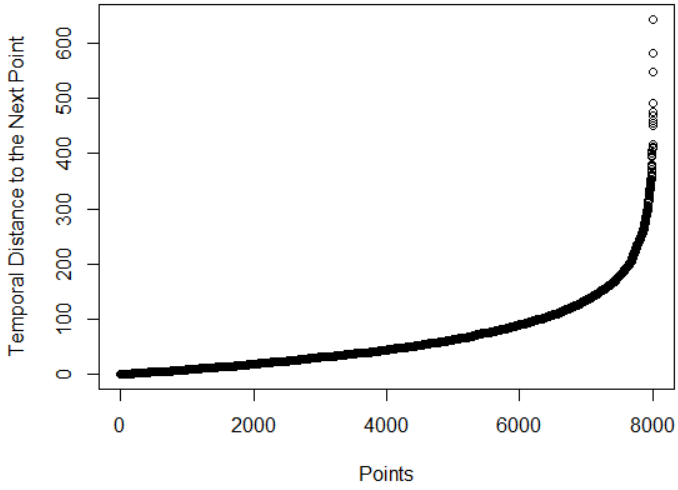

Figure 22 - Sorted Temporal Distances in t5.8k.a.

As can be seen in Figure 22 and Figure 23, the other two dimensions (temporal and semantic attribute) showed a similar behaviour, many points were very close to each other at the beginning and, at a specific point, (around 70-80% for the temporal distances and around 90% for the semantic attribute distances) the graph starts to grow exponentially. The sorted attribute graphic is a little different from the spatial and temporal graphics because it started to grow after the 70-80% point. This pattern appeared in all studied datasets that had a semantic attribute. For this reason, it was created the rule mentioned earlier that states that when 80% decile value is still 0 the algorithm successively increments the 80% value by 1% until a difference different than 0 is found.

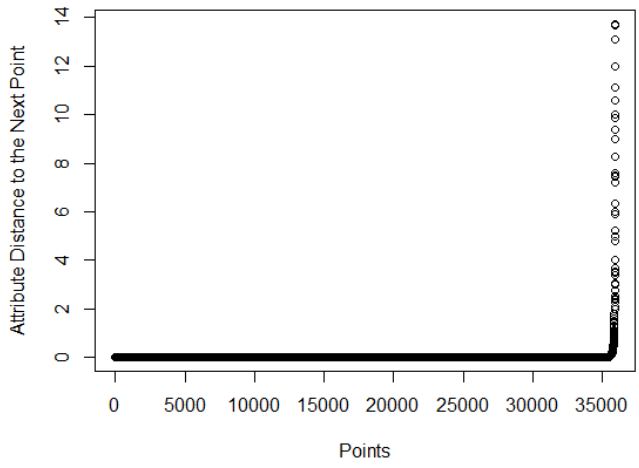

Figure 23 - Sorted Attribute Distances in Fires 2011 Dataset.

This approach was based in the last two approaches (*k* interval and valleys) as it uses the quartiles method but with deciles and defines where the "break" to use as the normalization parameter is, instead of trying to search for it in the dataset. This approach presented the best clustering results in this work and they will be shown in the next chapter.

Next, some results that this approach gives are presented. Figure 24 shows the result for the spatial dimension with t5.8k dataset.

```
> quantile(sortedSpatialDifferences, probs = seq (0,1,0.1))
          0%          10%          20%          30%          40%          50%          60%
0.000000000  0.007338237  0.015326559  0.024575189  0.035932544  0.049370356  0.067251413
         70%          80%          90%         100%
0.093116543  0.129618619  0.204888862 13.368486521
```

Figure 24 - Spatial Deciles Result for t5.8k.

As the t5.8k has two different temporal configurations, the temporal deciles reflect that difference. As can be seen in Figure 25 and Figure 26, t5.8k.a has greater amplitude and thus a greater value in the 80% decile than the 80% decile in the t5.8k.b (both t5.8k.a and t5.8k.b datasets are variations of the t5.8k dataset that will be explained in detail in the results chapter).

```
> quantile(sortedTimeDifferences, probs = seq (0,1,0.1))
  0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
   0    7   14   23   33   44   59   78  104  149  644
```

Figure 25 - Temporal Deciles Result for t5.8k.a.

```
> quantile(sortedTimeDifferences, probs = seq (0,1,0.1))
  0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
   0    2    4    7   10   14   19   25   34   50  250
```

Figure 26 - Temporal Deciles Result for t5.8k.b.

As the semantic attribute dimension normally has amplitude smaller than the spatial and temporal dimension, it can be difficult to find the usable value in the 80% decile. So, if the 80% value is 0, it increments by 1% until reaching a value different of 0. Figure 27 presents the result of this heuristic using the Fires 2011 dataset that will be described in more detail in the next chapter.

80%	81%	82%	83%	84%
0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
85%	86%	87%	88%	89%
0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
90%	91%	92%	93%	94%
0.0000e+00	0.0000e+00	0.0000e+00	1.0000e-04	3.0000e-04

Figure 27 - Attribute Decile Result for Fires 2011 Dataset.

4 - RESULTS

This chapter shows the results achieved with the proposed approach when clustering spatio-temporal events, geo-referenced variables and geo-referenced time series. At the end of this chapter, a brief discussion is presented with some conclusions. Several datasets were used and for each one the heuristic previously presented was used in order to understand what values to use as parameters *MaxS*, *MaxT* and *MaxA*.

One of the major problems of this work was the search for the optimal SNN parameters. These values influence greatly the outcome of the clustering process and there is no pre-designed method to achieve these parameters which makes this process more difficult, mostly in real (non-synthetic) datasets (Bouguessa, 2011). Because of that, for each dataset, the usual procedure of trial and error was used to try to achieve the optimal SNN parameters.

In order to have a starting point in the search for the SNN parameters, an approach proposed by G. Moreira, Santos, & Moura-Pires (2013) was used. In this work, the authors did an extensive series of tests on different datasets in order to perceive a pattern that could indicate what the *k* parameter value should be. After knowing *k*, this approach can search for suitable values for the other two SNN input parameters *(Eps* and *MinPts)*. With the values given by this approach, the first test was performed and according to the results, the SNN parameters were adjusted to the final parameters used in the results presented next.

4.1 - Distance Function

As reported in the previous chapter, this approach can handle any function that the user finds suitable. For the purpose of this work and for all the results presented next, the spatial distance function chosen was the Euclidean distance (Equation 7) to calculate the spatial distances between points.

Equation 7: $Ds(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

For both time (Equation 8) and semantic attribute (Equation 9), the absolute value of a simple subtraction was used since these are simpler dimensions (only one dimension) than the spatial dimension and this way some processing time could be diminished.

Equation 8: $Dt(p_1, p_2) = |t_1 - t_2|$

Equation 9: $Da(p_1, p_2) = |a_1 - a_2|$

4.2 - Spatio-temporal Clustering

This subchapter presents the synthetic datasets t5.8k and t4.8k and the real dataset Fires 2011. Then, the clustering results using the 4D^{+}SNN approach with these datasets will be shown. In this section, will the spatial and temporal dimensions of the datasets will be used.

4.2.1 - t5.8k

The first dataset used in this work was a synthetic one, it is named t5.8k and it was created by Karypis, Han, & Kumar (1999). It integrates 8009 points which spatial distribution is shown in Figure 28.

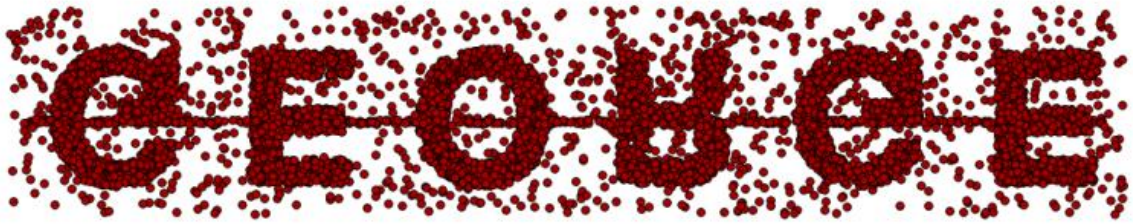

Figure 28 - Spatial Distribution of t5.8k Dataset.

To this dataset, two different modifications (named t5.8k.a and t5.8k.b in this work) were made to add a temporal dimension. The first modification (t5.8k.a) was to separate vertically the dataset in six days so that each letter was in a different day. For each day, the several points were randomly distributed along the day in minutes. The dataset is distributed equally along the time as it can be seen in Figure 29 where the temporal distribution of the dataset is presented.

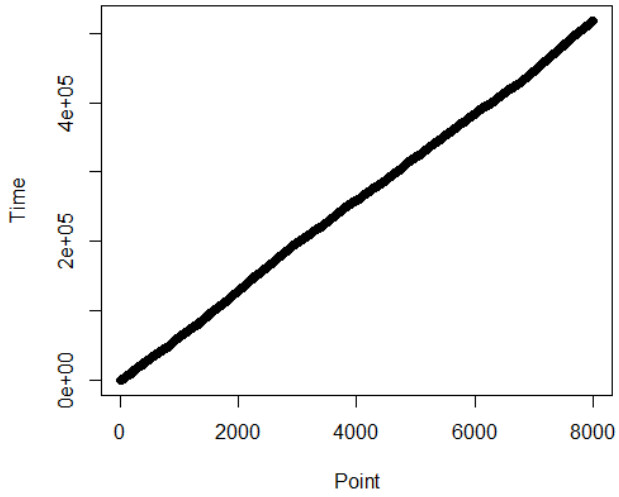

Figure 29 - Temporal Distribution of t5.8k.a.

The second transformation (t5.8k.b) was to assign the first four letters (G-E-O-R) to one day and the other two (G-E) to the following day. This separation was also done vertically and as such the points were also randomly distributed in each day. As can be seen in Figure 30, t5.8k.b has a different temporal distribution from t5.8k.a.

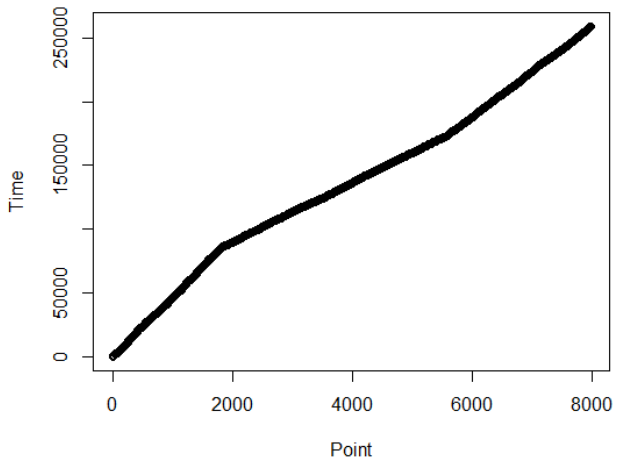

Figure 30 - Temporal Distribution of t5.8k.b.

