UNIVERSIDAD DE DEUSTO

KNOWLEDGE DISCOVERY TECHNIQUES TO IMPROVE THE SERVICES OF INTERNET ECONOMICS

Tesis doctoral presentada por Igor Ruiz Agúndez

dentro del Programa de Doctorado en Sistemas de Información

Dirigida por Dr. Pablo García Bringas

y Dr. Yoseba Koldobika Peña Landaburu

UNIVERSIDAD DE DEUSTO

KNOWLEDGE DISCOVERY TECHNIQUES TO IMPROVE THE SERVICES OF INTERNET ECONOMICS

Tesis doctoral presentada por Igor Ruiz Agúndez

dentro del Programa de Doctorado en Sistemas de Información

Dirigida por Dr. Pablo García Bringas

y Dr. Yoseba Koldobika Peña Landaburu

El doctorando El director El director

Bilbao, Febrero de 2012

Knowledge Discovery Techniques to Improve the Services of Internet Economics

Author: Igor Ruiz Agúndez
Advisor: Pablo García Bringas
Advisor: Yoseba Koldobika Peña Landaburu

The following web-page address contains up to date information about this dissertation and related topics:
`http://paginaspersonales.deusto.es/igor.ira/`

Text printed in Bilbao
First edition, February 2012

Nire gurasoei.

Abstract

Clustering algorithms in literature follow different strategies focused on extracting a certain kind of knowledge. In this way, their results are usually disparate. Such diversity makes choosing the optimal clustering algorithm for a definite purpose very difficult, since each model is tailored to a specific goal; indeed, model comparison is one of the classic clustering challenges. Traditionally, this enterprise has been tackled in non-contrasted ways of performing the validations manually, without a clear method, or based on formal knowledge.

One of the most promising fields to which clustering algorithms may contribute is Internet Economics: a research area that studies the services that users can consume. So far, its scope has been mainly concentrated around data transport accounting (customers usually pay for the use of network resources of certain access providers) but do not pay for any other type of service. The evolution of Internet already forces us to offer new and varied kinds of applications that lie on the transport infrastructure, using it. Therefore, Internet Economics must provide with suited tools and answers to these changes. For instance, the requirements of support systems have turned stricter and more demanding, pressuring service providers to improve their service quality, customer care, marketing, management, and so forth. This scenario hinders this discipline from achieving maturity, since it keeps procedures and semantics far from being completely standardised.

Thus, we have evaluated the application of clustering algorithms on service data in order to enrich support systems in Internet Economics. Hence, in order to find the technique that allows extracting the most significant knowledge, we have to cope with challenges related

to the difficulty of comparing algorithms with different goals, inputs, and outputs: structuring the input data, obtaining the models layout, determining the quality of the results, and so on. Indeed, each problem requires its own metric to obtain the best representation of knowledge in a problem-independent methodology.

Against this background, we propose a knowledge discovery methodology that analyses the data generated by a service and enables the extraction of an optimal model that best represents each Internet service so it can be used to improve the aforementioned support systems.

This methodology applies a number of clustering algorithms to a service dataset. First, it collects and compares the results of the clustering algorithms, subsequently obtaining an optimal model that represents the service. In order to compare these models, we have defined a common base attribute, a metric over it, and a criterion that picks up the best solution according to that metric. Still, trying to design a formal, theoretical way to accomplish this task for every possible problem specification is not feasible since metric, base attribute, and criterion are all area specific. Therefore, we propose the use of problem-independent metrics that enable the comparison of results by problem-specific criteria.

We have validated this novel methodology through a real Internet Economics use-case, namely a Voice-over-Internet protocol service. Such service may range from a simple call between two users to more complex practices including multiple-user teleconference, call transferring, call-centre functionalities, and so on. The used dataset corresponds to a medium-sized corporation and the extracted knowledge has been used to support and advise the infrastructure dimensioning system. In this way, we demonstrate how clustering techniques allow extracting significant knowledge that may assist the support systems of a service of Internet Economics in its task.

Resumen

Todos los algoritmos de clustering existentes utilizan estrategias distintas para extraer un cierto tipo de conocimiento. Por ello, sus resultados suelen ser dispares. Esta diversidad hace que la elección de qué algoritmo de clustering es más adecuado para un determinado problema sea muy difícil, ya que cada modelo está enfocado a un objetivo distinto; de hecho, la comparación de modelos es uno de los desafíos tradicionales del clustering. Hasta ahora esta tarea se ha llevado a cabo de forma no contrastada, realizando las validaciones de forma manual, sin un método claro, basándose en la intuición más que en un conocimiento formal.

Una de las áreas más prometedoras en las que los algoritmos de clustering podrían incidir de forma más notable es la de las ciencias económicas de Internet: el área de estudio de los servicios que el usuario puede consumir. Hasta el momento, se ha centrado principalmente en la economización de la transferencia de datos (los usuarios pagan al proveedor de acceso por el uso de los recursos de red que utilizan) pero no en otros tipos de servicios. La evolución de Internet apremia a ofrecer nuevas y variadas aplicaciones que se despliegan sobre la infraestructura de transporte, haciendo uso de ella. Por tanto, las ciencias económicas de Internet tienen que ofrecer un conjunto de herramientas y respuestas a estos desafíos. Por ejemplo, los requisitos de los sistemas de soporte se han vuelto más estrictos y exigentes, presionando a los proveedores de servicio a mejorar la calidad, la atención al cliente, el marketing y la gestión, entre otros, de sus servicios. Este escenario dificulta que esta disciplina alcance un nivel de madurez alto, ya que sus procesos y su semántica están lejos de establecerse completamente.

En consecuencia, hemos evaluado el uso de algoritmos de clustering sobre los datos de un servicio con la intención de enriquecer los sistemas de soporte de las ciencias económicas de Internet. Nuestro objetivo ha sido encontrar la técnica que permita la extracción del conocimiento más significativo entre los posibles algoritmos de clustering. Para ello, hemos de hacer frente a los desafíos que implica comparar algoritmos diseñados con distintos objetivos y que manejan distintas entradas y salidas: estructurar los datos de entrada, obtener la distribución del modelo resultante o determinar la calidad de los resultados. De hecho, a pesar de utilizar una metodología independiente del problema, cada uno requiere su propia métrica para obtener la mejor representación del conocimiento.

En este contexto, proponemos una metodología que analiza los datos generados por un servicio y que permite la extracción de un modelo óptimo que representa cada servicio de Internet, de forma que puede usarse para mejorar los sistemas de soporte mencionados.

Esta metodología aplica algoritmos de clustering al conjunto de datos de un servicio. Primero, recoge y compara los resultados de los algoritmos de clustering, obteniendo el modelo que mejor representa al servicio. Para poder comparar estos modelos, hemos de definir un atributo base común, una métrica sobre él, y un criterio que selecciona el mejor modelo respecto a esa métrica. Con todo, tratar de diseñar formalmente un modo teórico de completar esta tarea para cada posible especificación de un problema no resulta viable ya que cada métrica, atributo base y criterio son específicos a cada área. Por lo tanto, proponemos el uso de métricas independientes al problema que permiten la comparación de resultados mediante un criterio específico en cada caso.

Hemos validado esta novedosa metodología con un caso de uso real de las ciencias económicas de Internet; en concreto, hemos utilizado un servicio de voz sobre el protocolo de Internet. Este servicio permite usos diversos, desde la realización de una simple llamada entre

dos usuarios a usos más complejos como teleconferecias multiusuario, transferencias de llamada, y funcionalidades de call-centre, entre otros. El conjunto de datos utilizado corresponde a una corporación de tamaño medio y el conocimiento extraído ha sido utilizado para mejorar el dimensionado de infraestructura del sistema. De esta forma, hemos demostrado cómo las técnicas de clustering permiten la extracción de conocimiento significativo que puede ayudar a los sistemas de soporte de un servicio de las ciencias económicas de Internet a realizar su labor.

Laburpena

Existitzen diren clustering algoritmo bakoitzak jakinduria eskuratzeko estrategia bat erabiltzen du. Horregatik, dituzten emaitzak anitzak izan ohi dira. Aniztasun horrek arazo zehatz baten aurrean zein clustering algoritmo den egokiena erabakitzea zaildu egiten du, eredu bakoitza helburu ezberdin batera bideraturik baitago; izan ere, ereduen konparaketa da clusteringaren erronka tradizionaletako bat. Orain arte, eginkizun hori egiaztatu gabe burutu izan da, balioztapenak eskuz egin dira, metodo argirik gabe, ezagutzan baino intuizioan gehiago oinarrituz.

Clustering algoritmoek eragin handiena izan dezaketen eremuetako bat Interneteko zientzia ekonomikoa izan daiteke: erabiltzaileek kontsumitu ditzaketen zerbitzuak aztertzen dituena. Gaur egunera arte, datuen transferentziaren ekonomizazioan oinarritu da gehien bat (erabiltzaileek erabiltzen dituzten sare baliabideengatik sarbide hornitzaileei ordaintzen diete), bestelako zerbitzuak alde batean utzita. Interneten eraldaketak premiazko egiten duen aplikazio berri eta anitzen eskaintza, eta horiek garraio azpiegitura baliatzen dute. Hortaz, Interneteko zientzia ekonomikoak erronka horiei aurre egiteko tresnak eta erantzunak eskaini behar ditu. Adibidez, euskarri zerbitzuen betebeharrak zorrotzagoak eta hertsiak bilakatu dira, horregatik zerbitzuen hornitzaileak euren zerbitzuen kalitatea, bezeroaren arreta, marketina, kudeaketa, etab. hobetzera beharturik daude. Diziplina honen prozesuak eta semantika guztiz finkatzetik urrun daudenez, heldutasun altua lortzea zaila da.

Ondorioz, euskarri zerbitzuak hobetzeko asmoarekin zerbitzu baten datuen gainean clustering algoritmoen erabilera ebaluatu dugu. Gure helburua aplika daitezkeen clustering algoritmoen artean jakinduria

adierazgarriena lortu dezakeen teknika identifikatzea izan da. Horretarako, helburu ezberdinak eta sarrera eta irteera ezberdinak kudeatzen dituzten algoritmoen konparaketari aurre egin behar diogu: sarrera datuen egituraketa, emaitzaren ereduaren banaketa, emaitzen kalitatea zein den ebaztea, etab. Berez, nahiz eta arazoarekiko mendekotasunik ez duen metodologia bat erabili, arazo bakoitzak berariazko metrika baten beharra du, jakinduria eredurik onena eskuratzeko.

Erronka horien aurrean, zerbitzu batek sorturiko datuak analizatzen dituen metodologia bat proposatzen dugu, zerbitzu bakoitzaren eredu hoberena eskura dezakeena eta aurretik aipaturiko euskarri zerbitzuak hobetzeko gaitasuna izan dezakeena.

Metodologia honek zerbitzu baten datu multzo bati clustering algoritmoak aplikatzen dizkio. Lehenengo, clustering algoritmo horien emaitzak batu eta konparatu egiten dira, zerbitzua hoberen adierazten duen eredua eskuratzeko. Eredu horiek konparatu ahal izateko, oinarrizko ezaugarri komun bat definitu behar dugu, horren gainean metrika bat, eta eredu adierazgarriena aukeratzeko metrika horri buruzko irizpide bat. Horrekin guztiarekin, arazo orok izan ditzakeen berezitasunak asetzen dituen metodo teoriko baten diseinua ez da bideragarria, metrika, ezaugarri komun eta irizpide bakoitza arlo bakoitzaren araberakoa baita. Horregatik, arazoarekiko mendekotasunik ez duten metriken erabilpena proposatzen dugu, horrela irizpide baten bidez emaitzen konparaketa ahalbidetzen dugu.

Metodologia berritzaile hau Interneteko zientzia ekonomikoaren benetako zerbitzu batekin egiaztatu dugu. Zehazki, Interneteko protokoloaren gaineko ahots bidezko zerbitzua erabili dugu. Zerbitzu horrek erabilpen anitzak ditu: bi erabiltzaileren arteko deia, erabiltzaile askoren arteko telekonferentzia, deien transferentzia, call-center funtzioak, etab. Erabilitako datu multzoa tamaina erdiko korporazio bati dagokio eta sistemaren azpiegitura dimentsioa hobetzeko erauzitako jakintza erabili da. Ondorioz, clustering teknikek euskarri zerbitzuak hobetzeko erabili daitekeen jakinduria eskuratzeko erabil daitezkeela

frogatu dugu, Interneten zientzia ekonomikoaren zerbitzuen euskarri zerbitzuak hobetzeko baliagarria dena.

Acknowledgements

This work belongs to all people that have been with me over these years. Thanks to everyone.

I would like to start mentioning my advisors: Dr. Pablo García Bringas, who gave me the amazing opportunity of conducting this research, and always was full of comprehension and shown a possibilistic view; Dr. Yoseba Koldobika Peña Landaburu, was also enthusiastic, full of positive energy even in the most difficult moments. Both of you have helped me discover this wonderful scientific world and fed my vocation.

I am also grateful to María Jose Gil for her useful orientations and encouragement that she provided me with, especially at the early beginning of this dissertation, when everything seemed so diffuse.

I also want to thank all my colleges at DeustoTech Computing, working and sharing experiences with you has been an incredible experience.

Moreover, I am indebted with many of my colleagues of the Deusto Institute of Technology and the University of Deusto. Each one of you has contributed somehow in this work. Miguel Castiella deserves also a mention, since without his kind help the validation of the methodology would have been totally different.

The essential help of Izaskun has also made this research possible. She provided me with advice, edition supervision, patience and love.

And, finally, I would like to thank my parents, my father Guzmán and my mother Ana for their support, education, and love. This work is dedicated to you.

Eskerrik asko,

Igor Ruiz-Agundez

February 2012

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence

AIC Akaike Information Criterion

AVEDEV Average Deviation

BIC Bayesian Information Criterion

CDR Call/(Charging) Detail Record

CV Cross Validation

DNA DeoxyriboNucleic Acid

ID Identification

IP Internet Protocol

MEDIAN Median

MLE Maximum Likelihood Estimate

PBX Private Branch Exchange

PSTN Public Switched Telephone Network

QoS Quality of Service

SRM Structural Risk Minimization

STDEV Standard Deviation

VoIP Voice over Internet Protocol

Weka Waikato Environment for Knowledge Analysis

XML Extensible Markup Language

The beginning is the most important part of the work.

Plato

CHAPTER

1

Introduction

Internet has been evolving and changing continuously since its birth-date. Many technologies and services have been born and passed away these last years and it is not yet clear which facts have influenced the survival of a few.

What is clear is that every single service back-ends with its respective support system. Among others, these support systems enable the delivery, deployment, coordination, control, and economisation of Internet services.

