Improving water network management by efficient division into supply clusters

Antonio Manuel Herrera Fernández*

FluIng – Instituto U. de Matemática Multidisciplinar

Departamento de Ingeniería Hidráulica y Medio Ambiente

A thesis submitted for the degree of

PhD in Hydraulic Engineering and Environmental Studies

Valencia, July 2011

*Supervisors:

Prof. Dr. Rafael Pérez García

Prof. Dr. Joaquín Izquierdo Sebastián

External examiners:

1. Karina Gibert (Universitat Politècnica de Catalunya, Spain)

2. Seán McLoone (National University of Ireland Maynooth, Ireland)

3. Helena Ramos (Universidade Técnica de Lisboa, Portugal)

PhD committee:

1. Francesco Archetti (Università degli Studi di Milano-Bicocca, Italy)

2. Eusebio Arenal (Universidad de Valladolid, Spain)

3. Karina Gibert (Universitat Politècnica de Catalunya, Spain)

4. Alexandros Karatzoglou (Telefónica Research, Spain)

5. Michael Tung (Universitat Politècnica de València, Spain)

Oral examination: July 5^{th}, 2011

Signature of the President of the PhD committee:

Abstract

Water is a scarce resource and must be efficiently managed. One of the purposes of efficient management should be reducing water losses and increasing supply performance in the networks through a profound knowledge of water supply networks (WSN). Obtaining this knowledge in real networks is a complex task because distribution systems may consist of thousands of consumption nodes interconnected by thousands of lines and the necessary elements to feed the network. These networks are not usually the outcome of a single process of design and are the consequence of years of anarchic responses to continually rising new demands. As a result, layouts lack a clear structure from a topological point of view.

The division of a water supply network into supply clusters enables sufficient hydraulic knowledge to be gathered to carry out maintenance tasks and guarantee quantity and regularity to the final consumer. This approach divides large and highly interconnected distribution networks into smaller sub-networks. These smaller networks are virtually independent and fed by a prefixed number of sources. Independence can be physically enforced in a number of ways. For instance, by closing valves in existing pipes, by sectioning existing pipes, or by introducing new pipes to redistribute the flow. Each supply cluster inlet must be equipped with at least one flowmeter to accurately measure and record consumption in short time periods. However, gauges and meters must also be placed to measure and control pressure, chlorine concentrations, and other supply parameters.

Water network division into supply clusters should not be understood only in network configuration terms, but also as a permanent method of management. It is essential to provide the system with a main centre that can receive and sort daily data, as well as analyse other information (such as

financial, climatological, inventory, and maintenance data). In this way, we can establish the real performance of sectors and take appropriate decisions regarding both operational maintenance and investment. From a classical perspective, the division of a water supply network into sectors is used with the aim of controlling leaks, since it helps maintain a permanent pressure control system. Nevertheless, this target has recently become more ambitious and incorporates new operational and management tasks.

This thesis proposes a suitable framework to establish efficient methods to divide the network into sectors and manage a WSN by taking advantage of this structure. These tasks will be addressed using kernel methods and multi-agent systems. Spectral clustering and semi-supervised learning have been shown to behave well when defining a sectorised network with a minimum number of cut-off valves. However, their algorithms are slow (and sometimes infeasible) when tackling divisions in a large WSN. The multi-agent system approach, firstly created as an alternative solution, is an excellent complementary tool for clustering kernel methods that use a boosting methodology. It is therefore possible to achieve a division of WSN into supply clusters even in the case of large networks. This thesis also highlights other machine learning and kernel methods, such as support vector machines, to manage a single supply cluster and facilitate the detection, identification and monitoring of possible abnormalities in water supply. In the same sense, predictive models are more accurate in a supply cluster than in the whole network, as they avoid biases derived from producing forecasts in smaller areas. Finally, another variant of kernel-spectral methods, similar to Google PageRank, is adapted and developed to assess the relative importance of the nodes of a WSN by assessing vulnerabilities and proposing a working-line for approaching several management and operation tasks (including WSN division).

Resumen

Dentro del llamado desarrollo sostenible, el agua es un recurso escaso que, como tal, debe ser gestionado de manera eficiente. Así, uno de los propósitos de dicha gestión debiera ser la reducción de pérdidas de agua y la mejora del funcionamiento del abastecimiento. Para ello, es necesario crear un marco de trabajo basado en un conocimiento profundo de la redes de distribución. En los casos reales, llegar a este conocimiento es una tarea compleja debido a que estos sistemas pueden estar formados por miles de nodos de consumo, interconectados entre sí también por miles de tuberías y sus correspondientes elementos de alimentación. La mayoría de las veces, esas redes no son el producto de un solo proceso de diseño, sino la consecuencia de años de historia, que han dado respuesta a demandas de agua continuamente crecientes con el tiempo (y haciendo que, en general, las redes carezcan de una estructura clara, desde un punto de vista topológico).

La división de la red en lo que denominaremos *clusters de abastecimiento*, permite la obtención del conocimiento hidráulico adecuado para planificar y operar las tareas de gestión oportunas, que garanticen el abastecimiento al consumidor final en cantidad necesaria y de manera regular. Esta partición divide las redes de distribución en pequeñas sub-redes, que son virtualmente independientes y están alimentadas por un número prefijado de fuentes. Su independencia puede ser forzada, físicamente, de varias maneras: Una de ellas será mediante el cierre de válvulas en tuberías; pero también se llevará a cabo por la sección de las mismas o introduciendo otras nuevas, que redistribuyan el caudal. Las entradas a cada cluster de abastecimiento deben estar equipadas con al menos un caudalímetro para obtener, con la mayor precisión posible, su consumo y almacenarlo en espacios cortos de tiempo como un registro de su base de datos. Cada sector también debiera

contar con manómetros y medidores de la calidad del agua, que situados de una manera adecuada registren los valores de presión y cloro; o de cualquier otra propiedad que se quiera estudiar y controlar.

La división de la red en clusters de abastecimiento no debe ser entendida tan sólo en términos de su configuración, sino también como un método permanente de gestión de la red. Para ello, es esencial que el sistema posea una central de datos, capaz de recibir diariamente, de manera instantanea, la información de la red y analizarla junto con el resto de información disponible (e.g., variables financieras, climatológicas y de mantenimiento e inventario). De esta manera, se puede conocer el funcionamiento real de los sectores y tomar decisiones más apropiadas respecto tanto del mantenimiento de la explotación como de su gestión. Desde una perspectiva clásica, la división de la red en sectores se ha venido usando con el propósito de control de fugas, dado que éstos ayudan a mantener el control de la presión del sistema. Sin embargo, la tendencia de los últimos años ha sido la de hacer más ambicioso este objetivo, y se le han ido incorporando nuevas tareas de operación y gestión de la red.

Esta tesis propone un marco de trabajo adecuado en el establecimiento de vías eficientes tanto para dividir la red de abastecimiento en sectores, como para desarrollar nuevas actividades de operación, gestión y planificación, aprovechando esta estructura dividida. En cualquier caso, la propuesta de desarrollo de cada una de estas tareas será mediante el uso de métodos kernel y sistemas multi-agente. El spectral clustering y el aprendizaje semi-supervisado se mostrarán como métodos con buen comportamiento en el paradigma de encontrar una red sectorizada que necesite usar el número mínimo de válvulas de corte. No obstante, sus algoritmos se vuelven lentos (en ocasiones infactibles) dividiendo una red de abastecimiento grande. Así, la aproximación basada en sistemas multi-agente fue creada, inicialmente, como alternativa a las metodologías anteriores. Sin embargo, a través de procesos de remuestreo, los sistemas multi-agente surgen como un extraordinario complemento para el clustering basado en métodos kernel. De

esta manera, se hará posible obtener la división en clusters de abastecimiento, incluso en el caso de redes grandes. Además, esta tesis destaca otros métodos kernel y de Aprendizaje Automático, como las Máquinas de Vectores Soporte para gestionar un cluster de abastecimiento facilitando la detección, identificación y monitorización de las posibles anomalías en el abastecimiento de agua. De la misma forma, también en un cluster de abastecimiento los modelos predictivos de la demanda son más precisos que en la red completa; a la vez que evitan los sesgos derivados de la obtención de predicciones en áreas más pequeñas. Por último, otra variante de los métodos kernel-spectral, como es el PageRank de Google, es adaptada y desarrollada para obtener la importancia relativa de los nodos de una red de distribución de agua; evaluando sus vulnerabilidades y proponiendo una nueva línea de trabajo para estudiar varias tareas de gestión y operación de la red (incluida la sectorización).

Resum

Dins del cridat desenvolupament sostenible, l'aigua és un recurs escàs que, com a tal, ha de ser gestionat de manera eficient. Així, un dels propòsits d'aquesta gestió haguera de ser la reducció de pèrdues d'aigua i la millora del funcionament del proveïment. Per a això, és necessari crear un marc de treball basat en un coneixement profund de la xarxes de distribució. En els casos reals, arribar a aquest coneixement és una tasca complexa degut al fet que aquests sistemes poden estar formats per milers de nodes de consum, interconnectats entre si també per milers de canonades i els seus corresponents elements d'alimentació. La majoria de les vegades, aqueixes xarxes no són el producte d'un sol procés de disseny, sinó la conseqüència d'anys d'història, que han donat resposta a demandes d'aigua contínuament creixents amb el temps (i fent que, en general, les xarxes manquen d'una estructura clara, des d'un punt de vista topològic).

La divisió de la xarxa en el que denominarem *clusters de proveïment*, permet l'obtenció del coneixement hidràulic adequat per a planificar i operar les tasques de gestió oportunes, que garantisquen el proveïment al consumidor final en una quantitat necessària i de forma regular. Aquesta partició divideix les áltamente connectades xarxes de distribució en menudes subxarxes. aquestes són virtualment independents i són alimentades per un nombre prefixat de fonts. La seua independència pot ser forada, físicament, de diverses maneres: Una d'elles serà mitjanant el tancament de vàlvules en canonades; però també es portarà a terme per la secció de les mateixes o introduint altres noves, que redistribuïsquen el cabal. Les entrades a cada cluster de proveïment han d'estar equipades amb almenys un caudalímetro per a amidar, amb la major precisió possible, el seu consum i emmagatzemar-lo en espais curts de temps com un registre de la seua base

de dades. Cada sector també haguera de contar amb manómetros i medidores de la qualitat de l'aigua, perquè, situats d'una manera adequada registren els valors de pressió i clor; o de qualsevol altra propietat que es vulga estudiar i controlar.

La divisió de la xarxa en clusters de proveïment no ha de ser entesa tan sols en termes de la seua configuració, sinó també com un mètode permanent de gestió de la xarxa. Per a això, és essencial que el sistema posseïsca una central de dades, capa de rebre, cada dia i de manera instantanea, informació de la xarxa i analitzar-la juntament amb la resta d'informació disponible (e.g., variables financeres, climatològicas i de manteniment i inventari). D'aquesta manera, es pot conèixer el funcionament real dels sectors i prendre decisions més apropiades respecto tant del manteniment de l'explotació com de la seua gestió. Des d'una perspectiva clàssica, la divisió de la xarxa en sectors s'ha vingut usant, i s'usa, amb el propòsit del control de fugides, atès que aquests ajuden a mantenir el control de la pressió del sistema. No obstant això, la tendència dels últims anys ha estat la de fer més ambiciós aquest objectiu, i se li han anat incorporant noves tasques d'operació i gestió de la xarxa.

Aquesta tesi proposa un marc de treball adequat en l'establiment de vies eficients tant per a dividir la xarxa de proveïment en sectors, com per a desenvolupar noves activitats d'operació, gestió i planificació, aprofitant aquesta estructura dividida. En qualsevol cas, la proposta de desenvolupament de cadascuna d'aquestes tasques serà mitjanant l'ús de mètodes kernel i sistemes multi-agent. El spectral clustering i l'aprenentatge semi-supervisat seran mostrats com mètodes amb un bon comportament en el paradigma de trobar una xarxa sectorizada que necessita usar el nombre mínim de vàlvules de cort. Encara que els seus algorismes es tornen lents (i en ocasions infactibles) dividint una xarxa de proveïment gran. L'aproximació basada en sistemes multi-agent, al principi va ser creada com alternativa a les metodologies anteriors. No obstant això, a través de processos de remuestreo, els sistemes multi-agent sorgeixen com un extraordinari complement per al clustering basat en mètodes kernel. D'aquesta manera, es

farà possible obtenir la divisió en clusters de proveïment, fins i tot en el cas de xarxes grans. A més, aquesta tesi destaca altres mètodes kernel i d'Aprenentatge Automàtic, com les Màquines de Vectors Suport per a gestionar un cluster de proveïment facilitant la detecció, identificació i monitoratge de les possibles anomalies en el proveïment d'aigua. De la mateixa forma, també en un cluster de proveïment els models predictius de la demanda són més precisos que en la xarxa completa; alhora que eviten els biaixos derivats de l'obtenció de prediccions en àrees més menudes. Finalment, altra variant dels mètodes kernel-spectral, com és el PageRank de Google, és adaptada i desenvolupada per a obtenir la importància relativa dels nodes d'una xarxa de distribució d'aigua; avaluant les seues vulnerabilitats i proposant una nova línia de treball per a estudiar diverses tasques de gestió i operació de la xarxa (inclosa la sectorització).

Kenbeo kenmaro

Acknowledgements

This thesis was written during the time I was part of the CMMF and FluIng at the Institute for Multidisciplinary Mathematics (IMM) at the Universitat Politècnica de València.

I would like to thank those who have contributed to this thesis. Firstly, I would like to express my gratitude to my supervisors Rafael Pérez and Joaquín Izquierdo for their confidence and support. They guided me in every moment while giving me the freedom to choose my own research lines.

I would also like to express my gratitude to the Spanish Ministry of Innovation and Science with references to both the project IDAWAS DPI-2009-11591 and PhD grant BES-2005-9708; as well as the travel assistance that enabled me to carry out part of this work at the Laboratory of Artificial Intelligence and Decision Support (LIAAD) at the University of Porto; and at the Laboratory of Computer Science, Information Processing and Systems (LITIS) at INSA in Rouen. I was privileged during these stays to work with talented and supportive researchers. Therefore, a significant part of this thesis is due to the help I have received from Luís Torgo, Stéphane Canu, and Alexandros Karatzoglou - among many others.

Last but not least, my gratitude to the people that somehow have been involved in this thesis and have been like an immense blue sea of mood and good feelings.

Contents

List of Figures

List of Tables

LIST OF TABLES

Glossary

DMA District Metered Area. Virtual or physical sub-network of the water supply system disposed to achieve more accurate measurements and better control of the whole network.

EPANET Public domain water software developed by US-EPA. EPANET can perform steady-state and extended period simulation analyses for network elements, calculating flow of water, pressures, tank elevations, concentration of chemical species, water age, and source tracing.

MAS Multi-agent System. A multi-agent system consists of a population of autonomous entities (agents) situated in a shared structured entity (environment). MAS's are able to solve complex distributed problems.

NetLogo NetLogo is a cross-platform multi-agent programmable modelling environment. NetLogo is particularly well suited for modeling complex systems developing over time.

NRW Non-revenue water. It is water that has been produced and is lost before it reaches the consumer. High levels of NRW reflect huge volumes of water being lost through leaks (real/physical losses), water not being invoiced or not being accurately measured (apparent/commercial losses) or both.

R R Language is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

SSL Semi-supervised Learning. A class of machine learning techniques that makes use of both labelled and unlabelled data for training (typically working with a small amount of labelled data and a large amount of unlabelled data).

WSN Water Supply Network. This is the base object of the study. Real networks may be complex systems, needing frequently being partioned into sub-networks to be better studied and managed.

Part I

Introduction and proposal of hydraulic sectors

1

Introduction

Rational distribution of water in supply systems is a complex problem. This complexity increases if the system is large and the goal is to offer regular supply of clean water at the pressure required by consumers. Distribution systems may consist of thousands of consumption nodes interconnected by thousands of lines and the necessary elements to feed the network. Most of the time, these networks are not the outcome of a single process of design. They are the consequence of years of history sometimes giving anarchic response to continually rising new demands. As a result, their layouts lack a clear structure from a topological point of view. This renders these networks difficult to understand and control.

Sectorisation, understood as network partition into sub-networks with controlled inputs and outputs, is a strategic option which homogenises the elements, measurements, and design parameters of each sub-network. In this way, we gain accuracy and avoid bias in decision-making about supply management. Sectorisation facilitates the detection, identification and monitoring of possible abnormalities in the water supply due to inspection area reduction. In addition, knowing the users included in each sector, along with an adequate treatment of the database, makes sectorisation essential for explaining water demand behaviour.

The concept of managing water supply networks based on district metered areas (DMAs), using the division of supply networks into water sectors, was a novelty introduced for detection and control of leakage. This method was born in the early 1980s in the UK water industry by the Department of the Environment and the National Water Council. Since then, its development has been primarily based on practical

implementations, with little scientific contribution. We can highlight to the manual of the Water Research UK Ltd., published in 1999, UKWIR (1999). More recently (December 2007) the IWA Water Loss Task Force published a practical guide for the management of DMAs [IWWA-Loss-Group (2007)]. Furthermore, within a more conceptual and scientific framework, Walski *et al.* (2001) proposed the establishment of a sub-metering system, supporting the implementation of water network sectorisation. Tzatchkov *et al.* (2006) applied graph theory to divide the supply network into different hydraulic zones. Hunaidi & Brothers (2007) collaborated with an article that seeks the optimal size of DMAs based on different criteria and taking into account their economic cost. Izquierdo *et al.* (2008) assessed the relative importance of pipes in a water supply network; based on this it is possible to establish a criterion formalising a zone network division. These same authors, Izquierdo *et al.* (2009), have developed software support to sectorisation, basing their work on multi-agent systems applications.

The location of leaks is the most developed use of hydraulic sectors. The works by Covas & Ramos (1999), from the late 1990s until now, have been devoted to this field of DMAs application. Yet, other ways to take advantage of working with sectors have also emerged. This is the case of the doctoral dissertation by Misiunas (2005), in which he discusses various aspects of the failures that can arise in a water network divided into DMAs. In the same year, Bougadis *et al.* (2005), highlighted differences in the behaviour of drinking water consumption by different areas. They proposed a hydraulic classification of the water uses to develop predictive models of demand. Later, Herrera *et al.* (2010d) conducted a comprehensive comparative study of predictive models and their efficient application to a DMA.

Division of a water supply system into water sectors has not only been introduced by scientific and technical developments, but also has practical implementation in different countries of Europe, Asia and America. In the case of Spain, various sectorisation actions have been accomplished in cities such as Barcelona (2004), Madrid (2005), Córdoba (2007) and San Sebastián (2008).

This thesis addresses the problem of water network partition into so-called *supply clusters*, which are hydraulic sectors, produced by the use of intelligent tools and machine learning methods. We claim that DMAs are produced in a more efficient way, thus improving on the ambiguous recommendations of the classical approaches. To this end, we introduce different abstractions for water networks and their components.

All of them are based on graphs, where edges are pipes and nodes are consumption points. There are special points, such as water tanks or pumping stations that are treated as graph nodes with some peculiar characteristics. Once this graph abstraction is accomplished, we focus the analysis on two lines:

- One is by semi-supervised learning (SSL) methods, Chapelle *et al.* (2006). SSL is a class of machine learning techniques that makes use of both labelled and unlabelled data for training (we typically work with a small amount of labelled data and a large amount of unlabelled data). Then, we propose the use of kernel graph and spectral clustering methods to divide the network into clusters. These are established under certain supply constraints, which are related to the labels of the SSL approach [Herrera *et al.* (2010a)].

- Another is by multi-agent systems (MAS), Shoham & Leyton-Brown (2009). MAS are systems composed of multiple interacting intelligent agents. An agent is any entity in a system which can generate events that affect itself and other agents, following certain behaviours. This proposes a new management framework where each system element cooperates with others towards their own individual targets, thereby achieving a global solution [Herrera *et al.* (2010b)].

An important SSL process is spectral clustering. This produces high-quality cluster configurations on small data sets but has some difficulties in large-scale problems. To approach a solution we propose re-sampling representative subgraphs of the network. This boosting process is based on multi-agent simulations, thereby facilitating the merging of the above methods [Herrera *et al.* (2010c)].

We have already mentioned the main goal of this thesis: to establish hydraulic sectors in some more efficient way, proposing a suitable framework for water network management and thereby covering an important scientific gap in the hydraulic research literature. This would have enough weight to be the main part of the current work, but interesting urban water management topics arise from a well sectorised supply network (and by the intelligent and machine learning methods used to manage them). Topics such as predictive methods in water demand [Herrera *et al.* (2010d)], anomaly control by causal models [Herrera *et al.* (2009b)] or sensor location, which offer new possibilities when operated by sectors. These new trends in methods for water network

management present a line of work to be followed in the future. An introduction to these items, and some new ways to approach them, closes this thesis.

1.1 The aims of the thesis

The main aims of this thesis are to:

- Propose new points of view about water system management by hydraulic sectors. These comprise traditional issues, but go beyond them, providing us with an overall and a partial system management.
- Divide, efficiently, the water network by a partition based on independent supply clusters. This takes advantage of graphical and vector information.
- Extract classification features to classify and define each sector.
- Offer new methodologies for working in network configuration terms. It also proposes suitable methods to manage them.
- Remedy the lack of scientific results about the establishment and management of hydraulic sectors.

1.2 Outline of the thesis

This thesis has eight chapters:

- **Chapter 1** corresponds to this Introduction.
- **Chapter 2** describes the basics of urban water management and the advantages and disadvantages of working with sectorised networks. Then it reviews the current state-of-the-art of hydraulic sectors establishment and demonstrates the necessity of improving its ambiguous methodologies. Clustering-based methods are our working hypothesis. We propose that an adequate clustering process will be enough to address the classical targets associated with DMA and hydraulic sectors, extending them to new ways to manage a water supply system.

- **Chapter 3** introduces and develops kernel spaces to augment clustering capacities, while using both graphical and vector information. To do this, we propose adding the various supply constraints to the adjacency matrix of the graph, thus gathering the reality of the hydraulic zones in a single matrix. The next step splits the network, applying a spectral clustering algorithm. This methodology offers an adequate solution to the hydraulic zones paradigm by clustering, allowing the conditions for the zones to become small quasi-independent water supply networks.

- **Chapter 4** uses a multi-agent approach to establish supply clusters in water networks. Multi-agent techniques have proven to be highly efficient in the solution of very complex problems of distributed nature, such as the one we consider here. In the simulation proposed, consumption nodes are agents of a certain breed, while pipes are links, which connect two different nodes. Both, along with the source points breed, have some associated variables. Supply clusters are the result of simulated cooperative interactions of these different agents' behaviours.

- **Chapter 5** develops a boosting methodology to establish supply clusters in large water networks. To approach a solution, we propose re-sampling representative subgraphs of the network. This boosting process is based on multi-agent simulations of virus propagation behaviour to obtain a final subgraph of the 'infected nodes'. Next, we iteratively build semi-supervised supply cluster partitions on them. After obtaining each individualised solution we re-weight the data to obtain the next sample in an adaptive way. This approach to the establishment of supply clusters proposes merging both SSL and MAS methodologies.

- **Chapter 6** applies the proposed algorithms to a complex real case study.

- **Chapter 7** employs supply clusters to improve different management actions. Firstly, it is proposed using sectors in predictive models for forecasting water demand. This idea is motivated by the homogeneity and utility of these sectorised results. Yet, supply clusters are also proposed because their size is appropriate for carrying out successful searches for causes and effects of anomalies in water networks. Another management topic, which is also improved via supply sectors, is network sensor location, because its complexity diminishes with network size.

A second part of this chapter introduces an adaptive way of Google's PageRank to rank nodes in a water supply network. This creates a suitable framework to study the vulnerability and robustness of the network. Ranking nodes may be viewed as a useful tool for the process of hydraulic sectorisation.

- **Chapter 8** summarises the main conclusions and contributions of this work, enumerates the publications associated with this thesis, and offers ideas for extending further our research.

Figure 1.1 summarises the structure of this thesis:

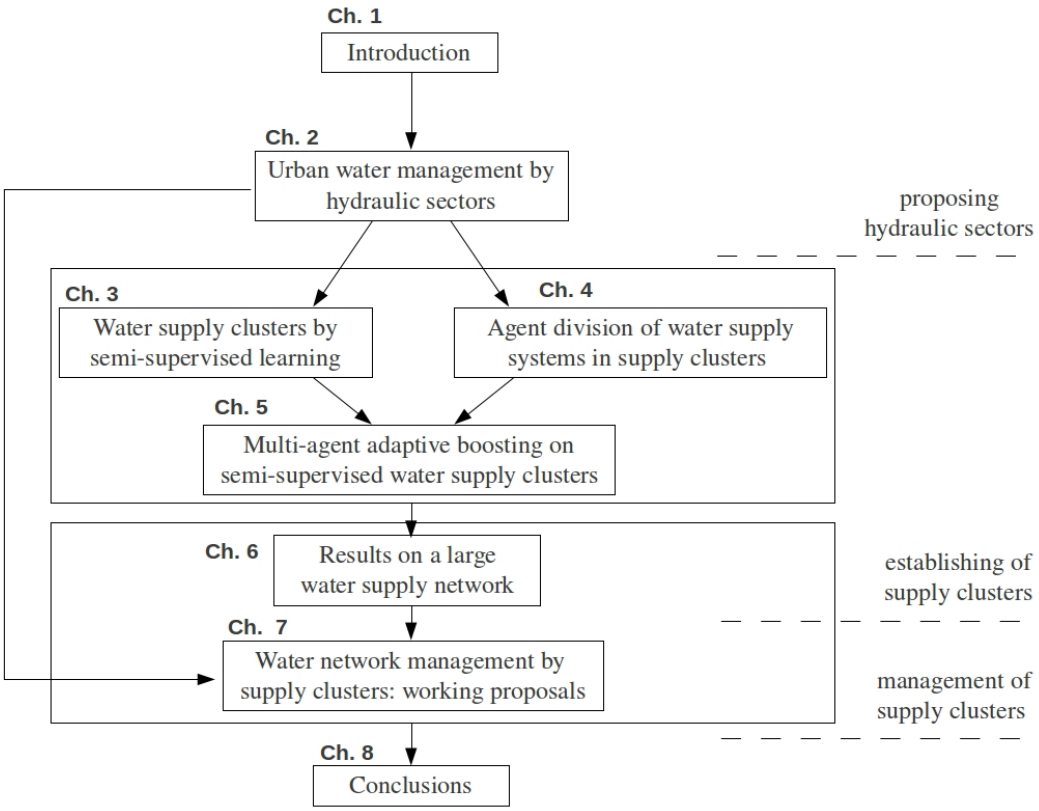

Figure 1.1: Thesis structure - showing the organisation of chapters

In addition to the main body of this thesis, there are two appendixes that are of interest for future reference: **Appendix A** expands on the software information. **Appendix B** offers an overview of the current predictive models to water demand

forecasting. It is based on Herrera *et al.* (2010d). **Appendix C** and **Appendix D** provide Spanish and Valencian versions of the conclusions of this thesis.

- **Appendix A** describes the software employed and develops parts of the code associated with our proposals.
  - **Appendix A.1** describes the NetLogo platform. In addition, we include part of the developed code and captions for the different multi-agent processes developed in this thesis.
  - **Appendix A.2** describes the R Language environment. We discuss some libraries for the kernel methods and graphs used in our analysis.
  - **Appendix A.3** provides an exchange framework of the software used.
    - **Appendix A.3.1** presents the NetLogo-R-Extension. It consists of NetLogo primitives for sending data between NetLogo and R and for calling R functions [Thiele & Grimm (2010)].
    - **Appendix A.3.2** introduces a new R library which is able to convert EPANET output to R input (and vice-versa). It will be a helpful tool for the analysis of hydraulic data and validation of results.
- **Appendix B** offers a brief summary of the predictive models used in Section 7.1 of Chapter 7. Every model will be introduced along with some keys about their R Language implementation.
- **Appendixes C and D** give the translations of the conclusions of this thesis.
  - **Appendix C**: Spanish translation.
  - **Appendix D**: Valencian translation.

1.3 Contributions of the thesis

The contributions of this thesis (see Section 8.1 for a detailed list) are divided into two parts. One is based on the intelligent and machine learning clustering algorithms developed to establish supply clusters in a water supply network. The other is about the methods to improve network management based on the accurate and unbiased

results achievable within the framework of a hydraulic sector. In both cases the proposed methods have the desirable properties of robustness, computational efficiency and statistical stability.

Classically, a division of a water supply network into hydraulic sectors aims at improving leakage detection using node elevation, pressure and demand information. In this thesis we propose augmenting, or changing, the perspective of this goal. This is accomplished by taking into account different information to be included within the criteria for the partition into clusters. In this sense, semi-supervised clustering offers the flexibility of including different inputs into the study and, thus, different criteria for dividing the network.

A multi-agent proposal is introduced as an alternative solution to clustering water supply networks. Multi-agent simulations are able to verify water distribution characteristics, addressing successfully the task of clustering under certain water supply constraints. Multi-agent methods can also be integrated elements for improving the semi-supervised process just introduced. This is of special importance in large real networks, where spectral-clustering computations become slower (sometimes infeasible). Then, sampling subgraphs by exploration, based on multi-agent simulations, is fundamental for applying the algorithm on the boosted subgraphs.

From the point of view of the performance of a network sector, we propose new alternatives to manage each established supply cluster in some efficient way (in line with the same tools used to previously establish them). One uses machine learning methods to forecast hourly water demand. The use of these algorithms, along with the medium size of a demand area, creates a suitable environment for making adequate decisions. Another management alternative is the automatic identification of unexpected or abnormal events characteristic of water demand. In this case, support vector machines and kernel causal algorithms may be used to explain the effects of different dis-functions of the water network elements. They also identify zones specially sensitive to leakage and other problematic areas, with the aim of including them in reliability plans. The third, and final management topic developed in this thesis, is to propose ranking nodes in a WSN. This approach is related to vulnerability analysis, rehabilitation plans and the sensor location paradigm, among others. Nevertheless, ranking nodes may also influence the establishment of supply clusters.

2

Urban water management by hydraulic sectors

Current management strategies, used by companies, hinge on the need to accurately capture infrastructure data sets. Real water distribution systems may consist of thousands of consumption nodes interconnected by thousands of lines and the necessary elements to feed the network. This complexity is augmented by a number of issues: water use behaviour, use of multiple water sources for different purposes, flows and leaks of water through a complicated technical system, the water supply being dependent on somewhat random weather patterns and complicated hydrological dynamics, as well as its dependence on pumping and allocation schedules, among others. Consequently, as with other complex systems, water supply networks demand deeper hydraulic knowledge to operate and to carry out tasks of maintenance, guaranteeing the quantity and the regularity of the supply to the final customer. The division of a network into hydraulic sectors is a strategic option for approaching this complexity. It is used in many cities worldwide to control and operate their systems, seeking the improvement of water supply networks management by working with each part in an isolated way.

Moreover, water supply should satisfy the final user demands in a suitable way to each of the different uses (personal/public, domestic/industrial). These uses are one of the main reasons that justify managing water networks by sectors proposing selective supply depending on the consumer behaviours and the quality requirements of each zone. Clearly it is not efficient to arrange the same supply conditions in an industrial area and in a residential neighbourhood. Nevertheless, there are still more reasons in

favour of a sectorised supply instead of working with the whole network: to preserve different pressure levels in the system, to maintain pipes of a wide range of ages and materials, to design rehabilitation plans, or to execute market research, are instances where it is easy to guess that more efficient results may be obtained by starting the analysis from a suitable network partition. This partition should take into account natural features of the area to be sectorised such as railways, rivers, roads and different topography elements. Besides these constraints, we have to add criteria about the main lines of WSN management if we want to design a system division efficient in some sense. Thus, it is necessary to have a sufficiently flexible division, working with the aforementioned management instances or others. This partitioning method should be extendible and able to adapt the sectorisation to the various objectives, in every district or different network.

The chapter is structured as follows. In Section 2.1 some preliminaries about the history of WSN are related. Section 2.2 introduces WSN management issues such as leak detection, demand management and network vulnerability analysis. A target area reduction allows approaching all of these tasks in an efficient way. This is the aim of sections 2.3 and 2.4, which present the implementation and performance of hydraulic sectors to achieve WSN management. Section 2.5 introduces a couple of case-studies which will be analysed throughout this thesis. Lastly, Section 2.6 summarises this introductory chapter.

2.1 Preliminaries

Water supply systems, as we know them today (with their structure and performance), are relatively new. Certainly, water distribution and transport, through pipes and other conduits, have an extensive history which dates back to the existence of gregarious communities. Hydraulic works carried out by the Roman Empire or the splendour period of the Arabs are well known examples. The inventiveness with which they solved water elevation, transport, and distribution problems arouses our curiosity. Nevertheless, it is in the middle of the XIX century when we find a clearly defined scheme of what we call an urban water supply system. It is in this historic moment of the Industrial Revolution, when rural population moves to big cities to find opportunities and suste-

nance. The increasing population density in these metropolises aims to reconsider such common services as supply and waste water systems.

Water supply system evolution may be studied under social and organisational perspectives. Thus, Matés (2001) introduces the urban technical networks. This concept refers to urban transport services, street lighting, energy, gas, water, and sanitation. These services are characterized by some permanent performance based on suitable technologies and a controlled organisation by public entities. Then, the main novelty is not the existence of such systems, but their general organisation following certain guidelines. These ideas may be traced back to the 1850s.

In the case of supply and waste water systems, the concept of network is intimately associated with technology innovation. Firstly, with the target of epidemics removal, improving water supply quality, and increasing the hygienic levels of the population. Next, by the progressive implementation of supply networks with high capacity able to distribute pressurised water.

Matés (2009) introduces an interesting contrast between classical and modern water systems. The characteristics of a classical system are summarised by:

- Low per capita consumption, located at a minimum which can be considered of biological nature (5 - 10 litres per inhabitant and day).
- Collective supplies such as irrigation canals and aqueducts. Individual supplies such as wells and domestic cisterns.
- Technical limitations by the lineal character of aqueducts and the impossibility of accessing to every urban property.

The transition to a modern system slowly developed for approximately a century, starting during the second half of the XIX century. It is a consequence of the appearance of new necessities triggered by the increasing of population size (growth of water demand and waste-water, among others). The main characteristics of a modern system are summarised by:

- Increasing consumption to previously unknown levels (80 - 300 litres per inhabitant and day).

- Predominance of collective networks inherently associated with urban space organisation.
- Innovations associated with the industrial era emerge: pressurised water, pipeline systems, extension to the whole population, and quality control.

In the late XIX century a legislative framework for water management is set up. It appears the necessity to impose fees on water use, in order to address investment and operational needs of the systems. During this stage, the participation of private firms is consolidated as an intermediate role regarding service delivery. The consideration of water as a social good starts in the early twentieth century, allowing a major public control of water management. Yet, it was in the 1960s when the development plans promoted the establishment of supply networks in small towns, and the States required again the involvement of private enterprises in the service. The subsequent evolution of water management brought out the severe indebtedness of municipalities, which has caused a new wave of service privatisations.

Along the time, focus on the study of water supply systems has been traditionally from the point of view of Engineering. Other disciplines such as Economy, Sociology, or History have not had special interest in water supplies until very recent dates. Nevertheless, we should take into account that Engineering applications are subject to complex purposes and necessities, in which social, economics, and normative elements participate. Thus, this bias towards Engineering has changed with the time and the current trend is to keep on evolving towards a multidisciplinary focus.

2.2 Essentials of water supply network management

Urban water systems are socio-technical systems that are a critical component in the functioning of cities [Moglia *et al.* (2010)]. In a general overview, one sees that their management tasks include the decision making support to deliver wholesome water to the consumer at adequate pressure in sufficient quantity at convenient points and achieve continuity and maximum coverage at affordable cost. To attain this objective we have to develop operating procedures to ensure that the system can be operated satisfactorily. In addition, each individual network (or each district on the network)

could have different management priorities depending on their particular characteristics. Thus, setting and choosing a suitable way to achieve these additional goals will be fundamental to addressing an appropriate administration of resources. So, the management process involves the planning and design of operational procedures required for inspecting, monitoring, testing, repairing and cleaning the system as well as for locating the buried pipes and valves. Network records and maps should be updated and have sufficient details of the system facilities, their condition, routine maintenance that is needed and done, problems found and corrective actions taken. Analysis of the records will help evaluate how well the installations are working and how effective its services are and hence assess their adequacy to meet the needs of consumers.

Other network management methods may also be found in the electricity field. However, the differences between them make that the majority of non-linear hydraulic processes may have a linear (or linearisable) electric counterpart. This is because there exist some simplifications in the energy distribution, comparing to water supply. These are regarding flow rates, head losses, and continuity conditions in its distribution, among others. In addition, electricity distribution networks are operated in a radial configuration [Lakervi & Holmes (1995)] (treelike structure). This means that each load point is connected to the point of supply through a single path. All of these enumerated issues make different hydraulic from electricity network management. Yet, some applications on the performance of WSN should be suitable to manage electricity networks under similar methodologies to the introduced in this thesis. Thus, electricity applications such as risk analysis [Oonsivilai & Greyson (2009)] or demand forecasting [Bozic & Stojanovic (2011)], use methodologies related to multi-agent systems and support vector machines (both are in the same direction that the approaches introduced in Chapter 7 of this thesis).

An urban water supply management system should at least include the following items:

- A description of the service area of the supplier, including current and projected population, climate, and other demographic factors affecting the supplier's water management planning;
- A study of the reliability of the water supply and vulnerability to seasonal or climatic shortage, providing data to forecast water demand in different scenarios.

The vulnerability analysis may be completed by the study of possible system failures or intentional damages;

- A quantification of past and current water use, describing and projecting it;
- A description of each water demand management issue (see Figure 2.1), including the steps necessary to implement any proposed action such as system water audits, leak detection, and repairs.

These and other management issues may be included as part, or consequence, of actions like *leak detection*, *demand management* and *screening of system vulnerabilities*. For instance, an optimal pumping scheduling is directly related to water demand forecasts [López-Ibáñez (2009)]; a suitable rehabilitation plan depends on leakage and pipe burst occurrences [Alonso (2010)]; and so on. This is the reason why this chapter includes brief introductions to these items in the next subsections.

2.2.1 Leak detection

Leakage is defined as non-controlled water output through any part of a WSN. Leaks can vary depending on soil type, quality of construction and materials used, age of facilities and operation and maintenance practices. There are also phenomena such as corrosion, which can increase this problem. In addition, pipe bursts also may occur due to:

- stress caused by vibration and surface charges;
- compression resulting from faulty construction;
- fatigue of materials, manufacturing defects or water hammer.

Leakage management practice follows two different approaches: passive and active control [Sturm & Thornton (2005)]. The passive way of control is based on taking actions when leakage becomes visible (when sometimes is too late to make decisions). On the other hand, active control (by acoustic instruments among other methods) may be an expensive and time-consuming activity [IWWA-Loss-Group (2007)], especially if the water network is too large. This is the main reason why a proactive leakage control has been developed and used during the last decade [Covas & Ramos (1999), Hunaidi

(2005), Fantozzi *et al.* (2009)]. The solution is a permanent control system whereby the network is divided into DMAs supplied by a limited number of key mains, on which flow meters are installed. The importance of defining a DMA is to control all the inflows and outflows in the area; it facilitates the creation of a permanent pressure control protocol. Thus, it is possible to detect and locate leaks using volumetric methods during the day or the minimum night flow approach. Moreover, in a sectorised network leaks that appear in one district may be repaired with a minimum impact on the rest of the sectors.

Leakage detection is a paradigm of growing interest because of its potential economic value. Moreover, current research trends related to this issue focus also on water quality problems, network security terms and the study of the relationship between water loss and environmental issues.

2.2.2 Demand management

Water demand is defined as the volume of water requested by users to satisfy their needs. In a simplified way it is often considered equal to water consumption, although conceptually the two terms do not have the same meaning because we have to take into account both real and apparent water losses, like non-revenue water (NRW) [IWWA-Loss-Group (2005)]. This last will be a helpful tool to detect water leaks and other forms of unbilled consumptions. Anyway, the more important purpose of water management is to match or balance the demand for water with its availability, through suitable water allocation arrangements.

There are a number of factors which can be related to water demand (Figure 2.1). Their current level of influence on demand and their specific behaviour trends are of great interest in the process of developing a demand management program. Some of these factors are enumerated next:

- water usage practices (including pricing, regulation restrictions, income levels, socio-cultural factors, knowledge and awareness, technical innovation and presence of water companies);
- water using equipment;
- demographics and land use;

- climate;
- water supply system;
- source substitution.

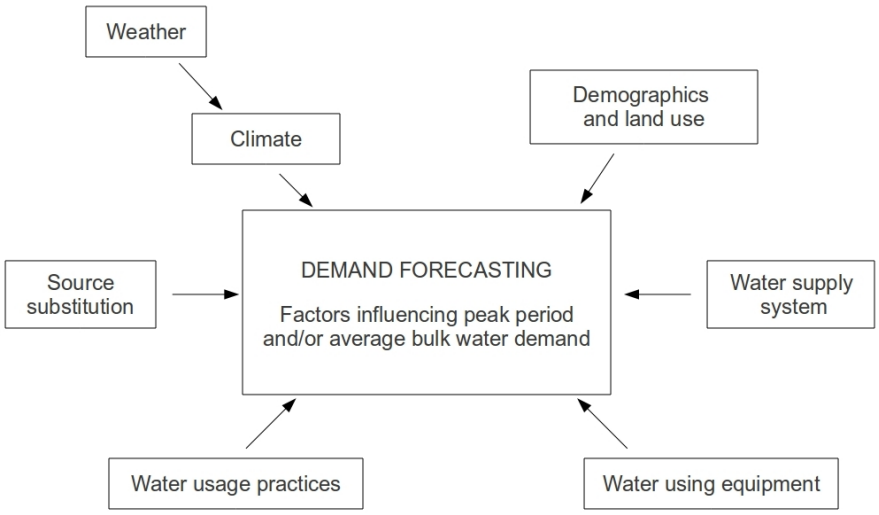

Figure 2.1: Water demand management scheme - based on White *et al.* (2003)

Demand management programs are designed to anticipate changes in consumer behaviour or changes in the stock of resource. This is key to establishing efficient pumping schemes [López-Ibáñez (2009)], which minimise energetic cost in some way. In addition, this demand knowledge also allows planning water pricing, temporary closing of pipes or detecting anomalies in the supply system [Herrera *et al.* (2009b)]. As we can see in Figure 2.1, these management decisions will be related to water demand forecast. One recent relevant study about this subject is Herrera *et al.* (2010d). This work starts with the decision of using only one hydraulic sector for the study. It is motivated by the homogeneity and utility of the results, the elimination of sources of bias and the avoidance of the impact of a small set of consumers that may incorrectly bias the forecasts due to unusual consumption profiles.

2.2.3 Screening of system vulnerabilities

Assessing the vulnerability of water supply systems tries to measure their resistance to a malicious attack, a failure or a disaster; and to determine their adverse consequences.

The main common elements of vulnerability assessments are as follows [Mays (2004)]:

- characterisation of the water system, including its mission and objectives;
- identification and prioritisation of the adverse consequences;
- determination of critical assets that might be subjected to malevolent acts that could result in undesired consequences;
- assessment of the likelihood of such malevolent acts.

The identification of critical components of the network, whose disruption or removal may result of significant negative consequences is key [Gutiérrez-Pérez (2010)]. The *per-se* importance of drinking water in human health, along with the spatial distribution of the supply network, makes necessary the development of early warning systems able to prevent and/or mitigate possible anomalies in WSNs [Hasan *et al.* (2004)]. Therefore, online monitoring systems are considered as possible solutions to protect WSNs against the impact of contamination events. This monitoring will be available by optimal location of control sensors on the network, which can be achieved by optimising economic costs, including the possibility of taking into account demand uncertainties [Shastri & Diwekar (2006)]. For this problem of sensor placement, the aforementioned uncertainty can manifest itself through changing population density/water demands at various junctions or through a varying probability of contamination at a node.

Summarising, identifying and prioritising vulnerabilities in a WSN is a complex problem; also placing sensors to control, prevent or mitigate undesirable events are complex in nature. Within this working line, Herrera *et al.* (2009b) have established a cause-effect relation in the detection and definition of anomalies in a DMA. Focusing the aforementioned management actions on a reduced sub-network of the whole water supply simplifies the complexity of the proposed methodologies. In addition, inspection area reduction can serve as a first approach to a network organised in sectors by the relative importance of pipes and junctions [Izquierdo *et al.* (2008)].

2.3 Implementation of hydraulic sectors in water supply networks

The division of a network into DMA follows a divide and conquer strategy that splits the large highly interconnected distribution network into smaller sub-networks. These smaller networks are virtually independent and are fed by a pre-fixed number of sources. This independence can be physically enforced in a number of ways. For instance, by closing valves in existing pipes, by sectioning existing pipes or by introducing new pipes that redistribute the flow.

The steps of a sectorisation process are:

- Preparation of a draft of the system;
- Design and implementation of a pilot sector;
- Expansion of the previous step to a complete partition of the network;
- Review the technical and economical feasibility of the proposed sectorisation;
- Integration of each part with the centralised control system.

2.3.1 Viability studies

There are two main characteristics which we should take into account in assessing the suitability of a sectorisation process:

- The whole network is functionally disaggregated into different levels of pressure.
- The system is a highly meshed network (this especially occurs in very populated areas).

Thus, it is essential to perform a viability study to check the changes made via sectorisation, which must not be too extreme with respect to the previous network. In addition, the supply level may be adequate in pressure and quality values. To do so, it is necessary to run hydraulic simulations of the sectorised model.