Using the 4D^{+}SNN approach, and for the t5.8k.a dataset, the *MaxS* used was 0.129618619 km and *MaxT* 104 seconds. Figure 31 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 (319 noise points). This result ensures the adequacy of the distance function defined to measure the similarity of the objects and confirms the correctness of the heuristic applied to identify the variables used in the normalization of each dimension, namely *MaxS* and *MaxT*.

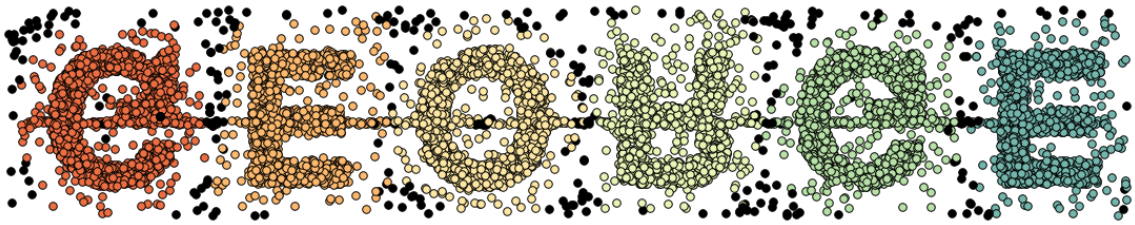

Figure 31 - Result of Spatio-temporal Clustering for t5.8k.a Using 50%-50% Weights (SNN Parameters, *k* = 40, *Eps* = 16, *MinPts* = 24).

For the dataset t5.8k.b, the *MaxS* used was the same as for the t5.8k.a (since they were spatially identical) and it was used a *MaxT* of 34 seconds. 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 32). Inverting the importance of each dimension in the clustering process, increasing the weight for space to 50% and 50% for time, giving more importance to where points are located and not when they were verified, allows the identification of a similar result to the one presented in Figure 31, only with small differences in the identified noise points (Figure 33).

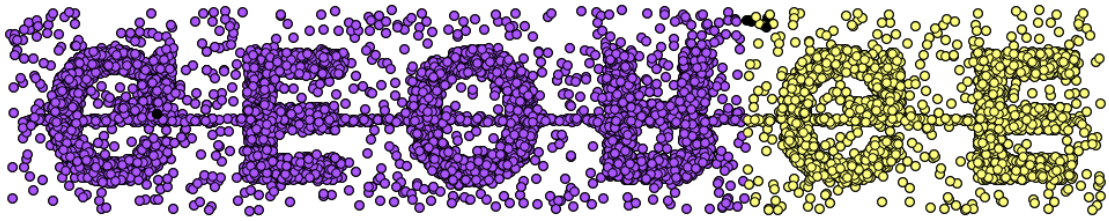

Figure 32 - Result of Spatio-temporal Clustering for t5.8k.b Using 20%-80% Weights (SNN Parameters, *k* = 40, *Eps* = 16, *MinPts* = 24).

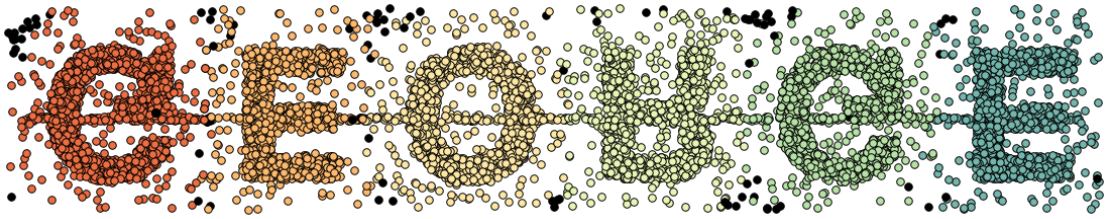

Figure 33 - Result of Spatio-temporal Clustering for t5.8k.b Using 50%-50% Weights (SNN Parameters, *k* = 40, *Eps* = 16, *MinPts* = 24).

Using the same weight for the two dimensions results in 6 clusters (letters) but with a very low number of noise points (107 points) identified because the temporal dimension is still joining points that are spatially distant (Figure 33). Giving 80% weight to the spatial dimension (Figure 34) is possible to have the 6 letters clearly identified (with 392 noise points) and have a result similar to 50%-50% weights for t5.8k.a (Figure 31).

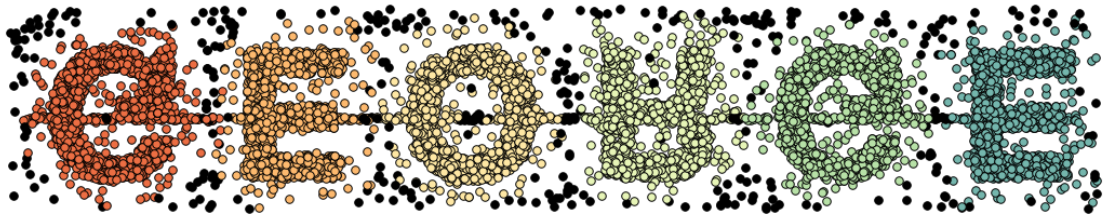

Figure 34 - Result of Spatio-temporal Clustering for t5.8k.b Using 80%-20% Weights (SNN Parameters, *k* = 40, *Eps* = 16, *MinPts* = 24).

Figure 32, Figure 33 and Figure 34 present the evolution of the clustering process using different weights, starting with a low weight value and finishing with a high weight value for the spatial dimension. The temporal dimension follows the contrary pattern. As can be seen, these weights influence the clustering process and are powerful parameters that can be tuned in order to achieve better results (according to the clustering objective).

4.2.2 - t4.8k

The second synthetic dataset used is t4.8k. As the previous one, it was created by Karypis et al. (1999) and it also has 8009 points. This dataset was temporally divided in 3 days, joining in the same day different spatial clusters as can be seen in Figure 35 where the blue lines separate consecutive days. This dataset had the same attributes as the t5.8k dataset and a *MaxS* of 0.10945188 km and a *MaxT* of 50 seconds was used in the clustering process.

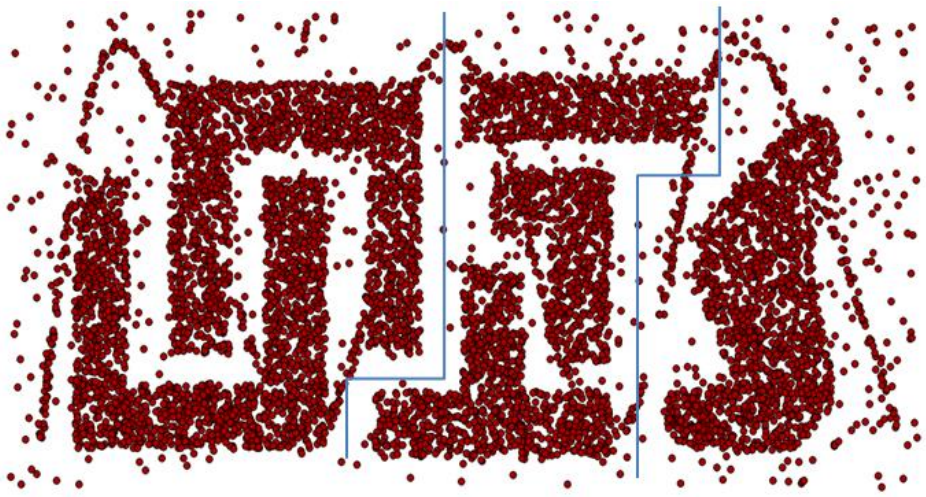

Figure 35 - Spatial Distribution of t4.8k.

As the division is similar in terms of space and time importance, using a similar weight for space and time produces a clustering result that reflects the division performed in the dataset (Figure 36).

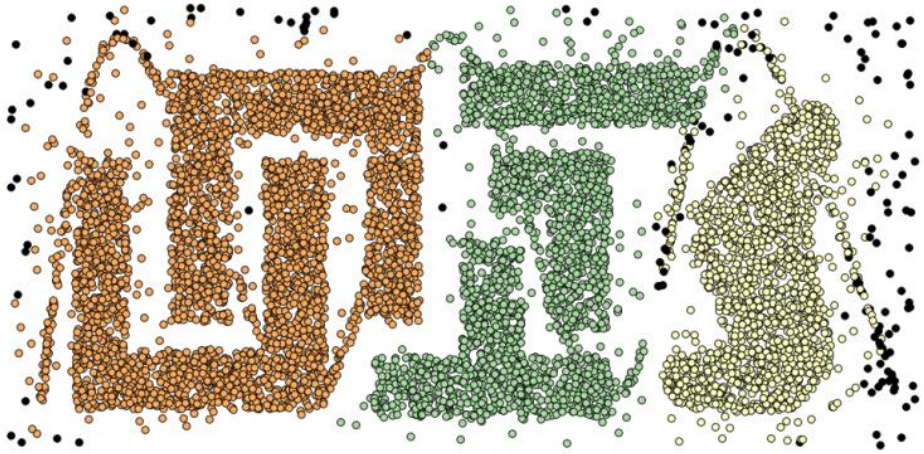

Figure 36 - Result of Spatio-temporal Clustering Using 50%-50% Weights (SNN Parameters, *k* = 40, *Eps* = 16, *MinPts* = 24).

Increasing the weight of space (75%) over time (25%), the 4D^{+}SNN approach is able to find the 6 expected clusters (Figure 37).