We consider that the orchestration of these systems is the killer factor of the success for the services in the complex world of Internet Economics.

Each time a service is used, big amounts of data are generated, most of times underutilised. We believe that these data could be used to improve the support systems, and hence the services themselves.

Moreover, based on this model, we may access service-specific knowledge that might potentially lead to an improvement in the service support systems.

Against this background, in this chapter, we introduce the motivation of this research and describe the current situation in Internet Economics. With this domain in place, the chapter continues to account the hypothesis and goals of this work, then details the structure of this dissertation.

1.1 Motivation

Internet has grown exponentially all boundaries over the last years, evolving in many directions. Its initial conception was completely different from what is now, and will for sure differ from what will be in the future. It is used in a wide range of areas such as academic, military, economic, and social. But, most of all, Internet is now mainly a business platform and has become a central part of social life [HNF+09].

The number and quality of the applications and services on the Internet has risen exponentially. From email and file transfer to an enormous variety of e-commerce systems, online news and entertainment programs [BEH09].

This change has also brought new stakeholders into the scene: the corporations trying to profit (i.e. generate value) from the Internet by providing new services.

Thus, Internet Economics has emerged as a research area, focusing on resource allocation, pricing schemes, responsive pricing, requirements of Internet services, and policies [MB95]. The requirements of each novel actor may be different and even opposed; therefore, there is a need to find a fair mechanism that satisfies everyone. Traditionally, there has been an economic conflict between the participants: operators, service providers and users.

In this way, Internet Economics aims at designing a scenario in which every actor of the economic process profits from it [HHH+09]. Therefore, new business models and pricing schemes must be designed for the sake of operators, providers, users, and other stakeholders. Moreover, legal and regulative issues such as network neutrality, privacy, and digital rights have to be addressed [TDG+09].

The scientific, academic, and industrial relevance of Internet Economics, has caused it to be in the development agenda of many research groups and consortia, with a hectic activity in terms of projects and publications.

Thereupon, the economisation of services is one of the most important tasks to be achieved by operators, and this is a complex process involving different actors with sometimes opposite interests. In addition, each service shows its own characteristics and requirements, making it harder, if possible, to integrate them in a uniform economic process.

Thus, Internet Economics integrates diverse aspects of other disciplines such as Engineering and Economics because it requires defining usage policies, sizing services, making use of statistics, market researching, obtaining investment returns or any other aspect of the business logic.

Given the complexity and the range of concepts of Internet Economics, this research focuses on support systems, the back-end of any service, that enables and assists it. They help at many different levels (management, operations, and planning).

These systems have traditionally maintained an inventory of the network, provided services, administrated the network elements, and controlled possible faults. Nevertheless, new service requirements have forced them to include other aspects such as assisting customers, processing bills or collecting payments.

Support systems generate data (e.g. documents, usage records, logs) that could potentially improve and help making intelligent decisions about the services. For instance, analysing a supermarket selling records we can find out that placing beers and diapers near by, sales increase [CFK03]. Nevertheless, these data need to be transformed into knowledge. It is important to remark that data are a set of symbols that do not provide any answer without an interpretation. On the other hand, knowledge answers to more complex questions (i.e. who, what, where, when, and how) about the problem that generates the data [Non94].

At this point is where knowledge discovery comes into action. It is a process that automatically looks for valuable patterns that can be considered knowledge from any type of data [FPSM92]. These data may be structured (e.g. relational databases, *Extensible Markup Language (XML)* schemas) or unstructured (e.g. documents, images).

Knowledge discovery is a broad scientific domain with multiple branches that can be organised in many ways. Here we focus mainly on data mining: like other knowledge discovery areas, it generates abstractions of the data that represent hidden knowledge. Data mining is considered, in its application to real use-cases, an essential part of marketing, surveillance, fraud detection, new discoveries, and many other areas. For instance, financial corporations have been deciding whether or not to approve loans and credit cards with these techniques.

Data mining is still a vast area that combines methods of Statistics and *Artificial Intelligence (AI)* with data management. We will concentrate on the area of AI that also overlaps with wider concepts of study, especially in the area of intelligent agents [JR99].

AI includes many other sub-areas such as deduction, reasoning, problem solving, knowledge representation, planning, learning, natural language processing, motion and manipulation, perception, social intelligence, creativity, or general intelligence.

From this set, we focus on learning or, more accurately, on machine learning, core aspect of AI since the early beginning [Tur50]. Machine learning addresses the creation of algorithms that enable computers to extract behaviours from data. The major goal of machine learning is to automatically recognise complex patterns and make intelligent decisions based on these data, that is, to perform a knowledge discovery.

Many different approaches exist in machine learning (e.g. decision tree learning, association rule learning, artificial neural networks, genetic programming, inductive logic programming, support vector machines, clustering, Bayesian networks, reinforcement learning, or representation learning) and can be classified in many different ways:

- **Unsupervised learning:** includes finding patterns in a stream of input.
- **Supervised learning:** includes classification and numerical regression.
- **Classification:** determines the category of a data instance with respect to learned models.
- **Regression:** discovers a function that models the input data.

Finally, from this list of *modus operandi*, and due to the characteristics of this dissertation, we hone in clustering. Cluster analysis, or clustering, consists of dividing a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense. It is used in many areas, such as pattern recognition, image analysis, information retrieval, and bioinformatics.

In summary, this dissertation pursues applying, for the very first time to our knowledge, Internet Economics and knowledge discovery to improve service support systems. We will propose and validate a methodology that combines both areas and applies it to a real world use-case.

1.2 Hypothesis and goals

The work presented here addresses two very important research areas: Knowledge Discovery and Internet Economics. The previous section introduced the challenges faced by both the scientific and the industrial community. Therefore, based on these challenges, we enunciate the main hypothesis of our research as follows:

> *It is possible to develop a method that extracts knowledge from Internet Economics service data. This knowledge can be represented as models that help us improve the support systems of these services.*

In order to validate the hypothesis, we must design a knowledge extraction model for the services of Internet Economics and validate it experimentally. *Ergo*, the general goal of our research is:

General goal 1 *The generation of an optimal model that represents the behaviour of a service in order to improve its support system(s).*

Further, three specific goals arise from the general goal, as described below:

Specific goal 1.1 *The definition of a standard research vocabulary in the area of Internet Economics.*

Specific goal 1.2 *The design of a methodology that obtains an optimal model representation of the usage records generated in that service.*

Specific goal 1.3 *The design of uses or applications that put in practice the obtained optimal models and improve support systems of that service.*

The first specific objective identifies the pillars of Internet Economics by establishing the working framework of this research. From all the areas of this framework, this research focuses on support systems since they are the key component of any service. The second objective copes with finding optimal models that represent the behaviour of the usage records of a given service. The third and final objective focuses on the applicability of these models to different areas of support services, aiming at improving them.

The importance of finding data profiles or models in the usage of services is determinant to identify behaviour patterns and trends. They provide new knowledge to conduct actions that will improve the use of the analysed service.

These specific goals will be accomplished, and hence the general goal and hypothesis, by means of a number of operational goals:

Operational goal 1 *To establish the state of the art in Internet Economics.*

Operational goal 2 *To establish a taxonomy to identify the concepts of the services of the Internet Economics.*

Operational goal 3 *To design and implement a data acquisition process that allows experimentation with any set of service data.*

Operational goal 4 *To design and implement a procedure for the analysis of the results of clustering algorithms.*

Operational goal 5 *To apply clustering algorithms to obtain model results.*

Operational goal 6 *To design and implement a set of metrics to measure the results of clustering algorithms.*

Operational goal 7 *To design and implement a set of criteria to select the best model, given the requirements of a given problem.*

Operational goal 8 *To test and validate the proposed methodology.*

Operational goal 9 *To design and implement experiments by using optimal models to improve support systems.*

1.3 Dissertation outline

At this point, we would like to highlight that the words *dissertation* and *thesis* had different semantics in English and were not interchangeable. When, at ancient universities, the lecturer had completed his lecture, there would traditionally follow a disputation, during which students could take up certain points and argue them. The thesis was the position that each one took during this disputation and the dissertation was the line of reasoning with which each one buttressed. In honour of the tradition, we would like to keep this ancient distinction and we are going to use the specific distinction of these words.

Having said that, the remainder of this dissertation is structured as follows:

- Chapter 2 discusses the state of the art by providing the required theoretical background. It presents the different aspects of Internet Economics and details the features of clustering algorithms.
- Chapter 3 presents the overall methodology of the dissertation. It outlines the planned steps in order to accomplish the hypothesis and goals. It introduces the proposed optimal clustering model selection methodology.
- Chapter 4 illustrates the empirical experiments and their results, which aim at validating the proposed methodology. This validation is conducted through a *Voice over Internet Protocol (VoIP)* use-case based on a real world scenario.
- Chapter 5 enumerates the contributions of the dissertation and lists the most relevant publications. It also summarises and discuses the contributions of this dissertation, and outlines the avenues of future work in the confluence of Internet Economics and the improvement of support systems by using clustering algorithms.

Personally, I think it does help, that it makes a beneficial difference, but the scientific literature on the subject is very messy.

Jeanne Petrek

CHAPTER

2

State of the Art

This work studies two disciplines that have not been brought together before. On one hand, the research area of Internet Economics provides the pillars on which to support this study. It defines the roots to specify the fundamental hypothesis, and outlines both general and specific objectives.

On the other hand, the need of improving Internet decision making and support systems leads to applying knowledge discovery techniques. Through they use, we may find ways to boost the services of Internet.

Knowledge discovery is a process that automatically searches large volumes of data for patterns that can be considered knowledge about it. Specifically, since we wish to study the behaviour of Internet services, we have addressed clustering algorithms to extract this knowledge.

In this way, we will introduce the concept of Internet Economics by defining the related work and the underlying concepts in this area by structuring them into a taxonomy. Furthermore, we will provide an overview of the most relevant clustering analysis techniques.

2.1 Internet Economics

Internet Economics deals with the services that users can consume, attempts to understand the Internet as an economic system [MB95]. These services are managed, deployed and exploited, trying to make it as efficient as possible. More accurately, it is a research area that overlaps concepts and procedures of Economics, Engineering, and Policy [MB97], creating new semantics and ideas from them.

This area provides an improvement in interoperability, revenue management, pricing policies, custom care, content deployment, and so forth.

In order to integrate all the concepts related to Internet Economics present in the literature, we will introduce a taxonomy describing the economic process. We will detail all the functions involved in it, paying especial attention to the relationships between them. Presenting this information in a taxonomy helps the learning, training and assessing processes in this area. Moreover, since it defines a common vocabulary, it is also useful for the definition of the requirements among different actors.

2.1.1 Related work on Internet Economics

The economic aspects of service usage is one of the main tasks of service providers in their operation and management processes, since it provides them with the necessary information for all the related functions. Although economic requirements are studied in many books and articles in deep detail, these works rarely define a clear taxonomy of the full Internet Economics process.

The Internet as an economic system is a scenario with multiple system stakeholders that interoperate. Each one of them will have a goal, that can be the same or opposite to other actors. These players have different roles that can be characterised as follows [SFPW98b]:

- Customer: is the user of the service. Can be an individual or an organisation.
- Provider: is the organisation that creates and offers the service.
- Administration: regulates and stipulates how the exchanges between the stakeholders take place.

- Carrier: it offers the required infrastructure to run the services.
- Service access utility: the software or hardware required to use the service.
- Trust center: the point where all the actors can trust in order to ensure the correct functionality of the economic exchanges.
- Financial clearing center: provide financial services if needed by the services.

Some authors address the terms accounting, pricing, charging, or billing to represent the complete process of detecting the specific usage of a service [KP02].

The concept of pricing schemes emerges at this point. They provide the price of the users' usages and are represented as a formula that expresses the pricing function. This formula consists of the pricing variables (consumption measure metrics of the session records) and several pricing coefficients [ZZC02] that can be organised in may ways:

- Time-based: pricing based on how long a service is used.
- Volume-based: pricing based on the volume of a metric (e.g. downloaded bytes).
- QoS-based: pricing depends on the hired quality of service.
- Flat-rate: a fixed tariff for a specified amount of time.
- Paris-Metro pricing: used for shared resources. Resources are split by the amount of users per split.
- Priority pricing: services are labelled and priced according to their priority.
- Smart market pricing: services are priced in an auction.
- Edge pricing: calculation is done based on the distance between the service and the user.
- Responsive pricing: charging is activated only on service congestion.
- Effective bandwidth pricing: charging is based on an expected usage function.

- Proportional fairness pricing: it is according to the user's willingness to pay and service optimization costs.
- Cumulus pricing: based on flat pricing and dynamically priced by using a credit point system.
- Session-oriented: based on the use given to the session.
- One-off charge per service: one charge per service session.
- Usage-based: based on the general use of the service for a period of time, e.g. a moth.
- Content-based: based on the accessed content.
- Location-based: based on the access point of the user.
- Service type: based on the usage of the service. Differentiation on time-of-day: pricing based on the hour when the service is used.

In addition, there also exist the following related concepts:

- Free of charge: no charge is applied for the services.
- Periodical fess: payment of time to time quantities for the use of a service.
- Discounts: reduction in the usual price
- Pre-paid: the payment of the service is done in advance.
- Post-paid: the payment of the service is done after the use.
- Online: the accounting performed while the user makes use of a service.
- Offline: the accounting process is done after a service is used.
- Static pricing: the pricing function does not change.
- Dynamic pricing: the pricing function changes on the fly, being adapted to the usage of the users.

Furthermore, many of these schemes can be combined to create new ones that inherit the properties of the originals. The possibilities are endless as new pricing schemes can be created to model more complex business models.

Moreover, depending on the application area of each author, they employ different terminology and semantics as shown for instance in accounting on packet-switched networks [KSSW00], micro-payments [Pá05], grid services [MWS06], mobile networks [KKA02], VoIP services [Der06] or Wi-Fi connections [DLB09].

The Internet Economics process needs to be disambiguated by employing a precise terminology and splitting clearly all the functions involved. We understand the Internet Economics process as a meta-concept that includes all the aforementioned functions.

Other research works have tackled the standardisation of the Internet Economics process on the Internet [MHR91] [Pro08] but none of them has considered all the previous work. Nevertheless, to our knowledge, there is none that has performed a full taxonomy of the Internet Economics requirements making the learning, teaching and assessing process much easier [ACM+00].

2.1.2 Accounting requirements

The Internet Economics implies a large number of challenges in the realms of technology, business, society and governance have to be overcome if the future development of the Internet [Uni08]. These changes imply that multiple service providers may be involved during the same session (e.g. a phone call, a data request or any other service) with users roaming through several networks and service providers. This fact implies that, consequently, a significant number of different charging schemes may also be involved simultaneously within a single session (e.g. charging based on data, time or content) [CH07].