2.3.2 Design and enforcement of sectors

Before starting to design, it is convenient to define the basic characteristics of sector sizing (average demand, number of consumers, network size, etc.). Thus we take into account the homogeneity in the division structure. Sectorisation turns an unstructured meshed network into a sector-structured meshed network; this has some drawbacks which must be minimised by simulating the final network topology. This simulation should guarantee a minimum supply pressure, whereby reinforced capacity must be provided if necessary. In addition, it is likely to impair the supply of an area because each sector has just one or few entries. Thus, it would be advisable to design these checkpoints with by-pass pipes and identify alternatives if they are necessary [Valdés & Castelló (2003)].

The factors that should be taken into account when designing a hydraulic sector are [IWWA-Loss-Group (2007)]:

- Size (geographical area and number of customer connections);
- Housing type (e.g., blocks of flats or single family occupancy housing);
- Variation in ground level;
- Water quality considerations;
- Pressure requirements and fire fighting capacity;
- Number of valves to be closed;
- Number of meters used to monitor flow (ideally minimised);
- Infrastructure condition.

Once the sectors are designed and enforced, it is important to update all information systems with the real status of the valves in use. In this sense, a sector boundary should not necessarily be considered definitive. With the change in operating conditions, it might be necessary to modify the boundary. For this reason it is usually better to create a boundary by closing valves rather than cutting pipes. However care must be taken to ensure that these valves are leak tight and that their accidental opening is avoided.

When a distribution network is configured and checkpoints are sized (in type and diameter), the sectorisation process passes to the civil works stage (e.g., the physical construction of manholes and installation of flowmeters). It is possible that a checkpoint requires relocation or even that we have to redesign some sector because of construction problems (given the limited space available in the middle of side-walks and urban walk-ways in which there are plenty of gas and electricity amenities available to the cities). Finally, it is necessary to have a control centre to receive data. This centre should be able to transform these data into updated information. This makes it possible to integrate each part with the system management, allowing suitable monitoring both of each sector and of the whole network and achieving efficient decision making regarding the aforementioned key issues in water supply management (see Section 2.2).

2.4 Classical and recent trends in hydraulic management by sectorised networks

From a classical perspective, the division of a water supply network into DMAs is used with the goal of leak control as in [Covas & Ramos (1999)], since it helps maintain a permanent pressure control system. This is the main reason for the IWWA-Loss-Group (2007) to recommend a DMA design related to the technique of leakage monitoring. This design has the goal of identification of leaks, maintaining the optimum level of pressure along the whole network. From this point of view, DMA division carries out a permanent leakage control system which maximises the accuracy of leakage within each DMA, facilitates the location of leaks and minimises the changes regarding the existing network [Hunaidi & Brothers (2007)]. But the sectorisation of WSN has other potential benefits:

- Containment of water quality incidents is much easier if an area can be rapidly isolated;
- Sectorisation can reduce the extent and complexity of mixing of different water sources;
- Interpretation of simple analysis data is easier.

Thus, the classical leakage monitoring goal has become more ambitious recently: now the main objective of sectorisation is to have enough distributed information, in a scale easy to handle. From this point of view, the following actions (some of them are coincident with the aforementioned aim of leakage monitoring) can be performed in each DMA [AVSA (2009)]:

- To carry out audits to know the hydraulic efficiency or NRW;
- To characterise the demand curve, especially the night flow;
- To detect, in a quick way, the possible leaks analysing the evolution of the minimum night flow;
- To detect frauds, under-registration or diverse errors of measurement;
- To diminish the maintenance costs;
- To plan the investments when guiding the stocks to sectors with more NRW.

Being more ambitious, it is possible to add to these objectives others such as improving demand management (cf. Subsection 2.2.2), looking for an optimal distribution of sensors on a water supply network or explaining system anomalies (cf. Subsection 2.2.3). This thesis proposes not only to achieve these goals, but also to design the sector division taking them into account. Thus, it will be possible to approach efficient network management starting from a suitable division into sectors, which takes into account the goals of the corresponding supply performance.

2.5 Case-study

Thoroughout this thesis some experimental and real data studies will be developed. They will illustrate algorithms, procedures and methodologies, which will be proposed in this work. However, it will give special treatment to the case study of Celaya, which is detailed next.

2.5.1 Main case-study

The municipality of Celaya belongs to the Guanajuato state, Mexico. It has an area of 579.3 km^2, within the geological system of the central plateau (see Figure 2.2). Its geographical coordinates are 20^o 31' 44" N, 100^o 48' 54" W and its mean elevation over the sea level is 1755 meters. The WSN for the city of Celaya is distributed in 32 districts, including the 86,312 connections across the network. The demand for different types of water consumption is covered by approximately 1345 km of main pipes, with a density of 64 connections per km of distribution lines. It has a supply level around 180 and 300 litres per capita per day. The water loss of the system is approximately 35% of the water supply. It is estimated that these leakage levels are higher in the centre because the WSN pipes of this area have an age between 40 and 50 years [Alonso (2010)].

Figure 2.2: Location of the case-study area - Celaya, Guanajuato, Mexico

The Centre Zone of Celaya is bounded on the North by the railway line, on the South by Constituyentes Avenue, on the East by 2 de Abril Avenue and at West by the railway line. It currently has 22,072 contracted water demand points, corresponding to a population of 110,360 inhabitants. The current water demand is around 350 l/s and the pipe age is close to 50 years. This causes various problems such as water losses, low pressures and contamination by different intrusions.

The sectorisation project of the Centre Zone of Celaya (see Figure 2.3) started in 2006. This raised a management working line that tried to establish DMAs or hydraulic sectors to supply drinking water in enough quantity and quality to costumers living in the area. Besides this, the project tries to assess the sectorisation efficiency and the decision making support for controlling and reducing leakages [Blanco-Figueroa (2009)].

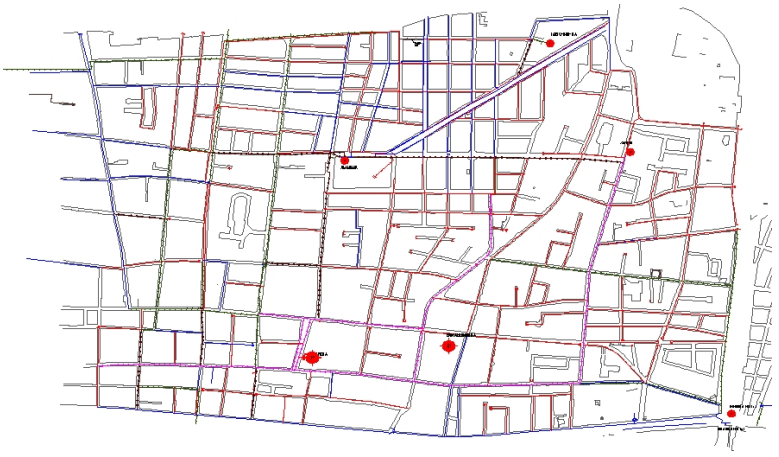

Figure 2.3: Layout of the case-study proposed - Central district of Celaya

The working line about this case-study, which the current thesis proposes, will try to:

- Propose a division of the Centre Zone of Celaya from an efficient supply management point of view. This should take into account different issues such as: pressure found in the area, age of pipes and rate of leakage, among others. In addition, the proposed division will benefit from the geographical, graphical and the rest of the vector information;
- Extract classification features to define each particular sector;
- Propose a water demand forecasting methodology to analyse in an accurate and unbiased way the water needs of the zone's customers;
- Develop new systems which explain anomalies and errors in the supply;
- Offer new methodologies to work in hydraulic and network configuration terms and also propose suitable methods to manage them;
- Improve the support to the decision making process regarding WSN management.

2.5.2 A simple real case

Contrary to the case of Celaya, which is widely treated in Chapter 6, we also introduce a simple case-study to appear together each presented process in this thesis. The main aim of this small network is to test, at a glimpse, the experimental performance of the new algorithms proposed. This allows to obtain fast checking about the performance of the initial targets of these processes. This case-study is (see Figure 2.4) fed by three reservoirs and made out of 132 lines and 104 consumption nodes; its total length is 9.1 km and the total consumed flowrate amounts to 47.1 l/s.

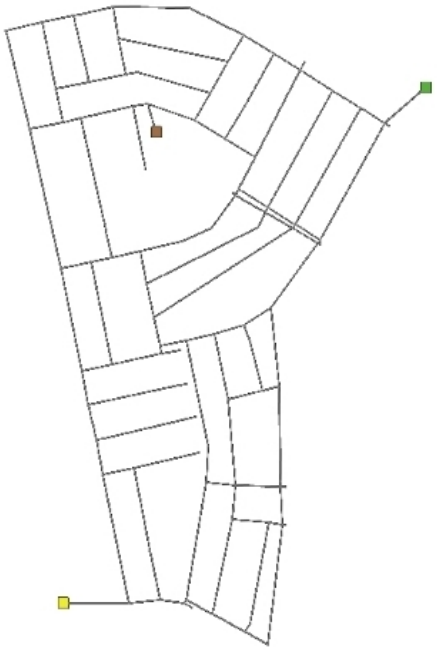

Figure 2.4: A simple real case study - Layout of the network

2.6 Summary and comments

This chapter has introduced WSN management and proposed an approach based on a suitable sectorisation of the network. Various fundamental issues can be improved by following adequate methodologies. In addition, developing these methods opens up the classical sectorisation goals to more ambitious ones. This completes and redefines the concept of sectorisation in a novel way to understand and manage a WSN. The chapter

closes by describing the main case-study employed in this thesis, which will serve to check the methodologies proposed to divide the network and manage the resulting sectors.

Part II

Establishment of supply clusters

3

Water supply clusters using semi-supervised learning

This chapter provides an efficient solution to the problem of designing hydraulic sectors in a water supply network (WSN). A statistical learning tool, based on clustering methods, is developed and adapted to the particulars of this specific division paradigm. Designing this division of a WSN should take into account the hydraulic constraints related to sectorisation and a final homogeneity criterion on their resultant structure. This homogeneity poses various issues: avoid abrupt changes in pressure levels, allow deep and accurate knowledge of the network and reduce pumping and maintenance costs, among others. A clustering methodology is proposed for both purposes. The main reason is precisely the coincidence with the fundamental cluster analysis aim: partitioning data observed into groups so that the pair-wise dissimilarities between those assigned to the same cluster tend to be smaller than those in different clusters. Thus, a clustering solution is proposed to carry out an optimal network division. But these algorithms are of a general character and must be adapted to the special features of a WSN, which will be considered as a particular graph where the links are the pipes and the nodes are the consumption points. Thus, the proposed clustering approach will be closely related to *graph clustering* algorithms [Schaeffer (2007)].

The main challenge of this chapter is to define the kernel matrix [Schölkopf & Smola (2002)] that captures the semantics inherent to the graph structure but, at the same time, is reasonably efficient for evaluation. The idea of constructing kernels on graphs was first proposed by Kondor & Lafferty (2002), and extended by Smola &

Kondor (2003). In the first instance, the affinity graph matrix is transformed into a kernel matrix, carrying out the correspondence kernel abstraction of the essential characteristics of the WSN. Next, spectral clustering techniques are applied to this new matrix [von Luxburg (2007)]. But constraints occurrence is natural for graphs and a certain amount of knowledge about WSN structure is available. This raises the idea of using graph-based semi-supervised learning methods [Kulis *et al.* (2005); Zhu *et al.* (2006)]. These learning methods can be viewed as imposing a mechanism of smoothness conditions on the target function with respect to a graph representing the data points to be labelled.

This approach finds solutions to the hydraulic sectorisation paradigm in some clustering methodologies; therefore we will call the proposed sectors *water supply clusters* (or just *supply clusters*).

The outline of the rest of the chapter is as follows. Firstly, clustering processes are introduced in Section 3.1 and analysed in deep in sections 3.2 and 3.3 where graph and kernel clustering are developed. This kind of solutions makes it possible to apply the semi-supervised algorithm presented in Section 3.4 to divide a WSN into hydraulic sectors. An experimental process is developed in Section 3.5 in order to check the performance of the proposed methodology. The conclusions, found in Section 3.6, close this chapter.

3.1 Clustering processes

Clustering (or cluster analysis) is related to grouping or segmenting a collection of objects into subsets or 'clusters', such that those within each cluster are more closely related to one another than objects assigned to different clusters [Hastie *et al.* (2001)]. A database object can be described by a set of measurements, or by its relation to other objects. Thus, items within a cluster are more 'similar' to each other than they are to items in the other clusters. Central to all of the goals of cluster analysis is the notion of degree of similarity (or dissimilarity) between the individual objects being clustered because clustering methods attempt to group objects based on the definition of similarity supplied to them. This notion of similarity can be expressed in very different ways, according to the purpose of the study, to domain-specific assumptions and to prior knowledge of the problem [Grira *et al.* (2004)]. For instance, it is possible

to understand this proximity as both a distance between objects and as the fact of containing the same underlying data structure [Schaeffer (2007)]. In a dynamic system, similarities can be calculated in different stages, leading to a class of evolving clustering procedures [Gibert *et al.* (2010)], which is able to integrate the results from each step into a unique global model.

Among others, clustering algorithms can be classified into two categories: partitional and hierarchical algorithms [Fung (2001)]. Most partitional clustering algorithms assume an a priori number of clusters, k, and a partition of the data set into k clusters. To get the correct partition, an objective function must be formulated that measures how good a partition is with respect to the data set. We will describe later the popular K-means algorithm, including the problem of choosing the number of clusters. Hierarchical clustering algorithms transform a proximity data set into a tree-like structure. The main drawbacks of these algorithms are their high computational cost and that they always suffer from the problem of not knowing where to prune the generated tree.

In real-life problems very large data sets containing variables of several types are typically found. This requires for a clustering algorithm to be scalable and capable of handling different attribute types. Classical methods are not the answer: for example, PAM (Partitioning Around Medoids*) algorithm [Kaufman & Rousseeauw (1990)] can handle various attribute types but is not efficient with large data sets. In contrast, K-means algorithms [Hartigan & Wong (1979), Likas *et al.* (2003)] can handle large data sets but deal with only data sets formed from interval-scaled variables. CLARA (Clustering Large Applications) algorithm [Kaufman & Rousseeauw (1990)] is a combination of a sampling approach and the PAM algorithm. Instead of finding medoids, each of which is the most centered object in a cluster for the entire data set, CLARA draws a sample from the data set and uses the PAM algorithm to select an optimal set of medoids from the sample [Wei *et al.* (2003)]. From another point of view, Gibert & Cortés (1997) proposed approaching the problem of working with heterogeneous information defining a family of metrics to measure distances that combine qualitative and quantitative variables. But the specific problem of dividing WSN into clusters also

*Medoid is similar in concept to mean or centroid, but being always a member of the data-set. In contrast, we will understand centroid as the geometric centre of points, which does not need to coincide with any point of the data-set. This last is usually associated with data in a continuous space.

needs handling graph data. Hence, in this thesis we propose working on both vector-based and graph-based data sets. This is the line proposed by Kamvar *et al.* (2003) with their results about spectral clustering and followed by Kulis *et al.* (2005), working with semi-supervised graph clustering. Approaching these two main ideas, we will try to adapt a suitable process to divide WSN into supply clusters [Herrera *et al.* (2010a)].

3.1.1 K-means algorithm

K-means and its variations (as proposed by Frahling & Sohler (2006) or Elkan (2003), among others) is an iterative descending algorithm in which all variables are of the quantitative type, and the squared Euclidean distance is chosen as a dissimilarity measure. Simplicity and speed obtaining useful results are the keys to the popularity of K-means. The algorithm (see details in Table 3.1 - source: Hastie *et al.* (2001) -) starts selecting k points as initial centroids. After points are assigned to a centroid, the centroid is then updated. Next, points are assigned to the updated centroids, and the centroids are updated again. The process finalises when there is no changes.

Table 3.1: A basic version of K-means algorithm

K-means clustering algorithm
1. For a given cluster assignment C, the total cluster variance is minimised with respect to $\{m_1, \ldots, m_k\}$ yielding the means of the currently assigned clusters.
2. Given a current set of means $\{m_1, \ldots, m_k\}$, the total cluster variance is minimised by assigning each observation to the closest (current) cluster mean. That is, $$C(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}} \|x_i - m_k\|^2$$
3. Steps 1 and 2 are iterated until the assignments do not change.

All these algorithms have a number of limitations with respect to finding different types of clusters. In particular, K-means has difficulty detecting the 'natural' clusters, when clusters have non-spherical shapes or widely different sizes or densities. Thus, this process does not work with categorical or graph data. However, we will consider some modifications that will be able to handle all type of data, such as the case of WSN

problem. Section 3.3 introduces the transform spaces where we will successfully apply these K-means modifications on the algorithms of spectral [Ng *et al.* (2001)] and semi-supervised [Kulis *et al.* (2005)] clustering. This will be more detailed in subsections 3.2.4 and 3.4.1.

3.1.2 Cluster evaluation

Cluster evaluation is not a well-developed or commonly used part of cluster analysis. There are a number of different types of clusters (in some sense, each clustering algorithm defines its own type of cluster), and it may seem that each situation might require a different evaluation measure. For instance, K-means clusters might be evaluated in terms of the SSE (sum of square errors), but this may not be used for other types of clustering. In general, measurements of cluster validity are often further divided into two classes: measurements of *cluster cohesion* (compactness, tightness), which determine how closely related the objects in a cluster are, and measurements of *cluster separation* (isolation), which determine how distinct or well-separated a cluster is from other clusters.

A classical criterion to cluster validation is based on the fundamental matrix equation: $T = W + B$, where W and B are the within-cluster and between-cluster variation, respectively [Everitt *et al.* (1988)]. T is, then, the total scatter matrix. From this point of view, the ideal form of T is a matrix built with a *small* W and a *large* B, so that the distances within the clusters are small compared with distances between clusters medoids. Then, an intuitive procedure for choosing clusters is to minimise the "size" of W and/or maximise B. Following this approach, McGregor *et al.* (2004) have developed new methodologies for validation results based on W. Barbará *et al.* (2002) have worked with entropy based measures for categorical data clustering.

But we can also evaluate the objects within a cluster in terms of their contribution to the overall cohesion or separation of the cluster. Objects that contribute more to the cohesion and separation are near the "interior" of the cluster. Those objects for which the opposite is true are probably near the 'edge' of the cluster (also called 'boundary' in Herrera *et al.* (2010b), where the authors raised a sensitivity analysis, of each object membership, based on this idea). Next, we are going to consider a cluster evaluation measure that uses an approach based on these ideas to evaluate points, clusters, and the entire set of clusters.

3.1.2.1 Analysis of the silhouette

The method of silhouette, introduced by Rousseeuw (1987), is a cluster validation and interpretation process which combines both cohesion and separation criteria. Silhouettes offer the advantage that they only depend on the current partition of clusters and not on the specific clustering algorithm. Then, in order to construct silhouettes, we just need the partition previously obtained and the data-set of proximities (or similarities) between objects. The next step combines these numbers into a plot (see Figure 3.1 for instance).

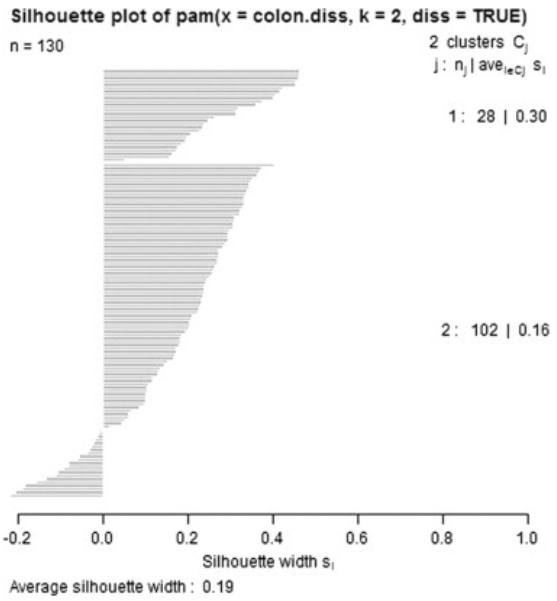

Figure 3.1: Silhouette plot of a clustering partition - Source: Herrera *et al.* (2009a)

The following steps explain how to compute the silhouette of the object i (where s_i is its associated measure of dissimilarity):

1. If cluster A contains other objects apart from i, let a_i be the average distance of i to all other objects in its cluster;

2. Consider a cluster C different from A and compute $d(i, C)$, the average dissimilarity of i to all objects of C. After computing $d(i, C)$ for all clusters $C \neq A$, seek the smallest one: $b_i = \min_{C \neq A}\{d(i, C)\}$;

3. For the object i, the silhouette coefficient is: $s_i = (b_i - a_i)/max(a_i, b_i)$.

The value of the silhouette coefficient can vary between -1 and 1. A negative value corresponds to a case in which a_i, the average distance to points in the cluster, is greater than b_i, the minimum average distance to points in another cluster. We want the silhouette coefficient to be positive ($a_i < b_i$), and a_i to be as close to 0 as possible, since this coefficient assumes its maximum value of 1 when $a_i = 0$. Based on this individual silhouette evaluation we are able to carry out sensitivity analysis of cluster configuration [Herrera *et al.* (2010b)]. This may propose a new membership criterion on some points, allowing performance with different clustering algorithms. Another possibly for applying these results is part of a boosting methodology [Herrera *et al.* (2010c)]. So, it will be easier to take into account poorly represented objects in their re-sampling associated process, improving the final cluster configuration. But we also can compute the average silhouette coefficient of a cluster by simply taking the average of the silhouette coefficients of the points belonging to the cluster. An overall measurement of the goodness of a clustering can be obtained by computing the average silhouette coefficient of all points. This will be the criterion followed in the current work.

3.2 Graph clustering and water supply network data

The starting point for creating DMAs in a WSN is to take into account all the available information of the network. First of all, a WSN must be considered as a particular graph where the edges are pipes and the nodes are consumption points. This fundamental information constitutes the structure of the graph. It includes geographical and connectivity information. In addition, the elevation of the nodes in the graph and the pipe diameter may be known by simply using the EPANET output [Rossman (2000)]. Usually, water companies have another supplemental and important information about the water demand in the consumption nodes.

3.2.1 Review of some graph theory topics

A graph G is a pair $G = (V, E)$, where V is the set of vertices and E contains the edges of the graph. In an undirected graph, E is an unordered pair of vertices (v, w). A graph definition is completed with $m = |E|$, the *size* of the graph and $n = |V|$ the

number of vertices (also called *order* of the graph). In the case of a weighted graph, which we will try later, a weight function $\omega : E \to \mathbb{R}$, should be defined.

If $(v, u) \in E$, we say that v is a neighbour of u. The set of neighbours for a given vertex, v, is called the *neighbourhood* of v and is denoted by $\Gamma(v)$.

The *adjacency* matrix A_G of a given graph $G = (V, E)$ of order n is an $n \times n$ matrix $A_G = (a^G_{v,u})$, where

$$a^G_{v,u} = \begin{cases} 1, & \text{if (v, u)} \in E \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

The number of edges incident on a given vertex v is the *degree* of v and is denoted by $deg(v)$. The diagonal degree matrix of a graph $G = (V, E)$ is:

$$D = \begin{pmatrix} deg(v_1) & 0 & \dots & 0 & 0 \\ 0 & deg(v_2) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & deg(v_{n-1}) & 0 \\ 0 & 0 & \dots & 0 & deg(v_n) \end{pmatrix}. \tag{3.2}$$

A partition of the vertices V of a graph $G = (V, E)$ into two non-empty sets S and $V \backslash S$ is called a *cut* and is denoted by $(S, V \backslash S)$. The *cut size* is the number of edges that connect vertices in S to vertices in $V \backslash S$:

$$c(S, V \backslash S) = |(u, v) \in E : u \in S, v \in V \backslash S|. \tag{3.3}$$

We denote by

$$deg(S) = \sum_{v \in S} deg(v), \tag{3.4}$$

the sum of degrees in a cut S. Note that in the presence of edge weights, the cut size is generally redefined as the *sum* of weights of the edges crossing the cut instead of using simply the number of edges that cross it.

A *path* from v to u in a graph $G = (V, E)$ is a sequence of edges in E starting at vertex $v_0 = v$ and ending at vertex $v_{k+1} = u$:

$$\{v, v_1\}, \{v_1, v_2\}, \dots, \{v_{k-1}, v_k\}, \{v_k, u\}. \tag{3.5}$$

The *length* of a path is the number of edges on it, and the distance between v and u is the length of the *shortest path* connecting them in G. A graph is *connected* if there

exist paths between all pairs of vertices. If there are vertices that cannot be reached from others, the graph is *disconnected*. The minimum number of edges that would need to be removed from G in order to make it disconnected is the *edge-connectivity* of the graph. A *cycle* is a simple path that begins and ends at the same vertex. A graph that contains no cycle is acyclic and is also called a *forest*. A connected forest is called a *tree*.

A *subgraph* G^S of $G = (V, E)$ is composed of a set of vertices $S \subseteq V$ and a set of edges $E_S \subseteq E$ such that $(v, u) \in E_S$ implies $v, u \in S$. An *induced subgraph* of $G = (V, E)$ is the graph with the vertex set $S \subseteq V$ with an edge set $E(S)$ that includes all such edges (v, u) in E with both vertices included in S. An induced subgraph is a *clique* if $(v, u) \in E \ \forall v, u \in S$.

There are more concepts and definitions associated with graph theory. We will introduce them throughout the current thesis. But, to end this introduction we would like to remark one of the central concept of this work, since it is related to graph theory. This is the notion of *spectrum* of a graph (cf. Subsection 3.2.4). The spectrum of a graph $G = (V, E)$ is defined as the list of eigenvalues (together with their multiplicities) of its adjacency matrix, A_G. It is often more convenient to study the eigenvalues of the *Laplacian matrix*: $L = I - A_G$ than those A_G itself (I is an $n \times n$ identity matrix). For a weighted graph, the expression for the Laplacian matrix will be:

$$L = D - A_G, \tag{3.6}$$

being D the diagonal degree matrix defined on Equation 3.2.

The *normalised Laplacian* is defined as:

$$\mathcal{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A_G D^{-1/2}. \tag{3.7}$$

As these matrices are symmetrical, their eigenvalues are real numbers. We note that the normalised Laplacian is also a positive semidefinite matrix [Cavers (2010)]. Using the normalised Laplacian is convenient as all the eigenvalues of $\mathcal{L}$ fall in the interval [0, 2]. The smallest eigenvalue is always zero, as the matrix is singular, and the corresponding eigenvector is simply a vector with each element being the square root of the degree of the corresponding vertex. Thus, we will be interested in the eigenvector associated with the second-smallest eigenvalue of the Laplacian matrix (so-called *Fiedler vector*), since

it plays a special role in spectral clustering, among other applications (cf. Subsection 3.2.4).

3.2.2 Graph clustering process and spectral methods

Graph clustering is the task of grouping the vertices of the graph into clusters taking into consideration the edge structure of the graph in such a way that there should be many edges within each cluster and relatively few between the clusters. Clustering is usually defined in terms of weighted, undirected graphs, where weights correspond to either similarity scores or distances. Unweighted graphs can be viewed as a special case of weighted graphs, where each edge has a weight of one. Dealing with sets of k-dimensional vectors, a set can be viewed as a complete, weighted graph, where each vector is a vertex, and the weight of each edge is the distance between the two vectors it connects. About this distance, it is usually Euclidean, but other choices are possible (e.g. correlation).

Unfortunately, the popular K-means algorithm is NP-hard when it is adapted to performance on a graph (even for the case of $k = 2$) [Aloise *et al.* (2009); Dhillon *et al.* (2004)]. But there are various methods to approach a solution and we highlight two of them:

- The *Markov Cluster algorithm* (MCL), which simulates flow *expansions* (to contact new node neighbours) and *contractions* (only the favourites neighbours are added to the cluster) on the graph [van Dongen (2000)]. For an unweighted graph, when one moves from one vertex to another choosing a neighbouring vertex uniformly at random, the transition matrix that results is the normalised adjacency matrix $D^{-1}A_G$ of the graph G. Then, it is possible to derive, from MCL, an expression for random walks on a graph in terms of the spectrum of the normalised Laplacian matrix of the graph [Orponen *et al.* (2008)]. Thus, MCL can be converted into a special case of spectral clustering.

- The *Spectral methods* are based on eigenvalues and eigenvectors of a block-diagonal matrix conveniently associated with the graph. They understand a graph as a collection of k disjoint cliques. Their normalised Laplacian is a block-diagonal matrix that has eigenvalue zero with multiplicity k and the corresponding eigenvectors serve as indicator functions of membership in the corresponding cliques.

Any deviations caused by introducing edges between the cliques causes k-1 of the k eigenvalues that were zero to become slightly larger than zero and also the corresponding eigenvectors change. This phenomenon is the basis of spectral clustering, where an eigenvector or a combination of several eigenvectors is used as a vertex similarity measure for computing the clusters.

Both methodologies find in spectral clustering (cf. Subsection 3.2.4) a main process to achieve their purposes of graph division.

3.2.3 Graphs and water supply network data

This thesis introduces a WSN as a special case of an undirected, weighted graph*. As a flow network, it has one or more *sources* and a certain number of *sinks*. In this case, sources correspond to water tanks and reservoirs. From them, water travels to sink nodes, which correspond to consumption points where the system tries to satisfy some water demand. Water travels through paths defined by links between nodes, or pipes between consumption points. Links are weighted by various pipe properties, such as their diameter.

Picking clustering up, we consider now the following equivalence relation: "WSN - flow networks - undirected, weighted graphs". Based on this equivalences we claim that *cut methods* are suitable for clustering configuration. For instance, dividing a graph by a minimal cut criterion is equivalent to maximising the flow on the same graph, viewed as a network; this corresponds to the network viewpoint. In addition, this minimal number of cut edges matches with the necessary number of cut valves splitting the WSN into sectors. Thus, this minimal cut will result in both economic (we minimise the financial investment in valves, as well as their maintenance and management) and energy savings (we have lower pumping needs when compared with the unsectorised network). From another point of view, knowledge of the pipes that offer this minimum cut provides an important measurement of the reliability of the WSN, assessing the consequences of the minimum number of links that may fail at most. Next we detail three cut algorithms with some modifications about this approach.

*Graphs are structures formed by a set of vertices and a set of edges. But, from the point of view of network analysis, we also use the terminology *nodes* for vertices and *links* to edges that are connections between pairs of vertices.

- *Min-cut*: The simplest and most direct way to construct a partition of the graph is to solve the min-cut problem [Wagner & Wagner (1993)]. Given the graph $G = (V, E)$ with a weighted adjacency matrix W and given two not necessarily disjoint sets S, $T \in V$. We define $W(S,T) = \sum_{i \in S, j \in T} w_{i,j}$ and $\overline{S}$ for the complement of S. Then, for k subsets, the min-cut approach consists in choosing a partition $A_1, \ldots, A_k$ such that minimises 3.8:

$$cut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} W(S_i, \overline{S}_i). \tag{3.8}$$

  The main drawback is that in many cases, the solution of this min-cut simply separates one individual vertex from the rest of the graph. This forces us to introduce size balancing conditions, which turn the simple min-cut into a NP-hard problem.

- *Ratio-cut*: One way to find a solution to the balancing min-cut clusters is the Ratio-cut algorithm [Hagen & Kahng (1992), Chan *et al.* (1994)]. This algorithm proposes a metric to allow freedom to meet partitions, minimising Equation 3.9. The numerator captures the min-cut criterion, while the denominator favors a balanced partition.

$$Rcut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(S_i, \overline{S}_i)}{|S_i|} = \sum_{i=1}^{k} \frac{cut(S_i, \overline{S}_i)}{|S_i|}. \tag{3.9}$$

  $|S_i|$ is the size of subgraph S_i in relation to its number of vertices.

- *Normalised-cut*: Another solution to min-cut drawbacks in the relative size of the clusters is the so called Normalised-cut [Shi & Malik (2000)]. It is based on the same fundamentals as Ratio-cut but takes as its denominator the volume in terms of the weights of the edges included on the involved subgraph. Thus, the proposal is to minimise Equation 3.10.

$$Ncut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(S_i, \overline{S}_i)}{vol(S_i)} = \sum_{i=1}^{k} \frac{cut(S_i, \overline{S}_i)}{vol(S_i)}. \tag{3.10}$$

Spectral clustering is a way to solve relaxed versions of these problems. We also will see that relaxing Normalised-cut leads to normalised spectral clustering, while relaxing Ratio-cut leads to unnormalised spectral clustering [von Luxburg (2007)].

3.2.4 Spectral clustering

Spectral clustering is a powerful technique in data analysis that has found increasing support and application in many areas. It improves the straightforward application of K-means, working well in non-convex spaces and taking into account the possible graph structure under study. Spectral clustering uses information obtained from computing the eigenvalues and eigenvectors of their Laplacian matrices for partitioning of graphs (cf. Subsection 3.2.1). The eigenvectors corresponding to the k smallest eigenvalues of this Laplacian are specially important, because they have some desirable properties. Several applications are based on them. For instance, these Laplacian eigenvalues govern the kinematic behaviour of a liquid flowing through a system of communicating pipes. Specifically, the second smallest eigenvalue determines the basic behaviour of the flow [Maas (1987)]. We could emphasise other physical and chemical applications [Mohar (1991)]. All of them are found on the special characteristics of these eigenvalues and their relation with the structure of the graphs. Specifically, the second smallest Laplacian eigenvalue, λ_2, measures the algebraic connectivity of the graph. Thus, a graph is connected if and only if its λ_2 is different from zero [de Abreu (2007)]. Moreover, $\lambda_1 = 0$ always and the corresponding eigenvector is $(1, 1, \ldots, 1)^t$. The multiplicity of 0 as an eigenvalue of L is equal to the number of components of G.

Besides these results, we have that the second smallest eigenvalue, λ_2, is related to the diameter and mean distance of a graph. This, together with its measure of graph connectivity, makes that one of the main approaches on spectral clustering has been based on the properties of λ_2. Thus, the second eigenvector of the graph's Laplacian is used to define a cut, which solves a relaxation of the NP-hard problem of graph partitioning. This method can be recursively applied to find k clusters [Spielman & Teng (1996)]. But, it also can be straightforwardly applied by using more eigenvectors (corresponding to smallest eigenvalues). Weiss (1999), Meila & Shi (2001) and Ng *et al.* (2001) propose algorithms that use k eigenvectors simultaneously to make clusters. We will use here the following version of a spectral clustering algorithm. It includes a slight modification regarding the one proposed by Ng *et al.* (2001)*.

*Ng *et al.* (2001) worked with $\mathcal{L}_{sym} = I - \mathcal{L}$. The analysis is parallel to the current one, but changing the λ_i eigenvalues to $(1 - \lambda_i)$. The target eigenvectors are the same, but now they should be associated with largest eigenvalues.

3.2.4.1 Spectral clustering algorithm

Given a set of points $X = \{x_1, ..., x_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

1. Build the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$ if $i \neq j$ and $A_{ii} = 0$.

2. Define D to be the degree diagonal matrix whose (i, i)-element is the sum of the entries in A's i-th row, and build the matrix $\mathcal{L} = I - D^{-1/2}AD^{-1/2}$.

3. Find $u_1, u_2, \ldots, u_k$ the k smallest eigenvectors of $\mathcal{L}$, and form the matrix $U = [u_1\ u_2 \ldots u_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.

4. Form the matrix U^* from U by renormalising each of U's rows to have unit length.

5. Treating each row of U^* as a point in $\mathbb{R}^k$, cluster them into k clusters.

6. Finally, assign the original point x_i to cluster j if and only if row i of the matrix U^* was assigned to cluster j.

The scaling parameter of step 1, σ^2, controls how rapidly the affinity A_{ij} falls within the distance between x_i and x_j. In step 5 we can apply k-means and then obtain an improvement of its straightforward implementation, avoiding convexity complications and running in a better computational way. The overall process is shown in Figure 3.2.

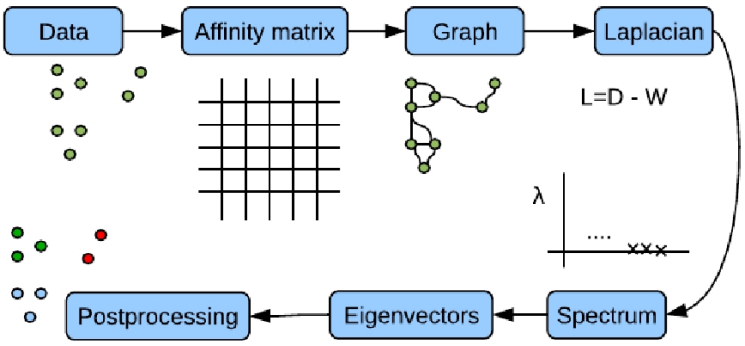

Figure 3.2: The process of spectral clustering - Source: Vejmelka (2009)

Summarising, the k eigenvectors, which correspond to the k smallest eigenvalues of the affinity matrix's Laplacian, are used to form an $n \times k$ matrix U^* where each column is normalised to unit length. Treating each row of this matrix as a data point, the

algorithm of k-means is finally used to cluster the points. Now, our goal is to adapt this graphical process to the data information of a WSN. This can be done by building a kernel weight matrix to represent all available WSN characteristics.

3.3 Graph clustering on kernel spaces

In the last section, spectral methods have been proposed as efficient processes for clustering WSN. They are able to work with undirected weighted graphs, proposing suitable relaxations to approach graph partition algorithms. The criterion to divide the WSN would be a kernelisation of the Laplacian matrix associated with the WSN graph, taking advantage of its structure [Gärtner (2003), Kashima *et al.* (2003)]. The solution may be enriched by adding hydraulic data to this kernel matrix. Thus, we are able to establish a complete framework where all possible information sources would be exploited properly.

3.3.1 Introduction to kernel methods

Kernel-based learning methods are a class of algorithms for pattern analysis [Shawe-Taylor & Cristianini (2006)]. They provide a powerful way of detecting nonlinear relations using well-understood linear algorithms in an appropriate space. Thus, each problem is approached by mapping the data, $\mathcal{X}$, into (a high dimensional) feature space, $\mathbb{H}$, defined by a *kernel function* [Hofmann *et al.* (2008); Karatzoglou (2006); Schölkopf & Smola (2002)]. The benefit of this process is that for nonlinear feature maps $\phi : \mathcal{X} \to \mathbb{H}$, we are able to produce nonlinear learning functions based on a linear approach. Furthermore, these methods enable us to work by inner products $k(x, x') = \phi(x) \cdot \phi(x') = \langle \phi(x), \phi(x') \rangle$. This is computationally simpler than explicitly working in the feature space. The function k is called the *kernel function*, and the above described approach is the so-called *kernel trick*. The kernel trick is based on Mercer's Theorem [Mercer (1909)] and was first referred by Aizerman *et al.* (1964); but their use have been a surge thanks to the most recent and diverse applications in pattern recognition algorithms developed by Schölkopf & Smola (2002) and Shawe-Taylor & Cristianini (2006), among others.

Expanding the concept of kernel functions, we can introduce the definition of a *kernel matrix* (also called Gram matrix). Given a kernel function, k, with inputs

$x_1, \ldots, x_n \in \mathcal{X}$, the $n \times n$ matrix of elements $k(x_i, x_j)$ is called *kernel matrix*. An important property of kernel matrices is that they are positive definite. That is, for any $c_1 = \cdots = c_n \in \mathbb{R}$ Equation 3.11 is satisfied:

$$\sum_{i,j} c_i c_j k(x_i, x_j) \geq 0. \tag{3.11}$$

Moreover, kernels are positive definite for any choice of ϕ, as we can see in Equation 3.12, and are therefore regarded as generalised inner products.

$$\sum_{i,j} c_i c_j \langle \phi(x_i)\phi(x_j) \rangle = \left\langle \sum_i c_i \phi(x_i), \sum_j c_j \phi(x_j) \right\rangle \geq 0. \tag{3.12}$$

3.3.1.1 Other properties of kernels

Given any space of samples, $\mathcal{X}$, and kernels $k_1(.,.)$ and $k_2(.,.)$ over $\mathcal{X}$. Then, $k(.,.)$ is a kernel in the next cases [Shawe-Taylor & Cristianini (2006)]:

i. $k(x, y) = k_1(x, y) + k_2(x, y)$

ii. $k(x, y) = ak_1(x, y)$, where $a > 0$

iii. $k(x, y) = f(x) \cdot f(y)$, for any function f on x

iv. $k(x, y) = k_1(x, y) \cdot k_2(x, y)$

v. $k(x, y) = p(k_1(x, y))$, where p is a polynomial with positive coefficients

vi. $k(x, y) = exp(k_1(x, y))$

vii. $k(x, y) = exp(-||x - y||^2/2\sigma^2)$

viii. $k(x, y) = x'By$, where B is a symmetric positive semi-definite matrix

3.3.1.2 Most common kernels

Due to the growing interest in kernel methods a large number of kernels have been conceived. We just highlight a couple of them, but there is a large number of different possibilities [Hofmann *et al.* (2008)].

- Polynomial kernel: They are of the type $k(x, x') =< x, x' >^p$, with $p \in \mathbb{N}$. The corresponding feature map [Poggio (1975)] can be calculated by:

$$< x, x' >= \left\langle \sum_{j=1}^{d} x_j x'_j \right\rangle^p = \sum_{j \in [d]^p} [x]_{j1} \dots [x]_{jp} \cdot [x']_{j1} \dots [x']_{jp} = \left\langle C_p(x), C_p(x') \right\rangle, \tag{3.13}$$

  where C_p maps $x \in \mathbb{R}^d$ to the vector $C_p(x)$ whose entries are all possible pth degree ordered products of the entries of x (note that $[d]$ is used as shorthand for $\{1, \dots, d\}$). A variation of this polynomial kernel is the inhomogeneous polynomial function $k(x, x') = (< x, x' > +c)^p$, where $c \geq 0$.

- Radial Basis Function (RBF) kernel: Most of these kernels are of general purpose and provide good performance on many learning problems. They can be written in the form:

$$k(x, x') = f(d(x, x')), \tag{3.14}$$

  where $d(x, x')$ is a metric on $\mathcal{X}$ and f is a function in $\mathbb{R}$. Usually the metric arises from the inner product $d(x, x') = ||x - x'||$ and in the case of RBF kernels are also translation invariant. The popular Gaussian kernel $k(x, x') = exp(-\sigma||x - x'||^2)$ [Vapnik (1998)] is an RBF kernel.

3.4 The proposed semi-supervised clustering algorithm

The proposed clustering algorithm is based on affinity graph matrices and their transformation into kernel matrices. This allows us to carry out the related kernel abstraction of the essential characteristics of the WSNs. Next, spectral clustering techniques are applied to this new matrix [von Luxburg (2007)]. But constraints occurrence is natural for graphs, and a certain amount of knowledge about WSN structure is available. This raises the idea of using graph-based semi-supervised learning methods [Kulis *et al.* (2005); Zhu *et al.* (2006)], which we introduce now.

3.4.1 Semi-supervised clustering

Semi-supervised learning (SSL) is a class of machine learning techniques that make use of both labelled and unlabelled data for training their associated algorithms (typically a small amount of labelled data with a large amount of unlabelled data). Thus, it

represents a halfway between supervised and unsupervised learning [Chapelle *et al.* (2006)], working with unlabelled data, but providing some supervised information. More specifically, in traditional clustering algorithms only unlabelled data is used to generate clusterings (unsupervised learning). But usually, some background knowledge about the cluster structure is available when we work on real-world problems (see a scheme of supervised, unsupervised and semi-supervised clustering in Figure 3.3).

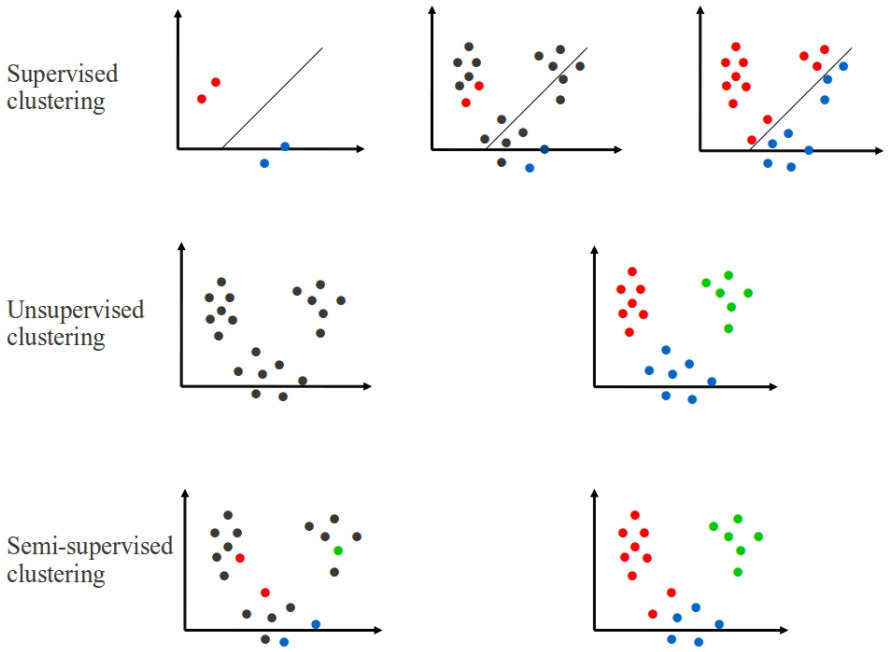

Figure 3.3: Naive example of semi-supervised clustering - comparation with supervised and unsupervised methods

Semi-supervised clustering [Kulis *et al.* (2009)] proposes incorporating prior information about clusters into the algorithm, in order to improve the clustering results. Research on semi-supervised clustering considers integrating this information (limited supervision) into the clustering process by both labelled points [Basu *et al.* (2002); Sinkkonen & Kaski (2002) and Chapter 5 of this thesis] and constraints [Basu *et al.* (2004a,b); Kamvar *et al.* (2003)]. In the development of the current chapter, we assume that knowledge comes in the form of pairwise *must-link* and *cannot-link* constraints (depending if a pair of points should belong to the same cluster or not). Typically, the

constraints are 'soft', that is, clusterings that violate them are undesirable (and penalty weights are specified, as in Kamvar *et al.* (2003)) but not prohibited. To this purpose, different components of the clustering algorithm may be adapted, such as the initialisation scheme, and the distance measure or the objective function [Handl & Knowles (2006)].