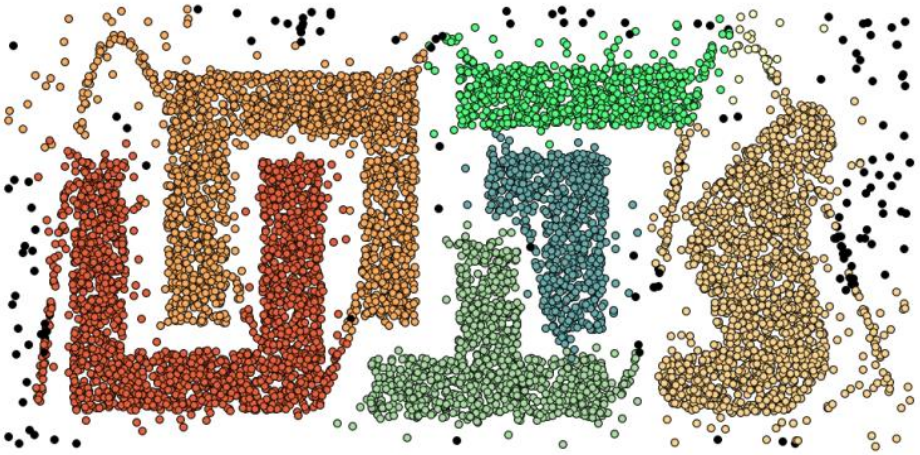

Figure 37 - Result of Spatio-temporal Clustering Using 75%-25% Weights (SNN Parameters, *k* = 40, *Eps* = 16, *MinPts* = 24).

The obtained results so far produced appropriate results when clustering datasets with spatial and temporal dimension. The modification of the synthetic datasets, including the time dimension, allowed the verification of the sensibility of the proposed approach to space and time, when those dimensions are analysed in an integrated way. Next, the results obtained when clustering the Fires 2011 dataset (a real dataset) will be presented.

4.2.3 - Fires 2011 Dataset

The first real dataset used in this work is the Fires 2011 dataset mentioned previously in Chapter 3. This dataset[4] was created by "Autoridade Nacional de Protecção Civil" and the data contained in it was treated and published by "Instituto da Conservação de Natureza e das Florestas". Figure 38 presents the spatial distribution of the 35941 records.

Figure 38 - Spatial Distribution of the Fires 2011 Dataset.

The dataset covers practically all the continental part of the country (it does not have records of fires in the Portuguese islands) presenting few areas with no records in centre and south zones. The north and the coast centre zone of the country are the zones with most fires records. The interior of the country has fewer incidents than the coastline.

The fires dataset was clustered considering events with the spatial and temporal dimension. The results shown in Figure 39 were obtained assigning the same weight to space and time and using a *MaxS* of 15.71958 km and a *MaxT* of 11 minutes, allowing the identification of clusters that combine the same spatial region with different periods of time or different regions in the same time. For a simpler understanding, the 3D graphics shown in this chapter will have numbers, with the same colour of the drawn cluster, that indicate the number of the cluster. The ticks at the temporal scale in the 3D graphics indicate the beginning of that season. All 3D graphics will not show noise points for clarity reasons.

[4] Available at http://www.icnf.pt/portal/florestas/dfci/inc/estatisticas/estatistica-sgif

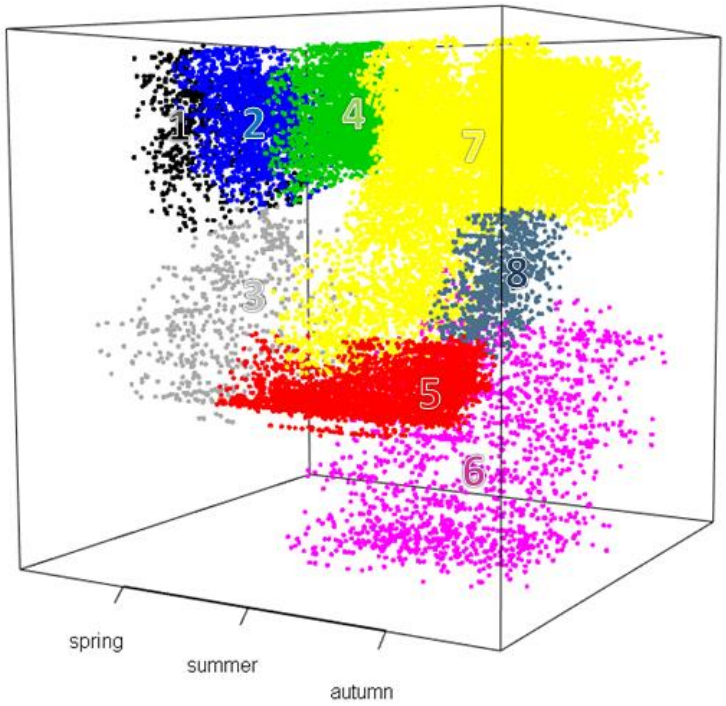

Figure 39 - Clustering of the Fires 2011 Dataset Using 50%-50% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200).

Eight clusters were detected. Six of them in the northern and centre part of the country (1, 2, 3, 4, 5 and 8), one in the south (6) and one big cluster (the yellow one, 7) with fires that occurred in the summer in the northern and centre part of the country. With the presented 3D visualization it is possible to verify when fires occur and in what regions they are more frequent. The North and South parts of the country present different seasoning behaviour, with the North part having fires almost all year long. To achieve a better understanding of the results, Figure 40 presents the resulting clusters along time (January to December). With this perspective it is possible to notice the temporal transitions throughout the year and the separation between north, centre and south of Portugal.

Looking at the obtained results, the eight clusters confirm the advantages of the SNN algorithm when used in spatio-temporal 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 the country.

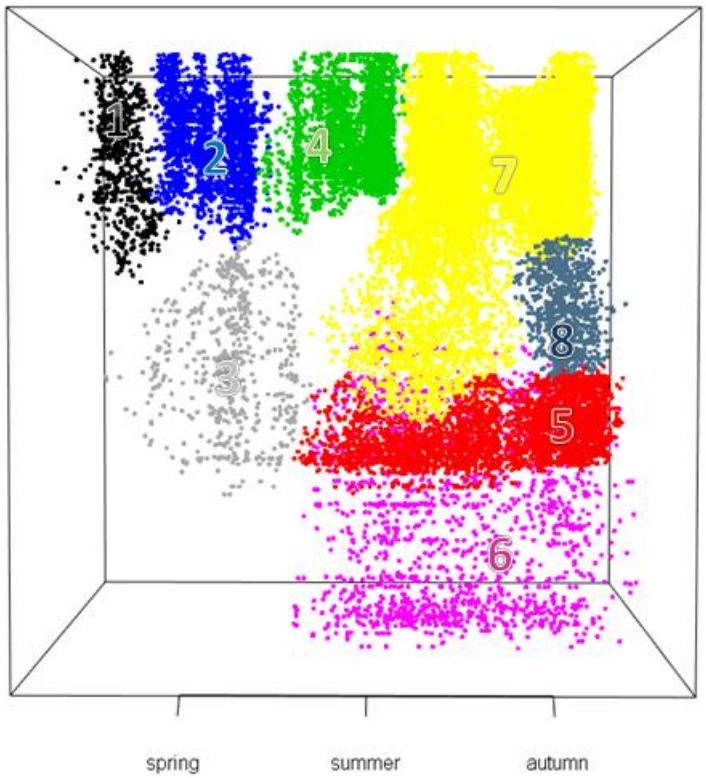

Figure 40 - Clustering of the Fires 2011 Dataset Using 50%-50% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200) (Temporal Perspective).

4.3 - Spatio-temporal and Semantic Attribute Clustering

In this subchapter, the clustering results with the Fires 2011 dataset using a semantic attribute will be presented. Furthermore, the Fires 2012 and Fires 2011-2012 datasets will be introduced as well as their clustering results. In these datasets, the spatial, temporal and semantic attribute dimensions will be used.

4.3.1 - Fires 2011 Dataset

After the clustering of datasets with a spatial and a temporal dimensions, the semantic attribute (the burnt area) presented in the previous chapter was added to increase the analytic capability of the user. In this process, it was used the same normalization parameters *MaxS* and *MaxT* (respectively 15.71958 km and 11 minutes) and a *MaxA* of 0.0001 ha. With these parameters and with the same weight for every dimension, 14 clusters were identified (Figure 41

and Figure 42). Each cluster is represented by a different colour and by the number of the cluster with the same colour as the cluster.

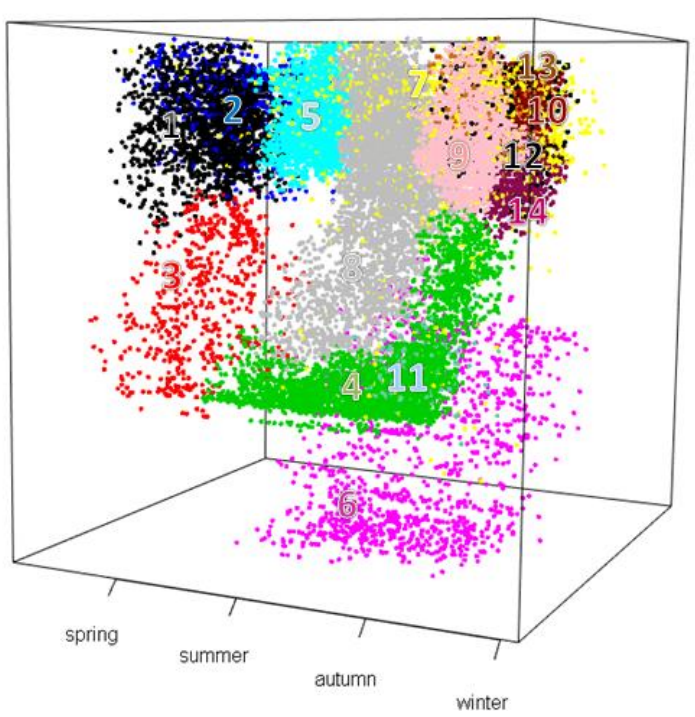

Figure 41 - Clustering of the Fires 2011 Dataset Using 33%-34%-33% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200).

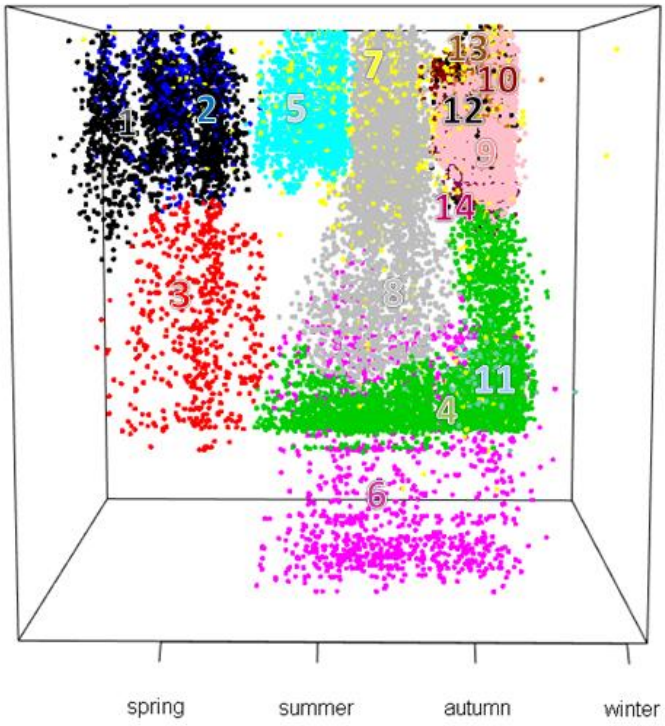

Figure 42 - Clustering of the Fires 2011 Dataset Using 33%-34%-33% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200) (Temporal Perspective).