In this way, these changes demand new working paradigms and new working requirements. Finding an economic model for each services is a real challenge due to the operator's structure and processes, being a key factor for their correct deployment. Despite the fast evolution of telecommunication technologies, many service providers think that past accounting systems are enough for the emerging

networks [Tak06] [BMAPO05]. Their use implies adapting them with the subsequent possibility of revenue reductions, system instability and slow new service introduction.

Within traditional models, the billing system operates in batch mode. It stores the so-called *Usage Detail Record (UDR)*-s and in this period there is a time window in which the provider has no control over the user behaviour. There should not be this time window since it may leave records of services unprocessed. This is why real-time billing is preferred: it allows to process information every time it is generated, enabling immediate business support systems operations. The real-time efficiency will depend on the available processing time and the available processing cost.

Further, traditional models present a centralised architecture: all UDRs are processed by only one rating engine and one invoicing engine that limit the scalability of the system. On the contrary, Internet Economics requires high scalability levels in order to support a large number of customers and inter-carrier settlement activities. Nevertheless, an standardised reporting protocol is needed in order to ensure distributed metering and cross-platform service providing [AKK03]. To present day, there are some attempts to standardise these interfaces [Pro08] [MHR91].

Moreover, classical economisation systems are designed service-specific. This is, traditional systems lack of flexibility and are not capable of accommodating content-based pricing that will replace current flat-rate charging models. Due to this flaw, in case a new service is introduced, it will also require a new accounting system. The services will have heterogeneous data from a larger number of systems. The end user demands a convergent billing that provides a unified view of the services that it consumes.

2.1.3 Taxonomy

The first taxonomy of History was accomplished by Carolus Linnaeus, to whom the following quote is attributed:

"The first step in wisdom is to know the things themselves; this notion consists in having a true idea of the objects; objects are distinguished and known by

classifying them methodically and giving them appropriate names. Therefore, classification and name-giving will be the foundation of our science."

Carolus Linnæus, Systema Naturæ (1735)

Originally, taxonomies were used only to classify organisms. Nowadays, they are used in other areas as. There can be economic, biological, and even military taxonomies, each one specifying its domain area. Taxonomies have been proved helpful organising content and easing the connections between people and the information they need [WB08]. Furthermore, taxonomies can show a tree, network or linear structure, all of them are useful for learning, teaching, and assessing [ACM+00], being a central part of most conceptual models since they enable relating disperse elements [WG01]. The taxonomies have been proved to help to organise content and make connections between people and the information they need [WF05]. In this way, they are especially helpful because they present limited views of a certain model for human interpretation, and play an essential role in reuse and integration tasks.

Prior to detail the taxonomy structure, it is worth mentioning that there have been two main Internet Economics branches: telecommunications and the Internet [KKA+04]. In telecommunications, an Internet Economics process consists of apportioning the charges between the home environment, the serving network, and the user (i.e. defining who is paying what). Regarding to the Internet, this process is defined as the set of functions that manages data detailing the use of the resources.

Traditionally, Internet Economics has been limited to data transport accounting: customers used to pay for the use of the network resources of certain access providers [Pá05] but not for any other type of service. Nevertheless, and despite their differences, these worlds are converging, with the subsequent refinement of the Internet Economics process.

So far, in this emerging new paradigm, accounting is the process in charge of collecting the resource usage data for capacity and trend analysis, cost allocation, auditing, and billing [AAH00]. The evolution of the Internet, however, renders

this definition insufficient and exiguous. It needs to aim at a broader concept, considering all possible functions and related concepts [PvBSP01], and meeting the requirements of Internet Economics.

2.1.3.1 An integrated vision of the Internet Economics process

Under these premises, we present a taxonomy of the full Internet Economics process, from the resource usage up to the financial clearing. This taxonomy has been defined following a developing method that ensures its quality and the maintainability of the knowledge to potential changes [WB08]. Further, we propose a unified and controlled vocabulary that can be employed in any operation related to Internet Economics.

The taxonomy was obtained as follows: we determined the requirements and identified the concepts involved in the area of Internet Economics. Then, we produced the first draft of the taxonomy. It was reviewed by users and experts in the field, which provided us with feedback for the subsequent refining process to obtain the final version. Finally, in order to culminate the model, we launched a maintaining process.

We realised that the terminology on the economic process has always been diffuse, sometimes involving contradictory semantics. The origin of this problem is not new and it dates back to the evolution of the Economy discipline through the years and the influence of the different application areas in which it has been applied. For instance, the economic processes in traditional telecommunications, energy systems or financial world differ.

Since the terminology of the Internet Economics process evolves without an established standard [KP02], we have studied the work carried out by other authors [SFPW98b] [KP02] [MHR91] [Pro08]. Each contribution depicts a different vision of the Internet Economics process, creating a set of mixed concepts. After analysing the most relevant Internet Economics process paradigms, however, we found out that they do share a bunch of common characteristics that can be re-factored in order to have the integrated Internet Economics process shown in Figure 2.1.

The process starts when a resource is used, registered by the metering function in its records. Afterwards, the mediation function intercedes by generating the

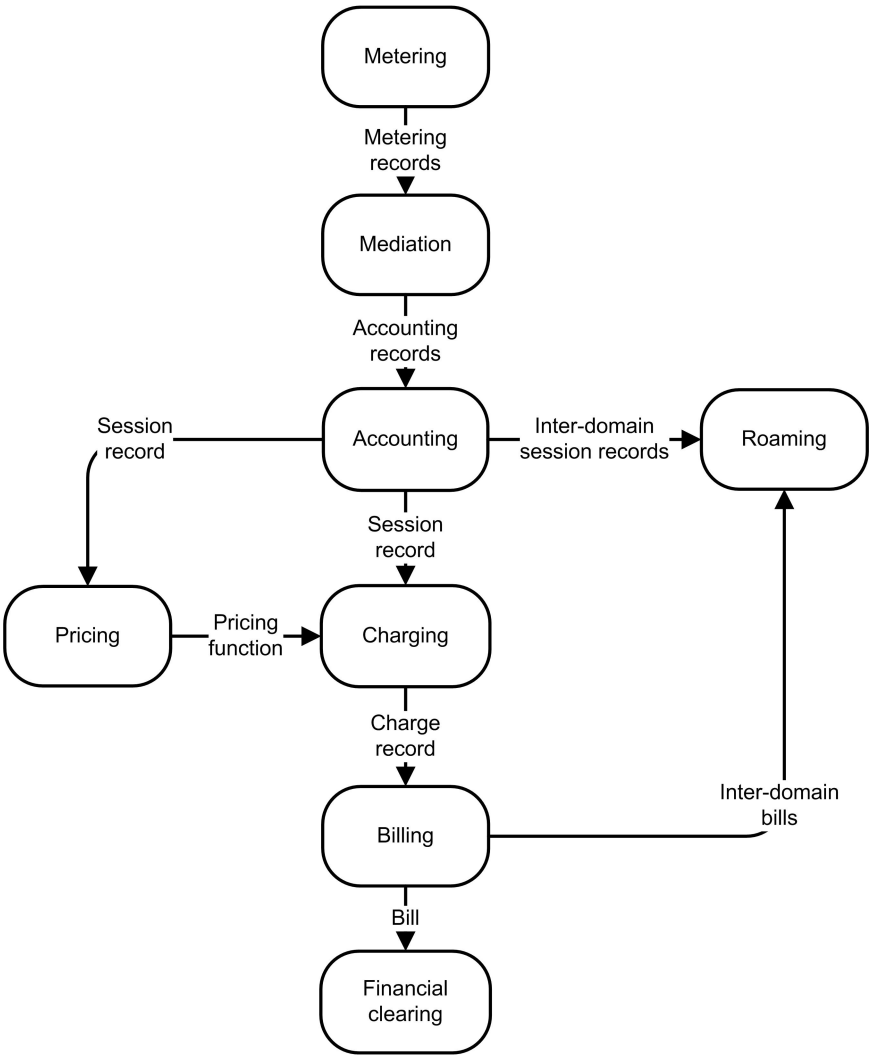

Figure 2.1: An integrated vision of the Internet Economics process.

accounting records for the accounting function. It issues session records, sent both to the pricing and charging functions. The first one generates a formula to define how to price session records, which is used by the second function. The flow continues with the charging step, which generates charge records for the billing function. There, the financial clearing function receives the final bill.

Throughout the whole process, there may occur inter-domain exchanges between organisations. These interchanges may happen at the accounting and billing steps, enabling roaming capabilities and inter-organisation collaboration.

2.1.3.2 **Metering**

This function collects the information flow regarding the resource usage of a certain service by a consumer and its behaviour. This measurement data consist of service usage metrics provided by the monitoring function. Figure 2.2 presents the metering function.

This information is technical, expressed therefore in measurable quantities of consumer resources [Pá05]. Examples of these measurable quantities are for instances the amount of data sent and received within an Internet connection (in bytes), the length of a telephone call (in seconds) or the amount of energy consumed (in kW).

This information is the starting point of the Internet Economics process and will be used along the entire process. It determines the particular usage of resources within end-systems or intermediate systems on a technical level, including *Quality of Service (QoS)*, management, and networking parameters [SGRF01].

This function is normally implemented in a meter reader at a certain infrastructure point known as the metering point [BMR99] where the resource usage data are buffered as long as the memory is able to.

Further, a meter reader can be classified as a consumer-side meter, in which the meter reader is placed at the consumer's organisation, or as a provider-side meter, in which the meter reader is hosted by the provider. In certain cases, meter readers are between the consumer and the provider, allocated in a third part, or in a neutral infrastructure that both consumer and provider trust.

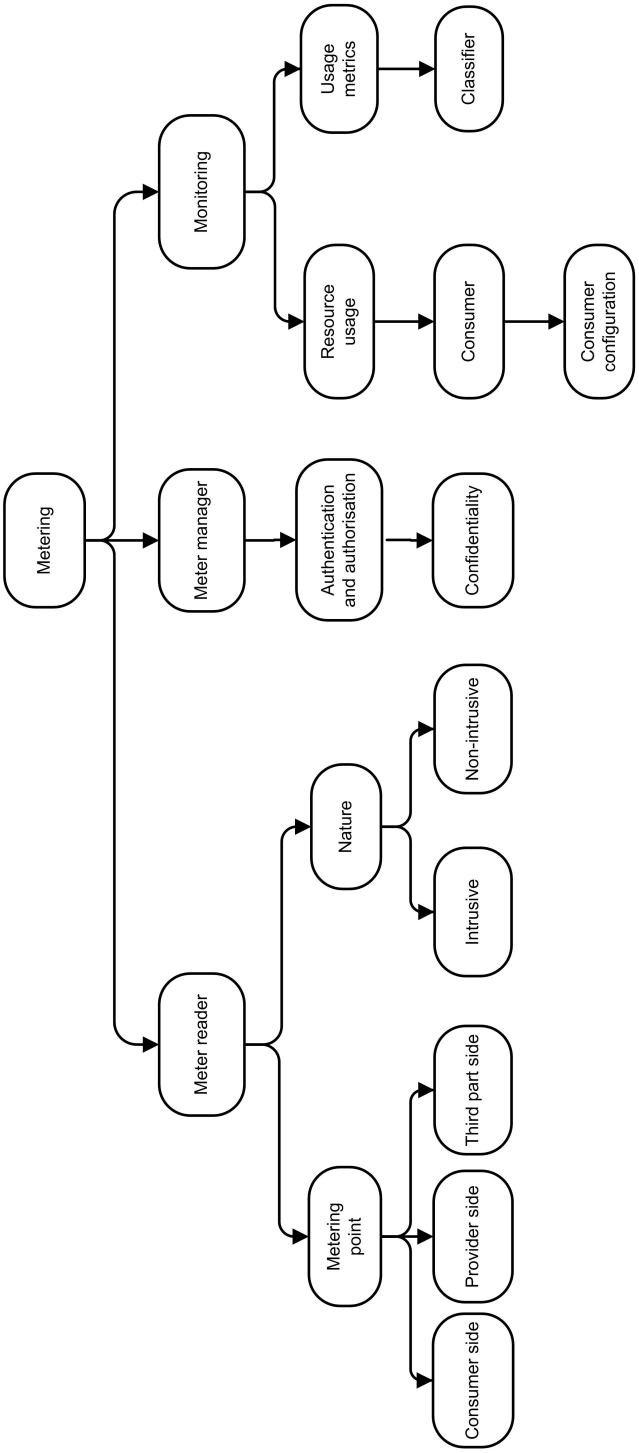

Figure 2.2: Metering function overview.

Meter readers can also be divided according to their nature into intrusive (in case there is an interface with the resource) or non-intrusive (in case there is no interface with the resource).

Additionally, there is a number of parameters that must be defined in order to achieve a correct resource usage measurement, depending on the resource itself. The general working procedure, however, persists among the different resources. The metering function is managed by a meter manager. It is responsible for the authentication and authorisation of the metering records, ensuring the confidentiality of the data.

2.1.3.2.1 Monitoring

This function collects the information of a resource usage as raw data and provides the metering function with usage metrics while being a part of it, as represented in Figure 2.2. These metrics reflect the use of a resource by a consumer (human, machine or other service) in measurable quantities, defining the rules that the monitoring device applies in a classifier in order to filter the usage data [BBC+98].

Further, consumer configuration refers to the function that configures a service for its use. Normally, this configuration is set after the user is authenticated in the service provider's infrastructure [ZZC02].

The monitoring function can be conditioned by the consumer's configuration. Therefore, different consumers may monitor different usage metrics (also known as data-points).

2.1.3.3 Mediation

Mediation is intended to filter, collect, generate, aggregate, correlate, and reconcile raw technical data by transforming these metering records into a data format that can be used to store and process [SGRF01] [Pro09]. The metering records generated by this function are usually stored in a homogeneous data format (known as accounting records).

In this way, data processing is easier and the different functions of the Internet Economics process require less mash-ups and conversions, resulting in a better performance [SFPW98a]. Figure 2.3 provides a scheme of the mediation function.