3.4.2 Kernel-based semi-supervised clustering

The existence of constraints is something natural for graphs. Pairwise relationships (representing these constraints) are explicitly captured via edges in a graph Kulis *et al.* (2005). This assumption is necessary in the problem of dividing the WSN into hydraulic zones. In this case, we should assure that each DMA is supplied by one or more water sources, such as tanks or reservoirs. This hydraulic feature should be translated as constraint implementation in the partition algorithm. However, most semi-supervised clustering processes with pairwise constraints assume that the input is in the form of data vectors. To deal with it, Kamvar *et al.* (2003) proposed the next spectral approach (similar to the one shown in Subsection 3.2.4.1, but with some changes in step 2). In any case, the direct application of the spectral clustering algorithm does not ensure its hydraulic feasibility.

1. Form the affinity matrix A: the entries of A are assumed to be normalised between 0 and 1.
2. For all points i, j that have a must-link constraint, set $A_{ij} = 1$; for all points i, j that have a cannot-link constraint, set $A_{ij} = 0$.
3. Re-normalise the matrix using additive normalisation: $N = \frac{1}{d_{max}}(A + d_{max}I - D)$
4. Take the top k eigenvectors of A to be the columns of the matrix V, and cluster rows of V.

d_{max} is the maximum row-sum of A and D is the degree matrix. Here, for a must-link constraint, if A_{ij} is the similarity between x_i and x_j, we set $W_{ij} = 1 - A_{ij}$ (and hence the corresponding value in $A + W$ is 1). Similarly, for cannot-links, we set $W_{ij} = -A_{ij}$ (and hence the corresponding value in $A + W$ is 0). With this particular choice of constraint weights, the matrix $A + W$ is identical to the matrix from spectral clustering before additive normalisation.

3.4.3 The proposed algorithm

The starting point of our proposal to create supply clusters into a WSN is to take into account all the available information of the network. This information will be available as input matrices. Next we see the construction and subsequent treatment of them.

A WSN must be considered as a particular graph including geographical and connectivity information (cf. Subsection 3.2.3). We start by building the affinity matrix associated with a WSN. Then, the next correspondences must be considered. First, graph nodes are the consumption points of the WSN and their weights are their water demands. Second, the graph edges are the pipes of the WSN and their weights are the diameters of these pipes. Using this information, an affinity matrix of the graph adapted to the needs of a WSN can be obtained. Other dissimilarity matrices, using different constraints and information about a water supply, can also be built. In particular, geographic information of the consumption nodes can be added by using the symmetric matrix that contains the distances between the nodes. Another possibility is to construct the dissimilarity matrix using the elevations of these nodes. Of course, it is possible to add as many matrices as there are information categories available.

Each supply cluster is really a small network supplied by one or at most two sources. To enforce it, this characteristic is represented in the form of must-link or cannot-link constraints in the graph of the WSN. As a result, all the sources of the network must be linked to their neighboring nodes. On the other hand, a closed valve will be represented by a cannot-link constraint (cf. Subsection 3.4.1). The isolation of hydraulic zones should be performed using the lowest possible number of valves; this number will depend on the fitness function of the clustering algorithm and can be obtained by applying some 'min-cut' algorithm (cf. Subsection 3.2.3).

As the entries of the affinity matrix are normalised between 0 and 1, we assign the following penalty weights: $A_{ij} = 1 \ \forall i, j$ that have a must-link constraint and $A_{ij} = 0 \ \forall i, j$ that have a cannot-link constraint. At this stage, we can continue with the kernel spectral clustering algorithm, looking for the kernel associated with the affinity matrix and taking the 'top' k eigenvectors (in the sense explained in Subsection 3.2.4) to be the columns of the matrix to cluster.

After building the matrices of input information, they are transformed into so-called kernel matrices. First, data are scaled between 0 and 1. Then, a diagonal of 1's

is plugged into the diagonal of each matrix. Next, matrices are mirrored through their diagonals to make them symmetric (the input matrices are triangular). There are two key properties that a kernel function must meet (cf. Subsection 3.3.1). Firstly, it should capture the measure of similarity approximate to the particular task and domain, and, secondly, its evaluation should require significantly less computation than it would be needed in an explicit evaluation of a corresponding feature mapping.

Furthermore, as the sum of kernel matrices is another kernel matrix, we propose to build an accumulative matrix, which is the weighted sum of the normalised dissimilarities in the different characteristics of the data (see Equation 3.15). Thus, to perform the partition, we work with so many dissimilarity matrices (transformed into kernel matrices) as variables are involved. In the case that a previous DMA exists, it will be quantified in another matrix of distances. In this way, we merge different information by adding other kernel matrices. This information contains values of the inputs under study, and the must-link and cannot-link constraints (cf. Subsection 3.4.1).This matrix will be transformed into a kernel matrix and will be treated as an additional input (suitably weighted against all the other matrices). Then, the desired information is combined by using the weighted sum of kernel matrices with graphical and vector information.

In this kernel based spectral clustering algorithm (cf. Subsection 3.2.4.1) we are interested in representing, in addition, the *a priori* information about the graph (i.e., previous DMA designs).

$$K = \lambda_A K_A + (1 - \lambda_A) \sum_{i \in I} \omega_i K_i. \tag{3.15}$$

K is the kernel matrix for clustering, K_A is the kernel matrix related to the affinity graph and $K_i,\ i \in I$, is the matrix associated with the inputs of our interest in the process of building hydraulic zones. Finally, λ_A represents the graph structure importance; and $\omega_i,\ i \in I$, are the weights entering the linear combination. The selection of λ_A is performed by using a cost analysis point of view. But the way to select the weighting of the rest of the matrices in 3.15 could depend on one or more hydraulic objectives (such as demand homogeneity, geographic distances or pipe diameters) in each case and moment. This will add some subjectivity to the problem, which we solve by assigning these weights by an analytic hierarchy process (AHP) criterion [Saaty & Hu

(1998)]. AHP is based on the Perron eigenvector of a well formed matrix of criteria for the evaluation of the different alternatives of one problem. The current AHP task will consist on assigning values to the ω_i, taking into account the relative importance of the variables which are introduced in 3.15. Ho (2008), in his AHP state-of-the-art revision, proposes different AHP applications. Delgado-Galván *et al.* (2010) and Srdjevic (2007) have addressed these procedurals into the Hydraulics area. In the next Section 3.5, we will specify more aspects of the AHP support for our data kernelisation process.

Starting from the information arranged in a suitable kernel matrix we can apply skills of analysis as spectral clustering. The overall process (detailed in these last sections) can be summarised by the algorithm in the Table 3.2. We can add some improvements about this methodology working with the cluster configuration quality.

Table 3.2: Overall kernel based semi-supervised process

Algorithm: water supply clusters by semi-supervised learning
1. abstraction of the water supply network as a graph
2. construction of Laplacian and dissimilarity matrices
3. *plug-in* of hydraulic constraints
4. data transformation into a single kernel matrix
5. calculation of the matrix spectrum
6. k-means for the 'top' eigenvectors
7. cluster re-assigning into the original data
8. hydraulic validation (EPANET)

3.5 Experimental process

In this section we check the proposed methodology with the simple case-study introduced in Chapter 2, Subsection 2.5.2. The aim is to divide the WSN of this experimental case-study into 2 supply clusters (each cluster supplied by at least one tank). To this purpose, the procedure of kernel based semi-supervised clustering algorithm explained in the current chapter and summarised in 3.2 will be applied.

3.5.1 Specifying the data matrices to kernelise

The variables involved in our data analysis are as follows. Firstly, the affinity matrix and the pipe diameters, which will be their weights. By kernelising this matrix, we

obtain our K_A of the Equation 3.15. Other graphical and vector variables will enter the rest of the equation, corresponding to the matrices K_i, for $i \in I$. These variables are node elevations, water demand and geographical coordinates (x and y). Following the Saaty scale (see Table 3.3) and based on our own targets (or hydraulic needs, depending on each case), we build the preference matrix of Table 3.4 to assign the AHP-weights, w_i.

Table 3.3: Saaty numerical scale for pairwise comparisons in AHP

Judgement term	**Saaty scale**
Absolute preference (element i over element j)	9
Very strong preference (i over j)	7
Strong preference (i over j)	5
Weak preference (i over j)	3
Indifference as regards i and j	1
Weak preference (j over i)	1/3
Strong preference (j over i)	1/5
Very strong preference (j over i)	1/7
Absolute preference (j over i)	1/9

Table 3.4: Preference matrix to assign weights by AHP

	elevation	demand	x-coor	y-coor
elevation	1	1/3	1/3	1/3
demand	3	1	3	3
x-coor	3	1/3	1	1
y-coor	3	1/3	1	1

As a result, the priority vector $W = (0.10, 0.42, 0.24, 0.24)^t$ represents the weight to *elevation*, *demand*, *x-coordinate* and *y-coordinate*, respectively. It has a consistency ratio of 0.06, which it is within acceptable limits for this measure (being less than 0.10) [Delgado-Galván *et al.* (2010)]. Then, Equation 3.15 is as follows: $K = \lambda_A K_A + (1 - \lambda_A)\{0.10K_l + 0.42K_d + 0.24K_x + 0.24K_y\}$.

To obtain λ_A we will carry out a cost analysis of their possible values, once we build the clusters. Thus, different values taken on $[0, 1]$ are checked. In the light of the results shown in Figure 3.4, it is possible to conclude that one of the more profitable options is to let $\lambda_A = 0.5$ to run the algorithm.

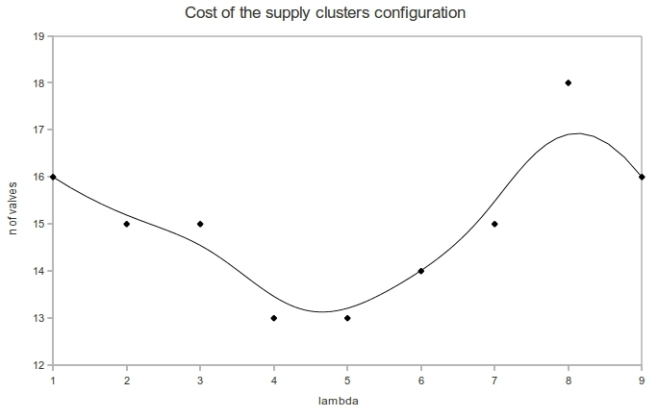

Figure 3.4: Cost of clustering by number of operations needed to isolate the clusters - Case-study divided into two supply clusters

3.5.2 Results

Figure 3.5 (calculated with the R Language [R-Development-Core-Team (2010)] function `specc` of the `kernlab` library [Karatzoglou (2006)] and represented with NetLogo [Wilensky (1999)]) shows the final sector division of the WSN into two supply clusters. This final configuration was successfully simulated in EPANET, thus validating our results.

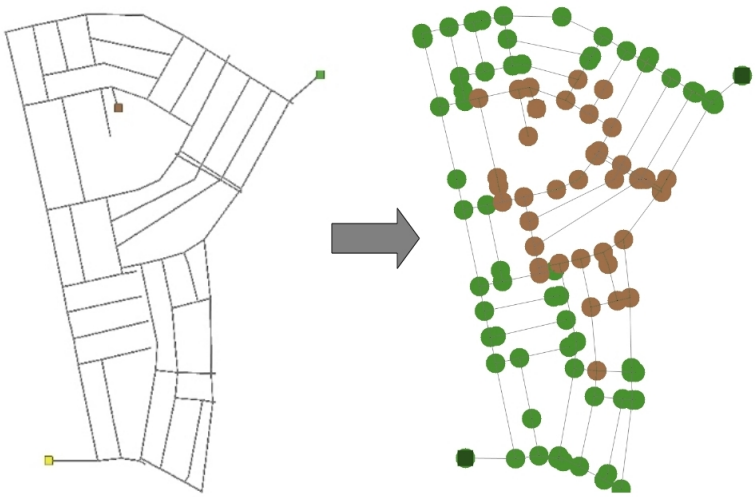

Figure 3.5: Water supply cluster configuration - Case-study divided into two supply clusters

This network is fed by three reservoirs and made out of 132 lines and 104 consump-

tion nodes; its total length is 9 km and the total consumed flowrate amounts to 47 l/s. The clustering process proposes to split this network into two sectors, having 71 and 49 pipes, for a total of 4.6 and 3.5 km of pipes, respectively. For this division, each sector is supplied by at least one tank. It is necessary to close 13 valves to isolate each cluster (see more details of these supply clusters in Table 3.5).

Table 3.5: Description of supply clusters of the case-study

sector	n nodes	n pipes	n sources	avg. elevation	avg. demand
Sector 1	64	71	2	73	42
Sector 2	40	49	1	71	53

3.6 Summary and comments

Classically, a division of a WSN into DMAs aims at improving leakage detection using node elevation, pressure and demand information. In the present work we propose augmenting, or changing, the perspective of this target. This can be done by taking into account different information to be included within criteria for the division of the WSN into hydraulic zones (clusters). Furthermore, one can use the diameter of the pipes and their age, weighting rehabilitation plans, among others. Other point of view considers the use of some index of vulnerability for the pipes, taking into account the effects of hazards in the construction of DMAs.

Compared to other methodologies, which only use graphical or vector information, semi-supervised clustering use both, and in a more efficient and robust way. The flexibility to include different inputs into the study, with different weights, is another improvement of the shown methodology. In addition, different modifications could be included in the clustering algorithm that could be compared on a work bench regarding their ability to build optimal models.

4

Agent-division of water distribution systems into supply clusters

Having introduced the supply cluster paradigm, this chapter tries to address it in a different way, via multi-agent systems [Shoham & Leyton-Brown (2009)]. Multi-agent systems are an approach to building complex distributed applications, such as the one considered here. Thus, the water supply network is decomposed into its elements, which are thought of as certain entities, or agents, that interact with each other, being able to build a network partition. Other authors have addressed water management tasks by a multi-agent based approach. This is the case with Gianetti *et al.* (2005), who used agents to control the physical equipment of a water supply. Maturana *et al.* (2006) followed this research line approaching a criterion to maintain continuity and reliability in the water supply. Cao *et al.* (2007), developed a modified multi-agent genetic algorithm to optimise water-using networks. Izquierdo *et al.* (2009) and Izquierdo *et al.* (2011) built a software environment to simulate district metered area (DMA) partitions in a WSN by a multi-agent metaphor. These works, along with that introduced by Herrera *et al.* (2010b), may be considered as the antecedents of the current chapter in relation to the methodology exposed (implemented) and the water problem approached.

This chapter is structured as follows. In sections 4.1 and 4.2 we present a brief introduction to multi-agent systems. Section 4.3 introduces a WSN agent-abstraction to focus on the sectorisation paradigm from the point of view of a multi-agent system.

Section 4.4 explains the clustering approach to establish a WSN division. Next, we propose an agent-based solution to obtain these supply clusters. In Section 4.5, an experimental study, based on a real network, is successfully developed with interesting results. Finally, Section 4.6 draws conclusions and lists a number of research challenges for further exploration of hydraulic applications of multi-agent systems.

4.1 Intelligent agents

This section tries to approach a (computational) synthetic world where their components can decide for themselves what they should do in order to satisfy their own necessities. These world inhabitants are know as agents and they are able to solve complex problems (with individual and social actions) when their aforementioned necessities meet with their design objectives. Currently, there is a large number of application areas for intelligent agents. In fact, intelligent agents in artificial intelligence are closely related to agents in Economics, and versions of the intelligent agent paradigm are studied in cognitive sciences, Ethics, health sciences*, as well as in many interdisciplinary socio-cognitive modelling and computer social simulations. The contribution of this chapter will be adapting them to achieve a solution to the division of a WSN into supply clusters.

4.1.1 A first approach to intelligent agents

It is not easy to formulate an accurate definition of the term *agent*. The next perhaps is the more repeated definition found in the literature: "An agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives" [Wooldridge & Jennings (1995)]. Figure 4.1 shows how an agent takes sensory input from the environment and produces output actions that affect it. In general, an agent will not have complete control over its environment, only influences it in some sense.

Besides, an agent can interact with others, influencing them and changing their behaviour. Thus, there are some properties that govern agent actions, which they

*We would like to highlight *K4Care* project. It is a Specific Targeted Research Project (STREP) in the thematic area of Information Society Technologies (IST) funded by the European Community under the Sixth Framework Program for Research and Technological Development. http://www.k4care.net

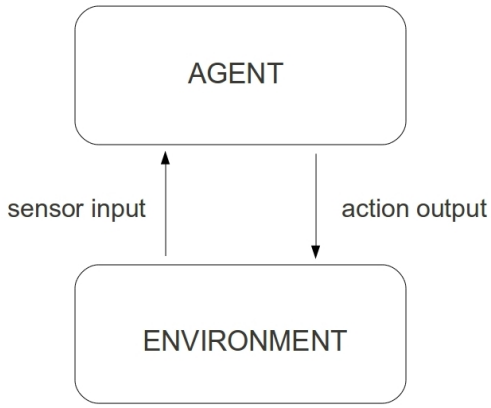

Figure 4.1: An agent in its environment - Source: Wooldridge (2002)

should satisfy [Wooldridge (2000)]:

- Reactivity: agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it, in order to satisfy their design objectives.
- Pro-activeness: agents are able to exhibit goal-directed behaviour by taking the initiative to satisfy their design objectives.
- Social ability: agents are capable of interacting with other agents (and possibly humans) in order to satisfy their design objectives.

These are important properties in practice. The main reason of this is because starting from them, we can build systems or societies of agents (cf. Section 4.2) to improve methodologies that solve complex problems.

4.1.2 Formal definitions of agents and their environment

Formalising the abstract view of agents, we define first the set $A = \{a_1, a_2, \dots\}$ of actions. They will be the result of how environment states, $S = \{s_1, s_2, \dots\}$, influence agents. Then, an agent can be represented as a function $S \to A$. This function maps sequences of environment states to actions in a deterministic way. Also, the responses of the environment to the agent actions will be the non-deterministic result of this action, $a \in A$, and the current state of the environment, $s \in S$, and is formalised as: $S \times A \to \rho(S)$.

We can also represent the interaction of an agent and the environment as a history, h:

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \ldots \xrightarrow{a_{u-1}} s_u \xrightarrow{a_u} \ldots$$

where s_0 is the initial state of the environment, a_u is the u^{th} action that the agent choose, and s_u is the u^{th} environment state.

These sets of agent behaviours can be complicated by the interaction with the current internal state of the agent, which will affect the next action. Thus, agents' decision-making will be influenced by their history, their environment evolution and their current internal state. We will consider four types of agents [Wooldridge (2002)]:

- Logic based agents: agents take their internal state by logical deduction.
- Reactive agents: decision making is a direct consequence of their environment and the current state of the agent.
- Belief-desire-intention agents: agents are influenced by their own beliefs, desires and intentions.
- Layered architectures: decision-making is consequence of their environment at different levels of abstraction.

However, it is not possible to understand what an agent is without their coexistence with others. Thus, agent behaviours are key to solve complex problems (final practical objective in the creation of this computational structure).

4.2 Multi-agent systems

A multi-agent system consists of a population of autonomous entities (agents) [Sycara (1998)] situated in a shared structured entity (environment) [Weyns & Holvoet (2005)]. These agents operate independently but are also able to interact with their environment, coordinating themselves with other agents. This coordination may imply cooperation if the agent society works towards common goals. Thus, in a cooperative community, agents usually have individual capabilities which, combined, will lead to solving the entire problem. But cooperation is not always possible and there are times in which

agents are competitive, having divergent goals. In this case, the agent should also take into account the actions of the others. However, even if the agents are able to act and achieve their goals by themselves, it may be beneficial to partially cooperate for better performance, thereby forming coalitions. Turning to coordinating activities, either in a cooperative or a competitive environment, one basic way to solve the potential conflicts that may arise between agents is by means of negotiation. Negotiation may be seen as the process of identifying interactions based on communication and reasoning regarding the state and intentions of other agents (cf. Figure 4.2 to obtain a scheme of possible agent actions).

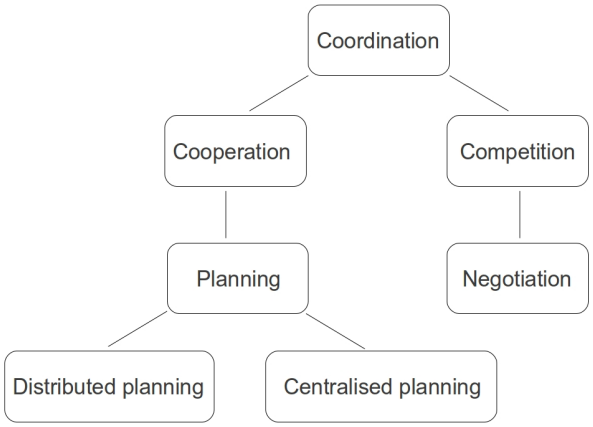

Figure 4.2: Classification of agent actions - Ways in which agents coordinate their behaviours

To produce coordinated systems, agents have a degree of autonomy in generating new actions and in deciding which goals to pursue next. There is a degree of uncertainty about each individual agent's action, which is mitigated with their integration on the global agent system. This system should coordinate the identification and classification of dependencies between agents (and their environment) and control decisions such as the agent influencing area (when an agent works alone or, working on commitments, it retains, rectifies or abandons). Another important system task is to manage the communication level between agents, taking into account that they are computationally efficient when concurrency of computation is exploited as long as communication is kept minimal. We dispose of agents with redundant characteristics which offers system

reliability. This system is easy to maintain because of the agent modularity that allows us to handle their properties locally. Agents are able to organise themselves to adapt their activity to different environments and even to solve different problems.

At this point, it is important to list the environment properties because it is here where agent interactions take place. A multi-agent system

- structures the multi-agent system as a whole.
- is in charge of managing resources and services.
- must be observable.
- must define concrete means for the agents to communicate.
- is responsible for maintaining ongoing processes in the system.
- can define different types of rules on all entities in the multi-agent system.

Once agents have been defined and their relationships established, a schedule of combined actions on these objects defines a process occurring, on their environment, over time. Thus, a multi-agent system (MAS) can be defined as various networks of problem solvers (each one is autonomous and can be heterogeneous) that interact to solve problems that are beyond the individual capabilities or knowledge of any one problem solver. Due to the characteristics of MASs, such as data decentralisation and asynchrony in computation, they are able to provide solutions distributed applications such as the problem of network division considered here.

4.3 Water network abstraction in a multi-agent environment

The applicability of multi-agent systems to obtain supply clusters from a WSN is due to their efficiency to work with distributed systems. These kinds of systems can be geographically distributed, can have many components, and can have a huge content (in number of concepts and/or amount of data), among other characteristics. A WSN meets all the above conditions.

A WSN is decomposed into its elements, which are thought of as certain agents that interact with each other, being able to achieve our target of the network partition. Thus,

consumption nodes are agents of a certain breed with a number of associated variables, elevation and demand being the most important. Pipes are links that connect two different nodes and have also the information of their diameter and length. Generally, pipes are open, allowing the transportation of the water flow from one end node to another (or consumption agent). But sometimes we can interrupt this flow transmission, making inoperative the link between nodes (thereby simulating the closure of their respective valves). Source points are another breed of agents, whose variables are the average demand they supply and the hydraulic sector they feed. All of these agents are spatially distributed and are not allowed to change their positions with time, but they can modify part of their characteristics. In our case, the most important evolving property is the membership to one or another hydraulic sector.

The MAS environment of the WSN is the space where the agents are distributed, representing the WSN. Each supply component usually inherits GIS model properties to locate it in a proportional position to their real coordinates. This environment lets us address the WSN as a whole, being correspondent via communication between agents, allowing to manage their operations and offering enough resources to maintain their processes in the system.

4.4 Graph multi-agent clustering

The goal of cluster analysis is to partition the observations into groups so that the pairwise dissimilarities (measurements of how different two elements are) between those assigned to the same cluster tend to be smaller than those in different clusters [Hastie *et al.* (2001)]. Most clustering algorithms assume an *a priori* number of clusters, c, and work to make a partition of the data set into these clusters. This methodology fits with WSN division based on DMA [Herrera *et al.* (2010b)]. Regarding this task, we are interested in the establishment of enough criteria to divide the network into homogeneous areas, obtaining an improvement in WSN management. The homogeneity concept has to do with these clustering similarities, which are measured by distances between the levels and demands associated with nodes. Besides, connectivity is another important issue to take into account when clustering graphs. In addition, distances are proportional to their difference in geographical coordinates, and neighbours are linked

nodes. Multi-agent alternatives facilitate the achievement of a clustering configuration, which contemplates those special graph characteristics.

In the current proposal, we *a priori* assume a given number of hydraulic sectors, relating them with the source points of the network. These will be the seeds for the corresponding districts. Then, these agents start a process of clustering by exploration, probing their neighbouring nodes and checking the likelihood of them being assimilated to the same supply cluster to which they belong. This likelihood is derived from a number of tests, which are performed based on sources, nodes and pipes properties [Izquierdo *et al.* (2009)]:

- The total length of the current DMA (sum of the length of its members) must be bounded by minimum and maximum values;
- The elevation of a new candidate must be within a certain range around the average elevation of the current supply cluster;
- The whole demand of the sector must be between certain prefixed limits;
- The geometrical properties of the area occupied by a supply cluster must exhibit certain basic requirements (connectivity, convexity, etc.);
- The associated sources must be able to provide that demand.

In each process step, neighbouring nodes for every hydraulic sector or division are explored. Each of these nodes is given a certain probability of belonging to a given district. This probability is proportional to the difference between its elevation and the average elevation of the district and to its demand and the average demand of the sector. Thus, the division is performed by weighting the difference between the node's elevation and demand with regard to the average elevation and demand of the calling sector. This weighted difference provides some resistance degree or dissimilarity, in terms of probability, to decide about its membership to the calling hydraulic sector. This way, the simple greedy competition based on the minimum distance among the districts is improved, what adds increased probabilistic richness to the process.

4.4.1 Negotiating the boundaries

The most distant points to the water sources are close to the boundaries between at least two districts. Besides, these points are usually the worst represented by the cluster to which they belong. We propose adding an energy criterion after the first division to re-assign these points to the boundaries in some efficient way. Until now, we are establishing sectors based on level and demand distances. We improve this dissimilarity measurement taking into account the lengths of paths from source points to the boundaries and changing the average districts level to the levels of source points. A sensitivity analysis is thereby proposed: negotiating the boundaries previously established from an energy point of view.

An instance of interaction protocol for negotiation may be such as found in Wooldridge (2000) (cf. subsections 4.5.1 and 4.5.2 to see the details of the supply clusters creation):

1. Agent1 proposes a course of action to Agent2;
   Agent2 evaluates the proposal and
2. sends acceptance to Agent1
   or
3. sends counterproposal to Agent1
   or
4. sends disagreement to Agent1
   or
5. sends rejection to Agent1

This process continues following our previous main target of homogeneity and minimal number of used valves. As a result, the global behaviour of the model agents, which perform a mixture of individual and collective actions, is able to delve into the best network layout providing efficient supply.

4.5 Agent algorithms and the experimental processes

This section checks the proposed methodology with the GIS model of a real network of moderate size introduced in Subsection 2.5.2. The aim is to divide the WSN of

this experimental case-study into 2 supply clusters (each cluster supplied at least by a tank). To this purpose, there will be applied the two procedures of multi-agent clustering algorithm developed in this section and finally proposed in next Subsection 4.5.1.

Graph nodes are considered agents with information about their level and demand. The final clustering configuration will be about the different associations that arise on these agents over the whole network. The necessary communication is by the set of their links and the connectivity degree of (each part of) the graph. Nodes connect with others by pipes (in this case, they are undirected links), which have some properties such as diameter and length. A neighbouring node may be in the same supply cluster or not. For this reason we enable the possibility of closing links that connect nodes in different areas and thereby achieve the isolation of each cluster. In our agent-world it is only necessary to forbid the passage of flow by certain links. This simulation has the hydraulic translation of closing valves in pipes.

4.5.1 Proposing a MAS clustering algorithm

The first part of the proposed process starts with the *a priori* assignment of each reservoir to a different DMA. Next, neighbouring nodes to such source points are scanned to decide whether to join to their clusters. This step is iterated a sufficient number of times, checking the neighbours' characteristics in each district and comparing these to decide (with a certain probability) the cluster membership of a new node. Our analyses are based on the level and demand of the new node and the average of these measures in each neighbouring cluster. Summarising, the general agent-behaviour tries to approach Equation 4.1:

$$Min \sum_{c=1}^{C} \left[\alpha_z \cdot (z_i - \overline{z}_c) + \alpha_d \cdot \left(d_i - \overline{d}_c \right) \right], \tag{4.1}$$

where α_z and α_d are the associated weights to level (z) and demand (d) to carry out the clustering configuration; $\overline{z}_c$ and $\overline{d}_c$ are level and demand average in cluster c, respectively. Finally, C is the total number of different supply clusters.

Once this first division is done, we propose to carry out the aforementioned sensitivity analysis. Nodes close to boundaries are activated to negotiate their cluster membership. The new paradigm consist in re-classifying these points to a different

membership if the associated cost to supply them is sufficiently great. Now, the general agent-clustering idea is based on the following Equation 4.2:

$$Min\left\{\lambda_\beta \sum_{c=1}^{C} \beta \cdot l_{i,c} + \lambda_\alpha \sum_{c=1}^{C} \left[\alpha_z \cdot (z_i - \overline{z}_c) + \alpha_d \cdot \left(d_i - \overline{d}_c\right)\right]\right\}, \tag{4.2}$$

where β is the associated weight to the distance ($l_{i,c}$) between the checking node and the source point of district c. λ_β is the importance of negotiating the energy effort of the hydraulic system and λ_α weights the above expression 4.1, homogenising cluster demands and levels.

4.5.2 MAS clustering algorithm implementation

The overall process can be summarised in two algorithms which are consecutively executed. The first is related to Equation 4.1 and tries to achieve homogeneity in the proposed clusters (Table 4.1).

Global:	0. Initial settings
Global:	1. Select source nodes: assign different groups to them
Agent:	2. Scan neighbouring cluster:
	2.1 Select a random node from their neighbourhood
	2.2 Is it on my cluster?
	Yes: Go to 2.1
	No: Continue
	3. Check selected neighbour:
	3.1 Is the difference of demands and levels $< Limit1$?
	Yes: Add to cluster and Go to 2
	No: Go to 2.1

Table 4.1: MAS-clustering algorithm I—homogeneity of the districts.

The second phase of the algorithm takes into account the energy costs associated with supply (see Table 4.2). Working with the same agents will provide continuity to the process. Nevertheless, these agents will change their point of view with respect to the clustering problem and will allow changing certain memberships if the distance from the sources to consumption nodes is large; this follows the idea behind Equation 4.2. These algorithms focus on global (universal) and agent points of view, which allow

Global:	0. Set the boundaries of the clustering solution
Global:	1. Select random nodes: verifying their cluster membership
Agent:	2. Scan neighbouring cluster:
	2.1 Select a random node from their neighbourhood
	2.2 Is it on my cluster?
	Yes: Go to 2.1
	No: Continue
	3. Check selected neighbour:
	3.1 Is distance to the proposed new water source
	$<$ current distance to source?
	Yes: Is difference of demands and levels $< Limit2$?
	Re-assign to cluster and Go to 2
	No: Go to 2.1

Table 4.2: MAS-clustering algorithm II—negotiating cluster boundaries.

us to approach the simulation in an easier way. The former is about general processes and the system can be managed by general decisions. The agent point of view endows our network with multiple (local) solvers, which cooperate and negotiate with others to achieve their clustering target.

Besides, in the algorithm we are working with two constraints defined by $Limit1$ and $Limit2$. Both have to do with the homogeneity condition to establish suitable clusters which can be translated to be efficient DMAs. $Limit1$ specifies the maximum allowed difference between node levels and demands and the same measurements (on average) for each cluster. This gives support to agents making a decision about to add or not a node to a cluster. But this decision is sometimes poorly achieved in the cluster boundaries. $Limit2$ offers the possibility to negotiate each membership on those areas, relaxing the homogeneity constraints to run the first algorithm but taking into account energy gains in the supply.

These algorithms have been implemented in NetLogo [Wilensky (1999)]. This software is an environment for the development of complex, multi-agent models, evolving in time (cf. Appendix A.1). It makes it possible to create populations of changing agents in a suitable grid of stable agents. The evolution of agents can take different forms: they can be created, move, change their properties, change their behaviour, change nature or breed, and even die. The model is created from GIS data defining

the physical and topological network characteristics. The area is divided into squares, patches. These represent the ground (underground) where pipes and nodes are buried, giving some raster format to the environment.

4.5.3 Results

After 20 runs of the model simulating the partition into hydraulic sectors of this network, the configuration shown in Figure 4.3 has been obtained in 80% of the cases. As a result, two sectors have been obtained that are isolated through 9 cut-off valves (links in red on part a) of Figure 4.4). These sectors have 62 and 63 pipes.

The NetLogo's output [Wilensky (1999)] (see part a) of Figure 4.4) shows that the average elevations for these sectors are 73 m for Sector 1 (represented in the Figure by brown nodes) and 69 m for Sector 2 (represented by the yellow ones). As shown by the plot in Figure 4.3 the architecture of the sectors stabilises along the simulation.

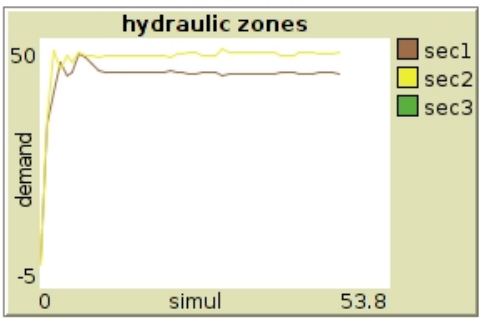

Figure 4.3: Evolution of demand running the MAS-algorithm I - Average demand by supply cluster

The final results of this first algorithm may be consulted in Table 4.3.

Table 4.3: Description of supply clusters with MAS-algorithm I

sector	n nodes	n pipes	n sources	avg. elevation	avg. demand
Sector 1	58	62	2	73	45
Sector 2	56	63	1	69	50

Now, the next phase of the process tries to identify interactions based on communication and reasoning regarding the membership and intentions of the agents on areas close to the boundaries. The behaviour of these agents could change following energy

premises instead of homogeneity. These changes have consequences in the supply cluster memberships, as can be seen in Figure 4.4. The final results may be consulted in Table 4.4.

Table 4.4: Description of supply clusters with MAS-algorithm II

sector	n nodes	n pipes	n sources	avg. elevation	avg. demand
Sector 1	44	51	2	72	46
Sector 2	70	73	1	70	49

Comparing both tables 4.4 and 4.3, significative changes between sectors are observed. Sector 1 (brown in Figure 4.4) diminishes in number of assigned nodes after negotiation, mainly in the Northwest of the network. This is the consequence of their higher average elevation. If this situation is accompanied by a greater dispersion in these levels it should be convenient to achieve this solution of cluster area reduction.

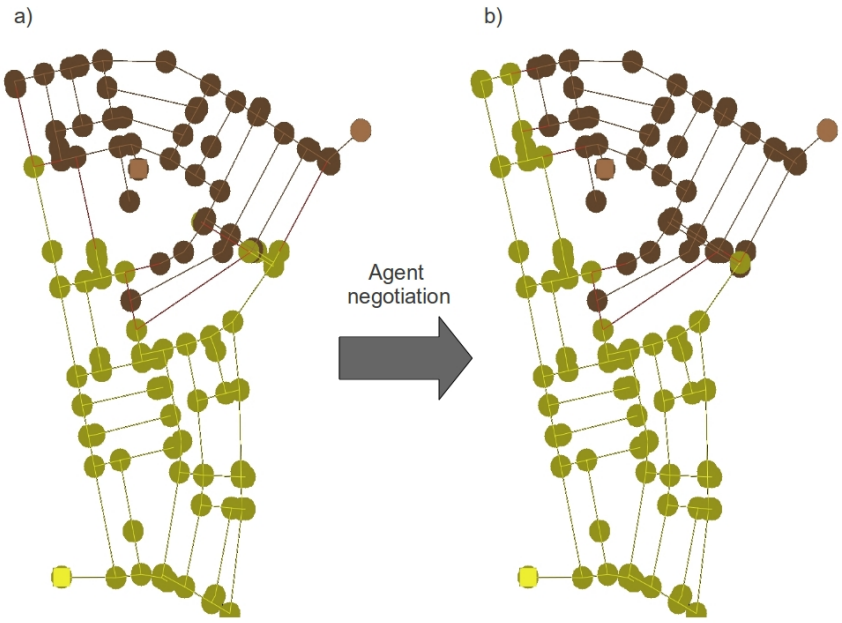

Figure 4.4: Final distribution of supply clusters - a) algorithm I b) negotiating: algorithm II

4.6 Summary and comments

The multi-agent metaphor has been introduced in this chapter to divide a WSN into supply clusters. In addition to a traditional centralised architecture of single reasoning agent (the computing counterpart of human decision support), this chapter shows that is possible to use systems of reasoning agents, or to apply multi-agent simulations to verify hypotheses about the different processes in water distribution. The inclusion of a negotiation behaviour of agents directly proposes a sensitivity analysis of the solution previously found via agent cooperation and competition. Nevertheless, it is the starting point of a working line on multi-objective approaches to be developed in the future.

The model may be applied to larger networks. Indeed, taking into account that the considered network in the proposed case-study is medium-sized and running times are slow (ranging between 10 and 20 seconds on a PC with an Intel Core 2 Duo T5500 1.66 GHz processor for the case considered), no added difficulties are foreseen.

Among the different scenarios using multi-agent systems in the scope of decision support for the water management company, this chapter focuses on the division of a WSN into district metered areas. Next research will focus on development of implementations of other scenarios of multi-agent applications in the water supply field, including aspects related to water quality, location of sensors and other managerial issues.

5

Multi-agent adaptive boosting on semi-supervised water supply clusters

As already mentioned in previous chapters, clustering plays an important role in many fields and can be used both for preliminary and descriptive data analysis and unsupervised classification. The intuitive goal of clustering is to divide the data points into groups such that points in the same group are similar and points in different groups are dissimilar to each other. For this reason, it is important to distinguish whether the data contains, or does not contain, graph information, because clustering algorithms (and similarity measures) should be sensitive to the nature of the data. The graph clustering [Shi & Malik (2000)] paradigm may be approached by similarity graphs and some associated spectral clustering algorithms. Spectral methods capture a variety of geometries being more flexible than classical algorithms, strongly based on the Euclidean distance. Nevertheless, when working with real-world data, graphs to cluster may be much too large for straightforward spectral clustering to be feasible.

This chapter deals with the application of spectral clustering to split a water supply network (WSN) into isolated supply clusters. A semi-supervised approach to spectral clustering is able to take into account the graph structure of a network and incorporate the corresponding hydraulic constraints and the rest of the available vector information [Herrera *et al.* (2010a)] of the WSN. But some disadvantages of these methodologies stem from the large size of the usual WSN and the associated computational complexity.

This could generate applicability problems. One way to manage this is by taking low-rank matrix approximations [Fowlkes *et al.* (2004)]. Another approach to speed up the algorithm is by using preprocessors that minimise the matrix distortion (viz to, minimise the effect of data reduction in the spectral clustering performance) to obtain approximate cluster configurations [Ng *et al.* (2001); Yan *et al.* (2009)]. This chapter works with the option of subsampling graph data [Bordino *et al.* (2008); Hübler *et al.* (2008); Kolaczyk (2009); Leskovec & Faloutsos (2006)] to run successive weak clusters and build a single robust cluster configuration. The background to these boosting methods is found in the works of Kudo *et al.* (2004); Tsuda & Kudo (2006) about graph classification and graph weighted substructure mining.

The road-map of the chapter is as follows. In Section 5.1 the spectral clustering algorithm and different methodologies to work with large graphs (such as probably is the case with real WSN) are introduced. This section also proposes a multi-agent methodology to sampling graphs. The first part of Section 5.2 introduces boosting methods and boosting on graphs paradigm. Secondly, it develops semi-supervised clustering under a boosting methodology. Understanding the algorithm proposed in next Section 5.3 (which is detailed step by step) is key for this chapter. The chapter finalises applying this semi-supervised boosting process to a real case-study in Section 5.4. Lastly, the results and conclusions are discussed in Section 5.5.

5.1 Clustering large graphs: the case of real WSN division into supply clusters

The starting point for creating supply clusters in a WSN is to take into account all the available network information. First of all, a WSN must be considered as a particular graph where the edges are pipes and the nodes are consumption points (such as is considered in this thesis; cf. Section 3.2 of Chapter 3). A WSN object can be described by a set of measurements (geographical and connectivity information, node elevations or pipe diameters) or by their relation to other objects. Thus, items within a cluster are more similar to each other than they are to items in other clusters. Central to all of the goals of cluster analysis is the notion of this degree of similarity (or dissimilarity) among the individual objects being clustered - because clustering methods attempt to group objects based on the definition of similarity supplied to them. This notion of

similarity can be expressed in different ways according to the purpose of the study, the domain-specific assumptions, and the prior knowledge of the problem.

A semi-supervised approach to spectral clustering takes into account the graph structure of a network, and incorporates the corresponding hydraulic constraints and remaining vector information [Herrera *et al.* (2010a)] from the WSN. Given a data set consisting of n data points, spectral clustering algorithms form an $n \times n$ affinity matrix and compute the eigenvectors, a task with computational complexity of $O(n^c)$, where in general $c \geq 3$. For applications with n on the order of thousands, spectral clustering methods begin to become infeasible [Yan *et al.* (2009)]. Real water networks must usually be categorised as examples of these large graphs. One strategy to manage these networks is to develop algorithms to improve the c term in the runtime effort. Another solution is to reduce n by reducing the graph size. One way to achieve this reduction is by sampling a subgraph from the large graph such that this subgraph is a reasonable approximation of the original graph [Leskovec & Faloutsos (2006)]. This is the problem approached in this chapter to obtain a practical methodology to clustering large graphs. There are many options that can be considered for this preprocessing task. One option is to perform various forms of subsampling of the data, selecting data points at random or according to some form of stratification procedure. Another option is to replace the original data set with a small number of representative points that aim to capture the relevant structure. Another approach that is specifically available in the spectral clustering setting is to exploit the literature on low-rank matrix approximations. The next subsections propose some of these alternatives found in literature.

5.1.1 Improving number of operations

- *Singular Value Decomposition*: It solves a continuous relaxation of the k-means clustering algorithm for graphs, finding the k-dimensional subspace V that minimises the sum of squared distances from m points to V. This relaxation can be solved by computing the Singular Value Decomposition (SVD) of the $m \times n$ matrix A that represents the m points [Drineas *et al.* (2004)].

- *Nyström method*: This approach is based on a technique for the numerical solution of eigenfunction problems. This method allows one to extrapolate the complete grouping solution using only a small number of samples. In doing so, it leverages

the fact that there are far fewer coherent groups in the target database [Fowlkes *et al.* (2004)].

- *Approximate spectral clustering*: This methodology extends the range of spectral clustering by developing a general framework for fast approximate spectral clustering in which a distortion-minimising local transformation is first applied to the data [Yan *et al.* (2009)]. This framework is based on a theoretical analysis that provides a statistical characterisation of the effect of local distortion on the miss-clustering rate.

5.1.2 Sampling graphs techniques

- *Sampling by random node selection*: This is a straightforward way to create a sample graph by uniformly selecting at random a set of nodes, N. Thus, a sample is a graph induced by this set of nodes, N. This algorithm is called *Random Node* (RN) sampling [Stumpf *et al.* (2005)]. There are other techniques which employ non-uniform sampling strategies. With these, the probability of a node being selected will be proportional to some characteristic such as its degree or associated *PageRank* weight [Leskovec & Faloutsos (2006)].

- *Sampling by random edge selection*: Similarly to selecting nodes at random, one can also select edges uniformly at random. This algorithm is referred to as *Random Edge* (RE) sampling. There are some problems with this idea, because sampled graphs would be sparsely connected and it does not respect any community structure. A slight variation is *Random Node-Edge* (RNE) sampling, where we first uniformly at random pick a node and then uniformly at random pick an edge incident to the node.

- *Sampling by exploration*: In this family of sampling techniques, a node is first uniformly selected at random and then its neighbouring nodes are explored [Herrera *et al.* (2011)].

  - *Random node neighbour*: A sample node is selected uniformly at random along with all its neighbours reachable by its out-going links. It matches well the out-degree distribution, but fails in matching in-degrees and the community structure.

- *Random walk*: This process samples uniformly at random by picking a starting node and then simulating a random walk on the graph. With some probability this process will go back to the starting node. There is a problem of getting stuck, for instance, if the starting node is a sink, and/or it belongs to a group of isolated components.
- *Random jump*: It is similar to Random Walk technique, but randomly jumping to any node in the graph.
- *Forest fire*: It randomly picks an initial node and begins *burning* outgoing links and the corresponding nodes [Leskovec *et al.* (2005)]. If a link gets burned, the node at the other endpoint gets a chance to burn its own links, and so on recursively.

5.1.3 Multi-agent support to sampling subgraphs

In the sampling application to a real WSN, each subgraph should be somehow representative of the network, but maintaining a compromise of ease of construction. To achieve this target, a methodology of sampling by exploration simulated by a multi-agent environment [Izquierdo *et al.* (2009); Wooldridge (2002)] is proposed below.