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. Cluster 3 and 4 are located in the same region (around the centre 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 39) are now separated in different segments attending to the burnt area.

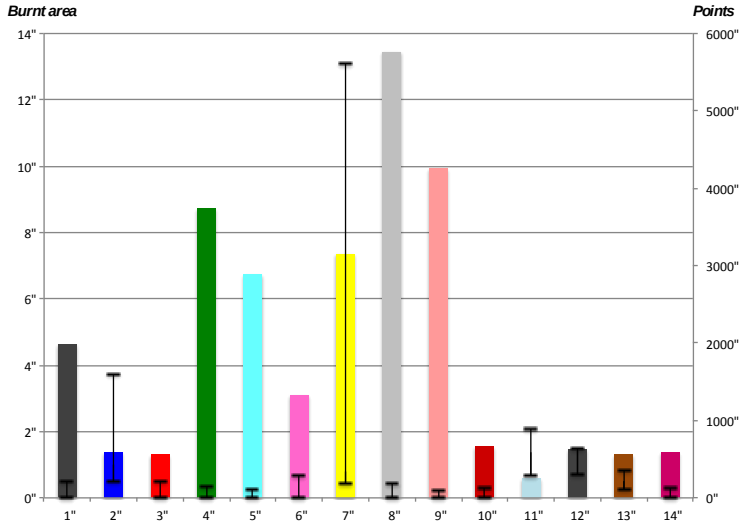

Figure 43 - Number of Points, Maximum and Minimum per Cluster (Same Weights for the 3 Dimensions).

Figure 43 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 coloured bars) shows how many points each cluster has. The colours in the bars are the same as in 3D graphics. Table 5 presents some statistics associated to each cluster, like the minimum, maximum, average and standard deviation of the burnt area. The results obtained 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 41, 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 separate one. In the 3D graphics with the clustering results with the semantic attribute, the tick that represents the beginning of the winter appears because in this result, contrarily to the result with only spatial and temporal dimensions, there is a cluster (7) with some fires that occurred in winter. These fires appear in this result because the semantic attribute in those fires is high and makes them join the cluster with high burnt area fires that occurred during summer and autumn (cluster 7).

Table 5 - Number of Points and Statistics of the Burnt Area per Cluster in Fires 2011 Dataset (33%-34%-33% Weights).

Cluster	Number of Points	Minimum	Maximum	Average	Standard Deviation
1	1989	0.00	0.50	0.05	0.08
2	581	0.50	3.74	1.58	0.75
3	555	0.00	0.52	0.03	0.07
4	3736	0.00	0.35	0.02	0.04
5	2887	0.00	0.25	0.02	0.04
6	1329	0.00	0.65	0.05	0.11
7	3150	0.40	13.10	3.33	2.76
8	5761	0.00	0.41	0.03	0.04
9	4252	0.00	0.20	0.02	0.03
10	666	0.00	0.30	0.02	0.04
11	249	0.68	2.07	1.26	0.36
12	623	0.70	1.51	1.01	0.10
13	559	0.25	0.82	0.48	0.11
14	587	0.00	0.30	0.03	0.05

Using the same semantic attribute, 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 44 presents the 21 clusters obtained. Although challenging due to the 21 colours used to plot each cluster, it is possible to verify that previous huge clusters (Figure 41) 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 45 presents the temporal perspective where the temporal separation of the formed clusters can be seen, as well as the separation between the north/centre and the south zones of Portugal.

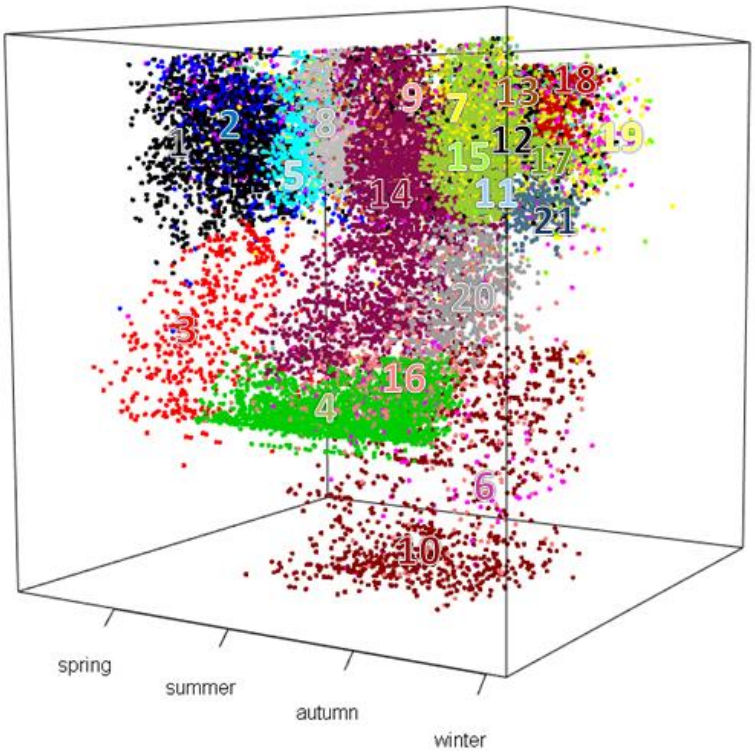

Figure 44 - Clustering of the Fires 2011 Dataset Using 20%-20%-60% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200).

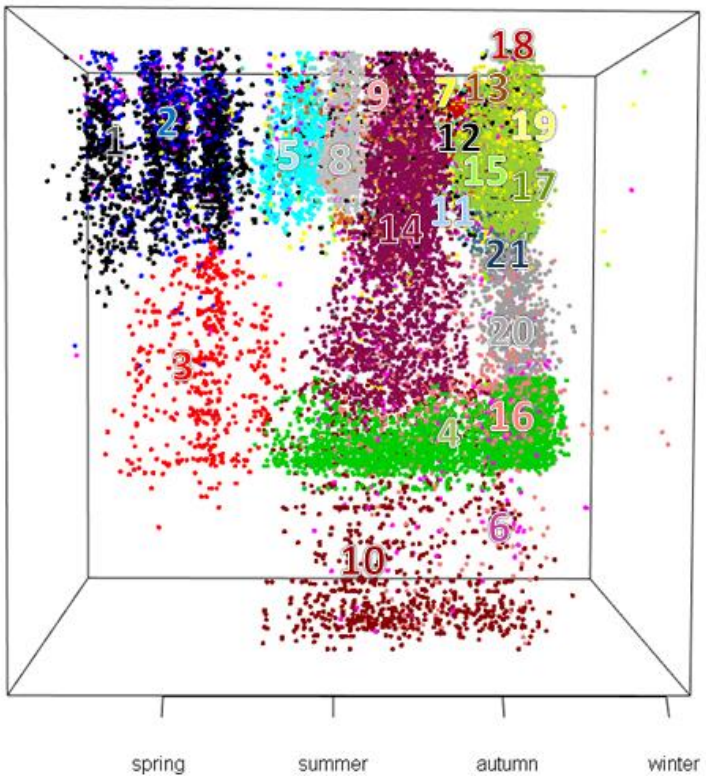

Figure 45 - Clustering of the Fires 2011 Dataset Using 20%-20%-60% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200) (Temporal Perspective).

Figure 46 presents a graphic (similar to Figure 43) with the number of incidents for each cluster and the maximum and minimum value of the burnt area for each cluster. Table 6 shows the statistics associated with each cluster, pointing the number of points and the minimum, maximum, average and standard deviation of the burnt area. 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 46.

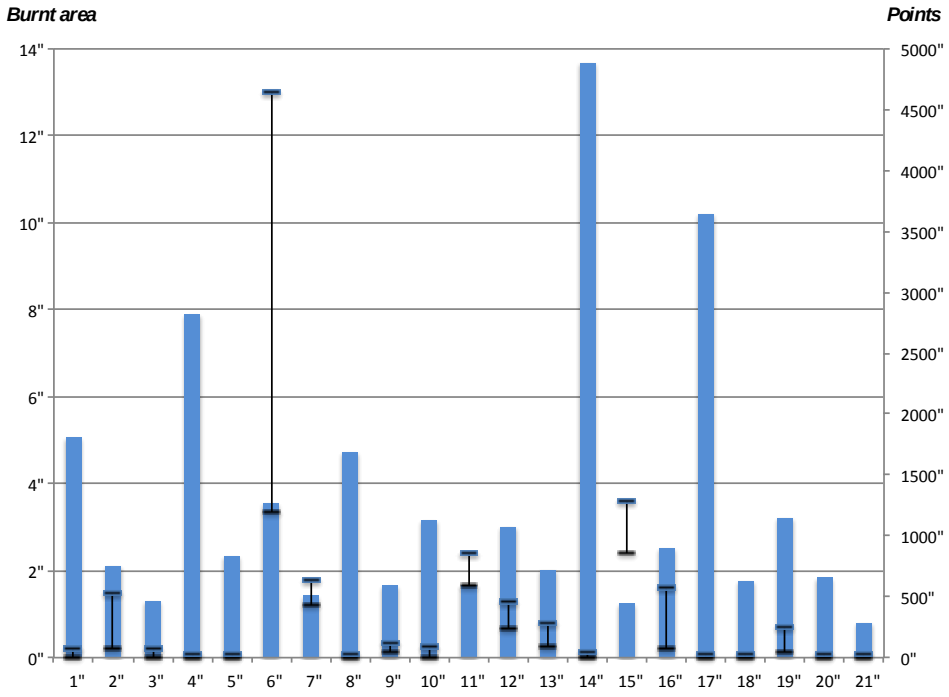

Figure 46 - Number of Points, Maximum and Minimum per Cluster Using 20%-20%-60% Weights.