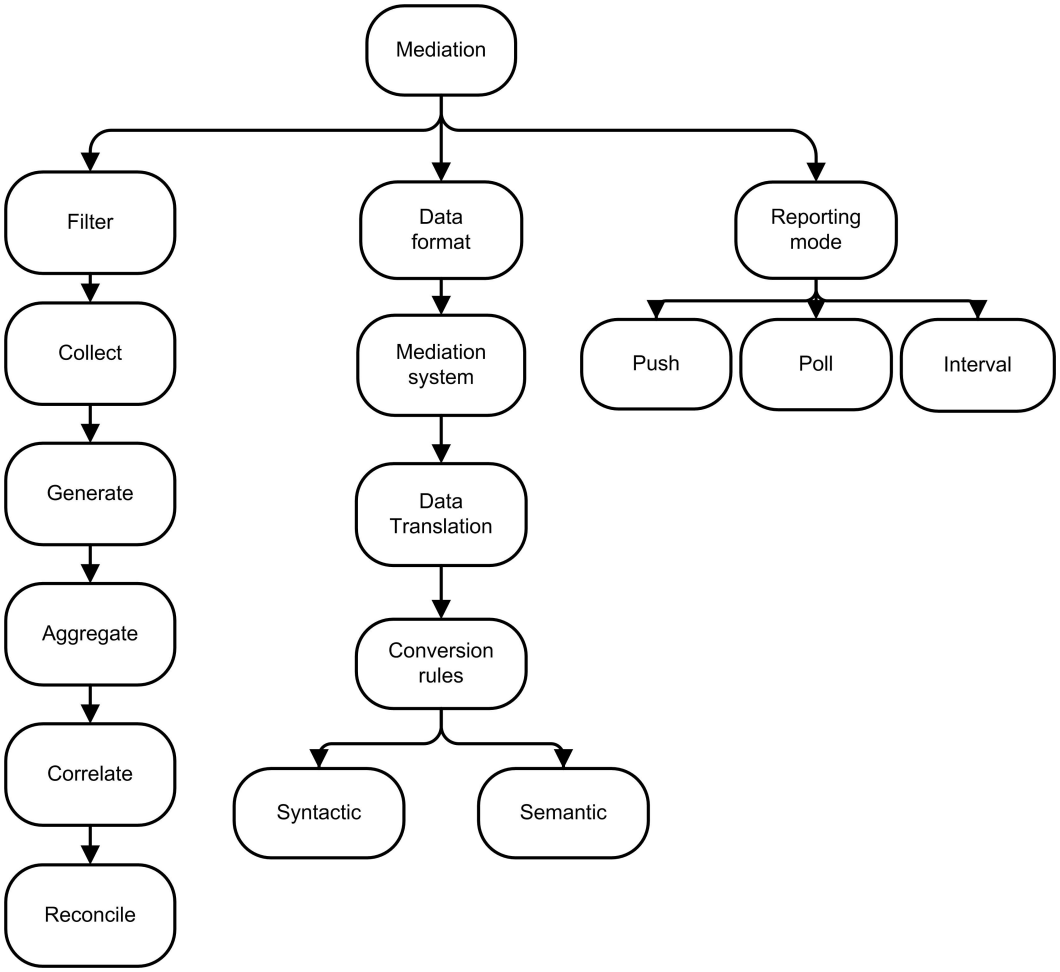

Figure 2.3: Mediation function overview.

In case different data formats are used, it is necessary to translate that data in order to have all the information in a homogeneous format as soon as possible. The conversion rules, both syntactic and semantic, required to guarantee the integrity of the transformed data are known as mediation systems [ZRM03], and its use is very common in the telecommunications world.

Furthermore, mediation reports to the accounting function in three different ways: push mode, poll mode or interval mode [MHR91] [Pro09]. In the push mode, the mediation function submits the accounting records as soon as it receives them. In the poll mode, the accounting function has to ask the mediation function for the accounting records. Finally, in the interval mode the mediation function reports each certain interval to the accounting function.

2.1.3.4 **Accounting**

Other taxonomies [AAH00] include resource usage measurement, rating, charging, billing, and invoicing in the accounting function. Nevertheless, we decided to split these functions in order to have a more representative organisation. Figure 2.4 outlines the accounting function.

Accounting is the process of filtering, collecting and aggregating the information regarding the resource usage by a certain consumer. This process will generate session records whose format will depend on the service infrastructure and the service provider [SFPW98a]. The session records represent the resource usage over a session.

Additionally, accounting gateways may create session records by processing interim accounting events or accounting events from several devices serving to the same user [AAH00].

Anyway, the data are expressed in metered resource consumption (e.g. for applications, calls, or any type of connections [SGRF01]) representing the technical specifications of the service. The representation may differ depending on the service provided. It includes meta-data such as the supervision of the data, gathering from the mediation function, the collection and the storage of these data [KKA+04]. Accounting policies define how these functions behave and are specified by a set of generation rules [ZZC02].

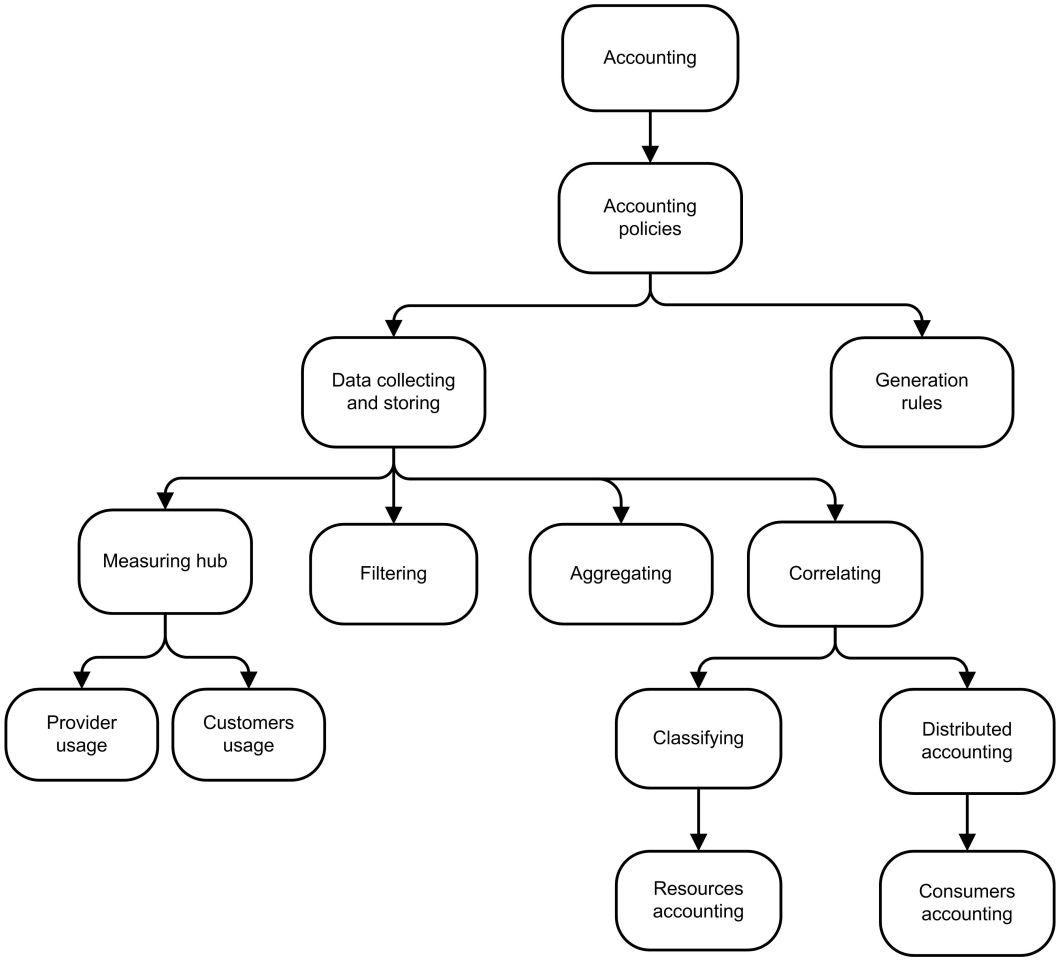

Figure 2.4: Accounting function overview.

The accounting data collection and storing are also known as archival accounting and is performed at a measuring hub [SFPW98b], where the data from the metering readers is collected (also known as storage point). Two winds of data arrive there: issued by the provider and by the customer. The first one is generated by internal and control meter, and is used to control the provider's infrastructure. The latter one represents usage consumption and is used along the whole Internet Economics process.

Moreover, data archival may be necessary due to memory limitations of meter readers or because the information may be needed during long periods of time. It is also used to reconstruct missing entries, and to prevent data lost. Legal or financial requirements frequently mandate archival accounting practices, and may also dictate those data to be kept confidential, regardless of whether it is to be used for billing purposes or not [AAH00].

Similarly, a measuring hub may be necessary to correlate information from distributed meter readers and to process the data solely in one point. Correlations are based on classifying functions that group accounting records by resources. All the available resource accounts are stored by the correlating function, which can also group the accounting records by grouping the data from a distributed accounting organizing the data from the different consumers.

2.1.3.5 **Roaming**

It allows using more than one provider while maintaining a formal, customer-vendor relationship just with one [ALA+97]. In order to offer roaming capabilities, providers require three main subsystems: the consumer's subsystem, which registers visiting consumers, the authentication subsystem, which validates the credential of the consumer, and the accounting subsystem, which has already been described in Section 2.1.3.4 [AZ99].

Further, providers must previously negotiate roaming agreements between them, including legal aspects of authentication, authorisation, and billing of the visiting subscriber. There exist several standards that define a work-field framework for such agreements [Hoc01] [Pro08].

Roaming can also be intra-domain or inter-domain [AAH00]. Being intra-domain implies that there is an exchange of session records between different accounting functions but always in the same provider or with the same administrative boundary. On the other hand, in the inter-domain roaming the session records travel from one provider to another crossing their administrative boundaries.

2.1.3.6 **Pricing**

It sets the price of using a certain resource, being a critical aspect in the full Internet Economics process because it defines the price that a basic quantity of the service will cost. Some authors name this rating or pricing policy [AAH00] since it determines the way a session record is rated. These records stem from the accounting functions and are correlated to the price that is normally represented in monetary units and depends on the pricing scheme used. Figure 2.5 shows an overview of the pricing function.

This process may combine technical considerations, such as resource consumption, and economical ones, such as applying tariffing theory or marketing methods [SGRF01]. The price can be calculated in many different ways (e.g. auctions, static pricing, dynamic pricing, priority pricing, cost-volume-profit analysis scheme or based on market situation analysis) [KSWS99] [CP01] [KP02]. Nevertheless, it will always reflect the results of cost and market analysis. This function translates the previous economic considerations into technical quantities that can be merged with the measurable quantities of consumer resource usages.

Further, pricing schemes are a critical part of the business and are related to cost and market analysis. Pricing schemes can be based on many different paradigms, such as pre-paid, post-paid, time-based, volume-based, flat-rate, usage-based or location-based, among others. Therefore, pricing can be seen as a function for calculating a price, containing pricing variables (consumption measure metrics of the session records) and pricing coefficients [ZZC02].

Tariffs are a especial case of pricing. They are normally regulated by a governmental institution and imply political as well as economic impacts. They have been classically applied to the traditional telephone network, energy or gas markets. Tariffs are defined by tariff models, determining the tariff function for a resource usage.

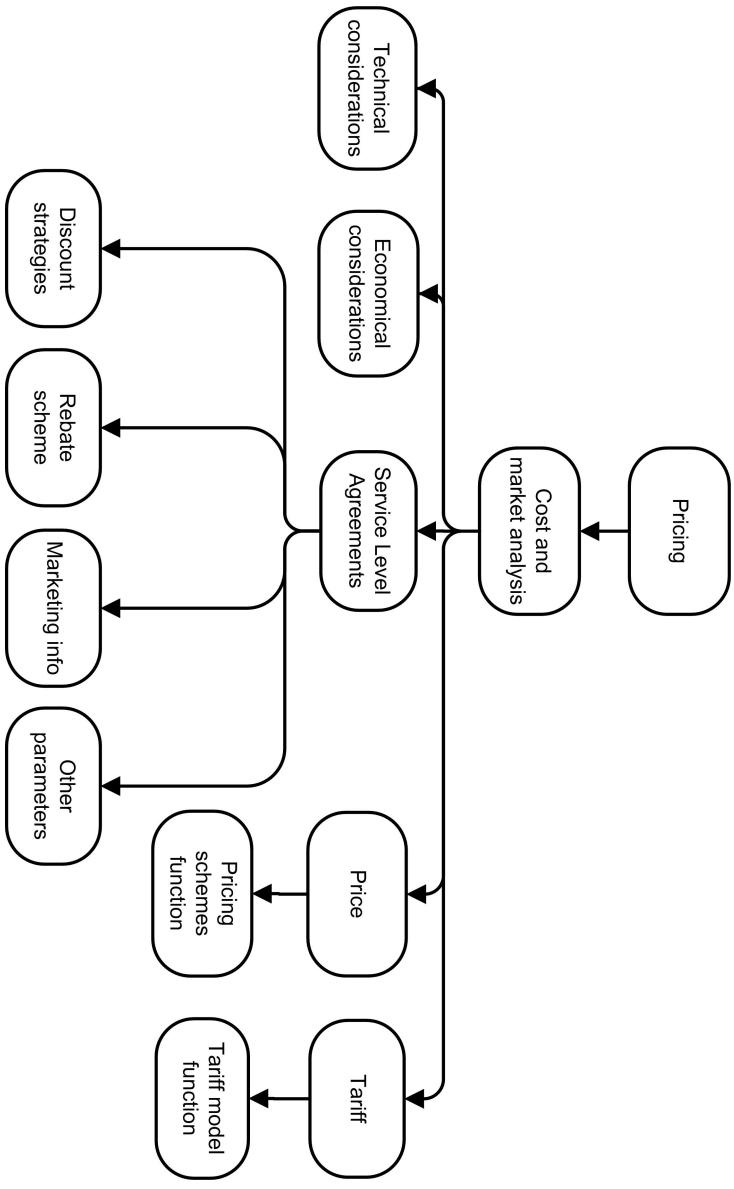

Figure 2.5: Pricing function overview.

Either, pricing or tariff functions are applied within the charging function. They can be modified by discount strategies, rebate schemes, marketing information or any other parameter defined by the service level agreements.

2.1.3.7 **Charging**

This is the process of calculating the cost of resource usage: the function that translates technical values into monetary units by applying a pricing function to the session records [SGRF01]. It correlates session records, from the accounting function, and resource usage unit price to generate charge records [SFPW98a] [KKA+04]. Figure 2.6 shows an overview of the pricing function.

Charging acts as an umbrella term for charging options and charging mechanisms. This separation emphasises both the technical and the economic aspects of charging [MWS06] (some authors refer to charging as billing); nevertheless, as we will see later on, billing implies some different processes, such as customer data management [SGRF01].

The charge records are formed by the technical quantities of a resource usage and their corresponding monetary units. They can be used for multiple purposes of support systems: statistical analysis, data mining, auditing, revenue estimation, financial planning, structure dimensioning, or any other support system.