As described in Section 4.1 of Chapter 4, an agent is any entity in a system which can generate events that affect itself and other agents. Once agents have been defined and their relationships established, a schedule of combined actions on these objects defines a processes occurring over time. These instructions are given to hundred or thousand of agents operating independently. In the problem we will consider here, the agents are consumption nodes and connecting pipes. The sampling subgraphs methodology will simulate virus propagation behaviour.

5.1.3.1 Sampling by simulation of virus propagation

Some steps should be taken into account if the aim is sampling WSN by simulation of virus propagation behaviour. The proposed sampling algorithm starts choosing uniformly at random a starting virus infected node. Next it may spread the virus through the network. Each time step, each infected node attempts to infect all of its neighbours. Susceptible neighbours will be infected with a given probability and then become transmitters of the virus to the neighbouring nodes susceptible to infection. The process

continues to a fixed number of iterations and finally we will obtain, as output, a subgraph sample of the infected nodes (see Table 5.1).

Table 5.1: Sampling subgraphs by multi-agent exploration methodology

pseudo-code: spread-virus to sampling subgraphs
1. set-up conditions
2. to spread-virus
ask nodes with [infected?]
[ask link-neighbours with [not resistant?]
[if random-float 100 < virus-spread-chance
[become-infected]]]
end
3. save subgraph

If necessary, this sampling methodology may be complicated by adding rules and agent-operations like *virus mutation*, *node-resistance* or offering the chance of *recovery* to the infected nodes. This opens possibilities to automatic changes in the sampling methodology and asses their adaptation to any graph boosting process (cf. Section 5.2). In addition, these new rules will achieve a more suitable fit of the algorithm to the special characteristics of a WSN.

5.2 Boost-clustering for semi-supervised subgraphs

Boosting is a technique to improve the performance of machine learning algorithms combining 'weak learners' to find a highly accurate classifier or better manage the training set [Schapire (1990)]. The boosting algorithm calls this 'base' learning algorithms repeatedly. Each time they are run on a different subset of the training instances, because they will be re-sampled under an evolutive structure of the data support (trying to improve the algorithm sequence in some way, forcing the base weak learners to focus attention on the 'hardest' examples). After enough iterations, the boosting process combine these weak results into a single output that will be much more accurate than any one of the others. Obtaining the final configuration may be possible by taking a weighted majority vote of the individual results. The *AdaBoost* algorithm is a practical implementation of boosting [Freund & Schapire (1999); Schapire (2003)]. This process is detailed in Table 5.2.

Table 5.2: Algorithmic implementation of AdaBoost

The boosting algorithm AdaBoost
Let $(x_1, y_1), \ldots (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, 1\}$
Initialise $D_1(i) = 1/m$
1. For $t = 1, \ldots T$
1.1 Train base learner using distribution D_t
1.2 Get base classifier $h_t : X \to \mathbb{R}$
1.3 Choose $\alpha_t \in \mathbb{R}$
1.4 Update:
$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_i y_i h_t(x_i))}{Z_t}$
where Z_t is a normalisation factor such that D_{t+1} is a probability distribution.
2. Output the final classifier:
$H(x) = sign(\sum \alpha_t h_t(x))$

Recently, the boosting algorithm has been successfully extended to tasks such as semi-supervised learning [Zheng *et al.* (2009)] and manifold learning [Loeff *et al.* (2008)], among others. This work adapts the AdaBoosting methodology to the special graph characteristics, modifying it to take advantage of both semi-supervised process accuracy and multi-agent framework efficiency.

5.2.1 Semi-supervised boosting and the pre-clustering phase

The key idea of semi-supervised learning [Chapelle *et al.* (2006)] is to use both labelled and unlabelled data (cf. Section 3.4). Labelled data can be understood as the prior information which is available in the target problem of boosting (it is graph clustering in the current case). One way to address the algorithm is by some kind of propagation of this previous knowledge. It may be approached spreading the labels to the rest of unlabelled data, obtaining so called *pseudo-labels**. The labelled data, along with the sampled pseudo-labelled data, are utilised in the next iteration for training the boosting algorithm on the second phase. To avoid possible bias in this process, the proposal to spread pseudo-labels will be by some type of fast clustering procedure.

*Pseudo-labels will contain information about previous sectorisations and the original supply nodes of each area.

The membership of every element to each cluster will propose the class of pseudo-label assigned. A multi-agent clustering methodology, similar to the one shown in Chapter 4, would be suitable to approach this task. In the current case, the process will continue running a spectral clustering algorithm on each sampled graph; loading these labels and pseudo-labels to the kernel matrix (cf. Chapter 3). This way, the sampled subgraphs will inherit some global characteristics of the graph through these pseudo-labels. Thus, this semi-supervised boosting phase of spreading labels may also be called a *pre-clustering* phase.

5.2.2 Semi-supervised boost-clustering in WSN

A suitable boost-clustering methodology exploits the general principles of boosting and increases the applicability of the main spectral clustering algorithm. At each boosting iteration, a new subgraph is sampled from the original network (cf. Subsection 5.1.3). Next, it applies straightforwardly the spectral clustering algorithm. However, dealing with real cases (such as is the case of WSNs), it should use some background knowledge about the cluster structure if it is available. In this sense, it tries to inherit in each subgraph the different nature of some nodes which represent supply sources of the WSN graph. Then, it considers a limited number of labelled data (sources) and several unlabelled examples (consumption nodes) bringing the current boosting algorithm into the class of semi-supervised learning methods [Mallapragada *et al.* (2009)]. Each source node is marked by a label and then we spread over the whole network as many different labels as water sources. This idea is based on the assumption that data samples with high similarity between them must share the same label. Thus, each node will have a pseudo-label starting from the water sources and propagating them based on a fast multi-agent clustering criterion [Izquierdo *et al.* (2009)]. As a result, a sampled subgraph (cf. Subsection 5.1.3) also gives the corresponding propagated pseudo-labels in every node. It will be enough to choose randomly a different label for each group of similar pseudo-labels to represent the new supply nodes in each subgraph*.

Next, to create a subgraph at each boosting iteration with labelled and unlabelled nodes, it is proposed to apply a spectral clustering method (cf. Chapter 3, Subsection 3.2.4) based on eigendecompositions of kernel matrices constructed from affinity and

*If some minimum number of different labels is not reached in the sampled subgraph (proportional to the number of original source nodes) it is necessary to allot them randomly.

dissimilarity matrices [Herrera *et al.* (2010a)]. This will take into account other information matrices that contain the values of the inputs under study and the must-link and cannot-link constraints related to the labelled source nodes (Table 5.3).

Table 5.3: Semi-supervised clustering in each subgraph

algorithm: water supply clusters by semi-supervised learning
1. abstraction of the water supply network as a subgraph
2. construction of Laplacian and dissimilarity matrices
3. data transformation into a single kernel matrix
4. *plug-in* of hydraulic constraints
5. calculus of the matrix spectrum
6. k-means into the top eigenvectors
7. cluster re-assignment into the original subgraph data
8. hydraulic validation (EPANET [Rossman (2000)])

5.3 The proposed MAS-boost clustering algorithm

The final clustering solution is produced by aggregating the multiple subgraph clustering results through a weighted voting system [Frossyniotis *et al.* (2004)] based on an adaptive boosting process [Dietterich (2001); Freund & Schapire (1997)]. In this joint case we propose proportional weights to the goodness of cluster representation of each node (border membership, cf. Subsection 5.3.5) and their iteration (weights being larger in more advanced iterations). The overall process is represented in Table 5.4.

The overall multi-agent boost-clustering algorithm is detailed, step by step, in the next subsections. Each developed phase is focused on the application case of WSN division into supply clusters.

5.3.1 Pre-clustering phase

In this stage, the process spreads pseudo-labels under similar considerations to those in Subsection 5.2.1. This is the moment to use a previous sectorisation, if it exists. Otherwise, this phase will employ a naive version of the multi-agent clustering developed in Chapter 4. In clustering WSN practice, it should be taken into account that the source nodes of the WSN will be the real labels to spread. The process continues

Table 5.4: Overall boosting semi-supervised clustering process

algorithm: semi-supervised clustering by boosting subgraphs
1. pre-clustering to label propagation (use previous sectorisation?)
2. sampling subgraphs by multi-agent graph exploration
2.1. assign labels (pseudo-source nodes)
2.2. semi-supervised clustering (see Table 5.3)
2.3. cluster evaluation
2.4. $t \leftarrow (t+1)$;
2.4.1. *if* $t < T$
[re-assign weights to sampling subgraphs
return to 2]
3. voting clusters summarising the boosting iterations

sampling subgraphs, once the labels have been assigned to each node. An instance of this phase results may be viewed by the case study of Figure 5.1.

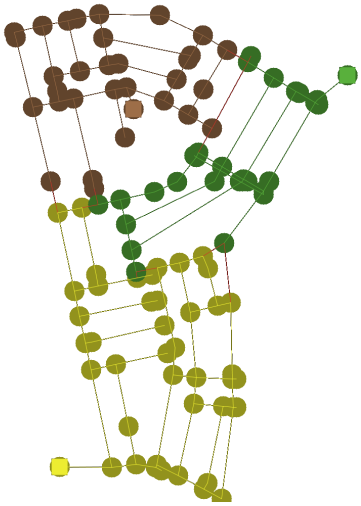

Figure 5.1: MAS pre-clustering phase - Results of the case-study

5.3.2 Sampling subgraph by multi-agent graph exploration

As described in Subsection 5.1.3 and more specifically in 5.1.3.1, the process starts choosing uniformly at random a starting virus infected node. The next sampling by exploration is thereby understood as a virus propagation phase (Figure 5.2 shows an

instance of a subgraph sampled by virus propagation - nodes in red colour - on a previously pre-clustered graph - clusters with nodes marked with green, yellow and brown -; the virus source node has been highlighted). Every node that is neighbour of an infected node, is susceptible to be sampled (infected). Special node characteristics, both natural (such as to be a source node) and added by the process (node immunity, resistance, etc.) should be taken into account at each sampling iteration.

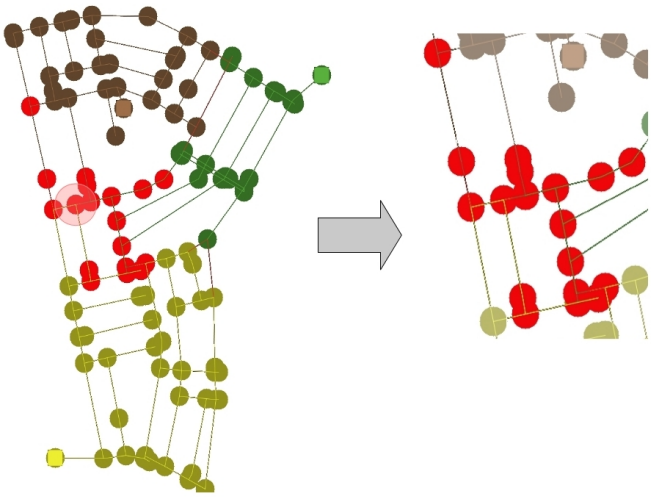

Figure 5.2: Sampling by simulation of a virus propagation - Results of the case-study

Other issues of this sampling methodology are rules and agent-operations like: *virus mutation* (it changes the strategy of the probability to sampling a whole subgraph), *node-resistance* (it varies the assigned weights in the probability distribution to sampling some selected nodes) and the chance of *recovery/immunity* (it is a temporary or definitive change in the status of some nodes). This opens possibilities to automatic changes in the sampling methodology and assesses their adaptation to any graph boosting process. They will have straightforward influence in the re-assignation of weights to sampling subgraphs (cf. Subsection 5.3.6).

5.3.3 Assigning pseudo-source nodes

The process of assigning pseudo-source nodes starts once the network sample has been selected. Next, their pseudo-labels are checked.

- If all possible supply clusters are represented by their respective pseudo-labels in the subgraph, then:
  - randomly choose one representative of each type of label to assign to be pseudo-source node. This selection should take into account that pseudo-labels contain information about the original supply source of each node. Thus, the selection will be one pseudo-source node for each different type of label.
- if not all possible supply clusters are represented:
  - randomly choose one representative of each type of label to assign to be pseudo-source node.

Figure 5.3 illustrates this part of the process.

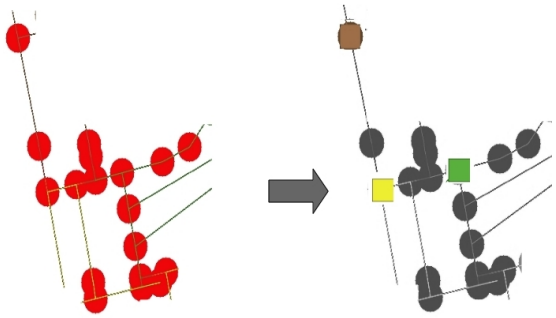

Figure 5.3: Assignment of pseudo-source nodes - Case-study instance

5.3.4 Semi-supervised clustering

A semi-supervised clustering algorithm runs on the sampled subgraph. This algorithm may be similar to the one described in Chapter 3, Section 3.4, and summarised in Table 5.3.

5.3.5 Cluster evaluation: silhouette for graphs

A modification of the method of silhouette, introduced in Chapter 3, Section 3.1.2, is the cluster validation process, which is chosen here, as cohesion-separation criteria. The changes proposed have to do with the mixture of different data-types that the process works with. Then the silhouette calculus is divided into two parts:

The first part is related to continuous variables of the database (demand, level, geographic coordinates). This measurement is assigned to each node in the clustering scheme and follows the next expression (eq. 5.1):

$$sc_i = \frac{(b_i - a_i)}{max(a_i, b_i)}, \tag{5.1}$$

where a_i is the average distance to object i to all other objects in its cluster and b_i is the minimum distance between object i and any other element of a different cluster to the current one.

The second part will be related solely to boundary nodes. These inherit the so-called property of *foreigner degree*. This index will be linked to the relative number of neighbouring nodes belonging to other clusters. Equation 5.2 represents this concept:

$$fd_i = \frac{nv_i}{(dg - 1) \cdot nt_i}, \tag{5.2}$$

where nv_i is the number of neighbour nodes of i classified in clusters different than i. The total number of neighbours of node i is represented by nt and the mean degree of the graph is dg.

The final expression of the silhouette will be:

$$sh_i = \frac{(b_i - a_i) - fd_i}{max(a_i, b_i)}. \tag{5.3}$$

This modified silhouette in Equation 5.3 will have an important role in this boosting process proposed, because it favours the appearance of worst represented nodes by the current cluster configuration, as observed in next samples.

5.3.6 Re-assigning weights to sampling subgraphs

Once an iteration is computed and the quality of the current cluster representation of the subgraph is obtained, the proposed boosting process changes the weights with which the nodes are sampled in the next iteration. The idea is to reduce the weight of well-clustered nodes and favour the sampling of badly clustered data (sampling more times the items harder to cluster in previous iterations). To do this, the system updates the weights relating the silhouette value of each node [Rousseeuw (1987); Tan *et al.* (2005)] with the *node-resistance* characteristic in the proposed sampling by virus exploration (cf. Subsection 5.1.3). The following Equation 5.4 details this update:

$$w_i^{t+1} = w_i^t \times \frac{\left(1 + \frac{1-sh_i^t}{2}\right)}{Z^t}, \tag{5.4}$$

where w_i^t is the weight associated with *node-resistance*, sh_i^t is the silhouette value and Z^t is a normalisation constant such that w_i^{t+1} is a sample of a distribution function, W^{t+1}, and then $\sum_i w_i^{t+1} = 1$. All items are referred to node i at the moment of moving from iteration t to $t+1$. This expression 5.4 favours sampling elements with lower silhouette value, taking into account that $-1 \leq sh_i \leq 1$.

5.3.7 Multi-agent voting system

Once the different clustering results on the subgraphs are available, it is necessary to develop a process that allows a combination among these results forming a common partition. The proposal is to do it by a multi-agent voting system in such a way that the global silhouette is improved (reducing error classification and achieving better results on the worst clustered data). The process is based on the same principles as Chapter 4, but taking into account the memberships to each cluster obtained above (cf. membership Table 5.5). This fact improves the results of the multi-agent approach with the resulting accuracy of the semi-supervised clustering process.

Table 5.5: Aggregated results of the sampling subgraphs

	cluster 1	cluster 2	...	cluster K
node 1	c_{11}	c_{12}	...	c_{1K}
node 2	c_{21}	c_{22}	...	c_{2K}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
node n	c_{n1}	c_{n2}	...	c_{nK}

The performance of this multi-agent voting system is based on the principles shown in sections 4.3 and 4.4. But now, new items are declared in relation to the current boosting process. They are the frequency of membership of the node, i, to each cluster, j: c_{ij} where $i = 1, \ldots, n$ and $j = 1 \ldots K$; n is the total number of nodes and K the number of clusters (cf. Table 5.5; in this matrix the rows are consumption nodes and the columns are cluster labels).

Now, the general voting idea is to aggregate similar nodes to the same clusters based on their boosting membership results. The following Equation 5.5 summarises the agent behaviour on this voting phase.

$$\min_{c_{ij}} \sum_{j \in N_i} [c_{ij} - \text{argmax}(c_{ij})], \ \forall i = 1, \ldots, n, \tag{5.5}$$

where N_i is the neighbourhood of node i.

Equation 5.5 tries to group nodes by similar high frequency of boosting classifications in the same cluster. If every node (or agent) works to obtain an equilibrium, it is possible to achieve an efficient voting system. In addition, this process avoids an hypothetical bias of the subgraph sampled by the agent's cluster membership negotiation in each node.

5.4 Experimental process

Now, we are going to check the proposed methodology for solving a WSN division. Then, we iteratively recycle the sampling subgraphs providing multiple clusterings and resulting in a common partition. This recycling process is key in the performance of the main algorithm (Table 5.4). Other questions are about the number of clusters C (a pre-fixed number between 1 and the number of source nodes in the WSN target) and the size of each subgraph sampled. Frossyniotis *et al.* (2004) and Mallapragada *et al.* (2009) proposed numbers between 20 and 50 per cent of the total size. Finally, to complete the boosting process, the maximum number of iterations, T, will be considered fixed. This criterion is for focusing on the quality of the cluster representation (silhouette) which is computed at each iteration. If this quality does not improve beyond a certain threshold, the process is stopped before iteration T.

In order to show the performance of the presented process, the simple real case introduced in Chapter 2, Subsection 2.5.2 is considered. The aim is to divide the WSN of this experimental case-study into 3 supply clusters* (each cluster is supplied by one tank). It is a very simple instance regarding the target of dividing large WSN, but it

*Note that a proposal of only 2 supply clusters (such as in previous chapters) loses interest in this case-study. Firstly, because it would subsample a small network with only 2 possible labels. Next, because the results would not be very different from those offered by the semi-supervised algorithm, proposed in Chapter 3.

will be enough to check the performance of the proposed boosting process. As already said, this network is fed by 3 reservoirs and made out of 132 lines and 104 consumption nodes; its total length is 9 km and the total consumed flowrate amounts to 47 l/s.

5.4.1 Results

First of all, it is necessary to establish conditions for approaching the division of the WSN into 3 supply clusters. Decisions such as how small a sample to take, how many times to iterate, the updating of the weights in sampling and how to summarise the final voting system are guided by the methodology shown. In this regard, the subgraph size should be approximately 30% of the total number of nodes. In addition, it is considered that the value of 10 is the maximum number of boosting iterations, T (this number is enough in this case due to the medium size of the original network). The way to update the weights in the boosting will be that of changing the *node-resistance* value by using (5.4).

This division assesses that each DMA is supplied by at least one tank (see more details about average elevation and total demand of these supply clusters in Table 5.6). Twelve cut-off valves are necessary to isolate the different supply clusters*.

Table 5.6: Description of supply clusters with MAS-boost clustering algorithm

sector	n nodes	n pipes	n sources	avg. elevation	avg. demand
Sector 1	31	36	1	73	45
Sector 2	34	38	1	69	50
Sector 3	49	60	1	72	47

In Figure 5.4 we can see, approximately, the cut planes dividing the WSN into three hydraulic zones. In addition, this figure shows the distribution of the different water suppliers in these supply clusters.

This final configuration was successfully simulated in EPANET, thus validating the results.

*These results are calculated with the R Language [R-Development-Core-Team (2010)] function `specc` of the `kernlab` library [Karatzoglou (2006)] along with NetLogo [Wilensky (1999)]

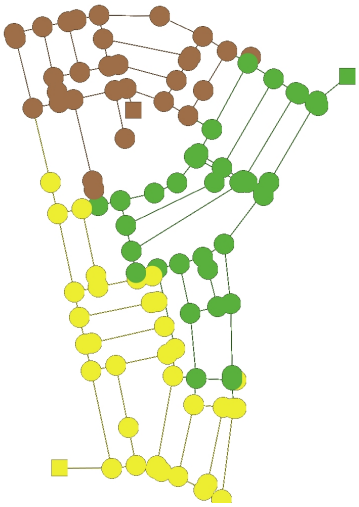

Figure 5.4: Layout of the WSN division into three supply clusters - Boosting algorithm application

5.5 Summary and comments

We have proposed an adequate framework to support the division of a WSN into DMAs. This division is scalable to whatever size of the network and takes into account the supply conditions necessary to turn these districts into real hydraulic sectors. In addition, this chapter proposes advances in various senses:

- It reduces the burden of computations in the spectral clustering paradigm, making it feasible by boosting, even in the case of large size graphs.
- It proposes a novel multi-agent approach to sampling subgraphs by exploration and to re-weighting boosting methods.
- It implements a fast multi-agent *pre-clustering* into the label propagation phase in semi-supervised clustering.
- Hydraulically, this chapter proposes the use of both graphical and vector information to improve the approach of dividing a WSN into DMAs.

Future research will focus on the improvement of the semi-boosting methodology employed. Among other things, this would include a guide sampling on subgraphs

formed by boundary elements of a previous network division. Through this sensitivity analysis costs of reforms of the DMAs already built may also be reduced.

Part III

Results and applications

6

Results of dividing a real water network into supply clusters

The main subject of this thesis has been the development, application and experimental analysis of clustering algorithms approaching the network division problem. The following sections test these proposed algorithms in the sectorisation proposal of the real-world case of Celaya (Guanajuato, Mexico) central district. A comparison of the different solutions opens the discussion about the first conclusions of this thesis.

6.1 Results and first conclusions

In order to test the performance of the proposed algorithms in chapters 3, 4 and 5, it is proposed to run them on the real-world case study proposed in Section 2.5 of Chapter 2, and analyse the results obtained.

The water supply network (WSN) of the Central district of Celaya is fed by one reservoir ($D1$) and five tanks ($E1, \ldots, E5$) with five pump stations. This network is made out of 479 lines and 333 consumption nodes; its total pipe length is 42.5 km and the node elevation average is 156 meters; the total consumed flowrate amounts to 91 l/s. The input database is composed of information about the node elevations, their water demand and geographic coordinates. In addition, information about pipes, such as their materials, diameters and lengths, is available.

Describing the study area, Figure 6.1 shows the GIS of the WSN under study along with the spatial distribution of the points with major water consumption (more than

2 l/s). The location of these points will influence the next results due to the important role of the demand in the proposed algorithms.

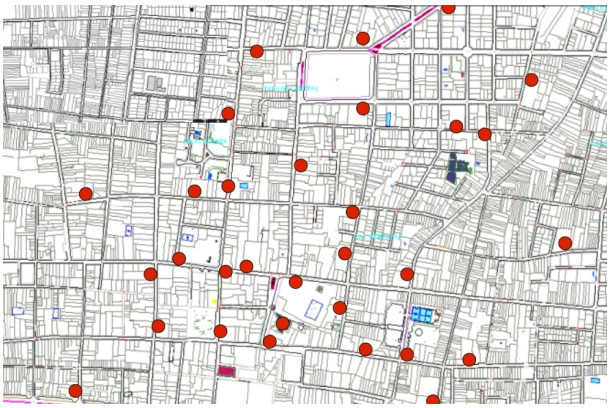

Figure 6.1: GIS map of the case-study proposed - major consumption nodes highlighted

The pipe materials distribution is represented in figure 6.2 (this variable may be related to the pipes age):

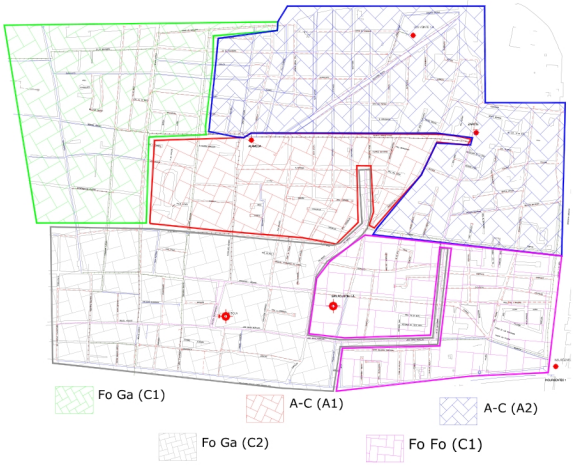

Figure 6.2: Layout of the case study regarding pipe materials - similar pipe material areas

where the legend for these materials is: Fo-Ga for galvanised iron, A-C for asbestos cement and Fo-Fo for cast iron.

The last co-variable that may influence the results is the layout of the main avenues of the district (cf. Figure 6.3). They can propose natural cuts into the WSN and may

be related to pipe variables such as diameter or stress.

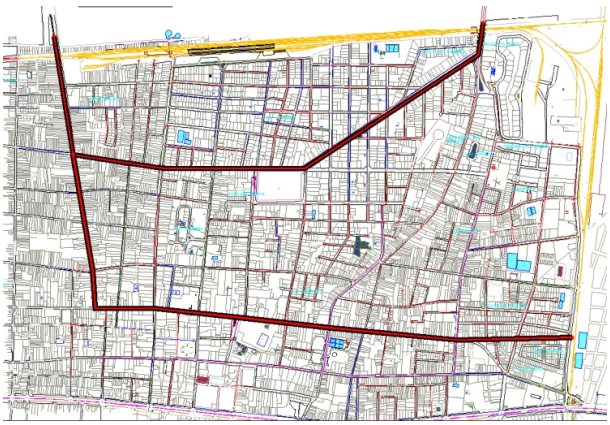

Figure 6.3: GIS map of the case-study proposed - layout of the main avenues

Finally, the network layout of the case-study is represented in Figure 6.4:

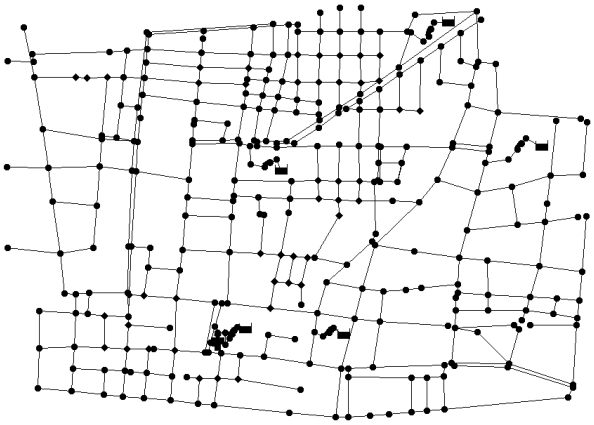

Figure 6.4: WSN layout of Centre Zone of Celaya - EPANET scheme

Once the WSN has been described and their variables introduced, the aim is to divide the WSN of our case study into 3 supply clusters (each one of them should be supplied by at least one tank or reservoir). This decision to propose 3 sectors is based on hydraulic reasons. Every sector should have a size bounded between a minimum and a maximum recommended [IWWA-Loss-Group (2007)], so we reject dividing the network into 2, 4 or 5 parts (we can not divide the network into any more parts that the number of sources of supply).

6.1.1 Application of semi-supervised clustering

We test the proposed semi-supervised learning algorithm, described in Chapter 3. In order to follow the kernalisation process of the involved variables (summarised in equation 3.15 of Subsection 3.4.3), they must be assigned their corresponding weights following the AHP criteria of Table 6.1.

Table 6.1: Preference matrix to assign weights by AHP

	elevation	demand	x-coor	y-coor
elevation	1	1/3	1/3	1/3
demand	3	1	3	3
x-coor	3	1/3	1	1
y-coor	3	1/3	1	1

As a result, the priority vector $W = (0.10, 0.42, 0.24, 0.24)^t$ represents the weight for *elevation*, *demand*, *x-coordinate* and *y-coordinate*, respectively. It has a consistency ratio of 0.06, Then, equation 3.15 is as follows: $K = \lambda_A K_A + (1-\lambda_A)\{0.10K_l + 0.42K_d + 0.24K_x + 0.24K_y\}$.

To obtain λ_A we will carry out a cost analysis of their possible values, once we build the clusters. Thus, different values belonging to $[0, 1]$ are checked. In the light of the results shown in Figure 6.5, it is possible to conclude that one of the more profitable options is to let $\lambda_A = 0.4$ to run the algorithm.

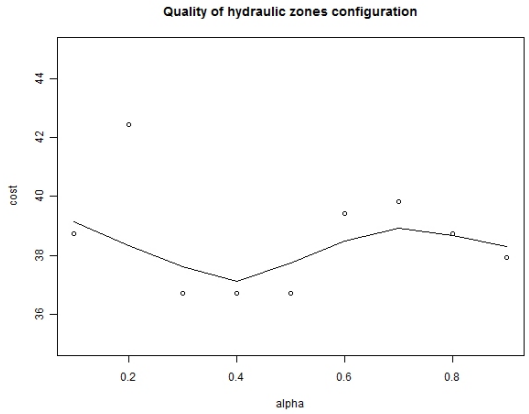

Figure 6.5: Cost of clustering by number of operations number of operations needed to isolate the clusters - Case-study divided into three supply clusters

After applying this process to the WSN data the following results are obtained. The size of each supply cluster in kilometres of pipes is 18, 9.5 and 15 km, respectively. The average diameter (in millimetres) of pipes per cluster is 144, 130 and 107, respectively. For this division, each sector is supplied by at least one tank (see more details about average elevation and total demand of these hydraulic zones in Table 6.2). It is necessary to close 26 valves to isolate each supply cluster. The objective function of the clustering algorithm guarantees that it is the minimum number of links (valves) that should be closed for carrying out this graph partition.

Table 6.2: Description of supply clusters by the semi-supervised algorithm

sector	n nodes	n pipes	sources	avg. elevation	total demand
sector 1	122	189	E1 + E3 + D1	157	35
sector 2	127	131	E4	155	25
sector 3	84	133	E2 + E5	156	31

Figure 6.6 (calculated with the R Language [R-Development-Core-Team (2010)] function `specc` of the `kernlab` library [Karatzoglou (2006)] and represented with NetLogo [Wilensky (1999)]) shows the final sector division of the WSN into three supply clusters. This final configuration was successfully simulated in EPANET, thus validating our results.

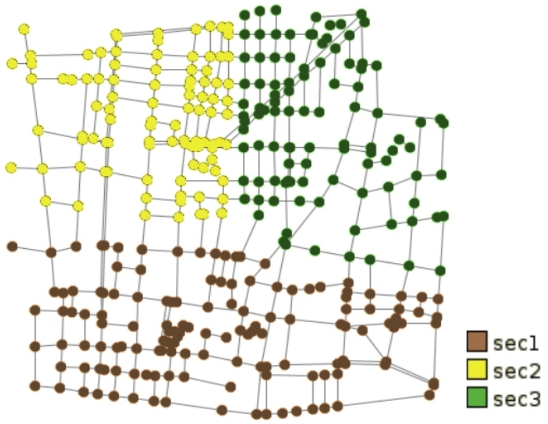

Figure 6.6: WSN division into supply clusters: SSL - Semi-supervised clustering approach

6.1.2 Application of multi-agent clustering

The first part of the proposed MAS clustering algorithm involves the *a priori* assignment of the reservoir and tanks to the different supply clusters. We have tested all the $\binom{6}{3}$ configurations to choose the one with the more suitable partition. Finally the tanks selected to be the algorithm' seeds were $E1$, $E2$ and $E5$.

After 20 runs of the model simulating the partition into hydraulic sectors of this network, the configuration shown in Figure 6.7 has been obtained in 90% of the cases. As a result, three sectors have been obtained that are isolated through 25 cut-off valves of Figure 6.8). These sectors have 187, 101 and 159 pipes.

The NetLogo's output [Wilensky (1999)] shows that the average elevations for these sectors are 153 m for Sector 1, 152 m for Sector 2 and 159 for Sector 3 (see Figure 6.8). As shown by the plot in Figure 6.7 the architecture of the sectors stabilises along the simulation.

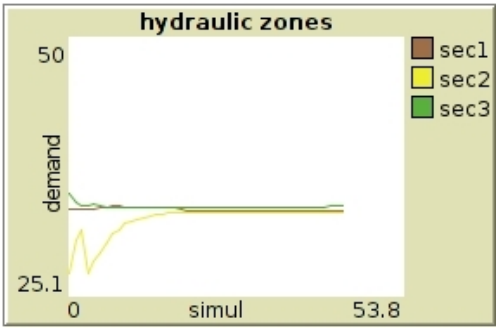

Figure 6.7: Evolution of demand running the MAS clustering algorithm - Average demand by supply cluster

The final results of this first algorithm may be consulted in Table 6.3.

Table 6.3: Description of supply clusters with MAS-algorithm I

sector	n nodes	n pipes	sources	avg. elevation	total demand
sector 1	135	187	E1 + D1 + E3 + E4	153	31
sector 2	96	101	E2	152	29
sector 3	102	159	E5	159	33

Now, the next phase of the process tries to identify interactions based on communication and reasoning regarding the membership and intentions of the agents on areas

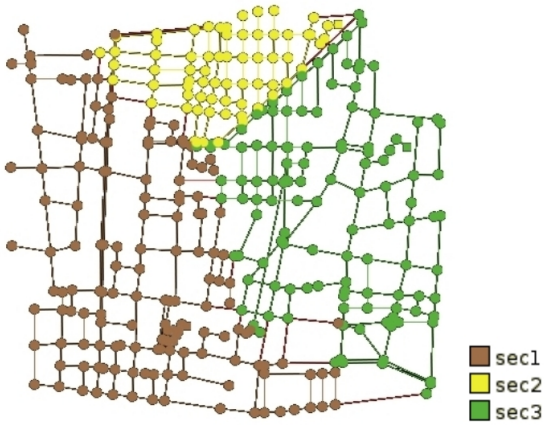

Figure 6.8: WSN division into supply clusters: MAS I - MAS-clustering algorithm I

close to the boundaries. The behaviour of these agents could change following energy premises instead of homogeneity. These changes have consequences in the supply cluster memberships, as can be seen in Figure 6.9. The final results may be consulted in Table 6.4.

Table 6.4: Description of supply clusters with MAS-algorithm II

sector	n nodes	n pipes	sources	avg. elevation	total demand
sector 1	55	62	E1 + D1	157	30
sector 2	89	130	E2 + E4	154	30
sector 3	189	245	E3 + E5	155	30

Comparing the results (Figure 6.10) of both algorithms, we can observe a decreasing size of Sector 1 and a major homogeneity in the elevation and the demand at the different clusters. These last results may offer a major efficiency in the supply. However, instead of 25 cut-off valves used in algorithm I, the negotiating algorithm needs a total of 29 to isolate their supply cluster proposal.

6.1.3 Application of MAS-boost clustering

This approach to solve the WSN division is by a boosting semi-supervised methodology, which iteratively recycles the sampling subgraphs providing multiple clusterings and resulting in a common partition (cf. Chapter 5). The subgraph size should be approximately 30% of the total number of nodes (approximately 100 nodes in this case-study).

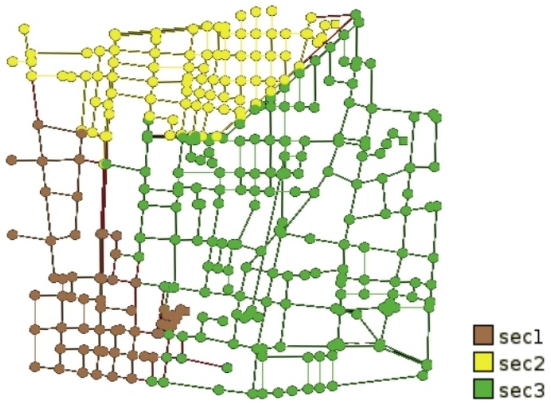

Figure 6.9: WSN division into supply clusters: MAS II - MAS-clustering algorithm II

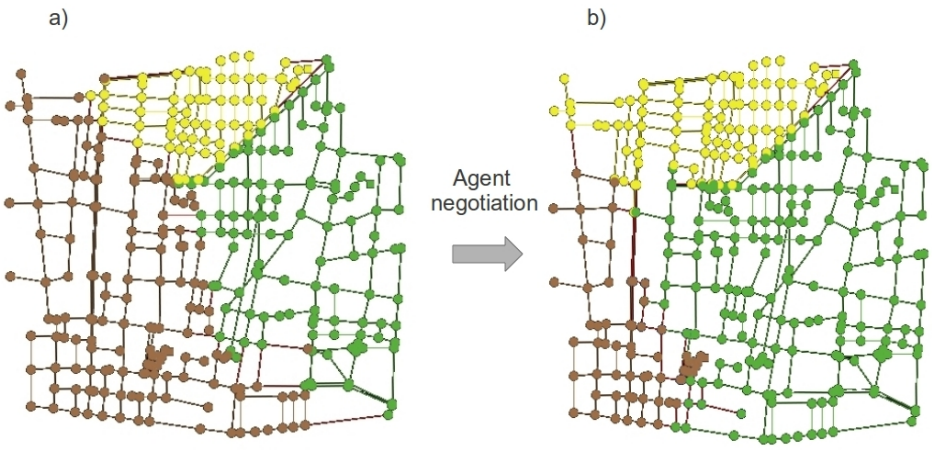

Figure 6.10: Final distribution of supply clusters - a) algorithm I b) negotiating: algorithm II

In addition, it is considered that the value of 20 is the maximum number of boosting iterations, T (this number is enough in this case due to the medium size of the original network). The way to update the weights in the boosting will be that of changing the *node-resistance* value by using equation 5.4 of the process described in Section 5.3 (Chapter 5).

This division assures that each sector is supplied by at least one tank (see more details about average elevation and total demand of these supply clusters in Table 6.5). Twenty-eight cut-off valves are necessary to isolate the different supply clusters*.

Table 6.5: Description of supply clusters with MAS-boost clustering algorithm

sector	n nodes	n pipes	n sources	avg. elevation	total demand
Sector 1	118	155	E1 + D1 + E3	157	37
Sector 2	127	192	E4	154	22
Sector 3	88	129	E2 + E5	154	31

In Figure 6.11 we can see the nodes of the central district of Celaya divided into the three supply clusters proposed by the MAS-boost algorithm.

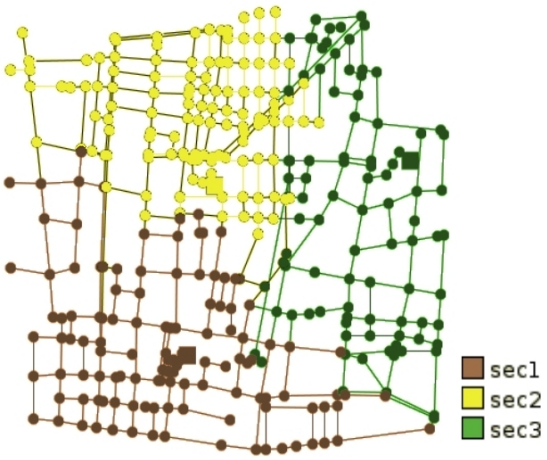

Figure 6.11: WSN division into supply clusters: MAS-boost - MAS-boost clustering algorithm

*These results are calculated with the R Language [R-Development-Core-Team (2010)] function `specc` of the `kernlab` library [Karatzoglou (2006)] along with NetLogo [Wilensky (1999)]

6.1.4 Comparison of the methodologies

In order to compare the above approaches, we firstly calculate the modified silhouette (cf. Chapter 5) in each case. In these terms, MAS clustering algorithm I (MAS I) is the method with high average silhouette (0.35) and it has very few negative weighted nodes (see Figure 6.12 - a). MAS-boost clustering has a silhouette of 0.33 (see Figure 6.12 - b), close to the first positioned method. This has more negative weighted nodes, but in general, the nodes with positive silhouette are represented by MAS-boost better than MAS I. In this first comparison the semi-supervised (SSL) and MAS-algorithm II (MAS II) approaches obtain poor results, with average silhouette widths of 0.23 and 0.21 respectively.

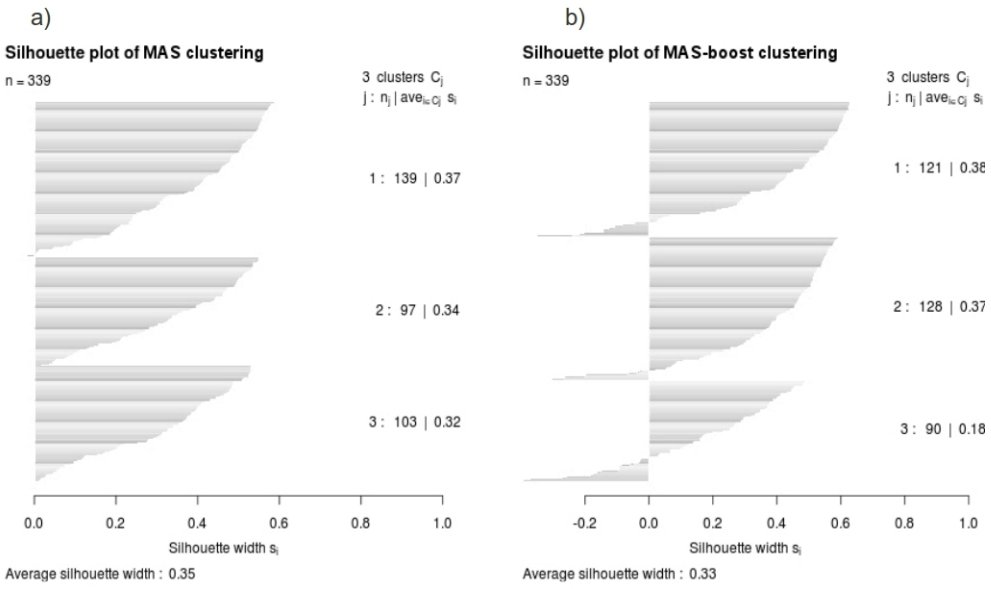

Figure 6.12: Best silhouette width of the proposed clustering algorithms - a) MAS algorithm I - b) MAS-boost clustering algorithm

There are other considerations to take into account along with silhouette criterion. It is the case of the measurement (elevation and demand) dispersions in each cluster. The reason is that one of the targets of the proposed sectorisation is searching homogeneous areas to these inputs. Table 6.6 has a comparison in this approach.

Table 6.6* shows the average *elevation* and the total of *demand*, both calculated

*As we see in the total sums of the table, the computations may have slight rounding errors associated. This fact is irrelevant to decision making and to the final conclusions.

Table 6.6: Comparison of the clustering algorithms presented in this thesis

	valves	avg./RMSE elev.	total/RMSE demand	silh.	dist.
SSL:	26				
Sector 1		157.34 / 1.33	35.38 / 7.03	0.15	351
Sector 2		154.75 / 2.58	25.33 / 5.44	0.53	249
Sector 3		155.54 / 0.13	31.04 / 4.86	0.17	255
Total		155.90 / 2.90	91.75 / 10.11	0.23	285
MAS I:	25				
Sector 1		153.51 / 0.19	30.52 / 4.65	0.37	308
Sector 2		152.39 / 2.08	29.03 / 4.59	0.34	353
Sector 3		158.76 / 1.53	32.60 / 9.80	0.32	323
Total		155.64 / 2.59	92.15 / 11.77	0.35	328
MAS II:	29				
Sector 1		156.86 / 1.80	30.02 / 5.64	0.21	186
Sector 2		153.53 / 1.17	31.00 / 6.23	0.28	336
Sector 3		154.65 / 1.07	30.31 / 5.23	0.14	339
Total		156.12 / 2.40	91.33 / 9.89	0.18	287
MAS-boost:	28				
Sector 1		156.70 / 0.21	37.20 / 5.37	0.33	315
Sector 2		154.12 / 1.58	22.33 / 4.37	0.37	276
Sector 3		153.77 / 1.83	31.09 / 5.39	0.18	70
Total		155.73 / 2.42	90.62 / 8.76	0.33	220

by sector. These measurements have associated their root mean square error (RMSE), respectively. Column *dist* contains the minimal value of the maximum distance (in geographic coordinates) between the nodes of each sector and their sources. Column

silh. shows the average silhouette width per sector, and column *valves* is the number of pipes proposed to be cut by each algorithm approach.

MAS-boost sectorisation takes advantage from the other approaches in cluster homogeneity and accuracy in the calculus of elevation and demand per cluster. Their associate average silhouette width is 0.33, only being slightly improved by MAS I algorithm (with an average silhouette width of 0.35). Regarding the number of necessary cut-off valves, all the results are within the same range of values (around 25 - 29). Thus, our candidate supply cluster configuration may be the one resulting from MAS-boost algorithm.

About the results of MAS-boost configuration, we can observe the layout of major-consumer points within Sector 1 (see Figure 6.13). The average elevation of the sectors is equally distributed, but Sector 3 may be the one with more internal difference in their elevation (due to a major variability of this measurement).