These results are confirmed by the Annual Report of Burnt Areas and Occurrences 2011 prepared by the Portuguese Forest Defence Unity Direction that states that the districts with greater number of occurrences in 2011 were the ones located in the north and in the coastline between the northern and centre part of the country: Porto, Braga, Aveiro, Viseu, Viana do Castelo and Vila Real (Direção de Unidade de Defesa da Floresta, 2012). In the results achieved, these zones have a very concentrated umber of clusters whereas the south zone of Portugal has only 2 clusters. Another aspect stated in this report is that, although with low number of incidents, the southern zone of Portugal was the zone with the largest burnt area of the country. This is reflected in cluster 6 as it has the higher average burnt area of all clusters.

Table 6 - Number of Points and Statistics of the Burnt Area per Cluster in Fires 2011 Dataset (20%-20%-60% Weights).

Cluster	Number of Points	Minimum	Maximum	Average	Standard Deviation
1	1811	0.00	0.20	0.03	0.04
2	755	0.20	1.50	0.66	0.30
3	467	0.00	0.20	0.02	0.03
4	2822	0.00	0.10	0.01	0.02
5	834	0.00	0.10	0.01	0.02
6	1269	3.37	13.00	6.29	2.39
7	510	1.20	1.80	1.48	0.11
8	1689	0.00	0.10	0.01	0.02
9	595	0.11	0.33	0.21	0.05
10	1128	0.00	0.25	0.02	0.04
11	596	1.65	2.40	2.00	0.08
12	1073	0.67	1.30	0.98	0.08
13	719	0.25	0.78	0.49	0.09
14	4884	0.00	0.14	0.01	0.02
15	446	2.40	3.58	2.96	0.23
16	899	0.20	1.60	0.74	0.34
17	3644	0.00	0.08	0.01	0.02
18	629	0.00	0.10	0.01	0.01
19	1146	0.12	0.70	0.37	0.15
20	658	0.00	0.10	0.01	0.02
21	284	0.00	0.08	0.01	0.01

4.3.2 - Fires 2012 Dataset

After the test with the Fires 2011 dataset, the same test was done but with the fires data from 2012 (named Fires 2012 dataset from now on) in order to compare the evolution in consecutive years. This dataset was similar to the previous one (same attributes) and had less records (30740 instead of 35941 from the 2011 Fires dataset). Figure 47 show the results (15 clusters) with this dataset using the same SNN parameters used with the Fires 2011 dataset and the same weights for every dimension. The normalization parameters used were 23.73865 km as *MaxS*, 14 minutes as *MaxT* and 0.0001 ha as *MaxA*.

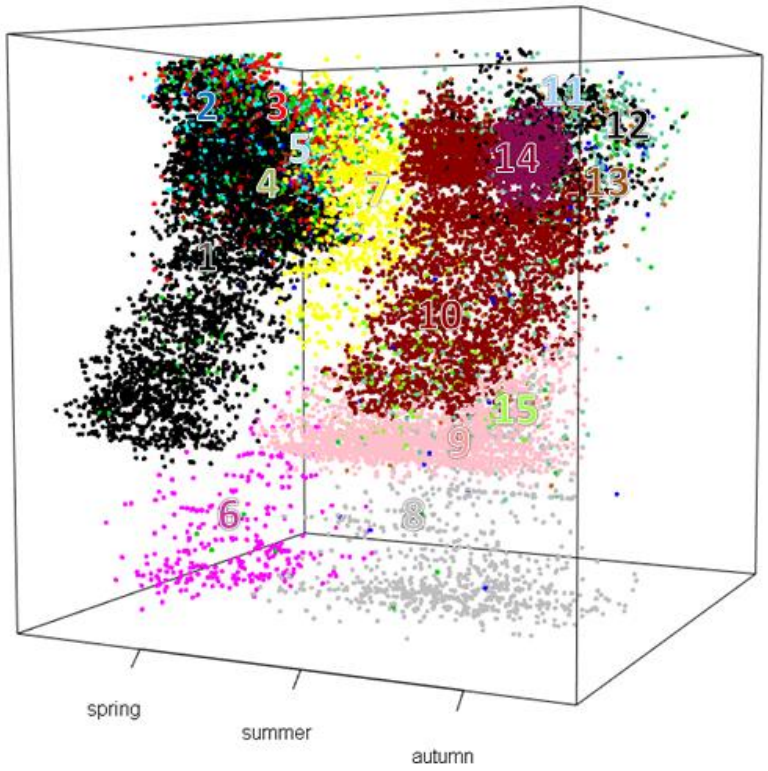

Figure 47 - Clustering of the Fires 2012 Dataset Using 33%-34%-33% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200).

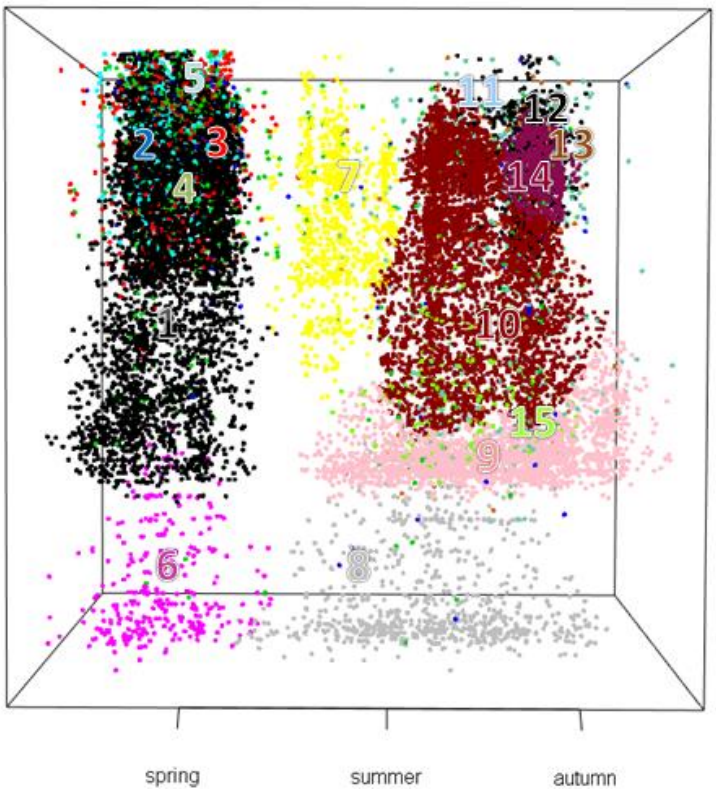

Figure 48 - Clustering of the Fires 2012 Dataset Using 33%-34%-33% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200) (Temporal Perspective).

Comparing with the previous year, it has more clusters in the first quarter of the year (winter and the beginning of spring), four clusters in the northern and center part of the country (with different averages of burnt areas values) and one in the south (which did not exist in this period of time in the previous year). The rest of the clusters are similar to the ones achieved using the 2011 fires dataset.

In these results, the clusters formed in the initial part of the year were a little strange because it is not a common thing to happen but according to the Annual Report of Burnt Areas and Occurrences 2012, this was the year with more fires in the first 13 weeks of the year since 2001. As the previous year, the districts with the highest number of occurrences were in the northern part of the country: Braga, Porto and Viseu (Instituto da Conservação da Natureza e das Florestas, 2013).

Table 7 - Number of Points and Statistics of the Burnt Area per Cluster in Fires 2012 Dataset (33%-34%-33% Weights).

Cluster	Number of Points	Minimum	Maximum	Average	Standard Deviation
1	6281	0.00	0.30	0.03	0.04
2	286	13.50	29.05	19.95	3.94
3	1275	0.70	2.26	1.39	0.44
4	1243	2.19	14.00	5.69	2.88
5	826	0.24	0.87	0.48	0.12
6	327	0.00	0.70	0.08	0.14
7	838	0.00	0.34	0.03	0.05
8	946	0.00	0.50	0.03	0.07
9	2222	0.00	0.25	0.01	0.03
10	4824	0.00	0.25	0.02	0.04
11	595	1.33	4.00	2.37	0.64
12	897	0.26	1.60	0.78	0.32
13	204	3.64	6.50	4.74	0.70
14	1713	0.00	0.15	0.02	0.03
15	203	0.60	1.75	1.02	0.20

4.3.3 - Fires 2011-2012 Dataset

Another test combined the two fires dataset (2011 and 2012) in order to understand if the implementation could identify the same clusters that were identified using separated datasets. It was used the same SNN parameters used in the previous tests (k = 230, *Eps* = 45 and *MinPts* = 200) and 9.875337 km as *MaxS*, 13 minutes as *MaxT* and 0.0001 ha as *MaxA*. The result can be observed in Figure 49 and Figure 50.

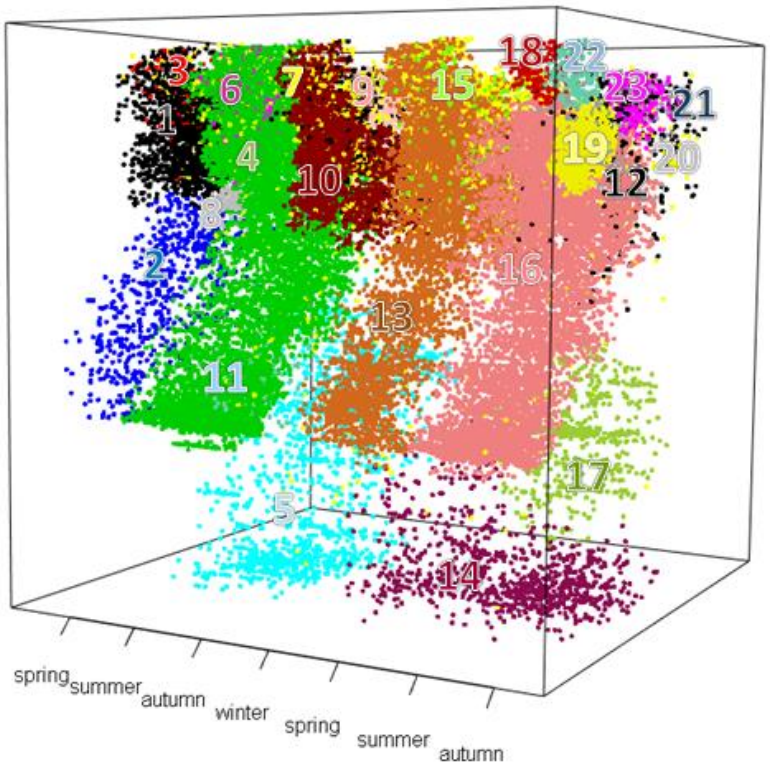

Figure 49 - Clustering of the Fires 2011-2012 Dataset Using 33%-34%-33% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200).