Charging policies define when and how the billing function is invoked. They specify the cost allocation frequency: every time accounting data are received, at regular intervals of time (e.g. daily, each moth or each two moths) or upon charging function request. They also set the granularity of the billing function (i.e. the degree of subdivision a data field presents). For instance, a postal address can be recorded with low granularity as a single field (address) or with high granularity as multiple fields (street address, city, postal code, country).

Besides, charging can be distributed between multiple parties following the distribution policy. It splits the costs between the different parties or consumers, allocating a previously known cost among several entities [AAH00]. Each party has its own profile containing its pricing function, discounts or especial offers.

Finally, consumers can also establish different business relationships with providers that will define the charging mode of the transaction (e.g. subscribers, pay-per-use or any of the previously introduced schemes).

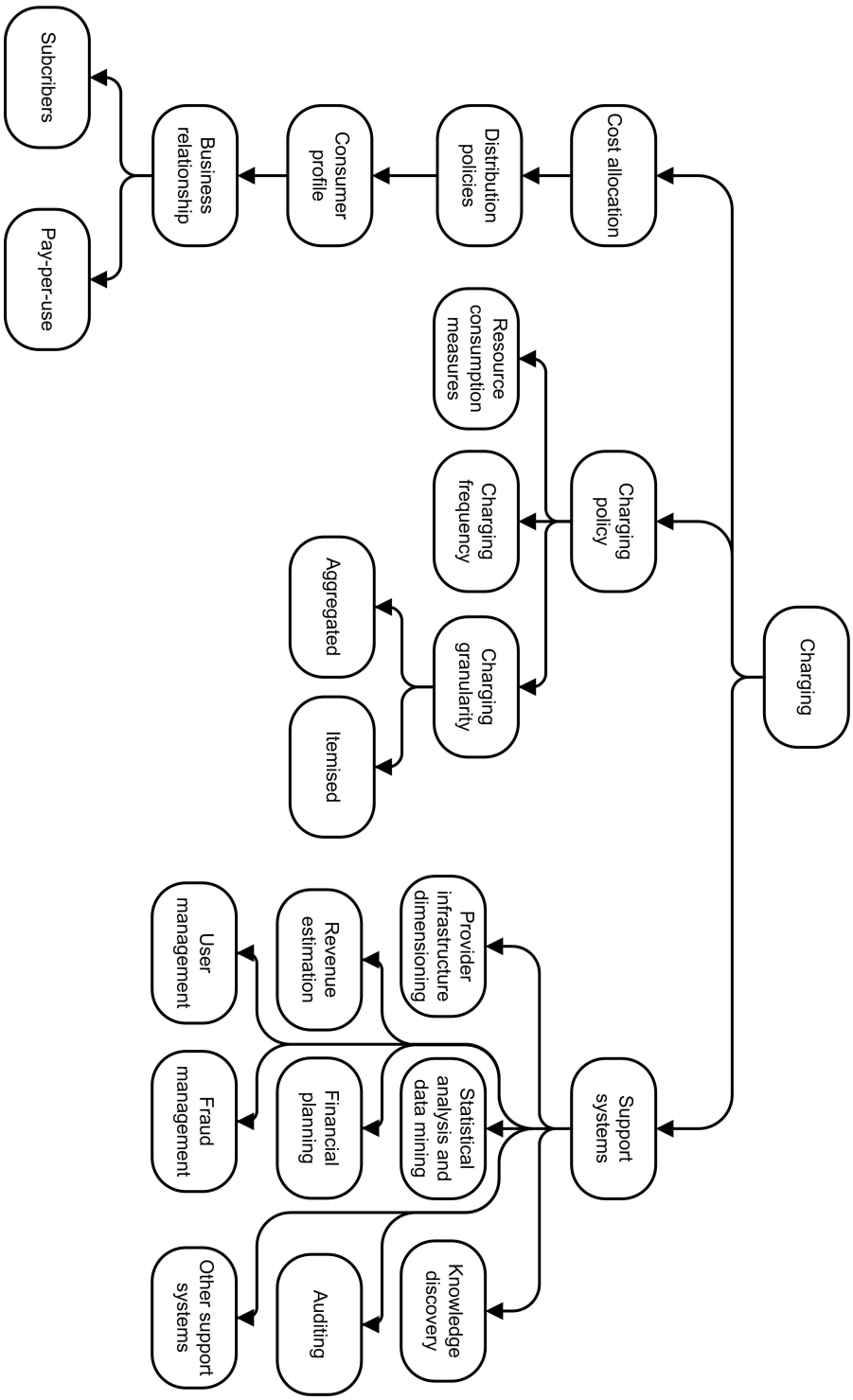

Figure 2.6: Charging function overview.

2.1.3.8 **Billing**

Also known as invoicing, this is the process of transforming charge records into the final bill, or invoice, summarising the charge records of a certain time period (usually a month) and indicating the amount of monetary units to be paid by the customer [KKA+04]. Figure 2.7 shows an overview of the billing function.

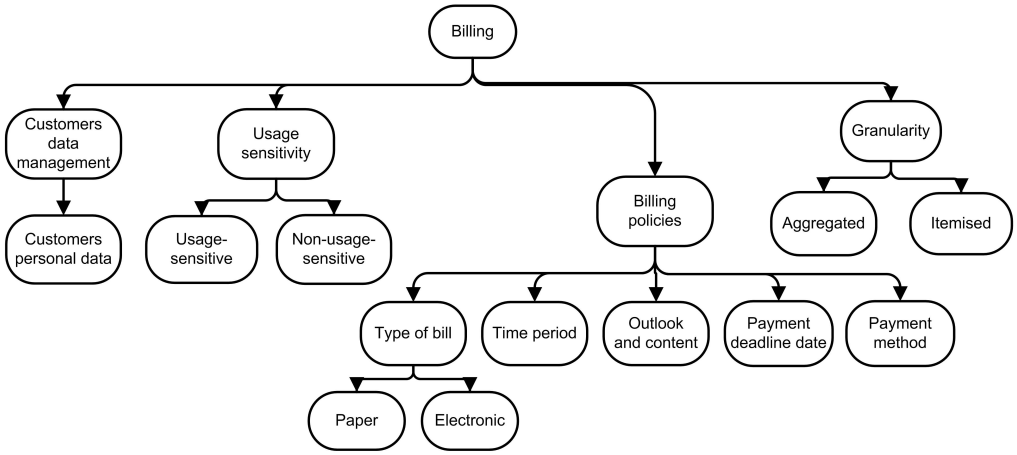

Figure 2.7: Billing function overview.

Billing may include information about the customer (gathered in the customers' data management system, which contains all the customers' personal data). The billing function is also usage-sensitive if it depends on customer's resource usage; otherwise it will be non-usage-sensitive [AAH00].

There exist also billing policies that specify the type of the bill (paper or electronic), the time period that the bill represents, the outlook and content, the payment deadline date and how the financial clearing is done, specifying the payment method [SFPW98a].

In the same way as charging, billing can also present different granularity: an aggregated bill represents two or more charges together and an itemised bill presents all the charges individualised.

2.1.3.9 **Financial clearing**

It includes activities from the commitment to a transaction to its settlement. In the case of resource accounting, this function implies the payment of a bill: transferring the client's agreed money amount to the service provider. Figure 2.8 shows an overview of the financial clearing function.

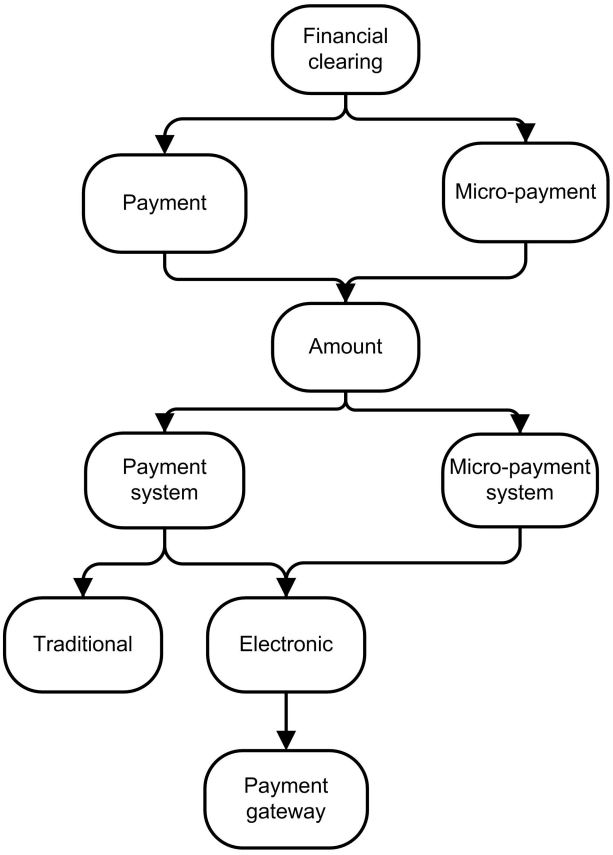

Figure 2.8: Financial clearing function overview.

Moreover, the payment function may use a traditional or electronic payment system. Please note that cash, paper checks and automatic bank clearances belong the group of the traditional payment systems. On the other hand, credit card systems are grouped in the electronic payment systems. The way money is exchanged between all the participants [SFPW98a] through a payment gateway is specified by a previously well-defined scheme.

There is also a especial type of payment, the micro-payments. They require high speed processing: delivery occurs immediately and in small sums of money, based on electronic means. A payment system also supports money transfers, which are smaller than the minimal economically feasible credit card payment [Pá05].

2.1.4 Scope of this dissertation

From the possible areas of work in the Internet Economics process, we decided to focus on charging. It is the most interesting one from our point of view, as we want to apply clustering algorithms in order to improve support systems. Figure 2.9 represents the integrated vision of the Internet Economics process stressing the charging area.

2.2 Clustering algorithms

Knowledge discovery helps to unveil patterns in large volumes of data that provide new information about that data. Our specific scope is concentrated on the subset of knowledge discovery known as data mining.

These techniques are a mature discipline [Lov83] devoted to the discovery of hidden predictive or value-adding information from large databases [TSK06]. They help to identify business components from domain business models [MZX05] and analysing data to feed decision support systems [CY00].

More accurately, we focus on clustering, a subset of data mining, applied in such diverse areas as energy systems [CNP06], *DeoxyriboNucleic Acid (DNA)* analysis [GWR$^+$06], document clustering [CH05], psychology and other social sciences, statistics, biology, pattern recognition, machine learning, information retrieval, and many others.

Clustering algorithms are statistical data analysis techniques used in many fields [Lov83] such as, pattern recognition, image analysis, document classification [CH05] and bioinformatics [GWR$^+$06], climate, business, medicine, and so on. Clustering algorithms select groups, or clusters, in a certain dataset according to their degree of similarity.

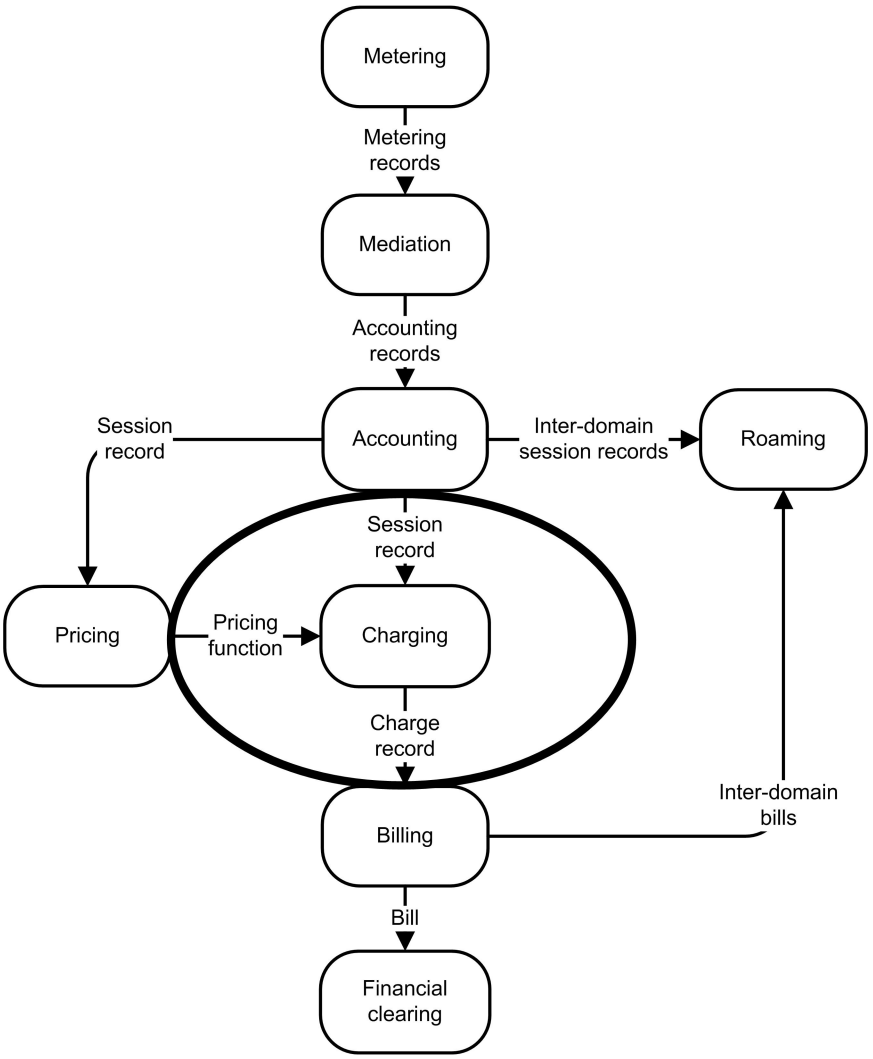

Figure 2.9: Area of study in the integrated vision of the Internet Economics process.

Indeed, clustering is the technique that separate data points, or instances, into clusters based on dataset instance values and their relationships. While it gathers similar or close instances, non similar instances are split [TSK06]. The aim is generating groups in which instances are as similar (say related) as possible and as different (say unrelated) as possible to the instances of the rest of the groups. The greater the similarity among the instances, and the greater the difference among them, the better (say more distinct) the resulting sets will be.

Clustering assumes that the obtained groups' subdivision will give an insight on the structure of the data, improving in this way the performance on support systems. The set of clusters, the data instances distribution and any other information that the clustering algorithms generate form an entity called a model: it provides an abstraction of the dataset instances into the clusters that contain them.

Further, clustering algorithms classify the instances of the dataset by labelling them with a class label. They do not use previously-known classes like discriminant analysis and decision analysis do [BK99]. These disciplines aim at finding rules for classifying instances in a set of pre-classified or trained objects. The study of these algorithms is out of the scope of this dissertation.