Figure 6.13: Distribution of major-consumption points by sectors - MAS-boost algorithm

Pipe materials (see Figure 6.2) are not involved in our cluster configuration. This occurs because this variable along with pipe ages or water uses in each cluster, have not been inputs in the current algorithm performance. Nevertheless, it is possible to observe a relationship between this clustering configuration and the relative importance of the nodes in the network if we run the PageRank algorithm proposed in Section 7.3 of Chapter 7. Once again, it is proposed to include PageRank vector as another part of

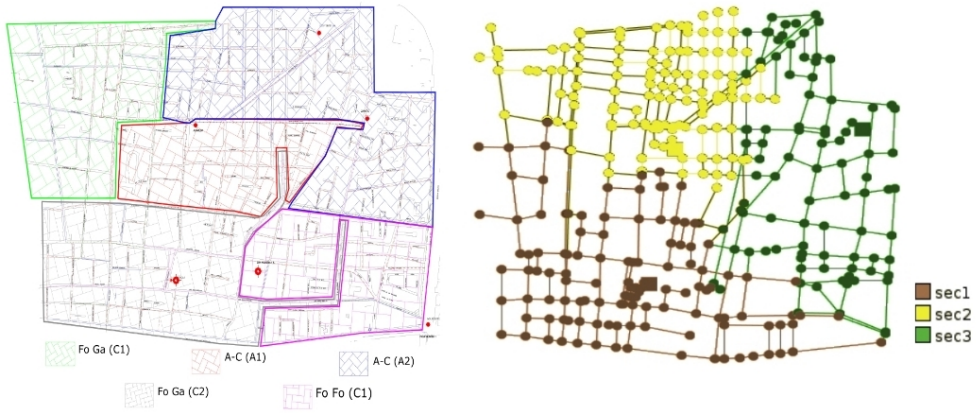

Figure 6.14: Comparison of sectors and pipe materials - MAS-boost algorithm

the input information in future supply clusters configurations. Figure 6.15 (left) shows the PageRank distribution in the case-study. Comparing this result with the right hand side of the same Figure 6.15, we can observe that Sector 3 concentrates 5 out of a total of 6 nodes with 'medium-high' PageRank.

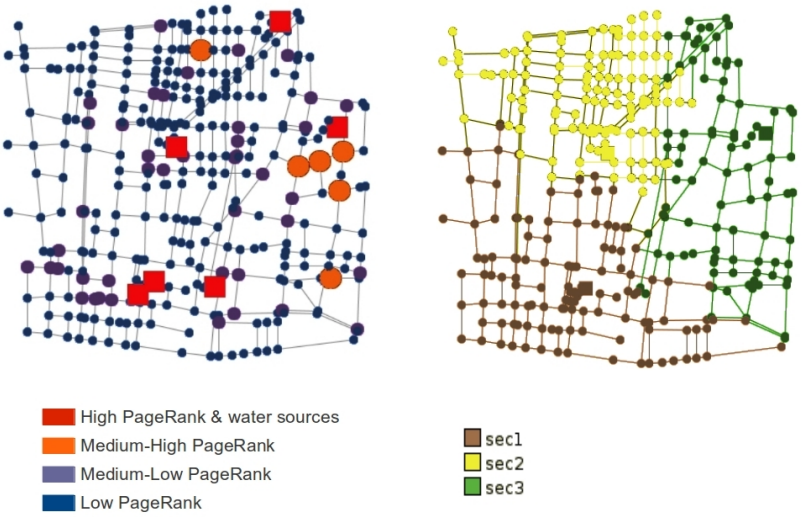

Figure 6.15: Graphical distribution of a WSN PageRank - comparison with the supply clusters

6.2 Summary and comments

This chapter completes the description of the case-study of the central hydrometric district of Celaya (Guanajuato, Mexico), introduced in Chapter 2; we have checked the different clustering algorithms (proposed in chapters 3, 4 and 5). All of them have shown an efficient performance and found suitable proposals to posterior sectorisations, within a target of demand and elevation homogeneity for each supply cluster. Other goals, such as cluster memberships of each node, the minimum number of cut-off valves used to encapsulate the sectors or the minimal distance from the water sources (tanks and/or reservoirs) to the farthest node to be supplied by them, are taken into account too. In this case, MAS clustering I and MAS-boost algorithms outperform the other methodologies. A detailed study was carried out with MAS-boost clustering configuration. The role of maximum demand points and the relative importance of each node (calculated following the PageRank introduced in Chapter 7, Section 7.3) explain part of the resulting partition.

7

Water network management based on supply clusters: working proposals

Once a water supply network (WSN) is divided into supply clusters, it is of key importance how to manage the system in a better way. This implies taking advantage of the reduced size of these working areas and their efficient way to be implemented. Besides, the methodologies employed to obtain these results may be adapted and/or extended to address management tasks. This chapter explores in some of these themes showing three different applications: predicting water demand, analysing system anomalies and ranking nodes in a WSN. The are two common points to these issues. Firstly, it is possible to obtain better modelling using only one sector for each study than the whole network. Second, in every case a suitable framework (usually related to kernel methods) to take into account all kind of available information is introduced. In addition, a management issue such as ranking nodes may be used in both cases, to study characteristics of only one sector and to support the final supply network configuration.

This chapter is organised as follows. Section 7.1 introduces the paradigm of water demand forecasting. A case study based on water demand historical hourly data from a hydraulic sector in a city is presented to evaluate the methodology employed and to provide comparisons between various predictive models. In Section 7.2 a methodology to detect the possible types of novelties within time series data is presented. We also develop a way of classifying these anomalies and to discuss their causes. Next Section

7.3 introduces the ranking algorithm proposed in this thesis. It is based on Google's PageRank. This algorithm is related to spectral methods and adapted to study a water supply network. The possibility to take into account PageRank in future supply cluster configuration is proposed. A Section of summary and conclusions close the chapter.

7.1 Predictive models of water demand*

The most important factor in planning and operating a water distribution system is satisfying consumer demand. This means continually providing users with quality water in adequate volumes at reasonable pressure, and so ensuring a reliable water distribution system. Efficiently operating and managing a water supply system requires short-term water demand forecasts; and the estimation of future municipal water demand is central to the planning of a regional water-supply system [Zhou *et al.* (2002)]. Water demand forecasting is becoming an essential tool for the design, operation, and management of water supply systems in activities such as: planning new developments or system expansion; estimating the size and operation of reservoirs and pumping stations; determining pipe capacities; and handling another urban water management issues (pricing policies, water use restrictions, etc.).

Long-term forecasting is required mainly for planning and design; while short-term forecasting is useful in operation and management. As mentioned by Bougadis *et al.* (2005), short-term demand projections help water managers make better informed water management decisions when balancing the needs of water supply, residential/industrial demands, and stream flows for fish and other habitats. Short-term demand forecasts help utilities plan and manage water demands for near-term events [Jain & Ormsbee (2002)].

In this section we tackle this kind of temporal landscape, taking our predictive output and using it as input of a previously calibrated water model (i.e. in EPANET [Rossman (2000)]). Availability of hourly predictions of water demand into a calibrated mathematical model is crucial due to a number of reasons:

*Adapted from Predictive models for forecasting hourly water demand, Journal of Hydrology 387 (1-2), 141-150, 2010.

- From an operative point of view, it enables water managers to determine optimal regulation and pumping schemes to supply the predicted demand. The aim is to improve the energetic efficiency through lower pumping energy consumption.
- From the quality point of view, the more suitable combination of water sources to obtain a given standard in the supplied water may be selected.
- From the vulnerability point of view, the comparison between the predicted and the real flow measurements can help pinpoint possible network failures (water leaks and pipe bursts). This provides the first step of a procedure for establishing early warning management.

7.1.1 Exploratory analysis of the data

The current study site is located in a hydraulic sector (or a hydraulic zone) in a city. It has a population of approximately 5000 consumers and an extension of nearly 8 km^2. The water demand average is 19 m^3/h with an associated standard deviation of approximately 8 m^3/h. The decision to use only one hydraulic sector for our study is motivated by the following: homogeneity and utility of the results; elimination of sources of bias; avoidance of the impact of a small set of consumers that may incorrectly bias the forecasts due to unusual consumption profiles.

The complete water supply network under study has been divided into hydraulic zones. Starting at a treatment plant, a water main distributes water to the sectors. Each sector has one or two sources and may or not have an output. A number of control valves isolating or communicating each zone with the whole network is essential. Water consumption in each sector is registered by flowmeters, and registered data are sent by radio-frequency to a central database for storage and posterior analysis. For this work, field measurements were collected from January 2005 through April 2005 on a hourly basis. In addition to water consumption values, we also have information concerning daily values of climate variables: temperature in Celsius, wind velocity in km/h, millimetres of rain, and atmospheric pressure (mean sea level pressure, measured in millibars).

All these factors are connected with the water demand behaviour. Temperature is the more relevant factor because it directly influences multiple sources of water consumption such as showers, water for gardens, etc. But water consumers also respond

to the occurrence of rainfall and other climate variables [An *et al.* (1995)]. Regarding this connection, Maidment & Miaou (1986) are critical about the linearity between water demand and weather variables. They suggest that rainfall has a dynamic effect in the sense that it reduces water demand initially, but the effect diminishes over time. These nonlinearities are also of concern to many authors, among others Arbúes *et al.* (2003) and Gato *et al.* (2007).

Previous analysis of data similar to our case included a distinction between weekdays and weekends. In the present study, we have instead used day of the week. The justification for this change lies in the observation that there is a clear difference in the demand profile for the different days; namely, Saturdays are clearly different from Sundays. This daily profile can be observed in Figure 7.1. It shows the average water consumption for the 24 hours of each day. Each average value was calculated from all the available data (i.e. an eight week period).

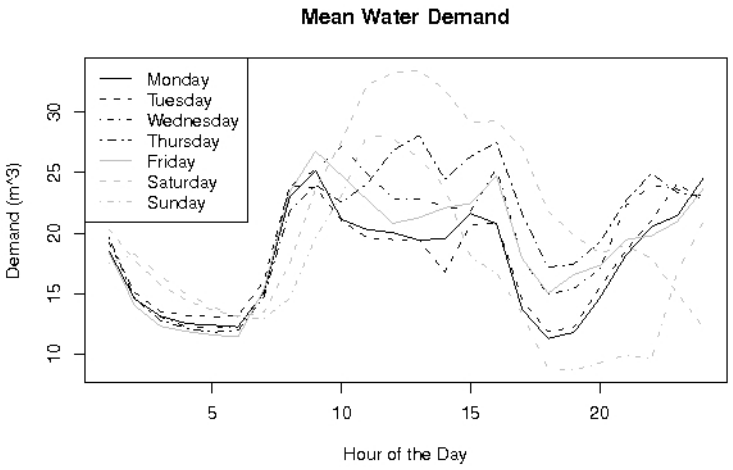

Figure 7.1: Evolution of the mean water demand - 24 hours of each day

The curves show a similar behaviour during the early morning. All curves grow from 6:00 am till 10:00 am. From 10:00 am to 4:00 pm, the behaviours are very different depending on the day. In the afternoon, all the curves have the same trend: first decreasing and then increasing (except on weekends). The maximum water demand values (Figure 7.2) may be useful for pumping and other water management actions. This, together with the information of the slopes in different time windows, is important to add to the forecasts.

In this way, the model based on the pattern curve [Herrera *et al.* (2010d)], may increase the sensitivity in detecting peaks in water demand. We have also carried out an analysis of the relationship between the weather information and the water demand. Figure 7.3 provides such an analysis in the form of paired graphs for several weather-related variables and also water demand. The left hand graphs show the time plots of the respective time series, while the right hand graphs are box plots of the values of each variable across the period under analysis. The box plots provide an idea of the distribution of the values of the variables.

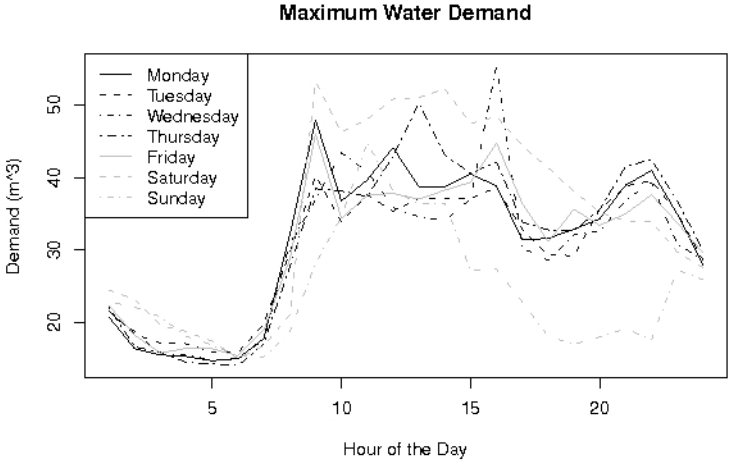

Figure 7.2: Maximum water demand per hour - 24 hours of each day

The box plots in Figure 7.3 show a very limited variability of the rain values, as well as the asymmetry in the distribution of the wind velocity values. The time plots of the various variables reveal different trends and enable us to observe, for example, the inverse association between the atmospheric pressure and the temperature for these months. We also observe that there is little water demand in the beginning of this period, coinciding with low temperatures. There is then a growth in the consumption followed by a growing variability period (coinciding with the irregular weather of the first days of spring). From these graphs it seems obvious that there is some influence of the temperature on water demand, as well as of the wind velocity and the volume of rain (these influences are shown significant after applying a Spearman rank correlation test). The influence of a rainy day in water demand is of special interest: just the day that it rains this demand increases on average and goes down this level the next day.

This may be attributed to the fact that people tend to stay longer at home during rainy days.

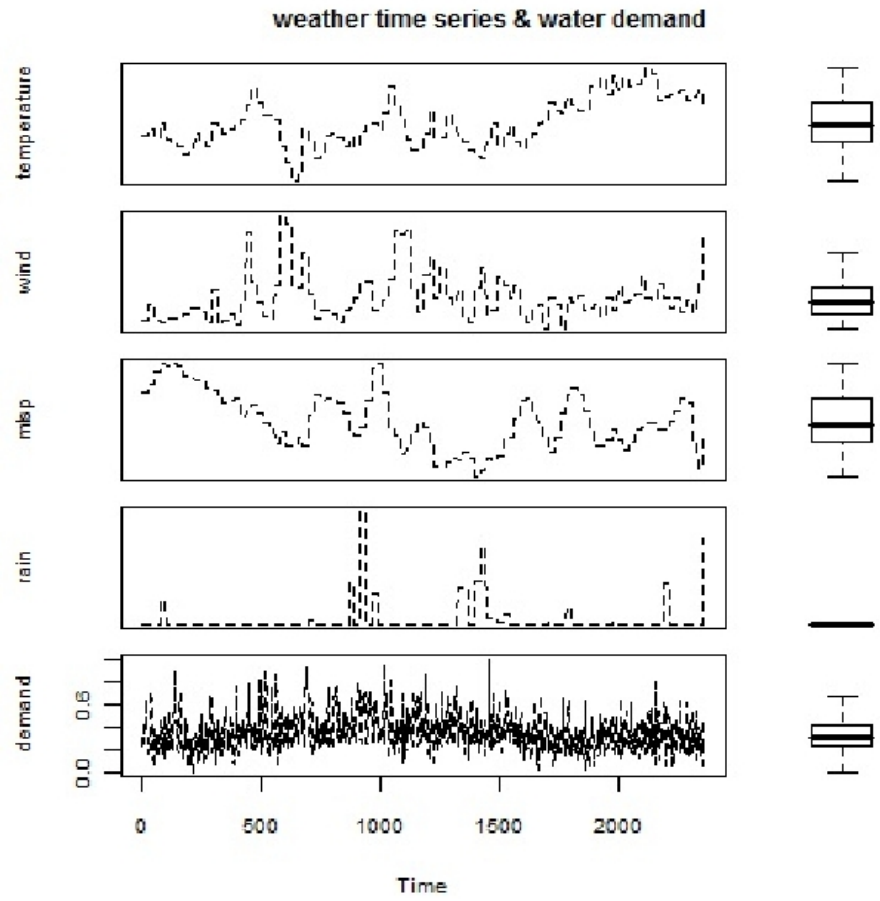

Figure 7.3: Impact of weather variables on water demand - Graphical visualisation

Given the multivariate time series database described previously, it is possible to use a multiple linear regression method to obtain a model that predicts the value of the future water demand from the past observed demand. From the point of view of multivariate time series theory, the first method of approaching the problem is to use vector AR (VAR) models, or state-space models; Durbin & Koopman (2001); Gilbert (2006); Wagner (1999) are good references for the fundamentals of these methods. Nevertheless, the behaviour of the autocorrelation function of filtered and differenced time series shows clear nonlinearities. In this context, we will approach the problem

using nonlinear forecasting methods, which will also allow to develop more flexible regression models, including for instance, the weather time series.

7.1.1.1 The prediction task

With respect to the target variable we have selected the water demand in the next hour as our forecasting goal. This means that at time i we will ask our candidate models to forecast the water demand for time $i + 1$ hour. Regards the predictor variables used to forecast this future water demand we have used our exploratory analysis described in the previous section to guide our selection. In this context, we have selected the current hour water demand, the previous hour water demand, the water demand for our target hour on the previous week, given the day of the week regularities that were observed, and also the current weather time series values. More formally, we are trying to approximate the unknown multiple regression function,

$$W_{i+1} = f(W_i, W_{i-1}, W_{i-7\times24+1}, t, vv, pnm, pt) \tag{7.1}$$

where W_i is value of the hourly water demand time series at hour i, and t, vv, pnm and pt are the last known values of the weather time series: temperature, wind velocity, atmospheric pressure and rain, respectively (Figure 7.3).

We have considered artificial neural networks (ANN), projection pursuit regression (PPR), multivariate adaptive regression splines (MARS), random forests (RF) and support vector regression (SVR). Apart from these models, we also propose a simple model based on the weighted demand profile resulting from our exploratory analysis of the data. All models were tested several times with different parametric configurations (cf. Section B.1 of Appendix B).

7.1.2 Experimental study

In this section we present the results of a series of experiments using the data presented in Section 7.1.1, namely for the predictive task described in Section 7.1.1.1. The main objective of these experiments is to compare different variants of the described models on the task of forecasting the next hour water demand (W_{i+1}) for our case study.

Given the time dependency between the observed values of water demand, we have been particularly careful with the experimental methodology we have used to estimate

the predictive performance of the different models. In effect, for this type of data, estimation processes that involve a random resampling step that changes the order of the observations, provide biased estimates of the predictive performance. These processes are based on resampling procedures that change the (time) order of the data due to their random nature. This may lead to models being obtained with data that is more recent (in terms of the time line) than the data on which they are tested. As the basic assumption of our prediction task is that it is possible to predict the future water demand looking at the past patterns of this variable, the use of the models with these setups will inevitably lead to over-optimistic estimates of the predictive performance of the models [Torgo (2010)]. In this context, we have followed a different estimation procedure in our experiments. Namely, we have used a Monte Carlo simulation designed to estimate the predictive performance of a model obtained on a set of data $D_{i\cdots j}$, on another set of data occurring later, i.e. $D_{m\cdots n}$, with $m = j + 1$. Giannella *et al.* (2003) and Lin *et al.* (2005) are two good references in the evaluation of models based on time granularity.

7.1.2.1 Monte Carlo estimates

Monte Carlo methods are a class of computational algorithms that can be used, among others, to obtain estimates of any variable by random repetition of a simulation experiment. In our concrete application we are interested in estimating the value of certain predictive performance evaluation statistics (cf. Section 7.1.2.2) that should be measured on a set of test data using models obtained on another set of data that occurred before in time.

In our experiments we have used the following experimental methodology to obtain our estimates. We start by randomly selecting a point in time, t, within the period for which we have data available*. We then use the previous $t - 1344$ data points to construct a training set with which the candidate models will be obtained. This means that we will use a training window of eight weeks (1344 hours) of data. The obtained models will then be tested on the subsequent 2 weeks, i.e. till point $t + 336$. This evaluation process is repeated 20 times to obtain 20 different estimates of the evaluation

*We actually have to reserve some initial and final periods of the data, to ensure that there is enough back data for training the models and also enough future data for testing purposes. Moreover, we have reserved the final week of our data set for our final sample test of the best model.

statistics for each model, each obtained at different points t in time to ensure a certain robustness of the estimates. The Monte Carlo estimates are obtained by averaging these 20 point estimates. To test the statistical significance of the observed differences among these averages we have carried out paired non-parametric Wilcoxon tests (cfr. Table 7.1).

7.1.2.2 The evaluation statistics

In terms of measures to assert the predictive accuracy of our candidate models we have selected the following,

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2}, \tag{7.2}$$

where n is the number of test cases, O_i the true value of the target variable for the case i, and P_i the respective prediction of the model for the same case, and

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|O_i - P_i|. \tag{7.3}$$

Both metrics are expressed in the same units as the target variable. We have also used two non-dimensional evaluation metrics: the Nash-Sutcliffe efficiency (Equation 7.4), and a modification of Nash-Sutcliffe (Equation 7.5), which are more sensitive to systematic errors Krause *et al.* (2005), and so may offer more information on the systematic and dynamic errors present in the models.

$$E_j = \frac{\sum_{i=1}^{n}|O_i - P_i|^j}{\sum_{i=1}^{n}|O_i - \bar{O}|^j}, \tag{7.4}$$

$$d_j = \frac{\sum_{i=1}^{n}|O_i - P_i|^j}{\sum_{i=1}^{n}(|P_i - \bar{O}| + |O_i - \bar{O}|)^j}, \tag{7.5}$$

with $j \in \mathbb{N}$.

The range of these criteria lies between 0 (perfect fit) and ∞. An efficiency of more than one indicates that the mean (median) value of the observed time series would have been a better predictor than the model. In particular, for $j = 1$, the overestimation of the demand water peaks is reduced significantly and results in a better overall evaluation Krause *et al.* (2005).

7.1.2.3 Model building

For each trial of a model in the Monte Carlo process described in Section 7.1.2.1, we are given a training set formed by 8 weeks of data and asked to make predictions for the next 2 weeks. In this context, the natural approach would be to obtain a model with the 8 weeks and then use it for obtaining the predictions for all data in the 2 test weeks. This procedure is acceptable in applications where we have no reason to suspect that the dynamics of the target time series varies with time. If that is the case, the model should maintain its accuracy during the two weeks. On problems with clear changes in regime (caused by several factors like weather or seasonal effects), this standard approach might not be the most adequate. In Section 7.1.2 we have observed that our target problem has clear changes of regime along the time. In this context we follow different approaches to obtain the models.

The growing window strategy (see top graph in Figure 7.4) consists in obtaining the model using the initially available data (8 weeks) and then use it for making predictions during a limited time window. When this window ends the test cases within this period are added to the initially available data and a new model is obtained with this larger data set. This new model is used to obtain predictions for another limited time window of the same size, and so on, till predictions for all test period are obtained. This means that this model building algorithm in effect develops several models (with an increasing number of observations), to obtain predictions for the test period. In the limit, each time a single prediction is made, a new model is obtained.

The sliding window strategy (bottom graph in Figure 7.4) again uses several models, with each model being used for predictions only during a limited time window. The difference for the growing window strategy lies in the fact that all models are obtained with a training window of the same size. So fresher data is added but at the same time the older data is being removed at the same proportion so that a constant size training window of 8 weeks is used to obtain all models. The motivation is to obtain models using only the most recent data.

In our experimental comparisons every model variant was obtained using different model building strategies. Namely, we have experimented with 3 different growing window approaches with the size of the test window for which the model is kept the same set to 1, 12 or 24, i.e. 1 hour, half day or a full day. With sliding windows we

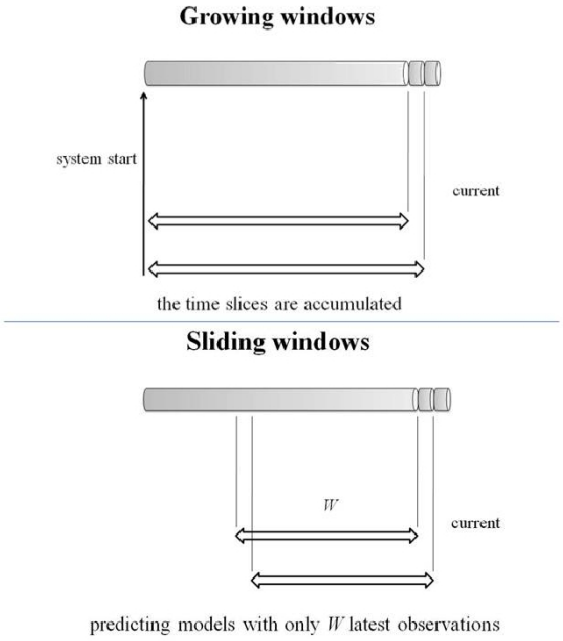

Figure 7.4: Two approaches to model building for time series prediction - Source: Herrera *et al.* (2010d)

have followed an equal approach in terms of timings for model updating, but for these methods the training window was always kept with the same size (8 weeks).

7.1.2.4 Results of the Monte Carlo comparisons

We first analyse the results of the Monte Carlo experimental comparison of all model variants described in Subsection 7.1.1.1 and detailed in B.1 of Appendix B. We provide a summary of our experimental results in the next subsections looking at them from different perspectives.

7.1.2.5 Comparing the model building strategies

In this section we address the question of which of the six different model building strategies that were considered, is more adequate for each of the modelling techniques. The six variants differ on the algorithm used (growing or sliding) and on the model updating frequency (1, 12 or 24 hours).

Table 7.1 shows a comparison in terms of MAE of the two top models for each of the building strategies across all learning algorithms. The final line indicates the statistical

Table 7.1: Difference between the best growing and sliding approaches.

		wPatt	nnet	mars	rf	svr	ppr
First	Strat.	slide	slide	grow	slide	slide	slide
	Up.Freq.	1	12	1	1	1	1
	MAE	9.31	6.24	4.34	4.37	4.33	4.34
Second	Strat.	grow	grow	slide	grow	grow	grow
	Up.Freq.	1	24	1	1	1	1
	MAE	9.62	6.3	4.37	4.37	4.33	4.34
	Signif.	++		+			

significance of the difference between the two best variants. Two signs represent a difference significant with the confidence level of 0.01, one sign represents a level of 0.05, while no sign represents lowest levels of significance.

Only two of the observed differences are statistically significant. Nevertheless, we can observe that with a single exception (MARS) all models obtain the best score using a sliding window approach*. Moreover, again with a single exception (NNET) all best scores are achieved using a model updating frequency of 1 hour. This is a clear indication of the existence of fast regime changes in the water demand time series of our problem.

In summary, the model building algorithm only seems relevant for the weighted pattern-based and MARS algorithms, although with opposite conclusions regarding which is the best. On the contrary, with respect to the model updating frequency, our results consistently indicate advantages on using a fast pace, with the best results almost always being obtained for 1 hour.

7.1.2.6 The best model variants for each evaluation metric

We now consider the question of which are the best variants in terms of the four used evaluation metrics. Figure 7.5 includes the distribution of the values on these statistics obtained by the best model variants of each algorithm, on the twenty repetitions of the Monte Carlo experiment. This distribution is presented by means of box plots.

The results on this figure show a group of models (SVR, Random Forests, PPR and MARS) as clearly better than both NNET and the weighted pattern-based model.

*Some results seem equal due to rounding effects, but only for SVR that really happens.

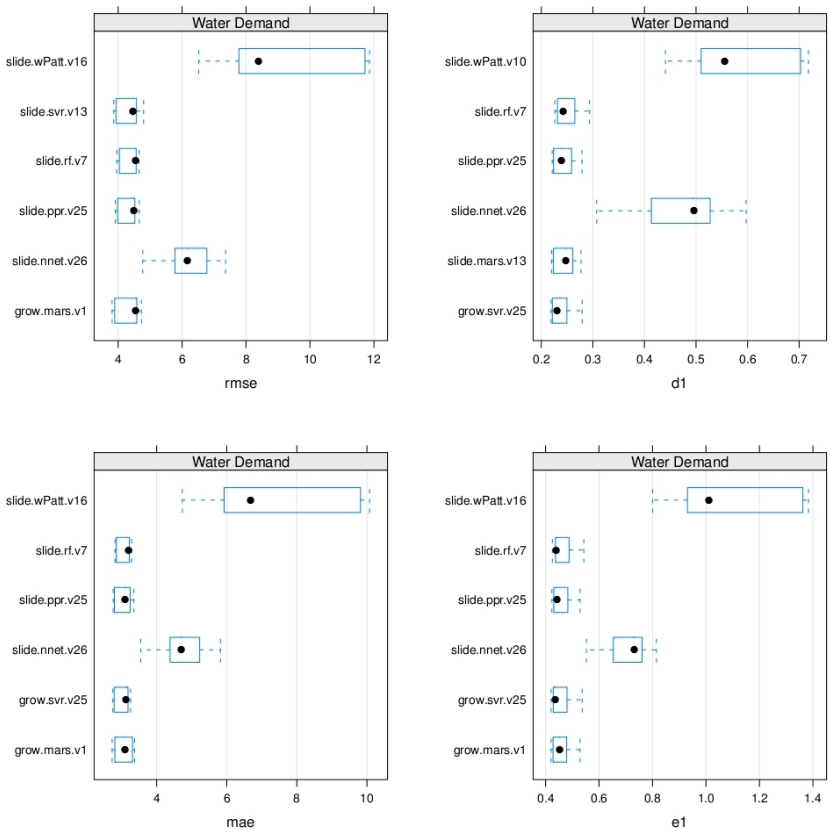

Figure 7.5: The best model variants for each evaluation statistic - A Monte Carlo comparison

Table 7.2: The best overall results.

Stat.	Score	Model ID	Model Parameters
MAE	3.0284422	grow.svr.v25	GrowWind.; Upd.Freq.=1; cost= 200; gamma =0.0005
RMSE	4.3314537	slide.svr.v13	SlideWind.; Upd.Freq.=1; cost= 150; gamma =0.001
e1	0.4537398	grow.svr.v25	GrowWind.; Upd.Freq.=1; cost= 200; gamma =0.0005
d1	0.2365621	grow.svr.v25	GrowWind.; Upd.Freq.=1; cost= 200; gamma =0.0005

Moreover, these differences are statistically significant with 99% confidence. It is also interesting to remark that these top models achieve rather consistent results across the twenty repetitions, as it can be seen by the tight boxes in the box plots.

Table 7.2 shows the top models on each statistic according to the Monte Carlo estimates. They are all variants of the SVR algorithm.

In terms of statistical significance, the difference between the top models shown in Table 7.2 and the remaining best model variants shown in Figure 7.5 is most of the times significant with a confidence of 99%. The exceptions are the following. For the MAE statistic there is lack of statistical significance to the scores obtained by the two MARS variants and also the PPR variant. With respect to the $RMSE$ metric the results of the best model are only significantly different from the results of the weighted pattern-based models and the NNET. In terms of the $e1$ statistic the conclusion is the same as for MAE. Finally, for the $d1$ metric the differences are always statistically significant (although only with 95% confidence with respect to the other SVR variant).

Summarising our results we can conclude that the SVR model seems clearly more adequate to this problem of forecasting the next hour water demand. Still, competitive results are obtained by MARS and PPR on this problem, as well as Random Forests.

7.1.2.7 Using the best model

We have intentionally left the last week of data out of our Monte Carlo experimental comparison. The goal is to use the model selected as the best in our comparisons to

Table 7.3: The results of the best model for the final week of data.

Train Size	MAE	$RMSE$	$e1$	$d1$
8 weeks	3.2555276	4.3831834	0.5551210	0.2938535
all data	3.3156837	4.4407355	0.5666435	0.2999423

obtain predictions for this last week. This section describes this small experiment with the "grow.svr.v25" model.

Using again the previous 8 weeks of data (1344 hours) we have obtained a SVR model with the parameters $cost = 200$ and $gamma = 0.0005$. Then, using a growing window approach with an updating frequency of 1 hour we have obtained the predictions of these models for the last week of data. In terms of the evaluation statistics the results are shown on the first line of Table 7.3. These scores are slightly worse than the estimates obtained by the Monte Carlo estimation method. Still, they are comparable. We then carried out the same experiment but instead of using only the 8 previous weeks of data, we have decided to include all previous data available. The results of this model are shown on the second line of the same table. As we may observe, the inclusion of more data made the results worse. This again confirms the existence of clear regime shifts on this data set, which makes the inclusion of too old data harmful for the prediction models. This result also provides evidence of the utility of the model building strategies we have used in our study.

In Figure 7.6 we plot the true values of the demand for this last week and also the predictions of the best model according to our experimental comparison. With few exceptions the predicted values seem to follow the trend of the observed water demand.

7.2 Anomaly causes in a water supply system*

Time series novelty or anomaly detection refers to automatic identification of novel or abnormal events embedded in normal time series points. In the case of water demand, these anomalies may be originated by external influences (such as climate factors, for example) or by internal causes (bad telemetry, pipe bursts, etc.). This section will focus on the development of markers of different possible types of anomalies in water

*Adapted from Scrutinizing changes in water demand behavior, Lecture notes in Control and Information Sciences, pp. 305-313, Springer, 2009.

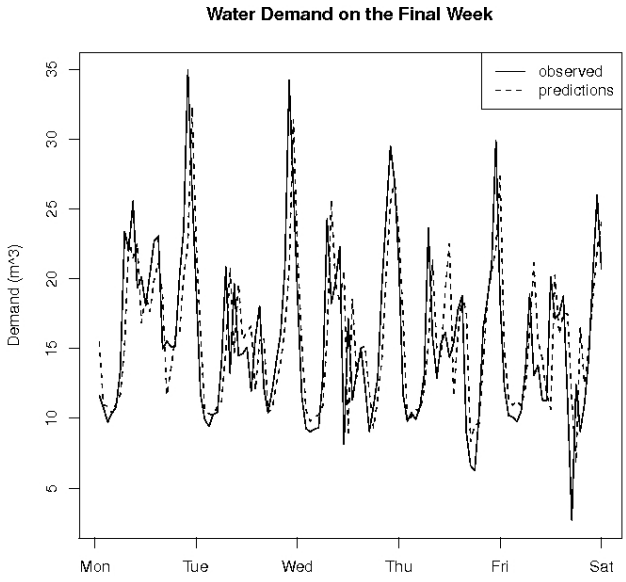

Figure 7.6: Difference between forecast and observed water demand values for the last week - A SVR approach

demand time series. The goal is to obtain early warning methods to identify, prevent, and mitigate likely damages in the water supply network, and to improve the current prediction model thorough adaptive processes. Besides, these methods may be used to explain the effects of different dis-functions of the water network elements and to identify zones especially sensitive to leakage and other problematic areas, with the aim of including them in reliability plans. In this chapter, we use a classical Support Vector Machine (SVM) algorithm to discriminate between normal and anomalous data. SVM algorithms for classification project low-dimensional training data into a higher dimensional feature space, where data separation is easier. Next, we adapt a causal learning algorithm, based on the reproduction of kernel Hilbert spaces (RKHS), to look for possible causes of the detected anomalies. This last algorithm and the SVM's projection are achieved by using kernel functions, which are necessarily symmetric and positive definite functions.

The anomaly detection of water demand time series aims to correct likely data errors in measures from telemetry systems. These systems are used by most water companies in big cities for control and operation purposes. This will allow more accurate

estimations that can be used to immediately detect severe anomalies, such as service disruption. Simultaneously, it can identify more rapidly light anomalies which can develop insidiously and progressively [Izquierdo *et al.* (2007)]. If no errors are found in data, the novelties significance is the occurrence of some physical change in the water supply network or in the demand behaviour caused by external influences, such as climate factors. Changes in time series behaviour may exhibit permanent or transitional effects. The causes of these are diverse, but could be divided into external and internal causes. Examples of the first class are weather or calendar factors. For the internal causes one can have wrong telemetry readings, water leakage or failure of one or more valves. There is a need to divide the problem into two phases: anomaly detection and action taking. In this way, one can obtain early warning methods to identify and mitigate likely damages in the water supply network or to improve the current prediction model thorough some adaptive process.

To distinguish between normal and abnormal deviations, novelties will be sought in three specific cases: when data loggers identify a disruption of service, when the discrepancies between the last observations and their prediction are significant, and when the last observations lack expected random characteristics. Here, we consider working with sliding time windows to include all possible cases. The sliding window method is based on a window size W; only the latest W observations are used for detection. As an observation arrives, the oldest observation in the sliding window expires. An alert processing method based on SVMs will be proposed to extract trends and to highlight punctual discrepancies between observed and predicted data [Rocco & Zio (2007)]. To look for possible causes of the detected anomalies, we propose using the recently developed causal learning algorithm based on the reproduction of kernel Hilbert spaces [Sun *et al.* (2007)]. By using this methodology the statistical dependences can always be detected by correlations after the data are mapped into an appropriate feature space. The algorithm is an improvement of the inductive causation (IC) algorithm [Spirtes & Gylmour (1991)], which generalises in several ways. The control of the consequences of novelties in the earlier stages can avoid, among other things, economic and water losses, which are of great importance from the point of view of water as a scarce resource. This section will focus on the development of markers of the different possible types of anomalies in water demand time series, explaining

their causes, and proposing a feasible integration mechanism in the prediction system. Figure 7.7 summarises the process.

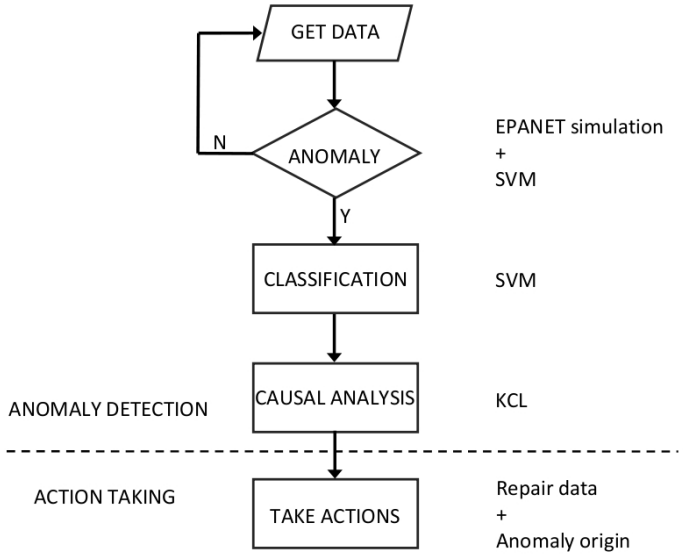

Figure 7.7: Scheme of the methodology proposed - Source: Herrera *et al.* (2009b)

Detecting novel events is an important ability of any signal classification scheme. This is the main reason for several models of novelty detection, which have proved to perform well on different data, to exist. It is clear that there is no single best model for novelty detection, and success depends not only on the type of method used but also on the statistical properties of the data handled. Thus, several applications have been published in the literature with the goal of detecting and classifying possible outliers or abnormal data. In the last years, Neural Networks approaches [Augusteijn & Folkert (2002)] have been replaced by Support Vector Machines applications in this regard [Ma & Perkins (2003); Rocco & Zio (2007)]. As different alternatives, other methods include the control charts, proposed by Nong & Qian (2003), and the techniques based on fuzzy rough clustering, tested by Chimphlee *et al.* (2006), to increase the detection rates and reduce false positive rates in the intrusion detection system. Herrera *et al.* (2007) have proposed hybrid nonlinear models for interpolation in the case of having problems with telemetry lectures of water consumption. Izquierdo *et al.* (2007) have presented a neuro-fuzzy approach to fault detection in water supply systems.

In addition to novelty detection, we also aim to identify its causes. Pearl (2000) has shown that, under reasonable assumptions, it is possible to get hints about causal relationships from non-experimental data. Schölkopf & Smola (2002) have proposed the idea of measuring dependences by reproducing kernel Hilbert spaces. Sun *et al.* (2007) and Fukumizu *et al.* (2007) have worked on an algorithm describing the causal learning method, which we will follow in this chapter. Being able to establish the cause-effect relationships in a water supply environment in the presence of anomalies, would certainly produce better understanding of the demand behaviour.

7.2.1 Methodology

To obtain abnormal data (anomaly events) in an easy way and to train correctly the Machine Learning procedures, we propose working with our real system replicated by its EPANET [Rossman (2000)] model. This way, we can run the water demand simulations under different novelty scenarios and check the response of our methodology. The next step will be to detect the abnormal data. Then, by using a kernel-based causal algorithm we will try to establish the causes of the observed anomalies.

7.2.1.1 EPANET simulation

The above methodology is tested on the simulated consumption of water by using EPANET. The first premise is to work with a correct pattern demand curve. We propose generating curves by using the current model for prediction. For the sake of simplicity we use a simple and novel weighted pattern-based model for water forecasting, which has been tested with very good results [Herrera *et al.* (2010d)]: this method is based on the pattern of the demand, which considers its seasonal properties.

This proposal contains two components: a first part that reflects the seasonal pattern of the water demand; and a second part that corrects/adjusts this initial forecast to account for the specificities of the day for which a prediction is being obtained. Both parts use exponentially decreasing weights, which give more importance to more recent values of the water demand. Equation 7.2.1.1 gives the formal definition of the model.

$$\hat{y_k} = \sum_{l=1}^{L} \alpha(1-\alpha)^{l-1} y_{k-24l} + \beta^{l-1}(1-\beta)^{l-2}\Delta_{k-24l}, \tag{7.6}$$

where $k = 25, 26, ..., L$ is the number of items to include in the predictor, $\Delta_h = y_h - \hat{y_h}$ and α and β are the exponential coefficients of the weights. These weights are independent but, seeking the stability of the model, usually the seasonality pattern part weight, α, will be higher than the error part weight, β. The model can discriminate between the different days of the week. All the characteristics of this model are reproduced as a generator system of the EPANET's pattern demand curve.

The different anomalies are also simulated in EPANET at randomised points of time: valves failures are schematised in a straightforward fashion (with a programmed change of their characteristics as a function of the loss ratio) and, for example, leaks may be modelled as shown in Figure 7.8.

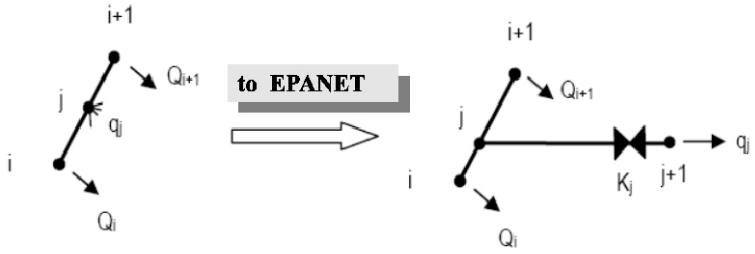

Figure 7.8: Leakage simulated under EPANET - Source: Herrera *et al.* (2009b)

From the hydraulic point of view, a leak can be simulated by a model consisting of a valve and a node with zero manometric pressure. The loss ratio of the valve will be proportional to the effective section of the fault, and depends on the coefficients of contraction and the velocity.

7.2.1.2 Detecting anomalies

Support Vector Machines provide a novel approach to the classification problem, learning to perform the classification task thorough a supervised learning procedure. Vapnik (1995, 1998) and Shawe-Taylor & Cristianini (2000) are two of the essential references for SVM. These are complemented with the works by Karatzoglou (2006); Karatzoglou *et al.* (2006), implementing SVM and kernel methods environment in R Language.

In this work, we propose a classical SVM to discriminate the normal and anomalous data obtained in the last sliding time window. The basis for the SVM algorithm for classification is the mapping of the low-dimensional training data in a higher dimensional feature space, since it is easier to separate the input data in this higher dimensional feature space. This mapping is achieved by using kernel functions. According to Mercer's theorem [Mercer (1909); Sun (2005)], kernel functions necessarily are symmetric and positive definite functions.

The proposed procedure involves the following steps:

1. measure the distance between the predicted and observed data within the last W-long performed array;
2. use the SVM algorithm to classify this array;
3. if an anomaly is detected then reclassify it as: outlier, trend or service disruption.

These are the necessary steps to complete the anomaly detection phase.

7.2.1.3 Kernel-based causal algorithm

As stated, the identification of the cause-effect relationships in a water supply environment in the presence of anomalies will produce better understanding of the demand behaviour. To achieve this we propose the application of the kernel-based causal learning algorithm (KCL) developed by Sun *et al.* (2007). This approach assumes that a variable Z is likely to be a common effect of X and Y, if conditioning on Z increases the dependence between X and Y. Based on this assumption, the algorithm collects "votes" for hypothetical causal directions and orients the edges by the majority principle. The algorithm is an improvement of inductive causation (IC) algorithm [Spirtes & Gylmour (1991)], generalising it in several ways. First, it handles both discrete and continuous variables. Next, it does not need the assumption of special kinds of distributions.

Let $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ and $(\mathcal{Y}, \mathcal{B}_\mathcal{Y})$ be measurable spaces and $(\mathcal{H}_\mathcal{X}, \mathcal{K}_\mathcal{X})$ and $(\mathcal{H}_\mathcal{Y}, \mathcal{K}_\mathcal{Y})$ be reproducing kernel Hilbert spaces of functions on $\mathcal{X}$ and $\mathcal{Y}$, with positive definite kernels $\mathcal{K}_\mathcal{X}$, $\mathcal{K}_\mathcal{Y}$. We consider a random vector (X, Y) on $\mathcal{X} \times \mathcal{Y}$ such that the expectations $E_X[K_X(X, X)]$ and $E_Y[K_Y(Y, Y)]$ are finite. We define Σ_{XY} as the cross-covariance operator and $\Sigma_{XY|Z}$ as the conditional cross-covariance operator. We have that $\Sigma_{XY} = 0 \iff X \perp Y$.

The strength of the marginal and conditional dependence can be defined by

$$\mathbb{H}_{XY} := ||\Sigma_{XY}||^2{}_{HS}, \tag{7.7}$$

$$\mathbb{H}_{XY|Z} := \beta_Z ||\Sigma_{XY}||^2{}_{HS}, \tag{7.8}$$

with $\beta_Z := 1/||T_Z||^2_{HS}$ and T_Z is defined by $\langle h_2, T_Z h_1 \rangle = E[h_1(Z)h_2(Z)]$ for arbitrary $h_1, h_2 \in \mathcal{H}_\mathcal{Z}$. Gretton *et al.* (2005) obtained consistent estimators of these dependences.