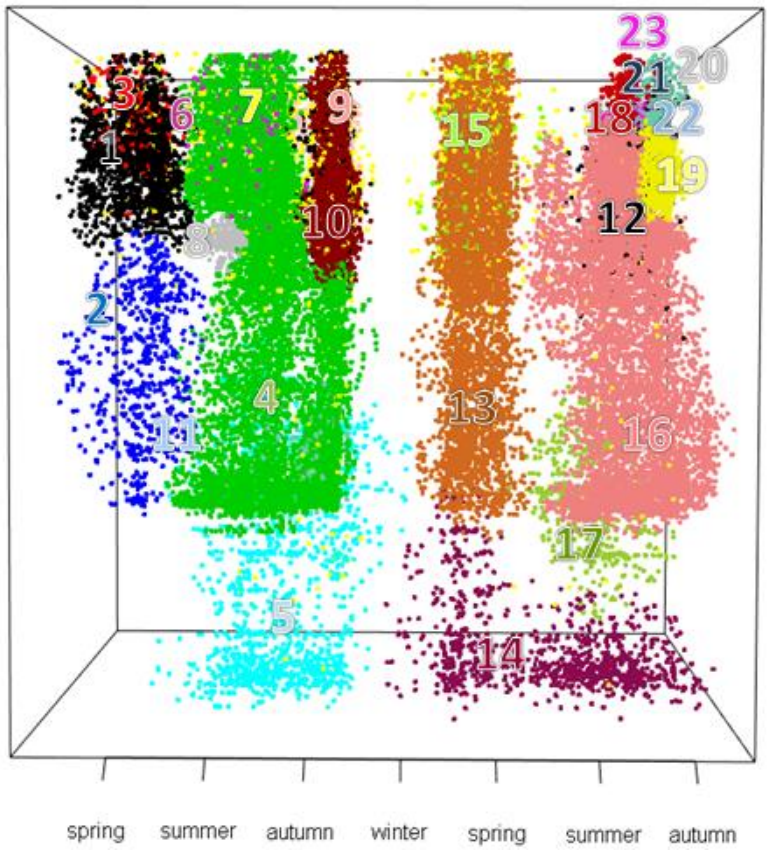

Figure 50 - Clustering of the Fires 2011-2012 Dataset Using 33%-34%-33% Weights (SNN Parameters, *k* = 230, *Eps* = 45, *MinPts* = 200).

Although some clusters are somewhat different (merging of some clusters that were separated in the separated clustering, especially in the south of the country in 2012) the final result is very similar, which shows that this approach can give coherent results using different datasets. For a better understanding of the results, Table 8 presents some information about the formed clusters.

Table 8 - Number of Points and Statistics of the Burnt Area per Cluster in Fires 2011-2012 Dataset (33%-34%-33% Weights).

Cluster	Number of Points	Minimum	Maximum	Average	Standard Deviation
1	1905	0.00	0.52	0.06	0.10
2	639	0.00	0.65	0.04	0.10
3	308	0.50	2.30	1.26	0.43
4	12911	0.00	0.50	0.03	0.05
5	1503	0.00	0.87	0.06	0.14
6	556	0.48	1.40	0.90	0.17
7	4734	1.20	45.00	7.78	8.61
8	201	0.00	0.24	0.02	0.04
9	680	0.00	0.40	0.03	0.07
10	5050	0.00	0.50	0.03	0.06
11	244	0.68	2.07	1.25	0.35
12	840	0.50	1.65	1.00	0.22
13	7345	0.00	0.50	0.04	0.07
14	943	0.00	1.00	0.07	0.17
15	956	0.40	1.70	0.97	0.26
16	8125	0.00	0.50	0.03	0.06
17	632	0.00	0.78	0.05	0.12
18	183	0.00	0.30	0.04	0.05
19	1821	0.00	0.23	0.03	0.04
20	389	0.56	1.41	0.95	0.15
21	460	1.26	5.00	2.62	0.85
22	234	0.00	0.30	0.03	0.04
23	286	0.00	0.20	0.02	0.04

Although some clusters identified in this result are similar to the clusters identified in the separate clustering, this result should be a little better since the number of clusters identified in the Fires 2011-2012 dataset (23) was different from the sum (31) of the clusters identified in the Fires 2011 (15) and Fires 2012 (16) datasets. This might have happened because of the SNN input parameters *(k*, *Eps* and *MinPts)*. As the Fires 2011-2012 dataset has almost twice the records of the separate datasets the SNN input parameters should be different in order to find a result that identified the same clusters as the separate clustering. Some tests were made using other SNN input parameters but the clustering results were not better than the result presented

above so, it was not possible to understand which were the optimal SNN input parameters for this dataset.

4.4 - Spatio-temporal and Two Semantic Attributes Clustering

The final test was done with a dataset about meteorological information which has 2555 records. These records were registered every day at 2 AM for a year (2007) in 7 different meteorological stations (that support the Portuguese agricultural notification network) in the northern zone of Portugal.

This dataset was provided by "Direção Regional de Agricultura e Pescas do Norte". Each record contains various meteorological information such as temperature, humidity, precipitation, etc. Using the main types of spatio-temporal data presented in Chapter 2, this can be classified as a dataset of geo-referenced time series since it is possible to see the attribute (temperature for the first test, temperature and humidity for the second test) variation of an object across time. Figure 51 presents the location of the meteorological stations in the map of Portugal. The blue square in the map at right is the zone of the map zoomed in on the map on the left side, and the orange marks are the meteorological stations location.

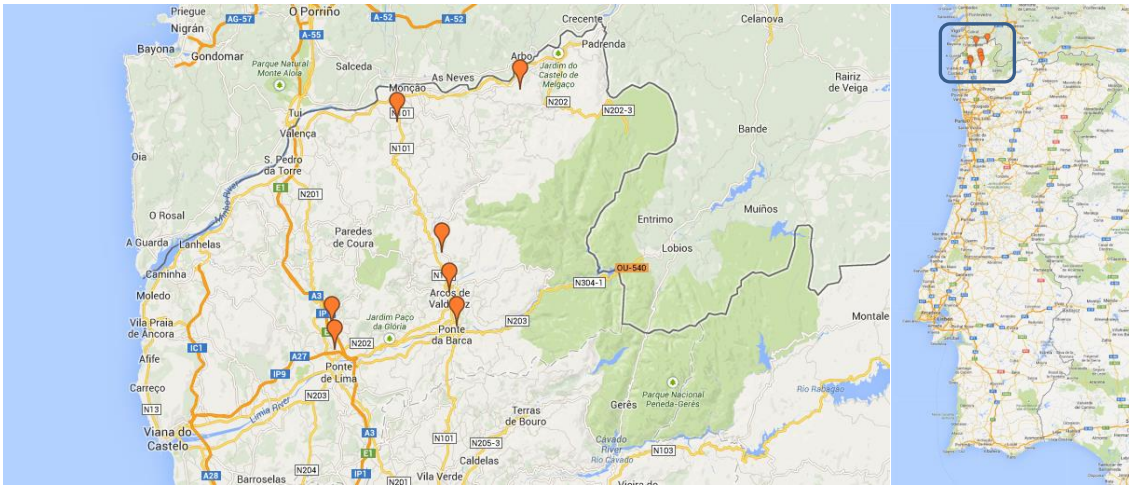

Figure 51 - Map of Portugal with the Meteorological Stations.

First, the spatial and temporal dimensions were used as well as one semantic attribute of the dataset (temperature). From now on, this dataset will be named Meteo1. The result of the clustering process (without noise points) and giving the same weight to every dimension can be seen in Figure 52. In this test, SNN parameters k = 50, *Eps* = 18 and *MinPts* = 45 and the normalization parameters *MaxS* = 7.146031 km, *MaxT* = 1 day and *MaxA* = 0.1° C were used.

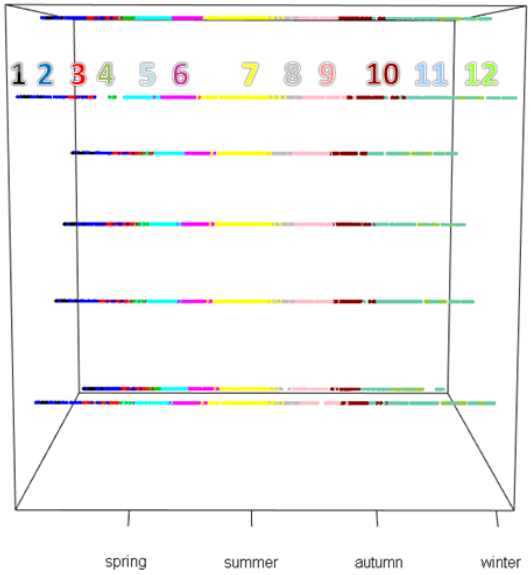

Figure 52 - Clustering of the Meteo1 Dataset Using 33%-34%-33% Weights (SNN Parameters, *k* = 50, *Eps* = 18, *MinPts* = 45).

Twelve clusters were formed. Clusters 1, 2, 3 and 4 are in the first season of the year (winter). The difference between them is the temperature, lower for clusters 2 and 4 and higher for clusters 1 and 3. In the spring it has two clusters (5 and 6) with different average temperatures. Clusters 7, 8 and 9 are in summer. Cluster 7 and 9 have the higher average temperatures in this result but are separated because cluster 8 has lower temperatures in the middle of them and thus, separating them temporally. Finally, in autumn, three clusters (10, 11 and 12) were created with different average temperatures. Table 9 presents some information about the created clusters.

Table 9 - Number of Points and Statistics of the Burnt Area per Cluster in Meteo1 Dataset (33%-34%-33% Weights).

Cluster	Number of Points	Minimum	Maximum	Average	Standard Deviation
1	67	8.70	14.90	11.38	1.57
2	218	-0.30	8.90	3.96	2.03
3	123	10.10	15.40	12.88	1.23
4	48	6.20	9.60	7.50	0.95
5	211	2.60	11.10	6.83	2.31
6	179	8.00	14.80	11.38	1.55
7	362	9.50	18.40	14.15	2.05
8	97	10.10	15.50	13.00	1.31
9	211	12.60	19.60	16.16	1.57
10	178	11.00	16.30	13.47	1.15
11	404	-4.60	11.60	4.56	4.01
12	94	6.60	11.80	9.32	1.01

The results show that the four seasons of the year are well defined and the only difference is the number of clusters created in each season.