There has been an intense research activity regarding clustering in recent years, resulting in several well-known algorithms, such as K-Means [Mac67], PAM [IAH08], CLARA [CPP+10], CLARANS [WLH03], BIRCH [PR07], CURE [QGZ03], ROCK [GRS00], DBSCAN [QGZ03], DENCLUE [HG07a], STING [WYM97], Wave-Cluster [SCZ98], Fuzzy C-Means [BE84] or EM [GI10]. See [HBV01] for an exhaustive review on this topic.

As clustering is applied in many areas, the number of clustering-related techniques and algorithms is vast in the literature. Therefore, in Section 2.2.1 we provide a brief description of the different clustering types, and in Section 2.2.2 introduce the most relevant clustering analysis techniques to our research. As each clustering algorithm presents its own properties, Section 2.2.3 compares the behaviour of the ones addressed in this dissertation. Finally, in Section 2.2.4 we study the existing cluster validation approaches.

2.2.1 Clustering approaches

There are many different ways of achieving clustering, with several configurations and implementations. This variety of possibilities enables diverse forms of classifying the clustering algorithms according to their type.

In this way, there exist a number of different sorting of the algorithms [AF07] [JD88] [JMF99] [HBV01] [Ber06] [XW+05]. Nevertheless, we consider that a general classification is enough for the purposes of this research:

- **Hierarchical clustering:** generates a nested sequence of instances with a unique cluster at the top, fathering the rest. At the bottom, clusters of just one instance may appear. The intermediate levels can be understood as the joint of two clusters of the lower level and, simultaneously, as a split of the clusters of the higher level. The algorithms merge or split the instances until a final dendrogram (tree of clusters) is created.

- **Partitional clustering:** obtains a one-level not nested distribution of the dataset. The algorithms in this category try to find all the clusters at once. The clustering is performed intending to optimise a model selection function.

- **Grid-based clustering:** the dataset is quantised in the space in a finite number of cells. These algorithms are mainly used in spatial data mining. They aim at finding the topology of the dataset from the quantised space and, to this end, they analyse the underlying attribute space by inducing instance membership.

- **Co-occurrence of categorical data clustering:** issue the clustering set according to the concept of variable-size transaction, it has a finite set of elements called items from the dataset and it is used to tackle datasets of categorical data presenting high dimensionality, significant zero values, and small number of common values between instances.

- **Constraint-based clustering:** applies the requirements and constrains that a certain problem may present. The constrains may be both in the individual or in the attributes instances.

- **Scalable clustering algorithms:** are especially designed to manage very large datasets. In turn, they are divided in incremental mining, data squashing or reliable sampling.
- **Density-based clustering:** groups dataset instances by using density conditions, connectivity's, and boundaries. They apply a metric based on spatial data clustering and are able to discover clusters of arbitrary shapes.
- **Distance and similarity clustering:** the data is described in a multidimensional vector. The measure factor is determined by the feature type (e.g. quantitative, qualitative, continuous, binary, nominal, ordinal).
- **Squared error-based clustering:** it is used as a criterion function. The goal in this clustering approach is to minimise the squared error as much as possible.
- **Mixture densities-based clustering:** instances are generated according to different probability distributions (e.g. the multivariate Gaussian). For well known distributions the finding the clusters is the same as estimating the parameters of the models.
- **Graph theory-based clustering:** uses the concepts of graph theory adapting it to the clustering problems. It detects edges between the nodes in order to perform the distribution.
- **Combinatorial search techniques-based clustering:** considers the clustering as a combinatorial optimisation problem. Given a set of instances this types of algorithms groups the clusters by a criterion function.
- **Fuzzy Clustering:** the instances in this clustering type can belong to all the clusters with a given degree of membership. It is especially used in those cases in which the boundary between the clusters is not clear. It may help finding unexpected relationships between the data.
- **Neural networks-based clustering:** uses neurons to activate and erase certain clustering regions. It has learning properties and can describe the data structure in the neural network layer by layer.

- **Kernel-based clustering:** transform complex and non-linear datasets into higher-dimensional vectorial spaces with the purpose of being able to operate linearly with this datasets.

2.2.2 Most relevant clustering analysis techniques

Next, we outline the most relevant clustering algorithms used in this research. Please follow the bibliographical references for a more detailed accounting of each one.

2.2.2.1 BIRCH

The *Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)* algorithm was specially designed for clustering large datasets hierarchically [ZRL96].

It optimises the computational and time resources, and in most of the executions it only requires a single analysis of the data. The assignment of clusters is performed without needing to scan all the instances and the rest of the clusters. It works using an incremental method that does not force to load the whole dataset, minimising the required memory and the operational costs.

The algorithm starts scanning all the dataset and building a tree within the memory constrains. Next, it removes the outliers and grouping the instances scans the leafs of the initial tree to rebuild a smaller tree. Finally, the leafs are clustered and hierarchically organised (the user can define the number of clusters). An additional step could be performed in order to optimise the results redistributing the instances.

2.2.2.2 CLARA

The *Clustering Large Applications (Clara)* algorithm is design to deal with large dataset [CPP+10]. CLARA is based on the K-medoids approach. It clusters a sample from the dataset and then distributes all the instances in the dataset to these clusters.

It uses the sampling approach to manage this datasets. It analyses small subsets from the dataset and generates optimal results using the average dissimilarity between instances in the entire dataset.

Internally, the subset is taken from the dataset and split into K clusters. The instances of this subset are selected to obtain the smallest possible average distance between them and their most similar representative instances. Them, a process to reduce the average distance by replacing the instances.

The sampling is repeated a user-defined number of times. The computational cost is proportional to the number of repetitions, thus, the user has a direct control over it.

2.2.2.3 **CLARANS**

The aim of the CLARANS algorithm is to identify spatial structures that may be present within a dataset [WLH03]. Experimental results prove that is an efficient and effective algorithm.

It is very related to the CLARA algorithm introduced in Section 2.2.2.2. Nevertheless, the internal functionality is different:

1. Input parameters are $numlocal$ and $maxneighbor$. Initialise i to 1, and $mincost$ to a large number.

2. Set current to an arbitrary node in G.

3. Set j to 1.

4. Consider a random neighbour S of current, and based on 5, calculate the cost differential of the two nodes.

5. If S has a lower cost, set current to S, and go to Step 3.

6. Otherwise, increment j by 1. If $jmaxneighbor$, go to Step 4.

7. Otherwise, when $j > maxneighbor$, compare the cost of current with $mincost$. If the former is less than $mincost$, set $mincost$ to the cost of current and set $bestnode$ to current.

8. Increment i by 1. If $i > numlocal$, output best node and halt. Otherwise, go to Step 2.

2.2.2.4 CLIQUE

The *Clustering in QUEst (CLIQUE)* algorithm identifies dense clusters in subspaces of high dimensionality. The user does not need to guess the subspaces that could be interesting [AGGR98].

The algorithm is insensitive to the instances order and does not assume that the dataset has a canonical distribution. It uses techniques that calculate the density grid and the lowest dimensionality identification.

The algorithms works at different levels. First, it determines 1-dimensional dense instances making a scan over the data. Second, it finds out the number of the dimensional dense units performing a pass over the data. Third, it returns a superset of the dataset of all the number of dimensional dense units. Finally, it discards the dense units from the clusters that do not correspond.

2.2.2.5 CLOPE

The CLOPE algorithm is not sensitive to data order, and neither does it require a big domain knowledge about the dataset. It is used as clustering algorithm and pre-processing algorithm with transactional data.

It is considered fast, scalable, and memory-saving in categorical data clustering, especially for large, sparse transactional databases with high dimensions [YGY02].

The algorithm strengthens the intra-cluster overlapping of transaction items by increasing the height-to-width ratio of the cluster histogram. This ability can be parameterised by a tightness control that will determine the number of clusters of the obtained model.

2.2.2.6 CLUES

The *Clustering Method Based on Local Shrinking (CLUES)* algorithm estimates automatically the number of clusters and only requires a convergence rule as input parameter. This rule will determining when to stop the algorithm.

CLUES organises the instances using similarity and dissimilarity measures based on that rule. It is used to partition large dataset into smaller and homogeneous groups.

The instances are moved to a specific distance toward a cluster centroid. The way they are moved is determined by the median of its K-nearest neighbours. This process is repeated until the input convergence rule triggers.

It is proven experimentally that the obtained results can be as high as with other clustering algorithms [WQZ07].

2.2.2.7 **COBWEB**

COBWEB is catalogued as conceptual clustering and organises the data in order to maximise the inference ability. It is incremental, computationally economic, and can be applied in various domains [Fis87]. The models generated by this particular type of clustering are especially useful for making inferences.

Conceptual clustering is an unsupervised classification algorithm [Mic80]. It differs from traditional clustering because it generates concept descriptions for each class. Normally, conceptual clustering implies the generation of hierarchical structure and is used in conjunction with concept analysis, decision tree learning and mixture model learning.

Further, it was the first clustering algorithm inspired by environmental and performance concerns. Until then, traditional clustering techniques solely focused on learning (i.e. clustering) and the resulting knowledge base (i.e. classification).

The COBWEB algorithm performs a hierarchical conceptual clustering. It accomplishes a hill-climbing search through the hierarchical classification space by using operators that enable bidirectional navigation through classification trees [GLF89].

2.2.2.8 **CURE**

The *Clustering Using REpresentatives (CURE)* algorithm is design to manage outlier instances, and to identify clusters that do not have spherical shape and have a wide variance in size [GRS98].

It represents each cluster with a certain fixed number of selected by well scattered points from the cluster and the shrinking them toward the centre of the cluster by a specified threshold.

CURE uses random sampling and partitioning. The datasets are split in small random subsets and they are partially clustered. Them, the outlier instances are removed and all the partial results pass a second phase to produce the final clusters. This enables the algorithm to adjust well to the shape of the clusters despite the geometry of the dataset. It correctly labels the instances to their corresponding cluster even in the aforementioned datasets.

2.2.2.9 **DENCLUE**

The *DENsity CLUstering Denclue (DENCLUE)* algorithm has evolved from its initial design [HK98], and now it has several more advantages [HG07b].

It uses a model based on the estimation of the kernel density. The clusters are formed based on a density function, and the instances that go to the same local maximum are assign to the same cluster.

The first version of the algorithm performed many unnecessary steps and did not converge properly to a maximum. Despite, the new version of the algorithm uses Gaussian kernels, which tune the process automatically with no extra computational cost. Further, the converge points are calculated with higher precision using a variant of the EM algorithm (see Section 2.2.2.11 for details).

2.2.2.10 **DBSCAN**

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm since it searches for the number of clusters starting from the estimated density distribution of corresponding nodes [EKSX96].

It was initially designed to work with spatial databases and aims at grouping the instances of datasets in subsets with intrinsic meaning, and at finding them a class identification.

Furthermore, this algorithm requires a minimal knowledge of the dataset domain to determine input parameters. This feature is especially important when a large amount of variables exists. It performs well in large datasets and can find clusters with an arbitrary shape (e.g. spherical, drawn-out, linear or elongated) due to the use of density-based notions.

Internally, the clusters generated with DBSCAN are composed of points that are mutually density-connected. In case a point is density-connected to any point of a cluster, it is also member of this cluster. This density-connection is calculated by the algorithm during the execution.

2.2.2.11 **EM**

The *Expectation-Maximization (EM)* algorithm associates a probability distribution to each of the instances by indicating how likely they may belong to any of the clusters of the model. It works iteratively, and each loop consists of an expectation (E) and a maximisation (M). It is part of the associated theory, which is simple and generalist, and has a wide range of examples [DLR$^+$77].

The formal description of the EM algorithm is defined as follows. Given a likelihood function $L(\theta; x, z)$, where θ is the parameter vector, x is the observed data, and z represents the unobserved latent data or missing values, the *maximum likelihood estimate (MLE)* is defined by the marginal likelihood of the observed data $L(\theta; x)$. Nevertheless, this amount is often not computationally feasible.

The EM algorithm iterates by trying to find the MLE with the following steps:

- *Expectation* step: Calculates the expected value of the log likelihood function, according to the conditional distribution of z and given x under the current estimate of the parameters $\theta^{(t)}$:

$$Q(\theta|\theta^{(t)}) = \mathrm{E}_{Z|x,\theta^{(t)}}\left[\log L(\theta; x, Z)\right]$$

- *Maximization* step: Looks for the parameter that maximizes this quantity:

$$\theta^{(t+1)} = arg\max_{\theta} Q(\theta|\theta^{(t)})$$

The EM algorithm uses *cross validation (CV)* to validate the number of clusters. *K-fold CV* is usually used in machine-learning evaluation [Bis06] [Koh95].

The most used in the literature are the following steps:

- The number of clusters is initially set to 1.

- The training set is split randomly into 10 folds or subsets as long as the number of instances in the training set is not smaller than 10, in which case, the amount of folds is set equal to the amount of instances. This division is an a standard practice in CV [Koh95].
- EM is performed 10 times using the 10 folds, as is usual in CV [Koh95]. That is, the dataset is split 10 times into 10 different sets of learning sets (66% of the total dataset) and testing sets (34% of the total dataset).
- The log likelihood is averaged over all 10 results.
- If the log likelihood has increased, the number of clusters is increased by 1, and the program continues to step 2.

2.2.2.12 **FarthestFirst**

The FarthestFirst algorithm starts working with any instance of the dataset. Afterwards, it selects the farthest point in it. Next, the farthest point from both of them, and so on until all the instances are analysed. These instances are taken as cluster centres and the rest of the instances are assigned to the closest centre [Das02].

This algorithm is considered the best possible heuristic for the K-Centre problem [HS85], and accepts variation if the problem metric has changed. For instance, the metric could be a distance function or a traversal one. It may be used to get a first approximate cluster distribution and might be useful as initialiser to the Simple-K-Means algorithm.

2.2.2.13 **Fuzzy C-means**

The Fuzzy C-means is a clustering algorithm that allows an instance to belong in more than one cluster. It distributes a dataset into a collection of C fuzzy clusters using a given criteria [Dun73].

Each of the instances are not associated with only one cluster, they have a certain degree of membership for each of the existing centroids (clusters). The algorithm uses a associativity matrix where the instances have a percentage of membership to each cluster.

If the dataset is homogeneous, the clusters will be grouped. This information helps the domain expert detecting which instances do not have a high degree of relevance.

2.2.2.14 **K-Means**

The K-Means algorithm aims at dividing part the instances of a dataset into N clusters, each instance belonging to the nearest mean cluster [HW79a]. It is similar to the EM algorithm introduced in Section 2.2.2.11, since both algorithms try to find the natural centres of the clusters in the dataset.