The algorithm is based on the following heuristics: conditioning on a common effect has the tendency to generate dependence between the causes. This is true when the unconditional dependences between the causes are small. Based on this, a voting-like procedure for orientation of edges is introduced: for any triple (X, Y, Z), one gets a vote for Z being a common effect of X and Y, if and only if $\mathbb{H}_{XY|Z} > \lambda \mathbb{H}_{XY}$, with appropriate $\lambda > 0$. By continuing with these votes we may direct most edges in the majority direction. We choose λ_1 very large in the first run and set $\lambda_2 := max\left\{\frac{\mathbb{H}_{ZX|Y}}{\mathbb{H}_{ZX}}, \frac{\mathbb{H}_{ZY|X}}{\mathbb{H}_{ZY}}\right\}$ in the second run. If the result is balanced, leave the edge undirected. See Sun *et al.* (2007) for further details.

7.2.1.4 Action taking phase

To take suitable actions when there is evidence of anomaly data detection we propose following the next steps:

1. Repairing the data with interpolation to continue working with the current prediction model
2. Analysing the anomaly origin
   (a) External cause
      i. Detect the cause type
      ii. Adapt the prediction model to these novelties
   (b) Internal cause
      i. Detect the cause type
      ii. Repair it
      iii. Prevent it, explaining characteristics and checking the common points with other anomalies in the water supply network

7.2.2 Results

We have tested this methodology in a hourly EPANET simulation of the conditions of a water supply network zone over 100 days. This is a real-world case study exhibiting high intensity in the presence of different malfunctions making it suitable for training the learning algorithms involved. Working with the demand variable, in a time window of 12 hours, the SVM algorithm is able to detect all the anomalies in the validation (40 days) and the testing data (10 days). In this case, we not only managed to detect the anomalies in the water demand behaviour, but also we have been able to find justifiable cause-effect relationships in the water demand environment. The causal model includes the continuous variables: pressure, valve position (that represents the water intake to our zone from other parts of the water supply network), diameter of the leakage and also the discrete inputs from the previous classification stage. This model offers deeper knowledge of the water consumption behaviour and the supply network and their elements, as seen in our working example of Figure 7.9. This graph shows some effects (for example, valves are related to outliers but are only a cause regarding trend novelties) that are obtained with the KCL algorithm. Future work will aim to obtain more information of the factors, to improve the identification of the anomaly causes, and to take actions to prevent them or mitigate their effects.

The kernel-based independence measures benefit from the power of detecting non-linear dependence and can keep, for example, type II or false negative errors (deciding independence when there is dependence) at a very low level. The methodology we have shown is a perfect supplement to improve the current prediction models of water demand, since it can be easily adapted to various anomaly scenarios. In the future, methods to screening the water network to find specially sensible or vulnerable zones, where abnormal events may have more important consequences, should be explored.

7.3 Ranking nodes in Water Supply Networks*

Ranking nodes in a Water Supply Network (WSN) is a very recent proposal. Addressing this issue, Grubesic *et al.* (2008) and Yazdani & Jeffrey (2010) investigated

*Adapted from Ajustes del modelo PageRank de Google en el estudio de la importancia relativa de los nodos de la red de abastecimiento, proceedings of X Iberoamerican Seminar in Planning, Design and Operation of Water Supply Systems (SEREA), 2011.

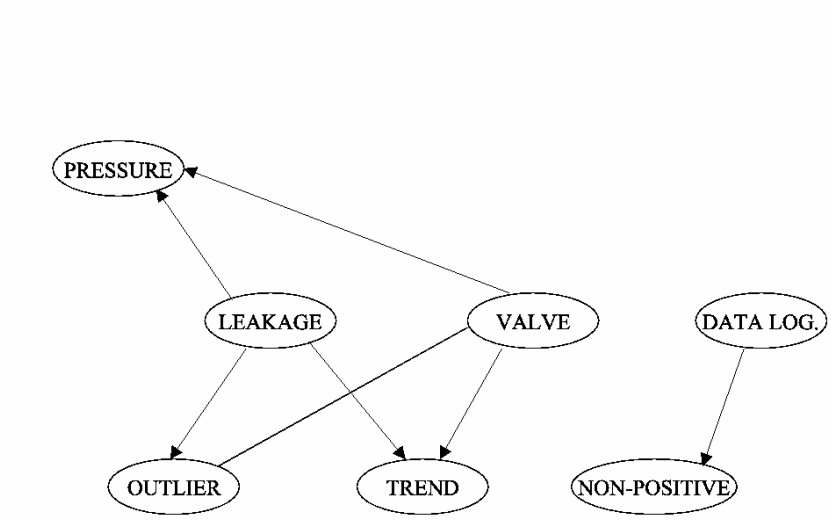

Figure 7.9: Final step of the KCL causal-effect structure - Source: Herrera *et al.* (2009b)

applications of graph theory and complex network principles in the analysis of vulnerability and robustness of WSNs. Both works reviewed metrics of indexing nodes based on statistics and spectral analysis. In this sense, statistical measurements are those which quantify organisational properties of the network based on the most frequent circumstances and structural patterns and relate them to network robustness and the dynamics on the network. Spectral metrics derive from the spectrum of the network adjacency matrix. These metrics quantify network invariants that, taken along with the described statistical measurements, reveal useful information on well-connectedness of the network, connectivity strength and failure tolerance [Yazdani & Jeffrey (2010)]. Another basic reference about ranking elements (nodes and other) of a water supply network is Michaud & Apostolakis (2006), where the authors introduce a scenario-based methodology. Izquierdo *et al.* (2008) focused on pipes, assessing their relative importance regarding the water distribution process.

This chapter tries to open new possibilities to the paradigm of ranking nodes in a WSN, which, besides being related to vulnerability indexes, is a first step to facilitate other key management tasks. Let us cite: to establish good performance reliability

indexes, to carry out water quality analysis, to locate network sensors and to study the viability of rehabilitation plans, among other possibilities. Ranking nodes could also be useful to support sectorisation works. Our proposal uses spectral-based methods (cf. Chapter 3, Section 3.3) based on the algebraical connectivity of the graph Laplacian matrix. In order to develop efficient models from these spectral methods, it will also be useful to adapt the PageRank methodology [Brin & Page (1997)], which is introduced below.

7.3.1 Brief introduction to Google's PageRank algorithm

PageRank [Brin & Page (1997)] is the initial calculus method that Google founders use to classify web pages by their importance. Until now, this algorithm has been improved and adapted to several fields of study [Chung & Zhao (2008)]. The aim of this methodology is to obtain the so-called PageRank vector that provides some relative importance of web pages.

PageRank classification can be understood as a Markov Process where the states are the pages and the transitions are their links. *A priori*, all of these transitions are equally probable. In this way, the idea is similar to reproducing a web user behaviour who moves from one page to another with a probability (clicking one of the links of the web site he or she is visiting). In a more formal way, let $G = (V, E)$ be a graph, the PageRank vector may be defined as π, which satisfies:

$$\pi = \alpha \frac{1}{n} + (1 - \alpha)\pi \, \mathbf{P}, \tag{7.9}$$

with $\pi \mathbf{1}^t = 1$ and where α (damping factor) is a constant taking values between 0 and 1. Usually α takes the value of 0.85 [Brin & Page (1997)]. $\mathbf{P}$ is a transition probability matrix of the graph. Besides, π is an eigenvector of the matrix G (eq. 7.10)

$$\mathbf{G} = \alpha \frac{\mathbf{1}\mathbf{1}^t}{n} + (1 - \alpha)\mathbf{P}, \tag{7.10}$$

where $\mathbf{G}$ is the so-called *Google Matrix*. The Perron-Frobenius theorem guarantees their existence, indicating that $\mathbf{P}$ would be strongly connected if all other eigenvalues have a modulus strictly lower than 1.

Some properties of this $\mathbf{G}$ matrix are [Langville & Meyer (2006)]:

- stochastic, because it is a linear combination of two stochastic matrices $\frac{\mathbf{1}\mathbf{1}^t}{n}$ and $\mathbf{P}$.
- irreducible, because every node is directly connected to every other node.
- aperiodic: $\mathbf{G}_{ii} > 0 \; \forall i$.
- primitive, because $\mathbf{G}^k > 0$ for some k. This implies that a unique positive π^t exists.

7.3.1.1 Computation of PageRank vector

To solve Equation 7.9, the following recurrence solution is proposed [Brin & Page (1997)]:

$$\pi_{t+1} = \alpha \frac{1}{n} + (1-\alpha)\pi_t \, \mathbf{P}, \tag{7.11}$$

starting from one suitable initial distribution, π_0. Thus, web PageRank is defined in a recursive way (*power method* [Langville & Meyer (2006)]); depending on the number and PageRank of their linked pages. There are other alternative calculus by using the spectrum of the transition matrix (and the role of eigenvalues and eigenvectors of this matrix). There are two interesting results: First, for stochastic matrices such as $\mathbf{G}$, $\lambda_1 = 1$, thus λ_2 achieves the convergence. Second, if the spectrum of $\mathbf{P}$ and $\mathbf{G}$, respectively, are $\sigma(\mathbf{P}) = 1, \mu_2, \ldots, \mu_m$ and $\sigma(\mathbf{G}) = 1, \lambda_2, \ldots, \lambda_m$, then [Haveliwala & Kamvar (2003); Langville & Meyer (2005)]:

$$\lambda_k = \alpha_k \mu_k \;\; \text{for} \;\; \mathrm{k} = 2, 3, \ldots, \mathrm{n}. \tag{7.12}$$

Then, Equation 7.12 guarantees convergence, with the appropriate accuracy, of this power method. Another works introduce alternatives to approach the PageRank vector; among them Bryan & Leise (2006), Langville & Meyer (2006) and Pedroche (2007).

7.3.2 Adaptation of PageRank algorithm to a WSN

Chung & Zhao (2008) proposed a generalisation of the PageRank algorithm to establish an ordering of the vertices in a graph. In addition, by understanding a WSN as a graph of special characteristics [Herrera *et al.* (2010a)], it is possible to abstract the concept of

web page looking at it as a consumption node in a WSN. Links between pages are now understood as pipes connecting different nodes. Finally, the random walk idea behind PageRank can be seen as a particle path along the network, starting at any node.

Previous simulations of the hydraulic model with EPANET [Rossman (2000)] are essential for calculating PageRank for WSNs, because it is necessary to estimate the flow direction through the network (thereby considering the network as a directed graph). Then, Equation 7.11 may be applied to a network as follows.

Let A be a consumption node with input flow pipes from nodes $T_1, \ldots, T_n$. Parameter d is the damping factor, which belongs to [0, 1] and (*a priori*) follows the recommended value, 0.85. $C(A)$ is defined as the number of output flow pipes from A. The PageRank (PR) of A is defined as:

$$PR(A) = (1 - d) + d \left[\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)} \right]. \tag{7.13}$$

The set of all PageRanks have a probability distribution on the set of all network nodes. Thus, the sum of all PageRanks must be equal to one.

7.3.3 Experimental study: PageRank on a real WSN

The case-study network is fed by three reservoirs and made out of 132 lines and 104 consumption nodes; its total length is 9 km and the total consumed flowrate amounts to 47 l/s. After running the introduced algorithm on this graph, a PageRank vector represented in Figure 7.10, calculated with the R Language [R-Development-Core-Team (2010)] library `igraph` [Csárdi & Nepusz (2006)] and represented with NetLogo [Wilensky (1999)], is obtained.

7.3.3.1 Results

The node diameters in Figure 7.10 are a function of their respective PageRank values. The node colours, showing an impact scale (blue-purple-orange-red, cf. Table 7.4), reinforce the understanding of this representation.

In view of the graphical results, it is clear that the nodes in the north have a higher combined value of PageRank. This conclusion is clearly reminiscent of the MAS-clustering output obtained with the same WSN (see Figure 7.11, where approximately 77% of the nodes with *Medium-High* and *High* PR are in the same cluster).

Table 7.4: Legend of the colour classification of the modified PageRank algorithm.

colour	PageRank	classification
blue	$PR < 0.001$	Low
violet	$0.001 \leq PR < 0.01$	Medium-Low
orange	$0.01 \leq PR < 0.05$	Medium-High
red	$PR \geq 0.05$	High

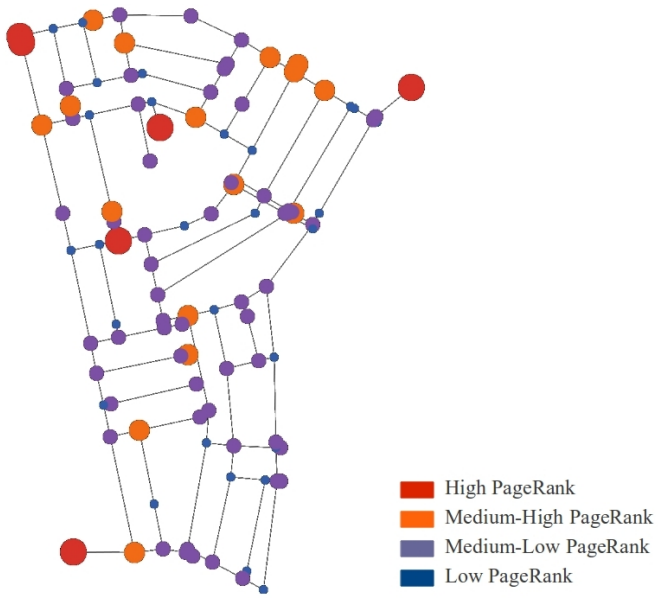

Figure 7.10: Graphical distribution of a WSN PageRank - Based on a real case-study

Consequently, the relative importance of these nodes will be higher and it should be taken into consideration in a hypothetical task of dividing the water network into supply clusters. Until now this fact solely was taken into account in indirect ways, such as the performance of semi-supervised clustering (cf. Chapter 3), which shares similar mathematical background to PageRank. Figure 7.12 shows a comparison of those two process output: semi-supervised clustering on the left hand side and nodes PageRank on the right hand side of the figure.

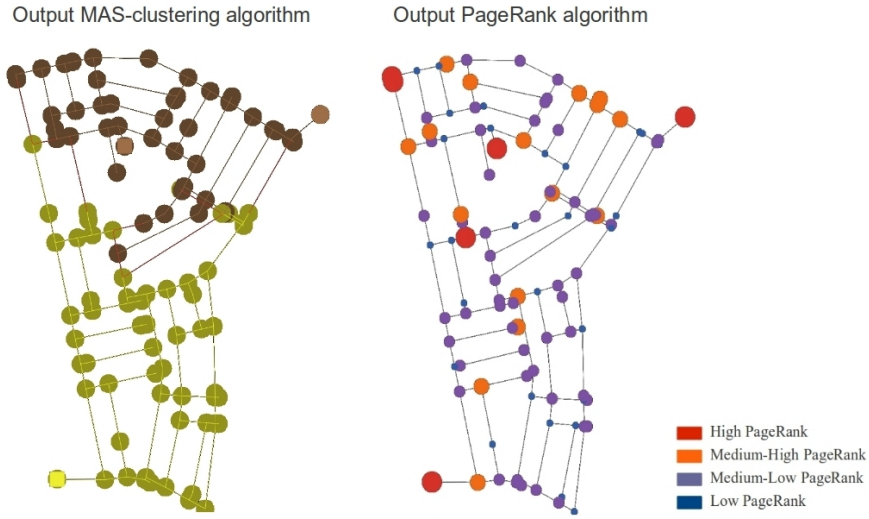

Figure 7.11: MAS-clustering and PageRank - Comparison

Figure 7.12 shows how an evident majority of nodes with *High* and *Medium-high* PageRank share the same supply cluster (approximately 68% of these high ranked nodes). In addition, the shape of the cluster configuration seems to inherit the spatial disposition of the rank spread. This suggests that future development should include ranking criteria together with the rest of variables when considering the problem of dividing a WSN into hydraulic sectors.

7.4 Summary and comments

In this chapter we have addressed the task of trying to predict the future water demand on an urban area of a city in south-eastern Spain. We have considered a large number of alternative machine learning methods to solve this prediction task. We have briefly described these techniques in Appendix B and carried out a thorough experimental

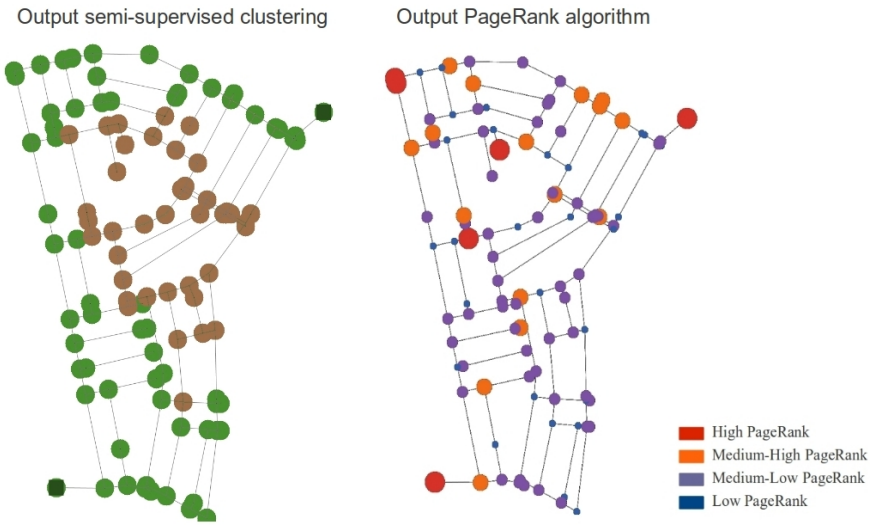

Figure 7.12: Semi-supervised clustering and PageRank - Comparison

comparison of several variants of these models using our case study data in this chapter. The results of this comparison have identified support vector regression models (SVR) as the most accurate models, closely followed by MARS, PPR and Random Forests. The experiments have also revealed a disappointing performance of the variants of neural networks that were considered. Finally, a heuristic model based on the empirical analysis of the regularities of the time series has shown its limitations when compared to these more sophisticated modelling approaches.

This chapter has also addressed the problem of correctly estimating the prediction performance of different models when facing time series data with clear changes of dynamics across the time. We have used a Monte Carlo simulation experiment that ensures unbiased estimates of the selected evaluation metrics, and have also considered and compared different model building strategies to handle the regime shifts of the data, which may be originated by the inclusion of too many data harmful for the prediction models.

All compared models take advantage of the reduced size of these working areas and their efficient way to be implemented. It is possible to obtain better modelling using only one sector for each study than the whole network.

Another application developed in this chapter is a causal model to detect and identify anomalies. This includes continuous variables and also the discrete inputs from

the previous classification stage. This model offers deeper knowledge of the water consumption behaviour, the supply network, and their elements. This will be a beneficial support to understand and satisfy the different necessities of each supply cluster. Once again, kernel methods (SVM and KCL) have been essential to develop new trends and possibilities in WSN management.

The final part of this chapter presents an adaptation of the Google PageRank algorithm to WSN graphs. This opens the possibility of working with a proven algorithm efficient in large databases after providing a simple adaptation to the particularities of a supply network. Undoubtedly, this chapter provides a scope for future developments in urban hydraulic applications, such as the management ideas proposed by this methodology. This methodology is not only related to indices of vulnerability but it also could be useful for water quality analysis, sensor location and the feasibility of rehabilitation plans, among others. In addition, establishing a ranking on nodes can be a very useful tool when looking for efficient criteria for sectorisation.

7. WATER NETWORK MANAGEMENT BASED ON SUPPLY CLUSTERS: WORKING PROPOSALS

Part IV

Conclusions

8

Conclusions

Efficient division into supply clusters is an important paradigm of water network management, and may result in substantial water and financial savings. An efficient network partition can be produced using kernel methods together with multi-agent systems. However, these methodologies go further and create a suitable framework for management tasks in a divided water supply network.

The main subject of this thesis has been the development, application, and experimental analysis of clustering algorithms for dealing with the network division problem. The following sections review the main conclusions obtained. The contributions of the thesis and publications arising from (or related to) this work are then summarised. The final section gives some insight into how the conclusions can be applied beyond this thesis and how this work could be extended in new directions.

8.1 Contributions of this thesis

From a hydraulic point of view, this thesis proposes WSN sectorisation to improve the performance of decision-making models in water supply. Nevertheless, for sectorisation to be efficient and enable a suitable network management, it must be consistently implemented. Classical criteria have been shown to be too coarse regarding this hypothesis; and as a new contribution we propose an approach that uses cluster analysis. The diversity of the available information (databases have continuous, discrete, and geographic information items) suggests the suitability of procedures to work with all the data describing the network graph. For this reason, we have developed clustering

algorithms using both multi-agent tools and semi-supervised learning. Both methodologies are flexible enough to fully mine all the graph information and adapt themselves to the graph structure. In the case of large networks, this thesis proposes integrating these approaches with a boosting methodology.

Once the network partition is established, it is possible to check if sectorisation has avoided the possible bias originated by more detailed analysis - while achieving more accurate results than when we work with the whole network. Based on this idea, we propose a certain methodological continuity bearing in mind that sectors are established to approach different management paradigms. Water demand prediction, anomaly detection, and ranking of network elements, represent instances of applications where management performance is improved by suitable sectorisation. All of these functions may be based on multi-agent systems and/or kernel methods.

The rest of this section presents a more detailed description of the contributions on sectorisation made in this thesis.

8.1.1 Water supply clusters made using semi-supervised learning

Classically, division of a WSN into DMAs aims at improving leakage detection using node elevation, as well as pressure and demand information. By using semi-supervised clustering (see Chapter 3) we propose increasing, or changing, the perspective of this target. This can be done by taking into account the information to be included within specific criteria for the division of the WSN into hydraulic zones (clusters).

When compared with methodologies that only use graphical or vector information, semi-supervised clustering (which uses both types of information) is a more efficient and robust approach. The flexibility to be able to include different inputs into the study, with different weights, is another improvement of the developed methodology.

8.1.2 Agent-division of water distribution systems into supply clusters

The multi-agent metaphor was introduced in Chapter 4 to divide a WSN into supply clusters. In addition to traditional centralised architecture (of single reasoning agents) as problem solvers, this chapter shows that it is possible to use systems of reasoning agents, or apply multi-agent simulations to verify hypotheses about the different processes in water distribution. The inclusion of negotiation behaviour for the agents

proposes a sensitivity analysis of the solution that was previously found via agent co-operation and competition.

8.1.3 MAS boosting semi-supervised supply clusters

The MAS-boost clustering presented in Chapter 5 proposed a scalable WSN division for any size of network and takes into account the supply conditions necessary to turn these districts into real hydraulic sectors. In addition, this chapter proposes advances in various senses:

- The burden of computations in the spectral clustering paradigm is reduced, making it feasible for boosting even in the case of large graphs.
- A novel multi-agent approach to sampling subgraphs by exploration and reweighting boosting methods is proposed.
- A fast multi-agent *pre-clustering* into the label propagation phase in semi-supervised clustering is implemented.
- Hydraulically, this chapter proposes the use of both graphical and vector information to improve the approach to dividing a WSN into sectors.

8.1.4 Water network management by supply clusters

A well sectorised WSN may improve the management of the whole network. Let us see in which sense the thesis supports this assertion.

8.1.4.1 Predictive models

In Chapter 7 (Section 7.1) we have considered a large number of alternative machine learning methods to solve the predictive task in water demand from a sector. We have briefly described these techniques (in Appendix B)and carried out a thorough experimental comparison of several variants of these models using our case study data. The results of this comparison have identified support vector regression models (SVR) as the most accurate models. The experiments have also revealed a disappointing performance for the considered neural network variants.

It also addressed the problem of correctly estimating the prediction performance of different models when facing time series data with clear changes in dynamics over

time. We used a Monte Carlo simulation experiment that ensured unbiased estimates of the selected evaluation metrics, and also considered and compared two different model building strategies (growing and sliding window models) to handle the regime data shifts. Our experiments confirmed the advantages of these learning approaches and established that few differences exist between the performance of growing and sliding windows. However, for this data set, we also found that model updating frequency should be as high as possible (in our case, every model is updated on an hourly basis as new data arrives).

8.1.4.2 Screening anomalies

The application developed in Chapter 7 (Section 7.2) is a causal model to detect and identify anomalies based on support vector machines and a kernel causal algorithm. The final model includes continuous variables and the discrete inputs from the previous classification stage. This model offers a more profound understanding of water consumption behaviour, and the supply network and its elements. This will improve our understanding and help us satisfy the different needs of each supply cluster.

8.1.4.3 Ranking nodes

Chapter 7 (Section 7.3) presents an adaptation of the Google PageRank algorithm to WSN graphs. This opens the possibility of working with a proven efficient algorithm in large databases with just a simple adaptation to the particularities of a supply network. This methodology is not only related with indices of vulnerability, but could also be useful for water quality analysis, sensor location, and rehabilitation plan feasibility studies, among others. In addition, ranking nodes and other network elements can be a very useful tool when looking for efficient sectorisation criteria.

Once again, kernel methods have been essential for developing new trends and possibilities in WSN management.

8.1.5 Developed and employed software

Appendix A introduces three programs used throughout this thesis: R Language [R-Development-Core-Team (2010)], NetLogo [Wilensky (1999)], and EPANET [Rossman (2000)]. The necessity to share information between these programs arose while the

proposed algorithms were being implemented. Thus, in addition to the use of the *NetLogo - R - Extension* of Thiele & Grimm (2010), two data exchange functions to share information between R Language and EPANET 2.0 were developed. These functions are RimpEpa and RexpEpa, to import data from EPANET to R and vice-versa, respectively. Both functions were developed in C Language and embedded in the R environment. Subsection A.3.2 of Appendix A details characteristics and use of these functions.

8.2 Publications in relation to this thesis

The main published contributions related to this thesis are shown below:

8.2.1 Journal papers

1. Herrera, M.; Izquierdo, J.; Pérez-García, R.; Montalvo, I. Multi-agent adaptive boosting on semi-supervised water supply clusters, *2011, under review* in Advances in Engineering Software

2. Herrera, M.; García-Díaz, J.C.; Izquierdo, J.; Pérez-García, R. (2011) Municipal water demand forecasting: Tools for intervention time series, *to appear* in Journal of Stochastic Analysis and Applications 29 (5)

3. Herrera, M.; Torgo, L.; Izquierdo, J.; Pérez-García, R. (2010) Predictive models for forecasting hourly water demand, in Journal of Hydrology, 387 (1-2), pp. 141–150

4. Herrera, M.; Izquierdo, J.; Montalvo, I.; García-Armengol, J.; Roig, J.V. (2009) Identification of surgical praxis patterns by evolutionary cluster analysis, in Mathematical and Computer Modelling, 50, pp. 705–712

5. Izquierdo, J.; Montalvo, I.; Pérez-García, R.; Herrera, M. (2008) Sensivity analysis to assess the relative importance of pipes in Water Distribution Networks, Mathematical and Computer Modelling, 48, pp. 268–278

8.2.2 Chapters of books

1. Izquierdo, J.; Herrera, M.; Montalvo, I.; Pérez-García, R. (2011) Division of water supply systems into district metered areas using a multi-agent based approach, in Lecture notes in Communications in Computer and Information Science (CCIS), Springer-Verlag, pp. 167–180 (ISBN: 978-3-642-20116-5)

2. Herrera, M.; Izquierdo, J.; Pérez-García, R.; Ayala-Cabrera, D. (2010) Integración de procesos de Aprendizaje Automático en la sectorización Hidráulica de redes de abastecimiento, in Retos Tecnológicos y Metodológicos en la Gestión Técnica de los Sistemas Urbanos de Agua, IMM-UPV, pp. 79–92 (ISBN: 978-84-89487-32-1)

3. Izquierdo, J.; Montalvo, I.; Herrera, M.; Pérez-García, R.; (2010) Utilización de modelos híbridos en hidráulica urbana, in Retos Tecnológicos y Metodológicos en la Gestión Técnica de los Sistemas Urbanos de Agua, IMM-UPV, pp. 149–179 (ISBN: 978-84-89487-32-1)

4. Herrera, M.; Izquierdo, J.; Montalvo, I.; Pérez-García, R. (2010) Simulación multi-agente de la sectorización de un sistema de abastecimiento de agua. in Planificación, proyecto y operación de sistemas de abastecimiento de agua, IMM-UPV, pp. 339–350 (ISBN: 978-84-89487-31-4)

5. Herrera, M.; Karatzoglou, A.; Canu, S.; Izquierdo, J.; Pérez-García, R. (2010) Representación kernel de la red de abastecimiento de agua para su sectorización eficiente, in Planificación, proyecto y operación de sistemas de abastecimiento de agua, IMM-UPV, pp. 351–360 (ISBN: 978-84-89487-31-4)

6. Herrera, M.; Pérez-García, R; Izquierdo, J; Montalvo, I. (2009) Scrutinizing changes in water demand behavior, in Lecture notes in Control and Information Sciences, pp. 305–313, Springer (ISBN: 978-3-642-02893-9)

7. Izquierdo, J.; Herrera, M.; Pérez-García, R.; López, P.A. (2007) Abastecimiento de água: o estado da arte e técnicas avançadas, cap. 20: Análisis Inteligente de Datos como Herramienta de Integración en la Gestión Técnica de los Abastecimientos, pp. 367-386 Editora Universitaria - UFPB Joao Pessoa, Brasil (ISBN: 978-85-7745-078-3)

8.2.3 Conference papers

2011

1. Gutiérrez-Pérez, J.A.; Herrera, M.; Pérez-García, R.; Ramos, E. Application of graph-spectral methods in the vulnerability assessment of water supply networks, *accepted in* Mathematical Modelling in Engineering & Human Behaviour (Sept. 2011, Valencia, Spain)

2. Herrera, M.; Izquierdo, J.; Pérez-García, R.; Ayala-Cabrera, D. La regularización del grafo de la red de abastecimiento de agua para la propuesta de su sectorización, *submitted to* II Jornadas de Ingeniería del agua (Oct. 2011, Barcelona, Spain)

3. Gutiérrez-Pérez, J.A.; Herrera, M.; Izquierdo, J.; Pérez-García, R. Uso de teoría de grafos para determinar zonas de vulnerabilidad en redes de abastecimiento de agua, *submitted to* II Jornadas de Ingeniería del agua (Oct. 2011, Barcelona, Spain)

4. Delgado-Galván, X.; Herrera, M.; Izquierdo, J.; Pérez-García, R. Aplicaciones de la metodología AHP para la toma de decisiones en la gestión de la red de abastecimiento, in X Ibero-american Seminar, SEREA (Jan. 2011, Morelia, Mexico)

5. Herrera, M.; Izquierdo, J.; Pérez-García, R.; Ayala-Cabrera, D. Boosting del grafo de la red de abastecimiento como soporte en el proceso de sectorización, in X Ibero-american Seminar, SEREA (Jan. 2011, Morelia, Mexico)

6. Herrera, M.; Gutiérrez-Pérez, J.A.; Izquierdo, J.; Pérez-García, R. Ajustes del modelo PageRank de Google en el estudio de la importancia relativa de los nodos de la red de abastecimiento, in X Ibero-american Seminar, SEREA (Jan. 2011, Morelia, Mexico)

7. Izquierdo, J.; Montalvo, I.; Herrera, M.; Ayala-Cabrera, D.; Pérez-García, R. Sistemas multi-agente. Aplicaciones en Hidráulica urbana, in X Ibero-american Seminar, SEREA (Jan. 2011, Morelia, Mexico)

2010

1. Herrera, M.; Izquierdo, J.; Pérez-García, R.; Montalvo, I. Water supply clusters based on a boosting semi-supervised learning methodology, in Civil-Comp Press, 2010. 7th International Conference on Engineering Computational Technology (Sep. 2010, Valencia, Spain)

2. Herrera, M.; Izquierdo, J.; Pérez-García, R., Ayala-Cabrera, D. Water supply clusters by multi-agent based approach, in Water Distribution System Analysis 2010 – WDSA2010 (Sep. 2010, Tucson, AZ, USA)

3. Izquierdo, J.; Montalvo, I.; Herrera, M.; Pérez-García, R. Multi-agent applications in urban hydraulics, in the 16th European Conference on Mathematics for Industry (Aug. 2010, Wuppertal, Germany)

4. Herrera, M.; Canu, S.; Karatzoglou, A.; Pérez-García, R.; Izquierdo, J. An approach to water supply clusters by semi-supervised learning, iEMSs 2010 (Jul. 2010, Ottawa, Canada)

2009

1. Herrera, M.; Karatzoglou, A.; Canu, S.; Izquierdo, J.; Pérez-García, R. Clusters de abastecimiento de agua basados en aprendizaje semi-supervisa-do, in IX Ibero-american Seminar, SEREA (Nov. 2009, Valencia, Spain)

2. Herrera, M.; Izquierdo, J.; Montalvo, I.; Pérez-García, R. Sectorización de redes de distribución de agua basada en técnicas multiagente, in IX Ibero-american Seminar, SEREA (Nov. 2009, Valencia, Spain)

3. Gutiérrez-Pérez, J.A.; Pérez-García, R.; Izquierdo, J.; Herrera, M. Reducción de la vulnerabilidad en redes de abastecimiento mediante la instalación de una red de sensores de calidad, in IX Ibero-american Seminar, SEREA (Nov. 2009, Valencia, Spain)

4. Herrera, M.; Pérez-García, R.; Izquierdo, J.; Montalvo, I. Scrutinizing changes in the water demand behavior, in 3rd Multidisciplinary International Symposium on Positive System: Theory and applications (Sep. 2009, Valencia, Spain)

5. Izquierdo, J.; Herrera, M.; Montalvo, I; Pérez-García, R. Agent-based division of water distribution networks into district metered areas, in 4th International Conference on Software and Data Technologies (Jul. 2009, Sofia, Bulgaria) – *Paper selected to publish by Springer-Verlag*

2008

1. Herrera, M.; Izquierdo, J.; Montalvo, I.; García-Armengol, J.; Roig, J.V. Identification of surgical praxis patterns by evolutionary cluster analysis, in the X Journals of Research and Interdisciplinarity of the University Polytechnic of Valencia (Sep. 2008, Valencia, Spain)

2. Herrera, M.; Salas, L.; Pérez-García, R.; Díaz, J.L. Métodos estadísticos para la caracterización de la sensibilidad a las fugas de una red de distribución, in VIII Ibero-american Seminar, SEREA (Jul. 2008, Lisbon, Portugal)

3. Montalvo,I.; Herrera, M.; Izquierdo, J; Díaz, J.L. Métodos heurísticos para el análisis cluster en una base de datos de abastecimiento de agua, in VIII Ibero-american Seminar, SEREA (Jul. 2008, Lisbon, Portugal)

2007

1. Herrera, M.; García-Díaz, J.C.; Pérez-García, R.; Martínez, J.F.; López, P.A. Hybrid models applied to intervened time series in forecasting urban water demand, in II Encuentro de Series Temporales No Lineales (Oct. 2007, La Manga, Murcia, Spain)

2. Herrera, M.; García-Díaz, J.C.; Pérez-García, R.; Martínez, J.F.; López, P.A. Métodos avanzados de previsión de la demanda en la gestión de un abastecimiento urbano de agua, in VII Ibero-american Seminar, SEREA (Jun. 2007, Morelia, Mexico)

3. Herrera, M.; Nudelman, M.; Pérez-García, R. Caracterización de la demanda urbana de agua, in VII Ibero-american Seminar, SEREA (Jun. 2007, Morelia, Mexico)

8.3 Future work

This thesis proposes some approaches to follow in the future. All of these lines are related with kernel methods, multi-agent systems, and some suitable integration of both. These algorithmic procedures could be more profoundly developed to include other network information as valuable input, such as the use of ranked nodes. In addition, it is possible to propose multi-objective criteria for the different functions to optimise. A third approach could exploit the property of sparseness associated with WSN graphs. In addition, WSN management could also take advantage of working with a pre-sectorised network with the prior ranking of nodes for predictive models and screening anomalies. Moreover, other management actions could also be tested: rehabilitation plans, sensor location, and water quality assessments are examples of promising future tasks that follow the line of work of this thesis. Details of this future work are introduced in the following sections.

8.3.1 Water supply clusters using semi-supervised learning

Regarding semi-supervised clustering, proposed in Chapter 3, we can use other optional inputs. This is the case of the pipe diameters and their ages, or weighting possible rehabilitation plans by sectors. An index of pipe vulnerability could also be used that takes into account the effects of hazards in the construction of sectors. Moreover, different modifications could be included in the clustering algorithm and these could be compared on a work bench with respect to building optimal models.

8.3.2 Agent-division of water distribution systems into supply clusters

Among the various scenarios using multi-agent systems in the scope of decision support for the water management company, Chapter 4 focuses on the division of a WSN into district-metered areas. Future research will focus on the development of implementations of other scenarios of multi-agent applications in the water supply field, including aspects related to water quality, location of sensors, and other managerial issues.

8.3.3 MAS boosting semi-supervised supply clusters

Future research in MAS-boost (see Chapter 5) will focus on the improvement of the semi-boosting methodology employed. This improvement would include a guide for sampling on subgraphs constituted by boundary elements of a previous network division. Through this sensitivity analysis we could reduce the costs of reforming existing DMAs.

8.3.4 Water network management by supply clusters

In a pre-sectorised network, it would be useful to locate sensors at specific major nodes. This important information could also be embedded in the sectorisation process as another important input variable. Other uses of ranking nodes include rehabilitation plan criteria, or WSN vulnerability management (this is the working-line of the Gutiérrez-Pérez *et al.* (2011) conference paper).

Further work in predictive models of water demand should include improving the weighted pattern-based method (see 7 and Appendix B), perhaps with the inclusion of different time support partitions. We also plan to develop a software tool to help in incorporating these modelling tools in a production environment that can help in water supply management.

We aim to detect anomalies in water demand behaviour (see Chapter 7, Section 7.2), as well as finding justifiable cause-effect relationships in the water demand environment.

Part V

Appendices

Appendix A

Developed and employed software

The software used in this thesis is based on three programs: R Language [R-Development-Core-Team (2010)], NetLogo [Wilensky (1999)] and EPANET [Rossman (2000)]. Thus, our approaches are based on algorithms which run under one of these three programs and the combination and communication between them. This point was crucial for our results because it allows us changing and testing information from statistical, multi-agent and hydraulic point of view at the same time. Figure A.1 summarises the environment where our models and algorithms have been developed.

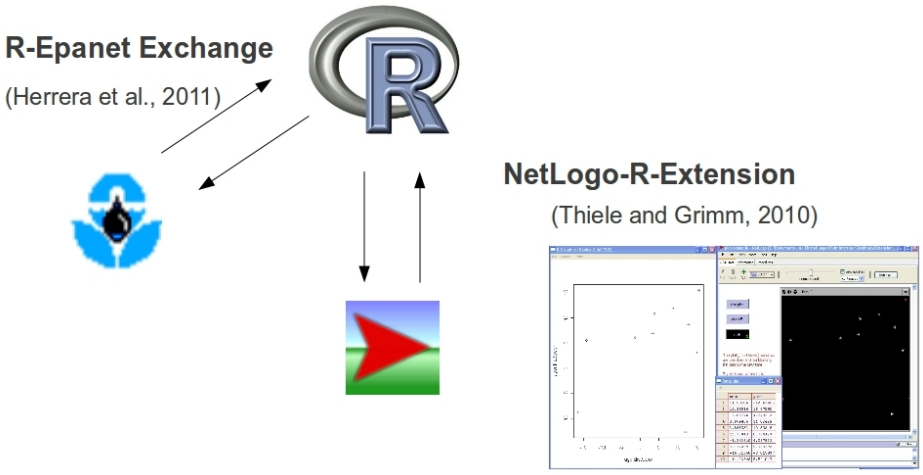

Figure A.1: Software used in this thesis - Creating an exchange framework

A.1 NetLogo and multi-agent systems

NetLogo* [Wilensky (1999)] is an environment for developing complex, multi-agent models that evolve over time. It is a Java-based high level programming language that comes packaged with its own IDE and graphing capabilities. A great advantage of NetLogo is its ease in learning and intuitive use of commands. With NetLogo it is possible to create populations of changing agents in a suitable grid of stable agents. The evolution of agents can take different forms. Agents can be created, move, change their properties, change their behaviour, change their nature or breed, and even die (cf. Chapter 3 of this thesis).

Our models are created from GIS data defining the physical and topological networks characteristics. The experimental data were obtained from GIS models of two real moderately-sized networks that have been studied by the authors within a joint research project with an international water company. These networks are parts of two water distribution systems in two Latin American cities. The area is divided into squares (patches), which gives some raster format to the environment. Patches represent the ground (underground) where pipes and nodes are buried. Figure A.2 shows a section of one network. Patches are used to define areas occupied by the different divisions that will be created. Consumption nodes (small circles) are agents (turtles) of a certain breed with a number of associated variables. Among the user-defined variables, elevation and demand are the most important. During the process, color is used to define the district metered area (DMA) that the agents belong to. Pipes (lines) are undirected links. Each pipe connects two different nodes and also has some associated variables. The main user-defined variables are diameter and length. Source points (squares) are another breed of turtles, whose variables are the average of the demand they supply and the DMAs they feed. Patches, sources, nodes, and pipes are spatially fixed agents in the sense that, obviously, they do not change their position with time. Instead, they change their properties, especially colour, and as a result they eventually belong to one DMA or another. Initially, sources, nodes, and pipes are presented in light grey, since no district structure is available at the setup. In this model, the user decides the number of DMAs to be built. Then, randomly the same number of source points are selected to be the seeds for the corresponding districts.

*NetLogo versions 4.1.1 and 4.1.2 were used in this thesis.

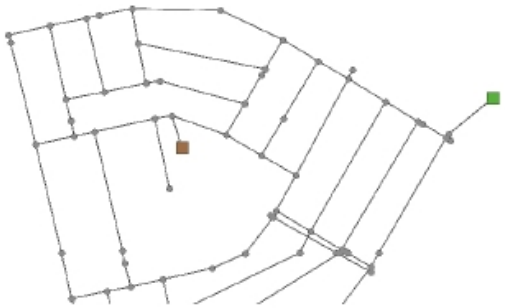

Figure A.2: Detail of a network - NetLogo environment

A.1.1 Practical implementation issues

From the point of view of NetLogo's programming, the functions `self` and `myself` are essential to define the list of inputs of the own agent (set of agents) that is enquired, and the agent (set of agents) making the enquiry. In other works, nodes are assigned to a given sector after some dialectic process of neighbouring enquiries among different sectors competing for them. The next NetLogo code shows the main procedure, which decides the assignment of a node to the winning sector. First, neighbouring nodes to the calling sector are scanned. Then a decision is made about the colour to be assigned to a neighbouring node. This decision is performed by weighting the difference between the node's elevation and demand with regard to the average elevation and demand of the calling sector (bold line in the next code). This weighted difference provides some resistance degree (resist) in terms of probability to decide about its membership to the calling hydraulic sector. The weights influencing this difference are selected by the user trough the slider called 'weight-demand'. Once a new configuration has been built other requisites (sector size, connectivity, etc.) are checked before validating the configuration.

```
to add-to-cluster
ifelse any?
link-neighbors with [color != [color] of myself and shape != square ]
[ ask one-of
link-neighbors with [color != [color] of myself and shape != square ][
   let color1 color
   set color [color] of myself
```

```
   if zoning [splotch]
   ask my-links [set color [color] of myself]
   let cota-cluster mean[cota] of turtles
      with [color = [color] of myself]
   let demand-cluster mean[demand] of turtles
      with [color = [color] of myself]
   let resist weight-cota * ([cota] of self - cota-cluster) +
      weight-demand * ([demand] of self - demand-cluster)
   if random negotiate < resist [
      set color color1
      ask my-links [set color color1]
   if zoning [splotch]
   ]
]
[ stop ]
end
```

One natural consequence of this process is that different DMAs are built and, with certain probability, some nodes will end up disconnected. The information about disconnected nodes can also be used by the network manager to detect errors in network data and to promote different actions aiming at improving the layout and/or the topology of the network.

Through the use of additional interface elements, the user can manage the course of the simulation by changing various parameters (Figure A.3 shows the interface for one of the studied networks).

The membership probability measurements of a node with respect to a district depend on elevation and demand, and can be modified by using the slider labeled 'weight-demand'. The user can also decide a priori the number of hydraulic sectors to be built by selecting an option from the chooser labelled 'n-clust'. By using the switch labelled 'zoning' the user can also ask the program to colour the patches occupied by the different hydraulic sectors. This option, as well as offering an interesting visual value, enables the user to decide if the districts displayed exhibit good topological properties. Certain convexity and/or compactness properties are desirable for districts. By default (option off) the different colours for the pipes and nodes make clear the division of the

hydraulic network into districts. By flipping the switch to 'on', patches are coloured according to the colour of the nodes and pipes they contain. This option is useful for revealing overlaps between sectors which, as explained before, can be used to produce suitable sensitivity analyses. The simulation results can be visualised on screens, plots, and graphs; and data can be stored for further processing in hydraulic simulation software and for decision-making support.