After this test, another attribute was added to the experiment, the humidity read by the sensors. The objective was to understand if the 4D^{+}SNN approach was indeed able to cluster more than 4 dimensions and justify the + symbol in its name. Using Equation 6 and with the same weight to every dimension (25%) the result is presented in Figure 53 (without noise points). This test used the same SNN input parameters and the same normalization parameters as the ones used in the previous test since the dataset is the same. Only the second semantic attribute with a normalization parameter *MaxA2* = 1 (%) for the humidity attribute was added. This dataset will be named Meteo2 from now on.

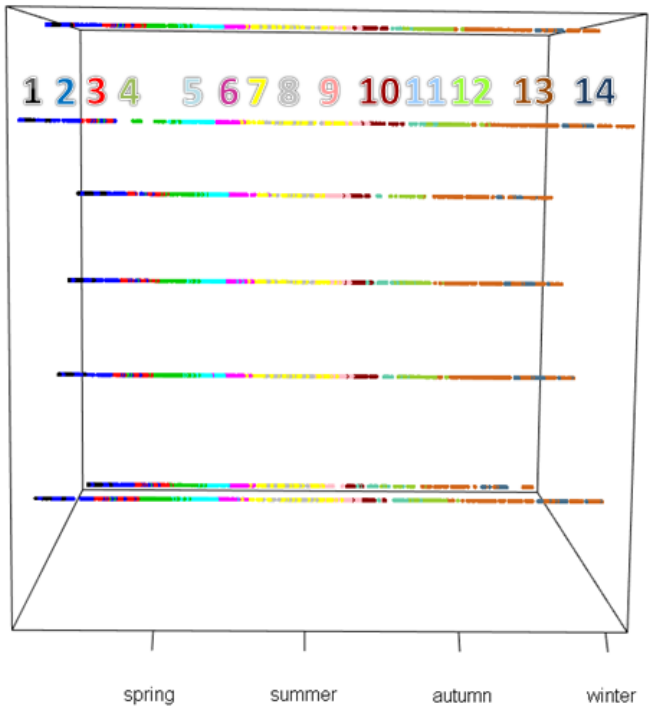

Figure 53 - Clustering of the Meteo2 Dataset Using 25% Weight for the 4 Dimensions (SNN Parameters, *k* = 50, *Eps* = 18, *MinPts* = 45).

As can be seen, the main difference to the previous test is at summer and autumn. Winter and spring had similar clusters because they were already separated by low and high temperatures and the humidity follows that pattern (higher humidity in cold days and lower humidity in hot days). In the summer and autumn the temperatures were not so different in consecutive days but with higher humidity (probably some rai in those days) it was sufficient to create new clusters that, although having similar average temperatures (differences of 2 or 3 degrees Celsius) present different percentages of humidity. These results are confirmed by the

report "The Climate in Continental Portugal in 2007" that states that this year had the most rainy summer of the last two decades (Instituto de Meteorologia, 2008). Table 10 presents more information about the clusters formed. This table is similar to the ones presented before in this work but with the addition of a second attribute (C means cluster, NP number of points, Min is the minimum, Max is the maximum, Avg is the average, SD is the standard deviation and A1 and A2 are, respectively, the attributes temperature and humidity).

Table 10 - Number of Points and Statistics of the Burnt Area per Cluster in Meteo2 Dataset (25%-25%-25%-25% Weights).

C	NP	Min A1	Max A1	Avg A1	SD A1	Min A2	Max A2	Avg A2	SD A2
1	71	8.70	14.90	11.49	1.69	68	100	94.54	7.47
2	208	-1.90	7.90	3.89	2.13	78	100	98.51	3.41
3	113	10.10	15.40	12.93	1.22	63	100	89.92	7.59
4	165	2.60	9.60	6.00	1.71	70	100	92.76	6.20
5	155	6.30	14.70	10.68	2.11	70	100	91.88	6.62
6	107	7.10	12.50	10.27	1.21	75	100	94.26	5.98
7	209	13.10	18.70	15.41	1.14	71	100	91.12	6.44
8	109	9.30	14.10	11.57	1.17	79	100	95.55	4.00
9	96	10.10	15.30	12.86	1.20	81	100	95.96	3.33
10	91	14.10	19.50	16.66	1.41	74	99	91.86	5.25
11	60	13.60	17.70	15.92	0.99	61	100	80.30	8.02
12	151	11.00	16.10	13.68	1.19	78	100	93.47	4.67
13	405	-5.30	12.70	4.90	4.26	80	100	97.22	3.23
14	80	5.90	12.10	9.08	1.24	83	100	97.83	4.04

This test shows that this implementation can handle with success more than one semantic continuous attribute (non-spatial and non-temporal). The problem about using more semantic attributes is that it is more difficult to understand the results given by the approach. That is a decision that the user must take, separate the attributes and do the clustering or cluster all the attributes at the same time.

4.5 - Discussion

After the results presentation, it is important to discuss some aspects of this work. The proposed approach had interesting and promising results, with only three dimensions (spatial and temporal) or four or more dimensions, because both events (without semantic attribute) and geo-referenced variables (with semantic attribute) were effectively clustered identifying relevant patterns.

The weighting factors used in Equation 4 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. On the other hand, 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, t5.8k.b and t4.8k. The weighting factors in Equation 5 and Equation 6, that include a semantic attribute, also affect the clustering results. In the experiments with Fires 2011 dataset, with a weight w_a = 33% the 14 clusters present an average range for the burnt area of 1.59 ha and with w_a = 60% the 21 clusters present an average range for the burnt area of 0.87 ha. The expected impact on the number of clusters and their densities (spatial, temporal and semantic attribute) depend on the used weights and on the dataset itself.

This approach has some advantages in relation to the other ones studied in Chapter 2. Specifically, ST-DBSCAN (Birant & Kut, 2007) and STSNN (Liu et al., 2012) were the two main approaches in this field of study. 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 (ST-DBSCAN adds two more input parameters and STSNN needs the user to know what is the time window for an object to be considered in the neighbourhood 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 these 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 with that dimension.

The main advantage of this approach is the ability to cluster several dimensions (spatial, temporal and various semantic attributes) at the same time which this work's investigation did not encounter any other approach that could do that during this work's search phase.

5 - CONCLUSION

In this work, the 4D^{+}SNN approach was presented. This is an approach to cluster datasets with four or more dimensions (spatial, temporal and semantic attributes). In an extensive literature review, the main types of spatio-temporal data and their characteristics, as well as the impact of these characteristics to the clustering process, were reported. After that, several approaches of various authors were presented in order to understand what was already done and what was missing in the current literature.

After delineating the objectives and expected results of this work, several approaches were created to try to answer the research question: "How can we integrate the space and time dimensions in the clustering of spatio-temporal data using the SNN algorithm?". Since this was not a trivial problem to solve, several approaches were followed. For this reason, the full process including all the approaches proposed, their results and their problems were reported in this document.

With an approach that fits the purpose of this work, several tests were done with various datasets. Synthetic datasets were used in order to understand if the space and time dimensions were properly clustered. From the presented results, it can be seen that this approach can effectively cluster spatio-temporal datasets.

After that, real datasets were used. First, it was confirmed that this approach can cluster datasets with both spatial and temporal dimensions as it was seen in Fires 2011 dataset. Then, the same dataset as well as other real datasets were used to confirm if the 4D^{+}SNN could handle the addition of a new dimension with a semantic attribute. These tests were also very interesting as significant patterns were discovered when clustering these datasets.

One final test was done introducing more than one semantic attribute and thus confirming the potential of this approach when clustering more than 4 dimensions.

The proposed approach, the 4D^{+}SNN, had interesting and promising results, with spatio-temporal data with or without semantic attributes, since both types of data were effectively clustered identifying relevant patterns. This approach has some advantages in relation to the

current approaches presented in the literature that cluster spatio-temporal data since the 4D^{+}SNN can cluster spatial and temporal dimensions in an integrated way 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 input parameters that this approach needs are the weighting factors that the user can tune in order to improve the results. This is possible because this approach can find the normalization parameters needed to make the dimensions equivalent, i.e., stop using measures and scales.

5.1 - Objectives and Expected Results

The research question of this work "How can we integrate the space and time dimensions in the clustering of spatio-temporal data using the SNN algorithm?" was answered. From the analysis of the clustering results achieved with the 4D^{+}SNN, this approach seems to have the ability to integrate both space and time dimensions and even one or more semantic attributes in the clustering of spatio-temporal data.

The objectives delineated for this work were all accomplished. In the literature review, the main types of spatio-temporal data and their characteristic were identified as well as the current approaches to cluster spatio-temporal data and their advantages and disadvantages. Then, several approaches were delineated in order to cluster spatio-temporal data. After that, it was implemented a prototype that has the ability to cluster spatio-temporal data with or without semantic attributes and, lastly, some tests were conducted to verify the quality of the obtained clusters.

There was one of the expected results that was not completed in time, the sensibility analysis of the influence of each input parameter of the SNN algorithm with the proposed approach.

5.2 - Limitations

This work, as any scientific work, has some limitations. First, the proposed heuristic uses an 80% decile value for the normalization parameters. This value was chosen because the analysis of several synthetic and real datasets allowed the identification of the distance value

given by the 80% decile as an appropriate value for the normalization parameters. The distance present in this decile splits the distances that are usually associated to neighbours values and those that start to be influenced by noise points but it is possible that there are other datasets that have a point's distribution that does not follow this pattern. If this happens, it could have a negative impact in the calculation of the normalization parameters, i.e., give a *MaxS*, *MaxT* and/or *MaxA* that could influence the clustering process.

This heuristic has another problem because of the usage of the bounding box. This box is needed in order to transform the spatial (2 dimensions) component of the records in a 1 dimension. Some experiences were performed using, instead of just one point (the lowest to the left), the four points of the bounding box and calculate the average distance between each point of the dataset to the four points of the bounding box. Other experiment was to add the middle point of the bounding box and calculate the average distance between the five points and each point of the dataset. The final experiment was to use the three "best" points of these five points to calculate the average distance to each point of the dataset. For each of these points, the three closer distances out of five were calculated in order to understand, which the best points were. Then, the average distance between the three points was calculated. These experiments were done because when calculating the *MaxS* parameter, the approach only uses the lowest to the left point of the bounding box and that could add some error to the calculations because points of the dataset that were at the same distance to the bounding box corner were considered the same point as they will have a difference of 0 when calculating the difference between consecutive positions of the distance list. To a better understanding of this problem, Figure 54 presents an example of this situation.