This algorithm can apply the Euclidean distance or the Manhattan distance for its internal calculations. With the Manhattan distance the centroids of the clusters are calculated with the component-wise median instead of with the mean.

Internally, the K-means algorithm splits M instances of the dataset in a N dimensions sum to pursue the minimisation of the within-cluster of squares. The complexity of the K-Means algorithm is:

- NP-hard in general Euclidean space d even for 2 clusters.
- NP-hard for a general number of clusters k even in the plane.
- If k and d are fixed, the problem can be exactly solved in time $O(ndk + 1logn)$, where n is the number of entities to be clustered.

Therefore, in order to optimise the execution of the algorithm, it calculates local optimal solutions.

2.2.2.15 **LVQ**

The *Learning Vector Quantization algorithm (LVQ)* algorithm is related to the artificial neural networks as internally they operate similarly.

The data is structured as a feature space and the algorithm calculates what is the best cluster for each instance using a given distance measure. This cluster assignation is then adapted. The cluster is moved closer if the instance is correctly assigned and moved away if it is not.

The models generated with LVQ are considered to be easy to user and verify by the domain experts [Koh03], and besides being a clustering algorithm, it can be also applied to classification problems.

2.2.2.16 **OPTICS**

The *Ordering Points To Identify the Clustering Structure (OPTICS)* algorithm finds density-based clusters in a spatial dataset. The conceptual idea is similar to DBSCAN, introduced in Section 2.2.2.10, but OPTICS overcomes the weakness of DBSCAN when detecting meaningful clusters in data of varying density.

It processes the instances of the database linearly by ordering the data spatially. In addition, each instance stores a distance representing the density that requires to be accepted for a cluster in order to build the distribution [ABKS99].

This algorithm tries to face the need of determining which input parameters are the most relevant for the clustering output. Furthermore, in many real life datasets, there is not domain expert that can determine which input generates the most accurate clustering model. OPTICS generates an ordering of the dataset that represents a density-based structure.

2.2.2.17 **OptiGrid**

The *Optimal Grid-Clustering (OptiGrid)* algorithms is conceived to deal with high-dimensional space datasets [HK99]. It performs a grid-partitioning of the datasets. It determines the best partition using hyperplanes and projections for each dimension.

Generally, in high-dimensional spaces the cluster centres cannot be easily calculated. Therefore, this approach clusters are split by a dimensional cutting plane and the instances of the clusters are spread over a grid cell. It also considers that calculating the connections for handling the effect of split clusters is not efficient. It offers a solution that guaranties the effectiveness preserving the efficiency.

Despite, this technique avoids all the effectiveness problems that other algorithms may have.

2.2.2.18 ORCLUS

The ORCLUS algorithm achieves subspace clustering [AY00]. It tries to redefine clustering for high dimensional dataset looking for hidden subspaces with clusters created with interrelations between instances.

It uses a arbitrarily oriented projected clusters approach. It is an effective solution to manage the dimensionality curse of the datasets. It eliminates most of the sparse subspaces of each cluster, and projects the points into these subspaces in which the greatest similarity occurs.

The algorithm only processes a random subset of instances each time to reduce the computational cost. In order to keep the accuracy, it associates the existing clusters with all the instances using robust merging decisions. In this way, the full dataset is hierarchically merged improving the response times.

2.2.2.19 Self-Organizing Map

The *Self-Organizing Map (SOM)*, also known as *Self-Organizing Feature Map (SOFM)*, are especially used to generate low dimensional results that represent the input dataset [Koh89].

This representation is called *"map"* and uses neighbourhood functions to preserve the topological properties of the data. This map is formed of nodes or neurons, each of them has associated a weight vector. The common representation of the dataset is a regular spacing in a hexagonal or rectangular grid.

For each instance the algorithm finds the node with the closest weight to the vector space. When the closest nodes are located, the values in the vector are assigned to the dataset.

2.2.2.20 WaveCluster

The WaveCluster algorithm is based on wavelet transformations (a special type of Fourier transformation [RG75]). It can identify arbitrary shape clusters with different degrees of accuracy.

It processes the dataset as it would be a signal processing problem [SCZ98]. The instances are organised in a feature space as a multidimensional signal. The high frequency instances of the signal correspond to the boundaries of the clusters,

and it is easy to change the cluster assignment. On the other hand, the low frequency instances have a higher amplitude that corresponds to the clusters themselves, the instances are more concentrated.

The algorithm makes use of the wavelet transform technique decomposing the signal into different frequency sub-bands. A one-dimensional transformation can be applied to multidimensional signals multiple times obtaining a feature space with dense regions (clusters).

2.2.2.21 **sIB**

The *sequential Information Bottleneck (sIB)* algorithm uses the sequential information bottleneck algorithm. It is guaranteed that the algorithm will converge to a local maximum with a significant improvement from the original information bottleneck algorithm, both in time and space complexity [SFT02].

The algorithm was initially designed to work with unsupervised document classification, facing a scenario where no labelled examples were provided. This strategy can be applied when there is no expert knowledge of the data. The low complexity of the algorithm enables the processing of large datasets, obtaining better results than with some supervised algorithms such as Naive Bayes.

2.2.2.22 **X-Means**

The X-Means algorithm is an extension of the K-Means algorithm. It uses statistically-based criteria to make local decisions that maximise the model's posterior probabilities [PM].

The algorithm tries to achieve three main contributions: improving the computational scalability, overcoming the need of defining the number of clusters in advance, and the trend to stack the searches in local minima.

The algorithm traverses the dataset space looking for cluster locations and the number of clusters to be used by using the *Bayesian Information Criterion (BIC)* or the *Akaike Information Criterion (AIC)* [Aka70] as measure techniques. With these measures, it makes local decisions about which subsets of the current centroids should split themselves in order to best fit the dataset.

2.2.3 Comparison

Each clustering algorithm presents its own properties and ways of dealing with data. Therefore, it is necessary to compare these properties in order to determine which one to use in each problem scenario.

In this way, we are going to compare solely the clustering algorithms that were relevant to our dissertation as they could be applied in the use-case experiment. The comparison is made according to three criteria. First, we will study the general capabilities to manage the input dataset. Second, we will provide an overview to the parameterisation options that each algorithm shows. Last but not least, we will introduce the output that each algorithm produces.

The described algorithms may present different variations in their capabilities, parameterisation options, and output depending on their actual implementation. This present comparison is based on the implementation of the clustering algorithms by the *Waikato Environment for Knowledge Analysis (Weka)* knowledge analysis machine learning suite [WF05].

In relation to the algorithms' capabilities to manage the input dataset, there are different categorisations [HBV01] [Hua97] [GRS00]: statistical (limited to numeric data), conceptual (based on concept categories), fuzzy clustering (instances can be classified in more than one cluster), crisp clustering (instances are in a cluster or not at all, non-overlapping), or Kohonen net clustering (based on neural networks).

Nevertheless, we are more interested in analysing the types of attributes that can be handled because in case it is not compatible with the dataset it is not possible to use it.

Table 2.1 summarises the described clustering algorithms' general capabilities. It can be observed that none of the algorithms is able to handle all the types of data attributes, and therefore, it is necessary to apply one or the other depending on the problem to be solved. The ones that show common attributes may handle similar datasets.

Nevertheless, as the internal working schemes are different, the results of the clustering process will be different. This fact forces the validation and comparison of the results of the clustering algorithms as detailed in Section 3.2. Finally, we also

consider an additional aspect common to all of them, the obvious need of having at least one instance to work with.

Table 2.1: Clustering algorithms general capabilities comparison.

Capabilities \ Algorithm	CLOPE	Cobweb	DBScan	EM	FarthestFirst	OPTICS	sIB	SimpleKMeans	Xmeans
Missing values	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Binary attributes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Empty nominal attributes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Nominal attributes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Unary attributes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Date attributes	No	Yes	Yes	No	Yes	Yes	No	No	Yes
Numeric attributes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Minimum number of instances	1	0	1	1	1	1	1	1	1

Each algorithm accepts different parameterisations depending on its internal specifications and the provided dataset. It is worth mentioning that there are standard or default values for each of them, representing the most common uses. Below, we show a comparison of the selected clustering algorithms parameterisation options:

- CLOPE
  - Repulsion: value of the repulsion to be used.
- Cobweb
  - Acuity: value of the minimum standard deviation for numeric attributes cutoff.
  - Cutoff: value of the category utility threshold by which to prune nodes seed.
  - Save Instance Data: indicates if the instance information is saved for visualization purposes or not.
  - Seed: value of the random number seed to be used.
- DBScan
  - Database Type: value of the used database.
  - Database DistanceType: value indicating the used distance type.

  - Epsilon: value of the radius of the epsilon range queries.
  - MinPoints: value of the minimun number of Data Objects required in an epsilon range query.

- EM
  - Debug: if set to true, the clusterer may output additional information.
  - Display Model In Old Format: indicates if the old format for model output is used or not. The old format is better when there are many clusters. The new format is better when there are fewer clusters and many attributes.
  - Max Iterations: value of the maximum number of iterations.
  - Min Std Dev: value of the minimum allowable standard deviation.
  - Num Clusters: value of the number of clusters. If the value is set to -1 the number of clusters is automatically defined by cross validation.
  - Seed: value of the random number seed to be used.

- FarthestFirst
  - Num Clusters: value of the number of clusters.
  - Seed: value of the random number seed to be used.

- OPTICS
  - Database Output: value of the optional output file for the generated database object.
  - Database Type: value of the used database.
  - Database distance Type: value of the used distance type.
  - Epsilon: value of the radius of the epsilon range queries.
  - Min Points: value of the minimum number of Data Objects required in an epsilon range query.
  - Show GUI: defines whether the OPTICS Visualizer is displayed after the clusterer has been built or not.
  - Write OPTICS results: indicates if the results are written to a file or not.

- sIB
  - Debug: if set to true, the clusterer may output additional information.
  - Max Iterations: value of the maximum number of iterations.
  - Min Change: value of the minimum number of changes.
  - Not Unify Norm: indicates if each instances has to be normalised to a unify prior probability.
  - numClusters: value of the number of clusters.
  - Num Restarts: value of the number of restarts.
  - Seed: value of the random number seed to be used.
- SimpleKMeans
  - Display Std Devs: indicates if it has to display standard deviations of numeric attributes and counts of nominal attributes.
  - Distance Function: value of the distance function to use for instances comparison.
  - Do not Replace Missing Values: indicates if it has to replace missing values globally with mean mode.
  - Max Iterations: value of the maximum number of iterations.
  - Num Clusters: value of the number of clusters.
  - Preserve Instances Order: indicates if the order of the instances is preserved or not.
  - Seed: value of the random number seed to be used.
- Xmeans
  - Bin Value: value of the attribute that represents a true value.
  - Cut Off Factor: value of the cut-off factor to use.
  - Debug Level: value of the debug level to use.
  - Debug Vectors File: value of the file containing the debug vectors.

- Distance F: value of the distance function to use.
- Input Center File: value of the file that contains the list of centres.
- Max Iterations: value of the maximum number of iterations to perform.
- Max K-Means: value of the maximum number of iterations to perform in K-Means.
- Max K-Means For Children: value of the maximum number of iterations K-Means that is performed on the child centres.
- Max Num Clusters: value of the maximum number of clusters.
- Min Num Clusters: value of the minimum number of clusters.
- Output Center File:value of the file to write the list of centres.
- Seed: value of the random number seed to be used.
- Use KDTree: indicates if the KDTree is used or not.

Table 2.2 presents the studied clustering algorithms' output comparison. Having different capabilities and parameterisation parameters, it is logical to expect that the output of the algorithms will also be unique for each one of them. Some of the algorithms provide with a lot of information, from the centroids of the clusters to the elapsed time since execution; others are more meagre in their results.

Depending on the problem we are facing, we will also need to consider the required output information. It is important to mention that if we are willing to compare or validate the models resulting from the clustering algorithms application, we will only be able to do it for each algorithm type or for those with common output. Furthermore, there is an output that joins to all of the algorithms: the clustered instance distribution. This fact will be analysed later on. These data represent how the algorithm assigns the instances to the discovered clusters.

Table 2.2: Clustering algorithms output comparison.

Output \ Algorithm	CLOPE	Cobweb	DBScan	EM	FarthestFirst	OPTICS	sIB	SimpleKMeans	Xmeans
Clustered instances	Yes	No	No	No	No	No	No	No	No
Clustered instances distribution	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Number of merges	No	Yes	No	No	No	No	No	No	No
Number of splits	No	Yes	No	No	No	No	No	No	Yes
Number of clusters	No	Yes	Yes	Yes	No	Yes	Yes	No	No
Tree	No	Yes	No	No	No	No	No	No	No
Elapsed time	No	No	Yes	No	No	Yes	No	No	No
Instance noise per cluster	No	No	Yes	No	No	No	No	No	No
Unclustered instances	No	No	Yes	No	No	Yes	No	No	No
Attribute mean and std. Dev per cluster	No	No	No	Yes	No	No	No	No	No
Loglikelihood	No	No	No	Yes	No	No	No	No	No
Cluster centroids	No	No	No	No	Yes	No	No	Yes	Yes
Cluster - Size - Prior probability	No	No	No	No	No	No	Yes	No	No
Cluster - Size - Prior probability - Attribute	No	No	No	No	No	No	Yes	No	No
Number of iterations	No	No	No	No	No	No	No	Yes	Yes
Within cluster sum of squared errors	No	No	No	No	No	No	No	Yes	No
Requested iterations	No	No	No	No	No	No	No	No	Yes
Splits prepared	No	No	No	No	No	No	No	No	Yes
Cutoff factor	No	No	No	No	No	No	No	No	Yes
Percentage of splits accepted by cutoff factor	No	No	No	No	No	No	No	No	Yes
Distortion	No	No	No	No	No	No	No	No	Yes
BIC-Value	No	No	No	No	No	No	No	No	Yes

2.2.4 Cluster validation

Nowadays, there is a wide list of well-known clustering techniques as a result of a hectic research activity in recent years. Nevertheless, each one presents its own features with different input data, parameterisation, architectures, approaches, and so on, yielding the selection of the most suited very hard (some authors consider that there are no best clustering results at all [Jai10]).

Cluster validation is a difficult interactive process, there is not a clear-cut prescription to perform the cluster validation [HKK05]. Generally, the validation is done differently depending on the problem.

This problem was tackled by changing the data and the algorithm until the results were acceptable [FPSSU96], yet the *acceptability* was *externally* defined (in this occasion, by the researcher). Others [Lia08] added further details such as the identification of the required resources and post-processing knowledge steps, and [HBV01] proceeded by selecting the features to study, selecting the proper clustering algorithm, validating the results, and interpreting them. Still, they both relied on *subjective* categories (e.g. *"properness"*) and neither considered the specific needs and requirements of the clustering algorithms as a whole.