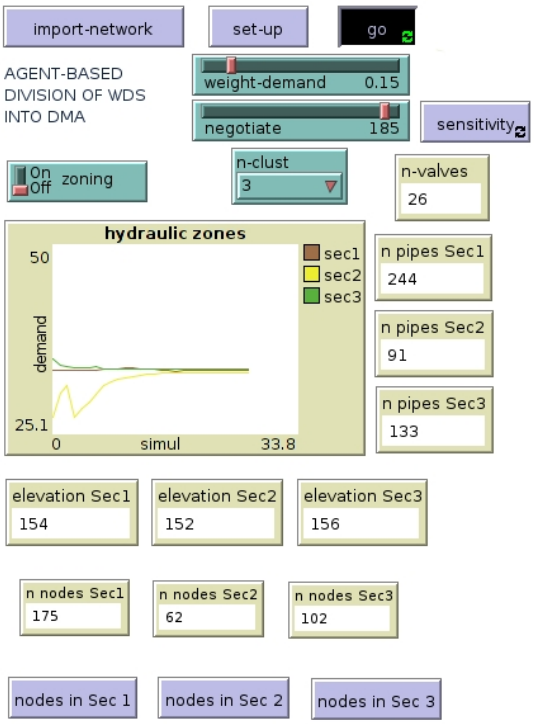

Figure A.3: Menu including parameter selectors and monitors - MAS clustering algorithms I and II developed in NetLogo

Figure A.3 presents several displays showing some of the used parameters, such as the average elevation of the different sectors and the number of pipes in the sectors. Of special importance is the display labelled 'n-valves' which shows the number of sectioning links connecting different sectors; these are pipes that enable isolation-communication between two sectors and provide the engineer with useful information about the candidate pipes for sectioning and where to install cut-off valves to isolate the districts. Engineers must make important decisions about the need to install closing valves in existing pipes, and about sectioning those pipes, and/or introducing new

pipes that redistribute the flow in more a reasonable manner. The process is able to find good solutions for the connectivity between DMAs. As a consequence, the number and location of the closure valves is optimised for a given layout. In addition, nodes are assigned to sectors in a remarkably stable way that further stabilises during the evolution of the process (see the demand plot on Figure A.3). In addition, the best partitions can be found with more frequency during different runs of the process. As a result, by repeating the process a certain number of times, the engineer can make a final decision that may, or may not, coincide with any of the obtained partitions - these being used as a basis for the decision. Our simulation model helps managers communicate with domain experts, because they can perform their analyses using solved modelled situations. Figure A.4 shows the interface of the developed software to the performance of MAS clustering algorithms I and II.

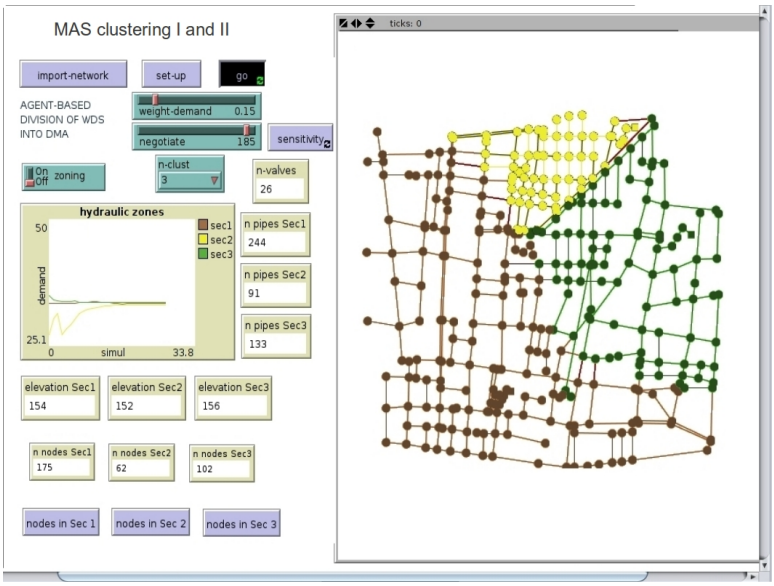

Figure A.4: Interface of MAS clustering algorithms I and II - Developed in NetLogo

MAS-boost clustering algorithm has a NetLogo base, as we can see in Figure A.5, where was developed the 'sampling subgraphs' phase. Figure A.6 shows the complete interface of the necessary MAS developed software part regards to the performance of MAS-boost clustering in the phase of 'voting'. This part is about pre-clustering phase, sampling by exploration and the final voting by MAS negotiation. A process of

information exchange with R Language (cf. next sections A.2 and A.3) will be necessary to complete the performance of this algorithm.

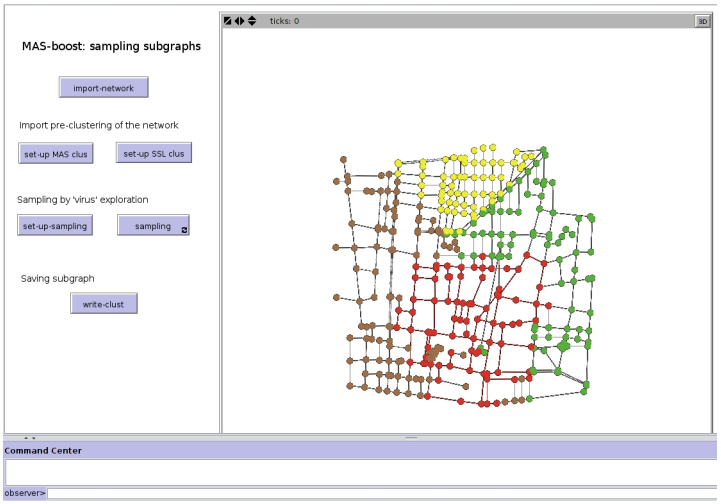

Figure A.5: Interface to sampling subgraphs in MAS-boost clustering - Developed in NetLogo

A.2 R Language: the flexible programming environment

R* [R-Development-Core-Team (2010)] is an integrated set of programs for statistical computation and graphics. It provides, among other things, a programming language, high level graphics, interfaces to other languages and debugging facilities. R can be seen as a dialect of the S language (developed at AT&T by Rick Becker, John Chambers and Allan Wilks).

The main R characteristics are:

- It has an effective data store and manipulation of them.
- It directly operates on indexed arrays and matrices.
- It has a large, coherent and integrated collection of data analysis tools, defining a complete work environment.
- It is a well developed program language: it includes cycles, conditionals, recursive functions and a diversity of input/output ways.

*R version 2.11.1 was used in this thesis

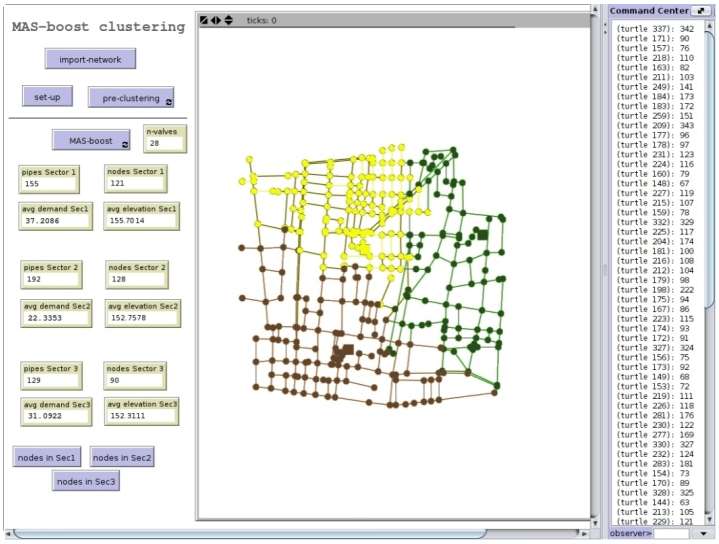

Figure A.6: Interface to voting in MAS-boost clustering - Developed in NetLogo

C is the code underlying R. But, R is able to communicate to different codes in languages like C++, Python or Java, among others*. Here, we will highlight the `rJava`† package, which provides a low-level bridge between R and Java (via JNI). It allows to create objects, call methods and access fields of Java objects from R. In a sense the inverse of `rJava` is JRI (Java/R Interface) which provides the opposite direction - calling R from Java. JRI is now shipped as a part of the `rJava` package, although it still can be used as a separate entity (especially for development). The NetLogo-R-Extension introduced in next Section A.3 will use `rJava` to communicate NetLogo and R.

In last years, R has experimented an evident raise in the field of Data Mining thanks to their connectivity to MySQL and to their web server interfaces, which makes R able to analyse large data stories [Torgo (2010)]. Thus, an important amount of libraries and functions have been developed. Appendix B shows a sample of this, regarding the libraries that may be used in the construction of predictive models in water demand. Nevertheless, there are used other two main R libraries in this thesis: `igraph` and `kernlab`.

*The Omega project for statistical computing plays an important role in the creation of new communication ways to other programs and applications. All of these R extensions can be consulted and downloaded in http://www.omegahat.org

†cf. http://www.rforge.net/rJava/

A.2.1 igraph library and page.rank function

`igraph` [Csárdi & Nepusz (2006)] is a free software package for creating and manipulating undirected and directed graphs. Its core is a software library written in C/C++, and it has interfaces to R and Python. It includes implementations for classic graph theory problems and also implements algorithms for some recent network analysis methods. `igraph` contains functions for generating regular and random graphs, manipulating graphs and assigning attributes to vertices and edges. It can calculate various structural properties, graph isomorphism, includes heuristics for community structure detection. It supports many file formats, such as GraphML, GML or Pajek.

The latest version of `igraph` (which was used in this thesis) is the 0.5.4 (Sep. 2010). One of the functions included in this version is the PageRank calculus*. This function is key to achieve the Chapter 7 (Section 7.3) results. The implementation of this function (`page.rank`) may be consulted in Csárdi & Nepusz (2006). A brief summary of this may be provided by the declaration of the function together their arguments:

```
page.rank (graph, vids = V(graph), directed = FALSE, damping = 0.85,
weights = NULL, options = igraph.arpack.default),
page.rank.old (graph, vids = V(graph), directed = TRUE,
niter = 1000, eps = 0.001, damping = 0.85, old = FALSE);
```

with the arguments defined by the next items:

- graph: The graph object.
- vids: The vertex of interest.
- directed: Boolean, whether to consider the directness of the edges. This is ignored for undirected graphs.
- damping: The damping factor ("d" in Chapter 7, Section 7.3).
- weights: Optional edge weights; it is either a null pointer, then the edges are not weighted, or a vector of the same length as the number of edges.
- niter: The maximum number of iterations to perform.

*This is the new PageRank implementation, based on the `ARPACK` library for calculating eigenvectors of sparse matrices.

- eps: The algorithm will consider the calculation as complete if the difference of PageRank values between iterations change less than this value for every node.
- old: Use or not, the previous version of PageRank.

A sample code for the PageRank calculus with `page.rank` would be the next:

```
...
edgelist <- cbind(pipes[,1],pipes[,2])
g1 <- graph( t(edgelist) )
plot.igraph(g1)
page.rank(g1)$vector
```

The PageRank's output vector will be added as input of the database which will be opened by NetLogo. Thus, we can represent interesting properties relative to the network graph, such as GIS information, along with the nodes size and colour to visualise the spatial distribution of PageRank.

A.2.2 kernlab library and specc function

`kernlab` [Karatzoglou (2006); Karatzoglou *et al.* (2006)] is an extensible package for kernel-based machine learning methods in R. It uses R's new S4* object model and provides a framework for creating and using kernel-based algorithms. Taking advantage of the inherent modularity of kernel-based methods, kernlab aims to allow the user to switch between kernels on an existing algorithm and even create and use own kernel functions for the kernel methods provided in the package. This last option was key to develop the algorithms proposed in the chapters 3 and 5 of this thesis.

The package contains dot product primitives (kernels), implementations of support vector machines and the relevance vector machine, Gaussian processes, a ranking algorithm, kernel PCA, kernel CCA, kernel feature analysis, on-line kernel methods and a spectral clustering algorithm. This last has been essential to our calculus of chapters 3 and 5. Thus, `specc` function is described in Karatzoglou *et al.* (2006) as a spectral clustering algorithm that clusters points using eigenvectors of kernel matrices derived from the data. A brief summary of this may be given by the declaration of the function

*S4 is a well defined and structured system class. It provides the way of implementing R code into a kind of class-oriented programming [Chambers (1998, 2006)].

together with their arguments. This declaration depends on the use and on the main inputs. We highlight the next two:

```
 ## S4 method for signature 'matrix'
specc(x, centers, kernel = ``rbfdot'', kpar = ``automatic'',
nystrom.red = FALSE, nystrom.sample = dim(x)[1]/6,
iterations = 200, mod.sample = 0.75, na.action = na.omit, ...)

 ## S4 method for signature 'kernelMatrix'
specc(x, centers, nystrom.red = FALSE, iterations = 200, ...)
```

with the main arguments defined by the next items:

- x: the matrix of data to be clustered or a symbolic description of the model to be fit.
- data: an optional data frame containing the variables in the model. By default the variables are taken from the environment which `specc` is called from.
- centers: Either the number of clusters or a set of initial cluster centers.
- kernel: The kernel function used in training and predicting. This parameter can be set to any function, of class kernel, which computes a dot product between two vector arguments. kernlab provides the most popular kernel functions, but the kernel may be defined by the user.
- kpar: a character string or the list of hyper-parameters (kernel parameters)*.
- nystrom.red: Nystrom method is used to calculate eigenvectors.

A sample code for spectral clustering with `specc` would be the next:

```
...
k <- 0.4*aff.matrix + 0.2*elev.matrix + 0.4*coord.matrix
k <- k + t(k)
diag(k) <- 1
k <- as.kernelMatrix(k)
specc(k, 3)
```

*cf. detailed information in Karatzoglou *et al.* (2006).

In Chapter 3 we take advantage of the fact that this function works with any kernel matrix to propose the semi-supervised clustering algorithm that mix into one kernel matrix both inputs and constraints. To represent (using colours) the node membership to each cluster, we will add this as input of the database used by NetLogo. Yet, this relation between R and NetLogo goes further in the MAS-boost algorithm proposed in Chapter 5. In effect, a flexible combination between both pieces of software is necessary. Firstly, NetLogo communicates to R the subgraphs sampled by multi-agent exploration. A semi-supervised clustering algorithm (Chapter 3) is applied to these subgraphs in R. Next, their corresponding silhouettes and memberships are sent back to NetLogo in order to update the weights of these nodes and to continue sampling subgraphs (in the proposed process of boosting).

A.3 Exchange framework between EPANET, R Language and NetLogo

An integrated methodological framework merging different kind of objectives and analysis will also require an integrated framework for the software used. First, we achieve this paradigm, applying the recently developed software of NetLogo-R-Extension. Then, we also create new R functionalities able to communicate with the hydraulic software of EPANET. Thus, this allows us to import and export data between R and EPANET, thus achieving the corresponding operations in R and validating the results in EPANET.

A.3.1 NetLogo - R - Extension

Thiele & Grimm (2010) present a Java extension of NetLogo that allows any R function (except functions with multi-line string return values) to be called directly from NetLogo programs. The interface created by this extension consists of NetLogo primitives, dealing with the communication between NetLogo and R via data and with calling R functions. It proposes to have access to direct interaction with the model by R functions. Thus, it opens the possibility to develop multi-agent based models, being we able to automatically test and validate them via R.

In this thesis, NetLogo-R-Extension is a valuable tool that speeds the associated calculations to the MAS-boost clustering algorithm (cf. Chapter 5). The most important application is the input/output exchange between both programs. Thus, the subgraphs

sampled by 'virus' exploration in NetLogo are the R inputs to apply semi-supervised clustering in every case. Other inter-programs calls could improve their performance by this extension. They are the updating of graph weights and the proposal to approach the MAS voting procedure (by the sum of cluster memberships in the database with aggregate results of the overall boosting process).

A.3.2 REPANET exchange

In relation to this thesis two data exchange functions to share information between R Language and EPANET 2.0 were developed. They are the functions `RimpEpa` and `RexpEpa`, to import data from EPANET to R and vice-versa, respectively. Both functions were developed in C Language and embed in the R environment. They are based on the structure of data output files of these respectively programs, to convert these output of one of them into data input of the other. Thus, on the one hand, R dataframes, representing water supply networks and the different properties calculated in R can be exported to a '.inp' (EPANET file extension) file. On the other hand '.inp' files are automatically manipulated to be as many '.data' files as inputs we wish to work in R environment. Instances of these files are related to nodes, pipes, coordinates or tanks and reservoirs characteristics, among others. Finally, we can merge all of these files into one '.Rdata' (R file extension) file.

A.3.2.1 RimpEpa.c

Part of the source code, in C, of the function `RimpEpa` is the next:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>


int main(int argc, char *argv[])
{ FILE *nodes, *pipes, *tanks, *reservoirs, *coords;
int c;
```

```
 FILE *f1 = fopen(argv[1], ‘‘r’’);

while ((c=fgetc(f1)) != EOF)
{
   if (c == ’[’)
   {
   if (fgetc(f1)==’J’)
         {
         fseek(f1, 12, SEEK_CUR);
         nodes = fopen(‘‘nodes.dat’’,‘‘w’’);
         while ((c=fgetc(f1)) != ’[’)
            fputc(c, nodes);

         fclose(nodes);
         }
   }
   else continue;
}
rewind(f1);
while ((c=fgetc(f1)) != EOF)
{
   if (c == ’[’)
   {
   if (fgetc(f1)==’P’ && fgetc(f1)==’I’)
         {
         fseek(f1, 7, SEEK_CUR);
         pipes = fopen(‘‘pipes.dat’’,‘‘w’’);
         while ((c=fgetc(f1)) != ’[’)
            fputc(c, pipes);

         fclose(pipes);
         }
   }
```

```
   else continue;
}
rewind(f1);
```

...

With this C code we are able to import to R nodes and pipes from the EPANET file. We could continue with a similar procedure to capture more WSN characteristics found in the source file: 'tanks', 'reservoirs', 'coordinates', 'elevations', etc. The next step should be to compile this function externally or embedded it in the R environment.

A.3.2.2 RexpEpa.c

Part of the source code, in C, of the function `RexpEpa` is the next:

```
#include <stdio.h>

int main(int argc, char *argv[])
{
int c, i;
FILE *rexport, *nodes, *reservoirs, *tanks, *pipes, *coordinates;
rexport=fopen(``RtoEpa.inp'', ``w'');
printf(``%d\n'',argc);

for (i=1; i<=argc-1; i=i+1) {
   if (argv[i][0] == 'n')
   {
      fprintf(rexport, ``\n[Junctions]\n;'');
      nodes=fopen(``nodes.dat'',``r'');

      while ((c=fgetc(nodes))!= EOF)
         fputc(c, rexport);