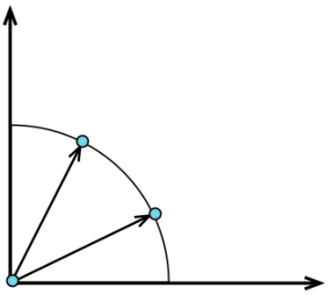

Figure 54 - Example of Bounding Box Problem.

As can be seen in the example, the result of the distance between the bounding box and the two points will be the same and in the next step of the heuristic, when calculating the distances between consecutive points the result will be 0 when, in reality, they are not in the same position.

Using more bounding box points than just one, it was expected to solve this problem but using three, four or five bounding box points the detection of outliers is much more difficult because the points are much closer than with just one point. So it was decided to abandon this strategy and continue the calculation of *MaxS* using just one point of the bounding box because the results were not affected by this constraint.

5.3 - Future Work

Although the results achieved with both synthetic and real datasets are very promising as spatio-temporal objects were effectively clustered identifying relevant patterns there are some aspects that can still be improved.

As future work, it would be very interesting to introduce a discrete attribute in the clustering process, opening new possibilities in the analysis of non-spatial attributes. As this new attribute has new characteristics that will influence the clustering process, some study about the impact of this new dimension would be necessary.

An investigation to improve the identification of the normalization factors would also be interesting because it is dependent on the 80% decile. An approach that uses the properties of the dataset to discover the normalization parameters would be a great addition to this approach. The current implementation could be complemented with a graphical user interface, for choosing the distance function to use and the attribution of weights, which would allow any user to take advantage of it.

Another study that could be made to this work is to perceive if the calculation of the normalization parameters can be done with a sample of the dataset, instead of using the whole dataset as it is now. If it is possible to use only a sample of the dataset and if that does not have an impact in the result of the normalization parameters, it would be interesting to use the sample as the user would gain in processing time.

In this work, there was one expected result not completed, a sensibility analysis on the influence of each input parameter of the SNN algorithm in the proposed approach. It would be useful to do a battery of tests to understand how each input parameter impacts the clustering process with spatio-temporal data. With these tests, it would be simpler in the future to

understand what the SNN input parameters to a determined dataset are without having to do a series of trial-and-error tests.

Finally, it would be interesting to do a direct comparison between this approach and the other approaches studied in the literature review. Although they only consider the spatial and temporal dimension, it would be interesting to see how the different approaches behave.

REFERENCES

Andrienko, G., Andrienko, N., Bak, P., Keim, D., Kisilevich, S., & Wrobel, S. (2011). A conceptual framework and taxonomy of techniques for analyzing movement. *Journal of Visual Languages & Computing*, *22*(3), 213–232. doi:10.1016/j.jvlc.2011.02.003

Ankerst, M., Breunig, M., Kriegel, H., & Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. *ACM SIGMOD International Conference on Management of Data* (pp. 49–60).

Antunes, A. (2012). *Análise Espacial de Grandes Quantidades de Dados de Movimento Usando Técnicas de Clustering Baseadas em Densidade* (Master Thesis). Universidade do Minho.

Auria, M. D., Nanni, M., & Pedreschi, D. (2006). Time-focused density-based clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, *27*(3), 267 – 289.

Berry, M., & Linoff, G. (2000). *Mastering Data Mining: The Art and Science of Customer Relationship Management*. John Wiley & Sons.

Berson, A., & Smith, S. (1997). *Data Warehousing, Data Mining & OLAP*. McGraw-Hill.

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, *60*(1), 208–221. doi:10.1016/j.datak.2006.01.013

Bivand, R. S., Pebesma, E. J., & Gómez-Rubio, V. (2008). *Applied Spatial Data Analysis with R*. Springer.

Bouguessa, M. (2011). A Practical Approach for Clustering Transaction Data. *Proceeding of the 7th International Conference on Machine Learning and Data Mining*. New York: Springer-Verlag.

Carvalho, M., & Natário, I. (2008). *Análise de Dados Espaciais*. Sociedade Portuguesa de Estatística.

Direção de Unidade de Defesa da Floresta. (2012). *Relatório Anual de Áreas Ardidas e Ocorrências 2011*. Retrieved 20, September 2013 from http://www.icnf.pt/portal/florestas/dfci/relat/rel-if/2011/relatorio-final-2011

Dodge, S., Weibel, R., & Lautenschütz, A.-K. (2008). Towards a taxonomy of movement patterns. *Information visualization*, *7*(3), 240–252. doi:10.1057/palgrave.ivs.9500182

Ertoz, L., Steinbach, M., & Kumar, V. (2002). Finding Clusters of Different Sizes, Shapes, and Densities in Noisy ,High Dimensional Data. *2nd SIAM International Conference on Data Mining*. San Francisco, EUA.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). 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).

Faustino, B. (2012). *Implementation for Spatial Data of the Shared Nearest Neighbour with Metric Data Structures* (Master Thesis). Universidade Nova de Lisboa.

Gonçalves, F. (2012). *Um Sistema de Informação Espaço-Temporal para Objectos Móveis* (Master Thesis). Universidade do Minho.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information*, 107–145.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques* (3rd Ed.). Morgan Kaufmann.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of Data Mining*. The MIT Press.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105.

Instituto da Conservação da Natureza e das Florestas. (2013). *Relatório Anual de Áreas Ardidas e Incêndios Florestais em Portugal Continental 2012*. Retrieved 20, September 2013 from http://www.icnf.pt/portal/florestas/dfci/relat/rel-if/2012/rel-fin

Instituto de Meteorologia. (2008). *Caracterização Climática 2007*.

Jain, A., Murty, M., & Flynn, P. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3).

Karypis, G., Han, E., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68–75.

Kisilevich, S., Mansmann, F., Nanni, M., & Rinzivillo, S. (2010). Spatio-Temporal Clustering: a Survey. *Data Mining and Knowledge Discovery Handbook* (pp. 855–875). Springer.

Laube, P., Kreveld, M. van, & Imfeld, S. (2005). Finding REMO—detecting relative motion patterns in geospatial lifelines. *11th International Symposium on Spatial Data Handling* (pp. 201–214). Springer Berlin Heidelberg.

Laube, P., Wolle, T., & Gudmundsson, J. (2007). Movement patterns in spatio-temporal data. *Encyclopedia of GIS*. Springer.

Lin, F., Xie, K., Song, G., & Wu, T. (2009). A Novel Spatio-temporal Clustering Approach by Process Similarity. *6th International Conference on Fuzzy Systems and Knowledge Discovery*. doi:10.1109/FSKD.2009.584

Liu, Q., Deng, M., Bi, J., & Yang, W. (2012). A novel method for discovering spatio- temporal clusters of different sizes, shapes, and densities in the presence of noise. *International Journal of Digital Earth*, (December), 1–20. doi:10.1080/17538947.2012.655256

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook* (2nd Ed.). Springer.

Manso, J., Times, V. C., Oliveira, G., Alvares, L. O., & Bogorny, V. (2010). DB-SMoT: A direction-based spatio-temporal clustering method. *5th IEEE International Conference Intelligent Systems* (pp. 114–119). IEEE International. doi:10.1109/IS.2010.5548396

Mcardle, G., Tahir, A., & Bertolotto, M. (2012). Spatio-Temporal Clustering of Movement Data: An Application to Trajectories Generated by Human-Computer Interaction. *XXII Congress of the International Society for Photogrammetry and Remote Sensing* (pp. 147–152).

Moreira, A., Santos, M. Y., & Carneiro, S. (2005). *Density-based clustering algorithms–DBSCAN and SNN*. Retrieved 10, December 2012 from http://andrey.savelyev.2009.homepage.auditory.ru/2006/Ivan.Ignatyev/AD/snn&dbscan.pdf

Moreira, G., Santos, M. Y., & Moura-Pires, J. (2013). SNN Input Parameters : how are they related ? *Crowd and Cloud Computing Workshop at International Conference on Parallel and Distributed Systems*. Seoul, Korea.

Peffers, K., Tuunanen, T., Rothenberger, M. a., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. doi:10.2753/MIS0742-1222240302

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

Rashid, B., & Hossain, A. (2012). Challenging Issues of Spatio-Temporal Data Mining. *Computer Engineering and Intelligent Systems*, *3*(4), 55–64.

Rinzivillo, S., Pedreschi, D., Nanni, M., Giannotti, F., Andrienko, N., & Andrienko, G. (2008). Visually driven analysis of movement data by progressive clustering. *Information Visualization*, *7*(3-4), 225–239. doi:10.1057/palgrave.ivs.9500183

Rosswog, J., & Ghose, K. (2008). Detecting and Tracking Spatio-Temporal Clusters with Adaptive History Filtering. *8th IEEE International Conference on Data Mining Workshops*. doi:10.1109/ICDM.Workshops.2008.122

Sander, J., Ester, M., Kriegel, H., & Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, *2*(2), 169–194.

Santos, M. F., & Azevedo, C. (2005). *Data Mining*. FCA - Editora de Informática, Lda.

Santos, M. Y., & Ramos, I. (2009). *Business Intelligence* (2nd Ed.). FCA - Editora de Informática, Lda.

Santos, M. Y., Silva, J. P., Moura-Pires, J., & Wachowicz, M. (2012). Automated Traffic Route Identification through the Shared Nearest Neighbour Algorithm. *Bridging the Geographic Information Sciences, International 15th AGILE'2012 Conference* (pp. 231–248). Avignon, France: Springer Berlin Heidelberg.

Silva, R., Moura-Pires, J., & Santos, M. Y. (2012). Spatial Clustering in SOLAP Systems to Enhance Map Visualization. *International Journal of Data Warehousing and Mining*, *8*(June), 23–43. doi:10.4018/jdwm.2012040102

Snow, J. (1855). *On the Mode of Communication of Cholera*. London: John Churchill.

Steinbach, M., Ertöz, L., & Kumar, V. (2004). The challenges of clustering high dimensional data. *New Directions in Statistical Physics*, 273–309.

Tork, H. (2012). Spatio-temporal clustering methods classification. *Doctoral Symposium on Informatics Engineering (26-27 January)*. University of Porto.

Turban, E., Shardam, R., & Delen, D. (2011). *Decision Support and Business Intelligence Systems* (9th Ed.). Prentice Hall.