      fclose(nodes);
```

```
    }

}
...
}
```

This sample code shows the C source to export the WSN 'nodes' from R to EPANET. Similarly, we can export 'reservoirs', 'tanks', 'pipes', 'coordinates' and any other variables of interest. Merging all of them in just one file (we propose 'RtoEpa.inp') we could obtain the EPANET '.inp' input file, containing R output.

Appendix B

Overview of predictive models developed in water demand

B.1 The used predictive models

This section describes the modelling tools we have used in the experimental comparisons performed with our hour demand data set of Chapter 7 (Section 7.1).

B.1.1 Artificial Neural Networks (ANN)

An artificial neural network (ANN) is an interconnected group of artificial neurons. Each neuron executes a non-linear computation based on the input values and the resulting value is fed to other neurons. Neurons are usually arranged as a series of interconnected layers. Based on the data presented to the network, an algorithm (usually back-propagation) is used to iteratively adjust the neuron connection weights in such a way that the predictive performance of the network [Bishop (2005)] is improved. The steps involved in developing an ANN model are detailed in Maier & Dandy (2000), among other references.

The most common ANN network is the feed-forward network, which uses the back-propagation algorithm for training [Bougadis *et al.* (2005)]. Obtaining this type of network is an iterative process in which each sample case is presented several times to the input neurons of the ANN.

Usually, a typical three-layer feed-forward model is used for forecasting purposes [Lingireddy & Ormsbee (1973)]. Hidden nodes (h in the next equation) with appro-

priate non-linear transfer functions are used to process the information received by the p input nodes, each associated with one of the predictors. Finally, the model can be written as [Zhang & Qi (2005)]:

$$Y_t = \alpha_0 + \sum_{j=1}^{p} \alpha_j f \left(\sum_{i=1}^{h} \beta_{ij} y_{t-j} + \beta_{0j} \right) + \epsilon_t \tag{B.1}$$

where p is the number of input nodes, h is the number of hidden nodes, f is a sigmoid transfer function; α_j, with $j = 0, 1, \ldots, h$, is the vector of the weights from the hidden to the output nodes and β_{ij}, with $i = 0, 1, \ldots, p$ and $j = 1, \ldots, h$, are the weights from the input to hidden nodes. α_0 and β_{0j} are the weights of the arcs leaving from the bias terms.

B.1.1.1 Tuning ANN

In our comparative study (cf. Chapter 7, Section 7.1), we have used feed-forward neural networks with one hidden layer and the back-propagation learning algorithm [Wang *et al.* (2006); Zealand *et al.* (2005)]. Specifically, we have used the implementation of this type of ANN available in the `nnet` package [Venables & Ripley (2002)] of the R environment [R-Development-Core-Team (2010)]. The input is normalised by subtracting each column of the data set by its mean value and dividing by the standard deviation, to obtain data in the same scale and in agreement with the support domain of the activation functions. In terms of the different parameters of the R function used to obtain the ANN models, we have considered nine different alternatives by varying the number of hidden nodes (parameter `size`) between 3, 5, and 7; and also varying the learning rate (parameter `decay`) between 0.0001, 0.001, and 0.1.

B.1.2 Projection Pursuit Regression (PPR)

Projection pursuit regression (PPR) is a powerful, non-parametric regression method proposed by Friedman and Stuelze in 1981 [Friedman & Stueltze (1981)]. Recently, Dahl & Hylleberg (2004) included the method in their comparative study of flexible regression models for industrial purposes. Storlie & Helton (2008a) describe PPR embedded in a sensitivity analysis of multiple predictor smoothing methods. In Storlie & Helton (2008b), the authors revise these techniques in various application environments.

PPR explains the target variable as a sum of spline functions of projections of the input variables. For many practical problems, the data is usually of high dimension. The most common practice is the use of dimension-reduction transformations, such as linear projections, to project the original high dimensional data into a lower-dimensional space, in an attempt to find the intrinsic structure for visual inspection. The PPR model can be written as:

$$y_t = \mu_{ppr}(x_t, Q) + \epsilon_t \tag{B.2}$$

where

$$\mu_{ppr}(x_t, Q) = {x'}_t\beta + \sum_{j=1}^{\nu} \omega_j \varphi_j(x'_t\phi_j) \tag{B.3}$$

and

$$Q = (\beta, \omega_1, \ldots, \omega_\nu, \phi_1, \ldots, \phi_\nu) \tag{B.4}$$

Parameters ϕ_j define the projection of the input vector x_t onto a set of planes indicated by j. These projections are transformed by the nonlinear activation functions, noted $\phi_j(.)$, and these, in turn, are linearly combined with weights ω_j and added to the linear part, $x'_t\beta$, to form the output variable y_t.

Friedman & Stueltze (1981) proposed an initial algorithm to obtain an estimate of Q. This algorithm consists of two components: a PP index and a PP algorithm. A PP index, $I(\alpha)$, is the objective function computed on the projected data set, and measures the "interestingness" of the projection α (the possible directions). $I(\alpha)$ is an estimate of the distance between the distribution of the projected data and an uninteresting distribution (implicitly, depending on the data). The larger the index value, the more interesting is the projection and thus the method tries to maximise the value. A PP algorithm is a numerical optimisation algorithm that varies the projection direction so as to find the optimal projections.

B.1.2.1 Tuning PPR

In our experiments (cf. Chapter 7, Section 7.1), we have again used an implementation of PPR available in R, through the function `ppr()` of package `stats` [R-Development-Core-Team (2010)]. In this function, the parameter `nterms` sets the number of linear combinations that will be included in the projection pursuit model. With the use

of parameter `max.terms` it is possible to set a maximum number of terms that will be tried. After adding this maximum number of terms, the "worse" terms will be iteratively removed by back-fitting until a model with `nterms` is reached. The levels of optimisation (argument `optlevel`) differ in how thoroughly the models are re-fitted during this process. At level 0, the existing ridge terms are not re-fitted. At level 1, the projection directions are not re-fitted, but ridge functions and regression coefficients are fitted. Level 2 re-fits all the terms. In our experiments, we have tried four different levels, together with four values of regression smoothing parameter `bass` (0, 2, 5, 7). This last parameter will increase the regression-smoothing as the values grow. In total, 12 PPR variants are checked.

B.1.3 Multivariate Adaptive Regression Splines (MARS)

Multivariate adaptive regression splines (MARS) were first introduced by Friedman & Stueltze (1981) in an attempt to overcome some of the limitations of regression trees. This procedure generalises recursive partitioning methods such as classification and regression trees (CART), while sharing their ability to capture high-order interactions. However, the procedure has more power and flexibility to model additive relationships [Hastie & Tibshirani (1990)]. MARS is a regression method applicable to high dimensional data, and is usually considered as an excellent example of a modern statistical approach to regression.

MARS models the target variable using a linear combination of splines, which are automatically built (matching the boundaries of each region) from an increasing set of piecewise-defined linear basic functions [Moisen & Frescino (2002)]. The model takes the form of an expansion in product spline basis functions, where the number of these basis functions and the parameters associated with each are determined by the data. The model can be represented in a form that separately identifies the additive contributions and those associated with different multivariate interactions. To avoid over-fitting, it is possible to subsequently apply a pruning mechanism to reduce model complexity. One form of writing the MARS model is the following,

$$\hat{f}(x) = a_0 + \sum_{k_m=1} f_i(x_i) + \sum_{k_m=2} f_{ij}(x_i, x_j) + \sum_{k_m=3} f_{ijk}(x_i, x_j, x_k) + \dots \tag{B.5}$$

The functions in the first sum are defined as,

$$f_i(x_i) = \sum_{k_m=1/i\in V(m)} a_m B_m(x_i) \tag{B.6}$$

where $V(m)$ is the variable associated with the mth basis function, B_m, that survives backward selection strategies.

The second sum is over all basis functions that involve two variables,

$$f_i(x_i, x_j) = \sum_{k_m=2/i,j\in V(m)} a_m B_m(x_i, x_j) \tag{B.7}$$

The third sum is over all basis functions that involve three variables, and so on.

B.1.3.1 Tuning MARS

MARS models are implemented in several R packages. In our experiments (cf. Chapter 7, Section 7.1) we have used the implementation in package `earth` [Milborrow (2009)], which is a re-implementation of the original MARS code by Trevor Hastie and Robert Tibshirani (1991) with similar, but not identical, results. We have considered 12 variants of these models formed by different combinations of the parameter `nk` that sets the maximum number of term before pruning (values 10 and 17); two variants (1 and 2) of the parameter `degree` that sets the maximum degree of interaction; and three variants (0.01,0.001 and 0.0005) of the parameter `thresh` specifying the forward stepwise stopping threshold.

B.1.4 Support Vector Regression (SVR)

Smola [Smola & Schölkopf (2004, 1998)] published a fundamental tutorial giving an overview of the basic ideas underlying SVM for function estimation. Vapnik (1995, 1998), and Shawe-Taylor & Cristianini (2000) are some of the essential references for SVM. These are complemented with the works of Karatzouglou [Karatzoglou (2006); Karatzoglou *et al.* (2006)] for implementing a SVM and kernel method environment in R Language; or Canu *et al.* (2003) for developing a MatLab toolbox.

In Support Vector Regression (SVR) the basic idea is to map the data x into a high dimensional feature space F via a nonlinear mapping Φ and obtain a linear regression model in this new space:

$$f(x) = (\omega \cdot \Phi(x)) + b \tag{B.8}$$

with $\Phi : R^n \rightarrow F$, $\omega \in F$, where b is a threshold.

Thus, linear regression in a high dimensional (feature) space corresponds to non–linear regression in the low dimensional input space R^n. Since Φ is fixed, ω is determined from the data by minimising the sum of empirical risk $R_{emp}[f]$ and a complexity term $||\omega||^2$, which enforces flatness in the feature space:

$$R_{reg}[f] = R_{emp}[f] + \lambda||\omega||^2 = \sum_{i=1}^{l} C(f(x_i) - y_i) + \lambda||\omega||^2 \tag{B.9}$$

where l denotes the sample size, $C(.)$ is a cost function (e.g. Vapnik's ϵ-insensitive loss function) and λ is a regularisation constant.

For a large set of cost functions, the previous equation can be minimised by solving a quadratic programming problem, which is uniquely solvable. It is possible to write the vector ω in terms of the data points:

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(\Phi(x_i)) \tag{B.10}$$

with α_i,α_i^* being the solution of the afore–mentioned quadratic programming problem.

The problem may be rewritten as products in the low dimensional space:

$$\omega = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(\Phi(x_i) \cdot \Phi(x)) + b = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{B.11}$$

In Equation B.11 the kernel function is introduced: $K(x_i, x_j) = (\Phi(x_i) \cdot \Phi(x_j))$. It can be shown that any asymmetric kernel function, K, satisfying Mercer's condition, corresponds to a dot product in some feature space. A common kernel is a Radial Basis Function (RBF) kernel:

$$K(x_i, x_j) = \exp\left(-\gamma \times ||x_i - x_j||^2\right) \tag{B.12}$$

being γ a user-defined parameter.

B.1.4.1 Tuning SVR

The R package `e1071` [Dimitriadou *et al.* (2009)], contains the `svm()` function that is mostly programmed in R but uses the optimisers found in *libsvm* [Chang & Lin (2001)], which provide a very efficient C++ version of the sequential minimisation optimisation (SMO). SVM is an excellent tool for classification, novelty detection, and regression.

In our comparative study (cf. Chapter 7, Section 7.1), we have focused on variants of the γ parameter of the RBF kernel (`gamma` parameter in the `svm()` function). Namely, we have tried the values 0.01,0.001 and 0.0005. Different values of `cost` parameter, namely values 10, 150 and 200. In summary, we have considered 9 variants of SVR models.

B.1.5 Random Forests

Random forests [Breiman (2001)] are formed by an ensemble of tree-based models [Breiman *et al.* (1984)]. They can be used for classification tasks in which the base models are classification trees, or regression tasks where the base models are regression trees. The particularity of random forests when compared to other ensemble strategies lies on the process by which the trees are built. Namely, at each split in a tree within the forest, the test is chosen from a randomly selected sub-set of the independent variables. Moreover, the obtained trees are not pruned.

Random forests have been proving to be outstanding predictive models in many classification and regression tasks. These methods can also be used for estimating the variable importance and also for outlier detection. They are reasonably fast to obtain and can be easily parallelised if more speed is required.

B.1.5.1 Tuning Random Forests

The R package `randomForest` [Liaw & Wiener (2002)] implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. In our experiments we have considered 3 variants of the parameter `ntree`, which controls the number of trees within the ensembles. For this parameter we have considered the values 250, 500 and 750.

B.1.6 Weighted pattern-based model for water demand forecasting

The final option that we have considered in our experiments was a simple and heuristic model based on observations from our exploratory analysis presented in Subsection 7.1.1 of Chapter 7. The main objective of this simple model is to serve as a kind of baseline from which we will compare the other more sophisticated models presented in the previous sections. This method is based on the pattern of demand, namely, its seasonal properties. There are some relationships with the method proposed by Alvisi *et al.* (2007). The basic idea is to explore the similarities in the hourly demand that were observed previously (cf. Figure 7.1), for each of the days of the week. Namely, a first term of the model includes the weighed average of the water demand for the hour and day of week for which a prediction is required. This average is calculated using past values of the water demand for the same hour and day of the week. This average value is then adjusted by a second term of the model in order to correct it for some specificities of the day in question (e.g. special weather conditions). This adjustment is carried out again by a weighted average of the errors committed by the strategy of the first term in the most recent hours for which a prediction was carried out. This means, for instance, that if today the water demand is consistently above what was on the past for the same hours and day of the week (which is captured by the average on the first term), then there will be a consistent error of this first term under-estimating the water demand. This error will then enter in the second term of the model so as to compensate the first term for the specificity of the current day.

From a theoretical point of view, this proposal can be seen as sharing some ideas with partially linear models [Härdle *et al.* (2000)]. As partial linear models, this proposal contains two components: an initial part that reflects the typical behaviour of the system; and a second part that corrects/adjusts this initial forecast to account for the specifics of the context for which a prediction is being obtained.

Formally, the prediction of this model for the water demand at time $i+1$ is given by,

$$W_{i+1} = f_1\left(\{W_{i+1-k\times 24\times 7}\}_{k=0}^{p}\right) + f_2\left(\{e_{i-l}\}_{l=0}^{q}\right) \tag{B.13}$$

where p and q are two parameters controlling the amount of memory the model has, while $f_1()$ and $f_2()$ are two averaging functions that can either be standard averages

or any form of weighted average that gives for instance more relevance to more recent values.

The e_i values in the model are calculated as follows,

$$e_i = f_1\left(\{W_{i-k\times 24\times 7}\}_{k=0}^{p}\right) - W_i \tag{B.14}$$

In our experiments (cf. Chapter 7, Section 7.1) we have tried several values for the different parameters of the model. Namely, we have used for the averaging functions the median of the values and also an exponential averaging function. For the value of p we have also used the maximum value allowed by the training set size, i.e. we calculate the average using all water demand values of the same day and hour. Finally, for the value of q we have tried the values 10, 18 and 24.

B. OVERVIEW OF PREDICTIVE MODELS DEVELOPED IN WATER DEMAND

Appendix C

Conclusiones

La división eficiente de las redes de abastecimiento de agua en los denominados *supply clusters* es un importante paradigma en su gestión, que puede dar lugar a importantes ahorros desde un punto de vista hidráulico y económico. Mediante métodos kernel y junto con sistemas multi-agente, se puede obtener esta partición de la red de manera eficiente. Pero además, estas metodologías van más allá, ofreciendo un marco de trabajo adecuado para llevar a cabo tareas de gestión en las redes ya divididas.

El tema principal de esta tesis ha sido el desarrollo, aplicación y análisis experimental de algoritmos de cluster que ofrezcan una solución del problema de división de la red. Las secciones siguientes revisan las principales conclusiones obtenidas. Después, se resumen brevemente las contribuciones aportadas por esta tesis. Finalmente, se discuten algunas ideas de cómo las conclusiones pueden aplicarse más allá de esta tesis y cómo este trabajo puede extenderse en nuevas direcciones.

C.1 Conclusiones

Desde un punto de vista hidráulico, la finalidad fundamental de esta tesis es llevar a cabo la sectorización de una red de abastecimiento para la posterior creación de modelos de toma de decisiones en el suministro de agua potable. Debido a este motivo, surge una hipótesis de trabajo consistente en la implementación de dichos sectores teniendo en cuenta la gestión final de la red. Los criterios clásicos resultan ambiguos en el desarrollo de esta hipótesis, por lo que se propone su creación mediante la técnica estadística del análisis cluster. La diversidad de la información disponible (las bases de datos de

los abastecimientos cuentan con ítems de naturaleza continua, discreta y provenientes de herramientas de información geográfica) hace que se busquen procedimientos adecuados para trabajar, en conjunto, con los datos que definan el grafo asociado a la red. Este es el motivo por el que se desarrollan algoritmos de agrupamiento basados tanto en técnicas multi-agente como de aprendizaje semi-supervisado. Ambas metodologías son lo suficientemente flexibles para explotar, plenamente, dicha información del abastecimiento de agua, además de ser capaces de adecuarse a su estructura de grafo. En el caso de redes de gran tamaño se propone integrar sus soluciones bajo una metodología boosting.

Una vez creada la división de la red de distribución de agua, se comprueba que en el estudio de su abastecimiento se evitan los sesgos de un análisis más detallado y se gana precisión respecto de los resultados que puedan obtenerse de trabajar con el total de la red. Partiendo de esta premisa, se propone una continuidad en la tendencia metodológica con que se establecieron los sectores, para así enfocar diferentes paradigmas de su gestión. La predicción de la demanda, la detección de anomalías en el suministro, la caracterización de las diferentes zonas para su clasificación y la ubicación de sensores en la red, representan ejemplos de aplicaciones que mejoran sus resultados en redes sectorizadas. Todas ellas se pueden fundamentar en sistemas multi-agente y/o modelos de aprendizaje automático basados en espacios kernel.

El resto de esta sección presenta una descripción más detallada de las contribuciones de esta tesis sobre la sectorización.

C.1.1 Clusters de abastecimiento de agua mediante aprendizaje semi-supervisado

Clásicamente, la división de una red de abastecimiento de agua en distritos hidrométricos tiene como finalidad la mejora en la detección de fugas usando información sobre la cota de cada nodo y su presión y demanda asociadas. Mediante el cluster semi-supervisado (cf. Capítulo 3) proponemos el aumento, o cambio, de la perspectiva de este objetivo. Esto puede ser realizado teniendo en cuenta otras fuentes de información diferentes, que sean incluidas dentro del criterio de división de la red en zonas (clusters) hidráulicas.

Comparada con otras metodologías que, o bien usan información gráfica o bien usan información numérica, el cluster semi-supervisado usa ambas y de una manera

más eficiente y robusta. La flexibilidad para incluir diferentes inputs en el estudio, con diferentes pesos asociados, es otra mejora de esta metodología.

C.1.2 División-agente de los sistemas de distribución de agua en clusters de abastecimiento

La metáfora de multi-agente fue introducida en el Capítulo 4 para dividir la red hidráulica en clusters de abastecimiento. Además de la arquitectura centralizada tradicional (de un solo agente) como solucionador de problemas, este Capítulo muestra que es posible usar sistemas de agentes inteligentes, o aplicar simulaciones multi-agente, para verificar hipótesis acerca de los diferentes procesos involucrados en la distribución de agua. La inclusión de un comportamiento negociador de los agentes propone, directamente, un análisis de sensibilidad de la solución encontrada con anterioridad mediante cooperación y competición entre agentes.

C.1.3 Remuestreo multi-agente de clusters semi-supervisados

El cluster MAS-boost presentado en el Capítulo 5 propuso una división de la red de abastecimiento escalable a cualquier tamaño de la misma y tiene en consideración las condiciones hidráulicas necesarias para que cada distrito en que parte la red sea, realmente, un sector de abastecimiento. Además, este capítulo propone avances en varios sentidos:

- La cantidad de cálculos en el paradigma de spectral-clustering se reduce, haciéndolo un algoritmo computacionalmente factible, mediante remuestreo, incluso en el caso de redes de gran tamaño.

- Se propone una solución multi-agente novedosa en el muestreo de grafos por exploración y en el sistema de reponderación de los métodos de remuestreo.

- Se implementa un pre-cluster rápido, basado en multi-agente, en la fase de propagación de etiquetas del cluster semi-supervisado.

- Hidráulicamente, este capítulo propone continuar aprovechando las ventajas del cluster semi-supervisado, haciendo uso tanto de información gráfica como numérica, para así mejorar la división de la red de abastecimiento en sectores.

C.1.4 Gestión de la red dividida en clusters de abastecimiento

Trabajar con una red bien sectorizada puede mejorar la gestión de toda la red. Veamos en qué sentido la tesis apoya esta afirmación.

C.1.4.1 Modelos predictivos

En el Capítulo 7 (Sección 7.1) consideramos varios métodos de aprendizaje automático alternativos para resolver las tareas de predicción de la demanda en un sector de abastecimiento. Se describieron brevemente estas técnicas (Apéndice B) y se llevaron a cabo la comparación experimental de un gran número de variantes de estos modelos, usando datos de un caso-estudio. Los resultados de esta comparación han identificado a los modelos de regresión de vectores soporte como los más precisos. Los experimentos también revelaron un cierto desajuste en el funcionamiento de algunas de las variantes de las redes neuronales consideradas.

También se abordó el problema de la correcta estimación del funcionamiento de diversos modelos de predicción con datos de series temporales que plantean cambios claros en su dinámica (a través del tiempo). Se desarrolló un experimento de simulación de Monte Carlo que asegura estimadores insesgados a la hora de seleccionar medidas de evaluación de dichos modelos de predicción, y también fueron considerados y comparados dos estrategias de construcción de modelos (modelos de ventanas de tiempo deslizantes y de ventanas de tiempo crecientes) que son capaces de trabajar con datos cuyo comportamiento puede ser cambiante a lo largo del tiempo (como puede ser el caso de modelos no estacionarios). Los experimentos que se llevaron a cabo confirmaron las ventajas de estos modelos de aprendizaje; si bien, entre ellos se encontraron pequeñas diferencias.

C.1.4.2 Detección de anomalías

La aplicación desarrollada en el Capítulo 7 (Sección 7.2) es un modelo causal para detectar e identificar anomalías, basado en máquinas de vectores soporte y un algoritmo kernel causal. El modelo final incluye variables continuas e inputs discretos, provenientes de una fase previa de clasificación de la anomalía. Este modelo ofrece un conocimiento profundo del comportamiento en el consumo de agua potable, incluyendo su red de abastecimiento y los elemento que la componen (expuestos a posibles fallos).

De esta manera, su aplicación puede resultar una ayuda beneficiosa para comprender y satisfacer las necesidad de cada sector de abastecimiento.

C.1.4.3 Importancia relativa de los nodos

El Capítulo 7 (Sección 7.3) presentó una adaptación del algoritmo PageRank de Google para los grafos de una red de abastecimiento. Esto abre la posibilidad de trabajar con un algoritmo de probada eficiencia en bases de datos de gran tamaño, teniendo una adaptación sencilla a las particularidades propias de una red de abastecimiento. Esta metodología no sólo está relacionada con índices de vulnerabilidad, sino que también podría ser útil para llevar a acabo modelos de calidad del agua, tratar problemas de localización de sensores o posibilitar diversos planes de rehabilitación, entre otras aplicaciones. Por otro lado, el establecimiento de un sistema de ranking en los nodos puede ser una herramienta útil en la búsqueda de criterios eficientes para la sectorización.

De nuevo, los métodos kernel han sido la base del desarrollo de nuevas tendencias y posibilidades en la gestión de los sistemas de abastecimiento de agua.

C.1.5 Software empleado y desarrollado

El apéndice A introduce los tres programas empleados a lo largo de esta tesis: Lenguaje R [R-Development-Core-Team (2010)], NetLogo [Wilensky (1999)] y EPANET [Rossman (2000)]. Mientras se implementaron los algoritmos propuestos, surge la necesidad de compartir información entre estos programas. Así, además de usar el *NetLogo - R - Extension* de Thiele & Grimm (2010) fueron desarrolladas dos funciones para que compartir información entre R y EPANET 2.0. Son las funciones `RimpEpa` and `RexpEpa`, que importan datos desde EPANET a R y viceversa, respectivamente. Ambas funciones fueron desarrolladas en C e incorporadas al entorno de R. La Subsección A.3.2 del Apéndice A detalla las características y el uso de estas funciones.

C.2 Trabajo futuro

Esta tesis propone algunas líneas de trabajo a seguir en un futuro. De nuevo, todas ellas están relacionadas con métodos kernel, sistemas multi-agente y alguna integración adecuada de los mismos. Estos procedimientos pueden ser desarrollados en mayor profundidad incluyendo más información de la red, como es el usar la importancia

de los nodos como otra valiosa variable input. Además, también es posible proponer criterios multi-objetivo en las diferentes funciones a optimizar. Una tercera vía de trabajo puede ser explotar la propiedad de 'dispersión' de los grafos asociados a la red de abastecimiento de agua. Además de esto, la gestión de la red aprovecha el trabajar con una red previamente sectorizada, tal como hemos comprobado en la tesis con el ranking de los nodos, proponiendo modelos de predicción de la demanda y detectando anomalías. Sin embargo, otras gestiones pueden ser probadas: planes de rehabilitación, localización de sensores, evaluación de la calidad del agua, son ejemplos de futuras tareas de resultados prometedores si continuamos la línea de trabajo de esta tesis. A continuación, se presenta este trabajo futuro en mayor detalle.

C.2.1 Clusters de abastecimiento de agua mediante aprendizaje semi-supervisado

Respecto del cluster semi-supervisado, propuesto en el Capítulo 3, podemos usar inputs diferentes de los propuestos. Este es el caso de los diámetros de las tuberías y su edad, que pueden emplearse como posible ponderación en la planificación de una rehabilitación de la red por sectores. Otro punto de vista es considerar el uso de índices de vulnerabilidad para las tuberías, teniendo en cuenta los efectos de hipotéticos riesgos en el establecimiento de los sectores. Además, se podrían incluir diversas modificaciones en el algoritmo de clustering para, posteriormente, ser comparadas en su habilidad para construir modelos.

C.2.2 División-agente de los sistemas de distribución de agua en clusters de abastecimiento

Dentro de los diferentes escenarios que emplean los sistemas multi-agente en el propósito de ayuda a la decisión (por ejemplo, en una compañía de agua), el Capítulo 4 abordó el problema de la división de la red de abastecimiento en sectores hidráulicos. Una revisión de estos métodos puede enfocarse en el desarrollo implementaciones de otros escenarios en las aplicaciones multi-agente en el campo del abastecimiento de agua, incluyendo aspectos relacionados con la calidad del agua, localización de sensores y cualquier otro asunto relacionado con su gestión.

C.2.3 Remuestreo multi-agente de clusters semi-supervisados

Las futuras investigaciones acerca del algoritmo MAS-boost (cf. Capítulo 5) se enfocarán en la mejora de la metodología de remuestreo empleada. Entre otras cosas, esto podría incluir una guía de muestreo en subgrafos formados por los elementos frontera de una sectorización previa de la red. A través de este análisis de sensibilidad podríamos reducir los costes asociados a las reformas a realizar sobre unos sectores ya construidos.

C.2.4 Gestión de la red dividida en clusters de abastecimiento

En una red ya sectorizada podría ser útil ubicar sensores según la importancia relativa de los nodos de la red (Capítulo 7, Sección 7.3). Pero esta importancia también puede ser interesante como criterio para el mismo proceso de sectorización, como otra variable más a tener en cuenta. Otros usos respecto de la importancia de los nodos puede ser su inclusión en planes de rehabilitación o en la gestión de las vulnerabilidades de la red.

Respecto de los modelos predictivos, un trabajo futuro debiera incluir la mejora del modelo ponderado de la demanda-patrón (cf. Capítulo 7 y Apéndice B); lo que quizá sea posible incluyendo, en su algoritmo, diferentes particiones de tiempo. En cuanto al estudio de las anomalías (Capítulo 7, Sección 7.2), no deseamos únicamente detectarlas en el comportamiento de la demanda de agua, sino que también debiéramos trabajar para ser capaces de encontrar relaciones causa-efecto justificadas en todo el entorno de la demanda de agua potable.

Appendix D

Conclusions

La divisió eficient de les xarxes de proveïment d'aigua en els denominats *supply clusters* és un important paradigma en la seua gestió, que pot donar lloc a importants estalvis des d'un punt de vista hidràulic i econòmic. Mitjançcant mètodes kernel i juntament amb sistemes multi-agent, es pot obtenir aquesta partició de la xarxa de manera eficient. Però a més, aquestes metodologies van més enllà, oferint un marc de treball adequat per a portar a terme tasques de gestió en les xarxes ja dividides.

El tema principal d'aquesta tesi ha estat el desenvolupament, aplicació i anàlisi experimental d'algorismes de cluster que oferisquen una solució del problema de divisió de la xarxa. Les seccions següents revisen les principals conclusions obtingudes. Després, es resumeixen breument les contribucions aportades per aquesta tesi. Finalment, es discuteixen algunes idees de com les conclusions poden aplicar-se més enllà d'aquesta tesi i com aquest treball pot estendre's en noves adreces.

D.1 Conclusions

Des d'un punt de vista hidràulic, la finalitat fonamental d'aquesta tesi és portar a terme la sectorització d'una xarxa de proveïment per a la posterior creació de models de presa de decisions en el subministrament d'aigua potable. A causa de aquest motiu, sorgeix una hipòtesi de treball consistent en la implementació d'aquests sectors tenint en compte la gestió final de la xarxa. Els criteris clàssics resulten ambigus en el desenvolupament d'aquesta hipòtesi, pel que es proposa la seua creació mitjançcant la tècnica estadística de l'anàlisi cluster. La diversitat de la informació disponible (les bases de dades dels

proveïments conten amb ítems de naturalesa contínua, discreta i provinents d'eines d'informació geogràfica) fa que se cerquen procediments adequats per a treballar, en conjunt, amb les dades que definisquen el grafo associat a la xarxa. Aquest és el motiu pel qual es desenvolupen algorismes d'agrupament basats tant en tècniques multi-agent com d'aprenentatge semi-supervisat. Ambdues metodologies són prou flexibles per a explotar, plenament, aquesta informació del proveïment d'aigua a més de ser capaces d'adequar-se a la seua estructura de grafo. En el cas de xarxes de gran grandária es proposa integrar les seues solucions sota una metodologia boosting.

Una vegada creada la divisió de la xarxa de distribució d'aigua, es comprova que en l'estudi del seu proveïment s'eviten els biaixos d'una anàlisi més detallada i es guanya precisió respecte dels resultats que puguen obtenir-se de treballar amb el total de la xarxa. Partint d'aquesta premissa, es proposa una continuïtat en la tendència metodològica amb que es van establir els sectors, per a així enfocar diferents paradigmes de la seua gestió. La predicció de la demanda, la detecció d'anomalies en el subministrament, la caracterització de les diferents zones per a la seua classificació i la ubicació de sensors en la xarxa, representen exemples d'aplicacions que milloren els seus resultats en xarxes sectorizadas. Totes elles es poden fonamentar en sistemes multi-agent i/o models d'aprenentatge automàtic basats en espais kernel.

La resta d'aquesta secci presenta una descripció més detallada de les contribucions d'aquesta tesi sobre la sectorització.

D.1.1 Clusters de proveïment d'aigua mitjançant aprenentatge semi-supervisat

Clàssicament, la divisió d'una xarxa de proveïment d'aigua en districtes hidromètricos té com finalitat la millora en la detecció de fugides usant informació sobre la cota de cada node i la seua pressió i demanda associades. Mitjançant el cluster semi-supervisat (cf. Capítol 3) proposem l'augment, o canvi, de la

Comparada amb altres metodologies que, o bé usen informació gràfica o bé usen informació numèrica, el cluster semi-supervisat usa ambdues i d'una manera més eficient i robusta. La flexibilitat per a incloure diferents inputs en l'estudi, amb diferents pesos associats, és altra millora d'aquesta metodologia.

D.1.2 Divisió-agent dels sistemes de distribució d'aigua en clusters de proveïment

La metàfora de multi-agent va ser introduïda en el Capítol 4 per a dividir la xarxa hidràulica en clusters de proveïment. A més de l'arquitectura centralitzada tradicional (d'un sol agent) com solucionador de problemes, aquest Capítol mostra que és possible usar sistemes d'agents intelligents, o aplicar simulacions multi-agent, per a verificar hipòtesi sobre els diferents processos involucrats en la distribució d'aigua. La inclusió d'un comportament negociador dels agents proposa, directament, una anàlisi de sensibilitat de la solució oposada amb anterioritat mitjançant cooperació i competició entre agents.

D.1.3 Remostreig multi-agent de clusters semi-supervisats

El cluster MAS-boost presentat en el Capítol 5 va proposar una divisió de la xarxa de proveïment escalable a qualsevol grandària de la mateixa i té en consideració les condicions hidràuliques necessàries perquè cada districte que parteix la xarxa siga, realment, un sector de proveïment. A més, aquest capítol proposa avanços en diversos sentits:

- La quantitat de càlculs en el paradigma de spectral-clustering es redueix, fent-lo un algorisme computacionalment factible, mitjançant re-mostreig, fins i tot en el cas de xarxes de gran grandària.
- Es proposa una solució multi-agent nova en el mostreig de grafos per exploració i en el sistema de re-ponderació dels mètodes de re-mostreig.
- S'implementa un pre-cluster ràpid, basat en multi-agent, en la fase de propagació d'etiquetes del cluster semi-supervisat.
- Hidràulicament, aquest capítol proposa continuar aprofitant els avantatges del cluster semi-supervisat, fent ús tant d'informació gràfica com numèrica, per a així millorar la divisió de la xarxa de proveïment en sectors.

D.1.4 Gestió de la xarxa dividida en clusters de proveïment

Treballar amb una xarxa ben sectorizada pot millorar la gestió de tota la xarxa. Vegem en quin sentit, la tesi, dóna suport aquesta afirmació.

D.1.4.1 Models predictius

En el Capítol 7 (Secció 7.1) considerem diversos mètodes d'aprenentatge automàtic alternatius per a resoldre les tasques de predicció de la demanda en un sector de proveïment. Es van descriure breument aquestes tècniques (Apèndix B) i es van portar a terme la comparança experimental d'un gran nombre de variants d'aquests models, usant dades d'un cas-estudi. Els resultats d'aquesta comparança han identificat als models de regressió de vectors suport com els més precisos. Els experiments també van revelar un cert desajustament en el funcionament d'algunes de les variants de les xarxes neuronals considerades.

També es va abordar el problema de la correcta estimació del funcionament de diversos models de predicció amb dades de sèries temporals que plantegen canvis clars en la seua dinàmica (a través del temps). Es va desenvolupar un experiment de simulació de Monte-Carlo que assegura estimadors insesgados a l'hora de seleccionar mesures d'avaluació d'aquests models de predicció, i també van ser considerats i comparats dues estratègies de construcció de models (models de finestres de temps lliscants i de finestres de temps creixents) que són capaços de treballar amb dades el comportament de les quals pot ser canviant al llarg del temps (com pot ser el cas de models no estacionaris). Els experiments que es van portar a terme van confirmar els avantatges d'aquests models d'aprenentatge si bé, entre ells es van trobar menudes diferències.

D.1.4.2 Detecció d'anomalies

L'aplicació desenvolupada en el Capítol 7 (Secció 7.2) és un model causal per a detectar i identificar anomalies, basat en màquines de vectors suport i un algorisme kernel causal. El model final inclou variables contínues i inputs discrets, provinents d'una fase prèvia de classificació de l'anomalia. Aquest model ofereix un coneixement profund del comportament en el consum de aigua potable, incloent la seua xarxa de proveïment i els element que la componen (exposats a possibles fallades). D'aquesta manera, la seua aplicació pot resultar una ajuda beneficiosa per a comprendre i satisfer les necessitat de cada sector de proveïment.

D.1.4.3 Importància relativa dels nodes

El Capítol 7 (Secció 7.3) va presentar una adaptació de l'algorisme PageRank de Google per als grafos d'una xarxa de proveïment. Açò obri la possibilitat de treballar amb un algorisme de provada eficiència en bases de dades de gran grandària, tenint una adaptació senzilla a les particularitats pròpies d'una xarxa de proveïment. Aquesta metodologia no només està relacionada amb índexs de vulnerabilitat, sinó que també podria ser útil per a dur a acabe models de qualitat de l'aigua, tractar problemes de localització de sensors o possibilitar diversos plans de rehabilitació, entre altres aplicacions. D'altra banda, l'establiment d'un sistema de rànquing en els nodes pot ser una eina útil en la recerca de criteris eficients per a la sectorització.

De nou, els mètodes kernel han estat la base del desenvolupament de noves tendències i possibilitats en la gestió dels sistemes de proveïment d'aigua.

D.1.5 Software emprat i desenvolupat

L'apèndix Aintrodueix els tres programes emprats al llarg d'aquesta tesi: Llenguatge R [R-Development-Core-Team (2010)], NetLogo [Wilensky (1999)] i EPANET [Rossman (2000)]. Mentre es van implementar els algorismes proposats, sorgeix la necessitat de compartir informació entre aquests programes. Així, a més d'usar el *NetLogo - R - Extension* de Thiele & Grimm (2010) an ser desenvolupades dues funcions perquè compartir informació entre R i EPANET 2.0. Són les funcions `RimpEpa` i `RexpEpa`, que importen dades des de EPANET a R i viceversa, respectivament. Ambdues funcions van ser desenvolupades en C i incorporades a l'entorn de R. La Subsección A.3.2 de l'Apèndix A detall les característiques i l'ús d'aquestes funcions.

D.2 Treball futur

Aquesta tesi proposa algunes línies de treball a seguir en un futur. De nou, totes elles estan relacionades amb mètodes kernel, sistemes multi-agent i alguna integració adequada dels mateixos. Aquests procediments poden ser desenvolupats en major profunditat incloent més informació de la xarxa, com és l'usar la importància dels nodes com altra valuosa variable input. A més, també és possible proposar criteris multi-objectiu en les diferents funcions a optimitzar. Una tercera via de treball pot ser explotar la propietat de 'dispersió' dels grafos associats a la xarxa de proveïment d'aigua. A més d'açò, la

gestió de la xarxa aprofita el treballar amb una xarxa prèviament sectorizada, tal com hem comprovat en la tesi amb el rànquing dels nodes, proposant models de predicció de la demanda i detectant anomalies. No obstant això, altres gestions poden ser provades: plans de rehabilitació, localització de sensors, avaluació de la qualitat de l'aigua, són exemples de futures tasques de resultats prometedors si continuem la línia de treball d'aquesta tesi. A continuació, es presenta aquest treball futur en major detall.

D.2.1 Clusters de proveïment d'aigua mitjançant aprenentatge semi-supervisat

Respecte del cluster semi-supervisat, proposat en el Capítol 3, podem usar inputs diferents dels proposats. Aquest és el cas dels diàmetres de les canonades i la seua edat, que poden emprar-se com possible ponderació en la planificació d'una rehabilitació de la xarxa per sectors. Altre punt de vista és considerar l'ús d'índexs de vulnerabilitat per a les canonades, tenint en compte els efectes d'hipotètics riscos en l'establiment dels sectors. A més, es podrien incloure diverses modificacions en l'algorisme de clustering per a, posteriorment, ser comparades en la seua habilitat per a construir models.

D.2.2 Divisió-agent dels sistemes de distribució d'aigua en clusters de proveïment

Dins dels diferents escenaris que usen els sistemes multi-agent en el propòsit d'ajuda a la decisió (per exemple, en una companyia d'aigua), el Capítol 4 va abordar el problema de la divisió de la xarxa de proveïment en sectors hidràulics. Una revisió d'aquests mètodes pot enfocar-se en el desenvolupament implementacions d'altres escenaris en les aplicacions multi-agent en el camp del proveïment d'aigua, incloent aspectes relacionats amb la qualitat de l'aigua localització de sensors i qualsevol altre assumpte relacionat amb la seua gestió.

D.2.3 Remostreig multi-agent de clusters semi-supervisats

Les futures investigacions sobre algorisme MAS-boost/ (cf. Capítol 5) s'enfocaran en la millora de la metodologia de re-mostreig emprada. Entre altres coses, açò podria incloure una guia de mostreig en subgrafos formats pels elements frontera d'una sectorització prèvia de la xarxa. A través d'aquesta anàlisi de sensibilitat podríem reduir els costos associats a les reformes a realitzar sobre uns sectors ja construïts.

D.2.4 Gestió de la xarxa dividida en clusters de proveïment

En una xarxa ja sectorizada podria ser útil situar sensors segons la importància relativa dels nodes de la xarxa (Capítol 7, Secció 7.3). Però aquesta importància també pot ser interessant com criteri per al mateix procés de sectorització, com altra variable més a tenir en compte. Altres usos respecte de la importància dels nodes pot ser la seua inclusió en plans de rehabilitació o en la gestió de les vulnerabilitats de la xarxa.

Respecte dels models predictius, un treball futur haguera d'incloure la millora del model ponderat de la demanda-patró (cf. Capítol 7 i Apèndix B); el que potser siga possible incloent, en el seu algorisme, diferents particions de temps. Quant a l'estudi de les anomalies (Capítol 7, Secció 7.2), no vam desitjar únicament detectar-les en el comportament de la demanda d'aigua, sinó que també haguérem de treballar per a ser capaces de trobar relacions causa-efecte justificades en tot l'entorn de la demanda d'aigua potable.

Part VI

References

References

Aizerman, M., Braverman, E. & Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, **25**, 821–837. 45

Aloise, D., Hansen, P. & Popat, P. (2009). NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, **75**, 245–249. 40

Alonso, C.D. (2010). *Modelo híbrido para la toma de decisiones en programas de rehabilitación de tuberías para sistemas de abastecimiento de agua: Aplicación a la ciudad de Celaya, Gto. (México)*. Ph.D. thesis, Departamento de Ingeniería Hidráulica y Medio Ambiente, Universidad Politécnica de Valencia, Valencia, Spain. 16, 24

Alvisi, S., Franchini, M. & Marinelli, A. (2007). A short-term, pattern-based model for water-demand forecasting. *Journal of Hydroinformatics*, **9**, 39–50. 178

An, A., N. Shan, C.C., Cercone, N. & Ziarko, W. (1995). Discovering rules form data for water demand prediction. In *Workshop on Machine Learning and Expert System (IJCAI'95)*, 187–202, Montreal, Canada. 110

Arbúes, F., García-Valiñas, M. & Martínez-Espiñeira, R. (2003). Estimation of residential water demand: a estate-of-the-art review. *Journal of SocioEconomics*, **32**, 81–102. 110

Augusteijn, M. & Folkert, B. (2002). Neural network classification and novelty detection. *International Journal of Remote Sensing*, **23(14)**, 2891–2902. 124

AVSA (2009). Sectorización. Avalaible online http://www.aguasdevalencia.es/ last accessed in May 2009. 23

Barbará, D., Couto, J. & Li, Y. (2002). An entropy-based algorithm for categorical clustering. In *Information and Knowledge Management (CIKM, 2002)*. 35

Basu, S., Banerjee, A. & Mooney, R. (2002). Semi-supervised clustering by seeding. In *19th International Conference on Machine Learning*, 19–26. 48

Basu, S., Banerjee, A. & Mooney, R. (2004a). Active semi-supervision for pairwise constrained clustering. In *4th SIAM International Conference on Data Mining*. 48

Basu, S., Bilenko, M. & Mooney, R. (2004b). A probabilistic framework for semi-supervised clustering. In *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 59–68. 48

Bishop, C.M. (2005). *Neural Networks for Pattern Recognition*. Oxford University Press. 171

Blanco-Figueroa, J. (2009). Proyecto de sectorización de la Zona Centro de Celaya, Guanajuato. *Aquaforum*, **52**, 20–23. 25

Bordino, I., Donato, D., Gionis, A. & Leonardi, S. (2008). Mining large networks with subgraph counting. In *IEEE International Conference on Data Mining*, Pisa, Italy. 74

Bougadis, J., Adamowski, K. & Diduch, R. (2005). Short-term municipal water demand forecasting. *Hydrological Processes*, **19**, 137–148. 4, 108, 171

Bozic, M. & Stojanovic, M. (2011). Application of SVM methods for mid-term load forecasting. *Serbian Journal of Electrical Engineering*, **8 (1)**, 73–83. 15

Breiman, L. (2001). Random forests. *Machine Learning*, **45**, 5–32. 177

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees. Statistics/Probability Series*. 177

Brin, S. & Page, L. (1997). The anatomy of a large-scale hypertextual web search engine. Available online at http://infolab.stanford.edu/backrub/google.html. 131, 132

Bryan, K. & Leise, T. (2006). The $ 25,000,000,000 eigenvector. the linear algebra behind google. *SIAM Review*, **48 (3)**, 569–582. 132

Canu, S., Grandvalet, Y. & Rakotomamonjy, A. (2003). Svm and kernel methods matlab toolbox. *Perception Systemes et Information*, http://asi.insa–rouen.fr/ arakotom/toolbox/index (Accessed on October 2008). 175

Cao, K., Feng, X. & Ma, H. (2007). Pinch multi-agent genetic algorithm for optimizing water-using networks. *Computers & Chemical Engineering*, **31 (12)**, 1565–1575. 57

Cavers, M.S. (2010). *The normalized laplacian matrix and general Randić index of graphs*. Ph.D. thesis, University of Regina, Canada. 39

Chambers, J.M., ed. (1998). *Programming with Data*. Springer, New York. 164

Chambers, J.M., ed. (2006). *How S4 methods work*. R Foundation for Statistical Computing, New York. 164

Chan, P., Schlag, M. & Zien, J. (1994). Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design*, **13 (9)**, 1088–1096. 42

Chang, C. & Lin, C. (2001). *libsvm: A Library for Support Vector Machines*. URL http://www.csie.ntu.edu.tw/ cjlin/libsvm (Accessed on October 2010). 177

Chapelle, O., Schölkopf, B. & Zien, A., eds. (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA. 5, 48, 79

Chimphlee, W., Abdullah, A., Sap, M., Srinoy, S. & Chimphlee, S. (2006). Anomaly-based intrusion detection using fuzzy rough clustering. In *2006 International Conference on Hybrid Information Technology (ICHIT'06)*, 329–334. 124

Chung, F. & Zhao, W. (2008). PageRank and random walks on graphs. In *Fete of Combinatorics and Computer Science Conference en honor de Laci Lovász*, Keszthely, Hungary. 131, 132

Covas, D. & Ramos, H. (1999). Practical methods for leakage control, detection and location in pressurised systems. In *13th International Conference on Pipeline Protection*, 135–149, Edinburg, Scotland. 4, 16, 22

Csárdi, G. & Nepusz, T. (2006). *igraph Reference Manual*. 133, 163

Dahl, C. & Hylleberg, S. (2004). Flexible regression models and relative forecast performance. *International Journal of Forecasting*, **20**, 201–217. 172

de Abreu, N. (2007). Old and new results on algebraic connectivity of graphs. *Linear Algebra and its Applications*, **423**, 53–73. 43

Delgado-Galván, X., Pérez-García, R., Izquierdo, J. & Mora-Rodríguez, J. (2010). An analytic hierarchy process for assessing externalities in water leakage management. *Mathematical and Computer Modelling*, **52 (7-8)**, 1194–1202. 52, 53

Dhillon, I., Guan, Y. & Kulis, B. (2004). A unified view of kernel k-means, spectral clustering and graph cuts. Technical report tr-04-25, University of Texas at Austin. 40

Dietterich, T. (2001). *Ensemble methods in machine learning*, vol. 1857. Springer. 81

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D. & Weingessel, A. (2009). e1071. In *Misc Functions of the Department of Statistics (e1071)*, TU Wien. 177

Drineas, P., Frieze, A., Kannnan, R., Vempala, S. & Vinay, V. (2004). Clustering large graphs via singular value decomposition. *Machine Learning*, **56**, 9–33. 75

Durbin, J. & Koopman, S. (2001). *Time series analysis by state space methods*. Oxford University Press. 112

Elkan, C. (2003). Using the triangle inequality to accelerate k-means. In *20th International Conference on Machine Learning (ICML)*, Washington, DC, USA. 34

Everitt, B.S., Landau, S., Leese, M. & Stahl (1988). *Cluster analysis*. Wiley. 35

Fantozzi, M., Calza, F. & Lambert, A. (2009). Experience and results achieved in introducing District Metered Areas (DMA) and Pressure Management Areas (PMA) at Enia utility (Italy). In *Water Loss Specialist Conference, International Water Association*, Cape Town, South Africa. 17

Fowlkes, C., Belongie, S., Chung, F. & Malik, J. (2004). Spectral grouping using the Nyström method. *IEE Transactions on Pattern Analysis and Machine Intelligence*, **26 (2)**, 214–225. 74, 76

Frahling, G. & Sohler, C. (2006). A fast k-means implementation using coresets. In *22nd Annual Symposium on Computational Geometry (SoCG)*, Sedona, AZ, USA. 34

Freund, Y. & Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Computer and System Sciences*, **55 (1)**, 119–139. 81

Freund, Y. & Schapire, R. (1999). A short introduction to boosting. *Japanese Society for Artificial Intelligence*, **14 (5)**, 771–780. 78

Friedman, J. & Stueltze, W. (1981). Projection pursuit regression. *Journal of American Statistical Association*, **76**, 817–823. 172, 173, 174

Frossyniotis, D., Likas, A. & Stafylopatis, A. (2004). A clustering method based on boosting. *Pattern Recognition Letters*, **25**, 641–654. 81, 87

Fukumizu, K., Bach, F. & Gretton, A. (2007). Statistical consistency of kernel canonical correlation analysis. *Machine Learning Research*, **8**, 361–383. 125

Fung, G. (2001). A comprehensive overview of basic clustering algorithms. 33

Gärtner, T. (2003). A survey of kernels for structured data. In *SIGKDD Explorations*, 49–58. 45

Gato, S., Jayasuriya, N. & Roberts, P. (2007). Temperature and rainfall thresholds for base urban water demand modelling. *Journal of Hydrology*, **337(3-4)**, 364–376. 110

Gianetti, L., Maturana, F. & Discenzo, F. (2005). *Agent-based control of a municipal water system*. Springer-Verlag. 57

Giannella, C., Han, J., Pei, J., Yan, X. & Yu, P. (2003). *Mining frequent patterns in data streams at multiple time granularities*. Ed. by H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.). 114

Gibert, K. & Cortés, U. (1997). Weighting quantitative and qualitative variables in clustering methods. *Mathware & Soft Computing*, **4**, 251–266. 33

Gibert, K., Rodríguez-Silva, R. & Rodríguez-Roda, I. (2010). Knowledge discovery with clustering based on rules by states: A water treatment application. *Environmental Modelling & Software*, **25**, 712–723. 33

Gilbert, P. (2006). *Brief user's guide: Dynamic System Estimation (DSE)*. R Foundation for Statistical Computing, http://cran.r-project.org (Accessed on October 2010). 112

Gretton, A., Bousquet, O., Smola, A. & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. 63–77, Springer-Verlag, Berlin, Germany. 128

Grira, N., Crucianu, M. & Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: a brief survey. 32

Grubesic, T.H., Matisziw, T.C., Murray, A.T. & Snediker, D. (2008). Comparative approaches for assessing network vulnerability. *International Regional Science Review*, 31–88. 129

Gutiérrez-Pérez, J. (2010). *Detección de eventos de contaminación en redes de abastecimiento de agua mediante el Control Estadístico de Procesos*. Master's thesis, Universidad Politécnica de Valencia, Spain. 19

Gutiérrez-Pérez, J., Herrera, M., Pérez-García, R. & Ramos, E. (2011). Application of graph-spectral methods in the vulnerability assessment of water supply networks. In *Mathematical Modelling in Engineering & Human Behaviour, 2011*. 151

Hagen, L. & Kahng, A. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Computer-Aided Design*, **11(9)**, 1074–1085. 42

Handl, J. & Knowles, J. (2006). On semi-supervised clustering via multiobjective optimization. In *8th Annual Conference on Genetic and Evolutionary Computation*. 49

Härdle, W., Liang, H. & Gao, J. (2000). *Partially Linear Models*. Physica-Verlag Heidelberg. 178

Hartigan, J. & Wong, M. (1979). A k-means clustering algorithm. *Applied Statistics*, **28**, 100–108. 33

Hasan, J., States, S. & Deininger, R. (2004). Safeguarding the security of public water supplies using early warning systems: A brief review. *Contemporary Water Research and Education*, **129**, 27–33. 19

Hastie, T. & Tibshirani, R. (1990). *Generalized Additive Models*. Chapman & Hall. 174

Hastie, T., Tibshirani, R. & Friedman, J., eds. (2001). *Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag. 32, 34, 63

Haveliwala, T. & Kamvar, S. (2003). *The second eigenvalue of the Google matrix*. Standford University Technical Report 2003-20. 132

Herrera, M., García-Díaz, J., Pérez-García, R., Martínez, J. & López, P. (2007). Interpolación con redes neuronales artificiales en series temporales intervenidas para la predicción de la demanda urbana de agua. In *NOLINEAL 2007*, Ciudad Real, Spain. 124

Herrera, M., Izquierdo, J., Montalvo, I., García-Armengol, J. & Roig, J. (2009a). Identification of surgical practice patterns using evolutionary cluster analysis. *Mathematical and Computer Modelling*, **50(5-6)**, 705–712. 36

Herrera, M., Pérez-García, R., Izquierdo, J. & Montalvo, I. (2009b). Scrutinizing changes in water demand behavior. In *Positive Systems. Lecture notes in Control and Information Sciences*, 305–313, Springer-Verlag, Valencia, Spain. 5, 18, 19, 124, 126, 130

Herrera, M., Canu, S., Karatzoglou, A., Pérez-García, R. & Izquierdo, J. (2010a). An approach to water supply clusters by semi-supervised learning. In *International Congress on International Environmental Modelling and Software, iEMSs 2010*, Ottawa, Canada. 5, 34, 73, 75, 81, 132

Herrera, M., Izquierdo, J., Pérez-García, R. & Ayala, D. (2010b). Water supply clusters by a multy-agent based approach. In *12th annual Water Distribution Systems Analysis conference, WDSA2010*, Tucson, USA. 5, 35, 37, 57, 63

Herrera, M., Izquierdo, J., Pérez-García, R. & Montalvo, I. (2010c). Water supply clusters based on a boosting semi-supervised learning methodology. In *7th International Conference on Engineering Computational Technology, ECT2010*, Valencia, Spain. 5, 37

Herrera, M., Torgo, L., Izquierdo, J. & Pérez-García, R. (2010d). Predictive models for forecasting hourly urban water demand. *Journal of Hydrology*, **387 (1-2)**, 121–130. 4, 5, 9, 18, 111, 117, 125

Herrera, M., Izquierdo, J., Pérez-García, R. & Montalvo, I. (2011). Multi-agent adaptive boosting on semi-supervised water supply clusters. *Advances in Engineering Software*, under review. 76

Ho, W. (2008). Integrated analytic hierarchy process and its applications - a literature review. *European Journal of Operational Research*, **186**, 211–228. 52

Hofmann, T., Schölkopf, B. & Smola, A. (2008). Kernel methods in Machine Learning. *Annals of Statistics*, **36 (3)**, 1171–1220. 45, 46

Hübler, C., Borgwardt, K., Kriegel, H. & Ghahramani, Z. (2008). Representative subgraph sampling using Markov Chain Monte Carlo methods. In *International Workshops on Mining and Learning with Graphs*, 322–336, Helsinki, Finland. 74

Hunaidi, O. (2005). Economic comparison of periodic acoustic surveys and DMA–based. In *Leakage 2005 Conference*, 322–336, Halifax N.S., Canada. 16

Hunaidi, O. & Brothers, K. (2007). Optimum size of District Metered Areas. In *Water Loss Specialist Conference, International Water Association*, 57–66, Bucharest, Romania. 4, 22

IWWA-Loss-Group (2005). *Best Practice Performance Indicators for Non Revenue Water and Water Loss Components: A Practical Approach*. IWA Eds. 17

IWWA-Loss-Group (2007). *District Metered Areas: Guidance Notes*. IWA Eds. 4, 16, 21, 22, 95

Izquierdo, J., López, P., Martínez, F. & Pérez-García, R. (2007). Fault detection in water supply systems using hybrid (theory and data–driven) modelling. *Mathematical and Computing Modelling*, **46**, 341–350. 123, 124

Izquierdo, J., Montalvo, I., Pérez-García, R. & Herrera, M. (2008). Sensitivity analysis to assess the relative importance of pipes in water distribution networks. *Mathematical and Computing Modelling*, **48**, 268–278. 4, 19, 130

Izquierdo, J., Herrera, M., Montalvo, I. & Pérez-García, R. (2009). Agent–based division of water distribution systems into District Metered Areas. In *4th International Conference on Software and Data Technologies, ICSOFT 2009*, Sofia, Bulgaria. 4, 57, 64, 77, 80

Izquierdo, J., Herrera, M., Montalvo, I. & Pérez-García, R. (2011). *Division of Water Supply Systems into District Metered Areas Using a Multi-agent Based Approach*. Springer-Verlag, Berlin Heidelberg. 57

Jain, A. & Ormsbee, L.E. (2002). Short-term water demand forecasting modelling techniques-conventional versus AI. *Journal AWWA*, **94**, 64–72. 108

Kamvar, S., Klein, D. & Manning, C. (2003). Spectral learning. In *17th International Joint Conference on Artificial Intelligence*, 561–566. 34, 48, 49

Karatzoglou, A. (2006). *Kernel methods software, algorithms and applications*. Ph.D. thesis, Technischen Universitat Wien, Austria. 45, 54, 88, 97, 101, 126, 164, 175

Karatzoglou, A., Meyer, D. & Hornik, K. (2006). Support vector machines in R. *Journal of Statistical Software*, **15(9)**, URL http://www.jstatsoft.org/v15/i09 (Accessed on January 2009). 126, 164, 165, 175

Kashima, H., Tsuda, K. & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. In *Intl. Conf. Machine Learning*, 321–328, Morgan Kaufmann, San Francisco, CA. 45

Kaufman, L. & Rousseeauw, P.J. (1990). *Finding groups in data: an introduction to cluster analysis*. Wiley. 33

Kolaczyk, E. (2009). *Statistical analysis of network data: methods and models*. Springer. 74

Kondor, R.I. & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. In *19th International Conference on Machine Learning*, 315–322, C. Sammut and A. Hofmann, editors. 31

Krause, P., Boyle, D. & Bäse, F. (2005). Comparison of different efficiency criteria for hydrological model assessment. *Advances in Geosciences*, **5**, 89–97. 115

Kudo, T., Maeda, E. & Matsumoto, Y. (2004). An application of boosting to graph classification. In *Advances in Neural Information Processing Systems (NIPS)*, Whistler, Canada. 74

Kulis, B., Basu, S., Dhillon, I. & Mooney, R. (2005). Semi-supervised graph clustering: a kernel approach. In *22nd International Conference on Machine Learning*, 457–464, ACM, Bonn, Germany. 32, 34, 35, 47, 49

Kulis, B., Basu, S., Dhillon, I. & Mooney, R. (2009). Semi-supervised graph clustering: a kernel approach. *Machine Learning*, **74**, 1–22. 48

Lakervi, E. & Holmes, E. (1995). *Electricity distribution network design*. Peter Peregrinus. 15

Langville, A. & Meyer, C. (2005). Deeper inside PageRank. *Internet Mathematical Journal*, **1 (3)**, 335–380. 132

Langville, A. & Meyer, C. (2006). *Google's PageRank and Beyond: The science of search engine rankings*. Princeton University Press. 131, 132

Leskovec, J. & Faloutsos, C. (2006). Sampling from large graphs. In *Knowledge Discovery and Data Mining*, Philadelphia, USA. 74, 75, 76

Leskovec, J., Kleinberg, J. & Faloutsos, C. (2005). Graphs over time: Densification laws, shrinking diameters and possible explanations. In *KDD*, 177–187, ACM Press. 77

Liaw, A. & Wiener, M. (2002). Classification and regression by random forest. *R news*, **2 (3)**, 18–22. 177

Likas, A., Vlassis, N. & Vebeek, J. (2003). The global K-means clustering algorithm. *Pattern Recognition*, **36**, 451–461. 33

Lin, H., Chiu, D., Wu, Y. & Chen, A. (2005). Mining frequent itemset from data streams with a time-sensitive sliding window. In *2005 SIAM International Conference on Data Mining*. 114

Lingireddy, S. & Ormsbee, L.E. (1973). *Neural Networks in Optimal Calibration of Water Distribution Systems*, vol. 3. Flood and N. Kartam (ASCE). 171

Loeff, N., Forsyth, D. & Ramachandran, D. (2008). ManifoldBoost: stagewise function approximation for fully-, semi- and un-supervised learning. In *ICML '08: 25th international conference on Machine learning*, 600–607, ACM, New York, NY, USA. 79

López-Ibáñez, M. (2009). *Operational Optimisation of Water Distribution Networks*. Ph.D. thesis, School of Engineering and the Built Environment, Edinburgh Napier University, UK. 16, 18

Ma, J. & Perkins, S. (2003). Time-series novelty detection using one-class support vector machines. In *International Joint Conference on Neural Networks*, vol. 3, 1741–1745. 124

Maas, C. (1987). Transportation in graphs and the admittance spectrum. *Discrete Applied Mathematics*, **16**, 31–49. 43

Maidment, D. & Miaou, S. (1986). Daily water use in nine cities. *Water Resources Research*, **22**, 845–851. 110

Maier, H.R. & Dandy, G.C. (2000). Application of artificial neural networks to forecasting of surface water quality variables: Issues, applications and challenges. *Environmental Modelling and Software*, **15**. 171

Mallapragada, P., Jin, R., Jain, A. & Liu, Y. (2009). Semiboost: Boosting for semi-supervised learning. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, **31(11)**, 2000–2014. 80, 87

Matés, J. (2001). Evolución y cambio en el abastecimiento urbano: del sistema clásico al moderno. In *VII Congreso de la Asociación de Historia Económica*, 13. 13

Matés, J. (2009). El desarrollo de las redes de agua potable: Modernización y cambio en el abastecimiento urbano. *Agenda Social*, **3-1**, 1–20. 13

Maturana, F., Kotina, R., Staron, R., Tichý, P. & Vrba, P. (2006). Agent-based water/waste water control system architecture. In *IADIS International Conference Applied Computing*. 57

Mays, L. (2004). *Water Supply Systems Security*. McGraw-Hill. 19

McGregor, A., Hall, M., Lorier, P. & Brunskill, J. (2004). *Flow Clustering Using Machine Learning Techniques*, vol. 3015/2004. Springer - Berlin. 35

Meila, M. & Shi, J. (2001). Learning segmentation by random walks. In *Neural Information Processing Systems 13*, MIT Press. 43

Mercer, J. (1909). Functions of positive and negative and their connection with the theory of integral equations. *Philos. Trans Royal Soc. London*, **209**, 415–446. 45, 127

Michaud, D. & Apostolakis, G.E. (2006). Methodology for ranking the elements of water-supply networks. *Journal of Infraestructure Systems*, **12 (4)**, 230–242. 130

Milborrow, S. (2009). earth: Multivariate adaptive regression spline models. In *derived from mda:mars by Trevor Hastie and Rob Tibshirani*. 175

Misiunas, D. (2005). *Failure monitoring and asset condition assessment in water supply systems*. Ph.D. thesis, Lund University, Sweden. 4

Moglia, M., Perez, P. & Burn, S. (2010). Modelling an urban water system on the edge of chaos. *Environmental Modelling & Software*, **25(12)**, 1528–1538. 14

Mohar, B. (1991). The Laplacian spectrum of graphs. *Graph Theory Combinatorics and Applications*, **2**, 871–898. 43

Moisen, G. & Frescino, T. (2002). Comparing five modelling techniques for predicting forest characteristics. *Ecological Modelling*, **157**, 209–225. 174

Ng, A.Y., Jordan, M.I. & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 849–856, MIT Press. 35, 43, 74

Nong, Y. & Qian, C. (2003). Computer intrusion detection through ewma for auto-correlated and uncorrelated data. *IEEE Transactions on Realibility*, **52(1)**, 75–82. 124

Oonsivilai, A. & Greyson, K.A. (2009). Power system contingency analysis using multiagent systems. *World Academy of Science, Engineering and Technology*, **60**, 355–360. 15

Orponen, P., Schaeffer, S. & Avalos-Gaytán, V. (2008). Locally computable approximations for spectral clustering and absorption times of random walks. Tech. rep., arXiv:0810.4061, arXiv.org. 40

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge University Press. 125

Pedroche, F. (2007). Métodos de cálculo del vector PageRank. *Boletín de la Sociedad Española de Matemática Aplicada*, **39**, 7–30. 132

Poggio, T. (1975). On optimal nonlinear associative recall. *Biological Cybernetics*, **19**, 201–209. 47

R-Development-Core-Team (2010). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. 54, 88, 97, 101, 133, 144, 155, 161, 172, 173, 185, 193

Rocco, M. & Zio, E. (2007). A support vector machine integrated system for the classification of operation anomalies in nuclear components and systems. *Reliability Eng. & System Safety*, **92**, 593–600. 123, 124

Rossman, L. (2000). *EPANET–User's Manual*. United States Environmental Protection Agency (EPA), Cincinnati, OH. 37, 81, 108, 125, 133, 144, 155, 185, 193

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, **20**, 53–65. 36, 85

Saaty, T. & Hu, G. (1998). Ranking by the eigenvector versus other methods in the analytic hierarchy process. *Applied Mathematical Letters*, **11 (4)**, 121–125. 51

Schaeffer, S. (2007). Graph clustering. *Computer Science Review*, **1**, 27–64. 31, 33

Schapire, R. (1990). The strength of weak learnability. *Machine Learning*, **5(2)**, 197–227. 78

Schapire, R. (2003). *The boosting approach to machine learning: An overview*. D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors - Springer. 78

Schölkopf, B. & Smola, A.J. (2002). *Learning with kernels*. MIT Press. 31, 45, 125

Shastri, Y. & Diwekar, U. (2006). Sensor placement in water networks: A stochastic programming approach. *Water Resources Planning and Management*, **132(3)**, 192–203. 19

Shawe-Taylor, J. & Cristianini, N. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press. 126, 175

Shawe-Taylor, J. & Cristianini, N. (2006). *Kernel Methods for Pattern Analysis*. Cambridge University Press. 45, 46

Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22(8)**, 888–905. 42, 73

Shoham, Y. & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press. 5, 57

Sinkkonen, J. & Kaski, S. (2002). Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, **14 (1)**, 217–239. 48

Smola, A. & Schölkopf, B. (2004). A tutorial on support vector regression. *Stat. Comput.*, **14**, 199–222. 175

Smola, A.J. & Kondor, R.I. (2003). Kernels and regularization on graph. In *16th Annual Conference on Computational Learning Theory and the 7th Kernel Workshop*. 31

Smola, A.J. & Schölkopf, B. (1998). A tutorial on support vector regression. In *NeuroCOLT Technical Report TR-98-030*, Royal Holloway College, University of London, UK. 175

Spielman, D. & Teng, S. (1996). Spectral partitioning works: Planar graphs and finite element meshes. In *37th Annual Symposium on Foundations of Computer Science*. 43

Spirtes, P. & Gylmour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, **9**, 67–72. 123, 127

Srdjevic, B. (2007). Linking analytic hierarchy process and social choice methods to support group decision-making in water management. *Decision Support Systems*, **42**, 2261–2273. 52

Storlie, C. & Helton, J. (2008a). Multiple predictor smoothing methods for sensitive analysis: description of techniques. *Reliability Engineering & System Safety*, **93**, 28–54. 172

Storlie, C. & Helton, J. (2008b). Multiple predictor smoothing methods for sensitive analysis: example results. *Reliability Engineering & System Safety*, **93**, 55–57. 172

Stumpf, M.P.H., Wiuf, C. & May, R.M. (2005). Subnets of scale-free networks are not scale-free: Sampling properties of networks. In *National Academy of Sciences of the United States of America, PNAS*, vol. 102, 4221–4224. 76

Sturm, R. & Thornton, J. (2005). Proactive leakage management using district metered areas (dma) and pressure management is it applicable in north america? In *Leakage 2005*. 16

Sun, H. (2005). Mercer theorem for RKHS on noncompact sets. *Journal of Complexity*, **21(3)**, 337–349. 127

Sun, X., Janzig, D., Schölkopf, B. & Fukumizu, K. (2007). A kernel–based causal learning algorithm. In *24th Annual International Conference on Machine Learning*, 855–862. 123, 125, 127, 128

Sycara, K. (1998). Multiagent systems. *American Association for Artificial Inteligence – AI Magazine*, **19**, 79–92. 60

Tan, P., Steinbach, M. & Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley. 85

Thiele, J.C. & Grimm, V. (2010). Netlogo meets R: Linking agent-based models with a toolbox for their analysis. *Environmental Modelling & Software*, **25(8)**, 972–974. 9, 145, 166, 185, 193

Torgo, L. (2010). *Data Mining with R: learning by case studies*. CRC Press. 114, 162

Tsuda, K. & Kudo, T. (2006). Clustering graphs by weighted substructure mining. In *International Conference on Machine Learning*, Pittsburgh, Pensilvania, USA. 74

Tzatchkov, V., Alcocer-Yamanaka, V. & Bourguett-Ortíz, V. (2006). Graph theory based algorithms for water distribution network sectorization projects. In *8th Annual Water Distribution Systems Analysis Symposium*, Cincinnati, Ohio, USA. 4

UKWIR (1999). *A Manual of DMA Practice*. UKWIR Eds., London. 4

Valdés, J. & Castelló, J. (2003). La gestión de redes por sectores y la experiencia de Barcelona. In *2nd International Conference on Efficient Use and Management of Urban Water Supply - Efficient 2003*. 21

van Dongen, S. (2000). *Graph clustering by flow simulation*. Ph.D. thesis, Universiteit Utrecht, Utrecht, The Netherlands. 40

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer. 126, 175

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley and Sons. 47, 126, 175

Vejmelka, M. (2009). Spectral graph clustering. In *Seminar z Umele Inteligence*, Prague. 44

Venables, W. & Ripley, B. (2002). *Modern Applied Statistics with S*. Springer. 172

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, **17(4)**, 395–416. 32, 42, 47

Wagner, D. & Wagner, F. (1993). Between min cut and graph bisection. In *18th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 744–750, Springer, London, UK. 42

Wagner, M. (1999). *VAR Cointegration in VARMA Models*. Institute for Advanced Studies, Vienna. 112

Walski, T.M., Gangemi, D., Kaufman, A. & Malos, W. (2001). Establishing a system submetering project. In *AWWA Annual Conference*, Washington, DC. 4

Wang, W., Gelder, P.V., Vrijling, J.K. & Ma, J. (2006). Forecasting daily streamflow using hybrid ANN models. *Journal of Hydrology*, **324**, 383–399. 172

Wei, C., Lee, Y. & Hsu, C. (2003). Empirical comparison of fast partitioning-based clustering algorithms for large data sets. *Experts Systems with Applications*, **24**, 351–363. 33

Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. In *International Conference on Computer Vision*. 43

Weyns, D. & Holvoet, T. (2005). On the role of environments in multiagent systems. *Informatica*, **29**, 405–421. 60

White, S., Robinson, J., Cordell, D., Jha, M. & Milne, G. (2003). *Occasional paper No 9*. Water Services Association of Australia. 18

Wilensky, U. (1999). Center for Connected Learning and Computer Based Modeling, Northwetern University, Evanston, IL. 54, 68, 69, 88, 97, 98, 101, 133, 144, 155, 156, 185, 193

Wooldridge, M. (2000). *Intelligent agents*. MIT Press, Cambridge, MAssachusetts, London, UK. 59, 65

Wooldridge, M. (2002). *An introduction to multiagent systems*. John Wiley and Sons, Chinchester, UK. 59, 60, 77

Wooldridge, M. & Jennings, N. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, **10 (2)**, 115–152. 58

Yan, D., Huang, L. & Jordan, M. (2009). Fast approximate spectral clustering. In *Knowledge Discovery and Data Mining*, Paris. 74, 75, 76

Yazdani, A. & Jeffrey, P. (2010). A complex network approach to robustness and vulnerability of spatially organized water distribution networks. In *12th annual Water Distribution Systems Analysis conference, WDSA2010*, Tucson, USA. 129, 130

Zealand, C.M., Burn, D.H. & Simonovic, S. (2005). Short term streamflow using artificial neural networks. *Journal of Hydrology*, **214**, 32–48. 172

Zhang, G. & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*, **160**, 501–514. 172

Zheng, L., Wang, S., Liu, Y. & Lee, C.H. (2009). Information theoretic regularization for semi-supervised boosting. In *15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, 1017–1026, ACM, New York, NY, USA. 79

Zhou, S.L., McMahon, T.A., Walton, A. & Lewis, J. (2002). Forecasting operational demand for an urban water supply zone. *Journal of Hydrology*, **259**, 189–202. 108

Zhu, X., Kandola, J., Lafferty, J. & Ghahramani, Z. (2006). *Graph kernels by spectral transforms*. MIT Press. 32, 47