2.2.4.1 Number of clusters

Other well known problem is determining the number of clusters of the model. In some algorithms the user can determine this number, K, using her domain knowledge. In other cases, this K is not defined and it is estimated from the dataset. This problem is even considered as the fundamental problem of cluster validation [Che10].

Different attempts have been tried to estimate the correct K:

- **Visualisation:** in the case of the datasets that can be represented in two or three dimensions, direct observation of the clustering results could be enough to determine the value of K. Nevertheless, in almost all the datasets the complexity of the data does not allow the use of this estimation.

- **Use of indices or stopping rules:** analyse the relation of the instances that are in a cluster against those that are not. The rules can be build with several different criteria and are problem specific.

- **Use of probabilistic function:** tries to fit the model resulting model and the dataset. The comparison is normally done with a criterion that optimises the problem.

- **Use of heuristics:** that range from the eigenvalue decomposition to the, neighbourhood analysis and many others.

Given the importance of defining the correct value of K, we detail some of the most relevant methods.

The Davies Bouldin Index [DB79] identifies the cluster solution that best minimises the intra-cluster distance, and at the same time, maximises the inter-cluster distances.

The gap statistic [TWH01] metric compares the quality of the clusters with a reference dataset with a uniform distribution. It uses a matrix of dissimilarities between the instances to work.

The Hartigan metric [HW79b] detects the optimum number of clusters using the K-Means algorithms. Specifically, it uses the existing intra-cluster dispersion.

The Krzanowsin and Lai index metric [KL88] detects the number of clusters produced by an elbow the the dispersion curve. It uses the number of attributes of the dataset and the dispersion matrix of the clustered data to determine the optimum K.

The Silhouette is another metric [Rou87] that indicates the quality of the clusters allocation. It uses the average distance of the elements in a cluster and the average distance between the instances of the closest cluster.

Furthermore, Several validation techniques can be combine [ASM11] simultaneously evaluating different subsets of the dataset.

2.2.4.2 Internal and external validations

Lately, there have been different approaches to face these questions, such as accurate evaluation of data labelling [CCC08], semi-supervised approaches that define constrains to the models [LDJ07] or consensus clustering [NC07]. Still, none of the solutions solves completely the challenges posed since each of them was designed to solve one requirement specifically, being just a problem solution approximation.

Closer to our approach, the so-called *internal validation* strategies rely on an index to assign the best score to the algorithm. This strategy can be based on different measures:

- **Combinations:** use more than one measure to perform the validation. Some well-known examples are SD-validity Index [HBV01], Dunn Index [Dun74], Dunn-like Indices [BP98], Davies-Bouldin Index [DB79], or Silhouette Width [Rou87]. For example, the production of clusters with high internal similarity within a cluster and low similarity between clusters [CAC10].

- **Compactness:** the validation measures are assessed studying the cluster compactness or homogeneity. The measurement can be done with numerous different metrics such as average or maximum pairwise intra-cluster distances or centroid-based similarities [BP98].

- **Compliance between a partitioning and distance information:** measures directly the distance between the instances in the original dataset and the data

after partitioning. This partition can be represented with a *cophenetic matrix* that shows the distances [Rom04].

- **Connectedness:** assesses how a partition agrees with the degree of a partitioning observed in local densities and instanced groups, together with their nearest neighbours in the dataset [HK05].

- **Predictive stability:** measures do not use any label information. They require additional access to the clustering algorithm, it repeatedly re-sample the original dataset and recalculate the clustering assignations. The consistency of the results gives an estimated significance of the results in the original dataset [BHEG$^+$02].

- **Separation:** quantifies the degree of separation between individual clusters. For instance, this quantification can be based on the minimum separation between individual clusters, or the distance between cluster centroids.

On the other hand, *external validation* compares the results of the algorithms according to some external benchmark:

- **Unary measures:** take a single clustering result as input and compare it with a known set of class labels to check the degree of consensus between them. The extremes cases in with the partitioning consist of singleton clusters or a one-cluster need to be controlled. The use of a confusion matrix, Fowlkes-Mallows index or F-measure [MRS10] assist in finding a proper equilibrium.

- **Binary measures:** assess the consensus between the instances partitioning and the contingency table of the pairwise assignment of the instances. The most relevant binary index is Rand Index [Ran71]. It determines the similarity between two clustering results as a function of agreements in pairwise cluster assignments.

Our methodology combines both internal (by using several datasets and measuring the distance of the optimal cluster set to the rest) and external validation (by applying the metric and criterion).

Historical methodology, as I see it, is a product of common sense applied to circumstances.

Samuel E. Morison

CHAPTER

3

Optimal clustering model selection methodology

Having introduced the pillars of the dissertation research areas, it is time to detail our overall methodology. It aims at obtaining the optimal clustering models in order to enable efficient knowledge discovery.

This chapter is devoted to the analysis of traditional knowledge discovery methodologies that normally use iterative approaches. Against these drawbacks, we propose a three-step methodology that makes possible finding the best cluster set for each problem in a quantifiable manner. First, it executes the selected (say applicable) clustering algorithms. Second, it selects the best model for the problem given and for each of the datasets. Third, operation with it applies the best global model to improve support systems. In order to ease the understanding of the methodology, we provide a simple synthetic use-case based on a travelling salesmen management company.

3.1 Traditional knowledge discovery methodologies

The first attempt to describe the steps that form a knowledge discovery process come out with an interactive and iterative process involving steps that include data

selection, pre-processing, integration and transformation (we include all of them under the common banner *"data acquisition"*), data mining algorithm selection and result interpretation. This process is repeated by changing the data and the algorithm until the results are acceptable [FPSSU96].

Other suggestions address more detailed methods, including the identification of the resources required and knowledge post-processing steps [Lia08]. Generally, most of the authors define the required steps as extracting knowledge by choosing the features we want to study, selecting the proper clustering algorithm, validating the results, and interpreting them [HBV01].

One of the hardest steps on clustering is the validation step. It is performed by a domain expert that studies the algorithm parameters and their results and selects the best model based on her knowledge. She analyses the attributes, ranges, distances between clusters, and many other aspects. Nevertheless, and due to the specifications of most of the problems (in which we may lack of a domain expert), we need to find a non-supervised way to validate the results of clustering algorithms.

3.2 Proposed methodology

We have tailored and extended the existing knowledge discovery methodologies to suit our needs, focusing specifically on the clustering algorithms. As Figure 3.1 shows, our methodology relies on a modular, parameterised architecture oriented to obtain comparable meta-results. These meta-results will be fed to the meta-learning process [BGCSV08], helping us obtaining the optimal knowledge model.

Clustering results is the phase devoted to the execution of the clustering algorithms. It receives the available datasets to be data-mined, an action plan that defines how to prepare the datasets for its use, and the parameters that each algorithm requires (we assume all datasets share the same structure). The resulting cluster sets are stored in a database that will be used in the next step. Please note that this phase will generate a different distribution set for each algorithm and parameter configuration. Let us enrich the explanation of the methodology with an example: think of a travelling salesmen management company operating in Britain. In this case, we would have 3 separate datasets (England, Scotland, and Wales) to record all the journeys of the respective salesmen within a certain period of time.

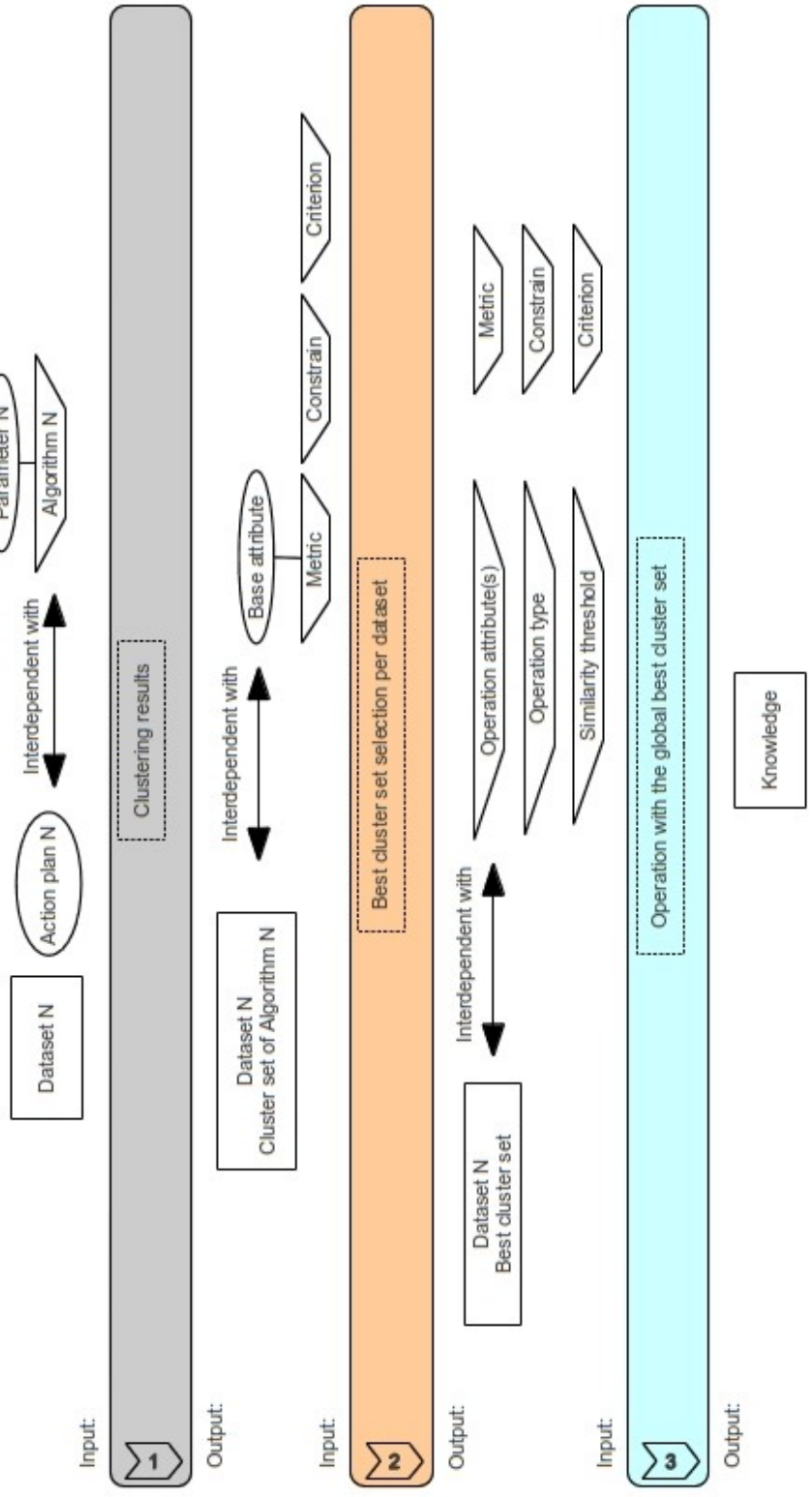

Figure 3.1: Optimal clustering model selection methodology.

The best cluster set selection module picks out a *base attribute* in these cluster sets that allows it to compare them. It must be present in all the algorithms and may be a piece of meta-data of the algorithms (e.g. number of instances) or an attribute of the datasets, like the mileage in our example. The *metric* will use it as the fitness function to measure cluster set performance (e.g. number of instances per cluster or mileage per month). This metric may include some *constrains* to be applied (e.g. monthly mileage must be under 5,000 miles).

Finally, a *criterion* sets the ranking rules (e.g. lowest mileage per month). The ultimate goal here is to select the best cluster set for each dataset, according to the criterion, and also to determine which one of them is best suited to represent the overall behaviour. Please note that we understand *"cluster set"* as a set of various elements: result information, cluster assignments to the instances, and other output information. In the example, after applying the lowest mileage per month criterion to journeys above 5,000 miles, we would obtain several distribution sets for England, Scotland, and Wales ranked from the lowest (and also best) to the highest mileage.

Now we are able to perform a global validation of the cluster set of each dataset by comparing their metric results. This task only shows whether the selected best model is correct; it does not provide with any further *"useful"* information. Moreover, this information will depend on the initial datasets. Thus, datasets could differ from one to another as this global validation does not include a deep attribute analysis.

The point here is modelling the general representation for all the datasets and using it to extract data-independent knowledge (i.e. applicable to the overall problem). For this purpose, in the operation stage we will use the best cluster set obtained in each dataset by focusing on a certain operation type (e.g. commercial efficiency of the travelling salesmen). This operation type will be specified by the operation attribute represented by the measure unit (e.g. sales). If the distance between the results of the dataset best cluster sets and the optimal global one shows a similarity above a pre-defined threshold, we can conclude that the selected optimal global cluster set is representative of the whole problem data.

In the example, we would calculate the commercial efficiency (i.e. sales/miles) of the best cluster set in England, Wales, and Scotland. Assuming for instance the

best cluster set of Wales as the optimal one, and that it is similar to English and Scottish clusters above the given threshold, we would conclude that the obtained sales/miles information for each of the groups of the cluster set can be exported to the whole Britain.

3.2.1 Clustering results

As aforementioned, the clustering result phase has three inputs: the datasets, the action plans and the clustering algorithms. Figure 3.2 shows in detail the clustering results step of the optimal clustering model selection methodology.

The datasets used in this function require a pre-processing step in order to make the data ready regardless of the algorithms that will be used [HBV01]. The data to be clustered undergo a feature transformation stage in which the dataset is prepared for the next step. These transformations are defined by the action plan.

At the same point, it is important that all the used datasets share a common structure so they can, somehow, share the representativeness of the resulting cluster sets.

A dataset is formed by multiple instances, and each instance will have one or more attributes that indicate its characteristics. These attributes will be defined by a value that determines their shape. These instance are also known as vectors; therefore, the dataset can be considered a multidimensional vector. Worth mentioning, representing the data through vectors is not the natural presentation of heterogeneous data. In these types of information, data are not represented by using a predefined feature vector length. Examples of such heterogeneous data are rank data, dynamic data, graph data, and relational data [Jai10].

In order to simplify the explanations of the different phases of this methodology, we use the example given in Section 3.2 with minor differences as a starting point. We will start by defining the dataset described in Table 3.1. This dataset contains data about a travelling salesmen management company operating in Britain. Each instance defines the operation nation in Britain for the salesmen, their name, the represented month, the mileage over this period and the achieved sales. Please note that in this example, and for the sake of simplicity, we are only going to use one dataset.

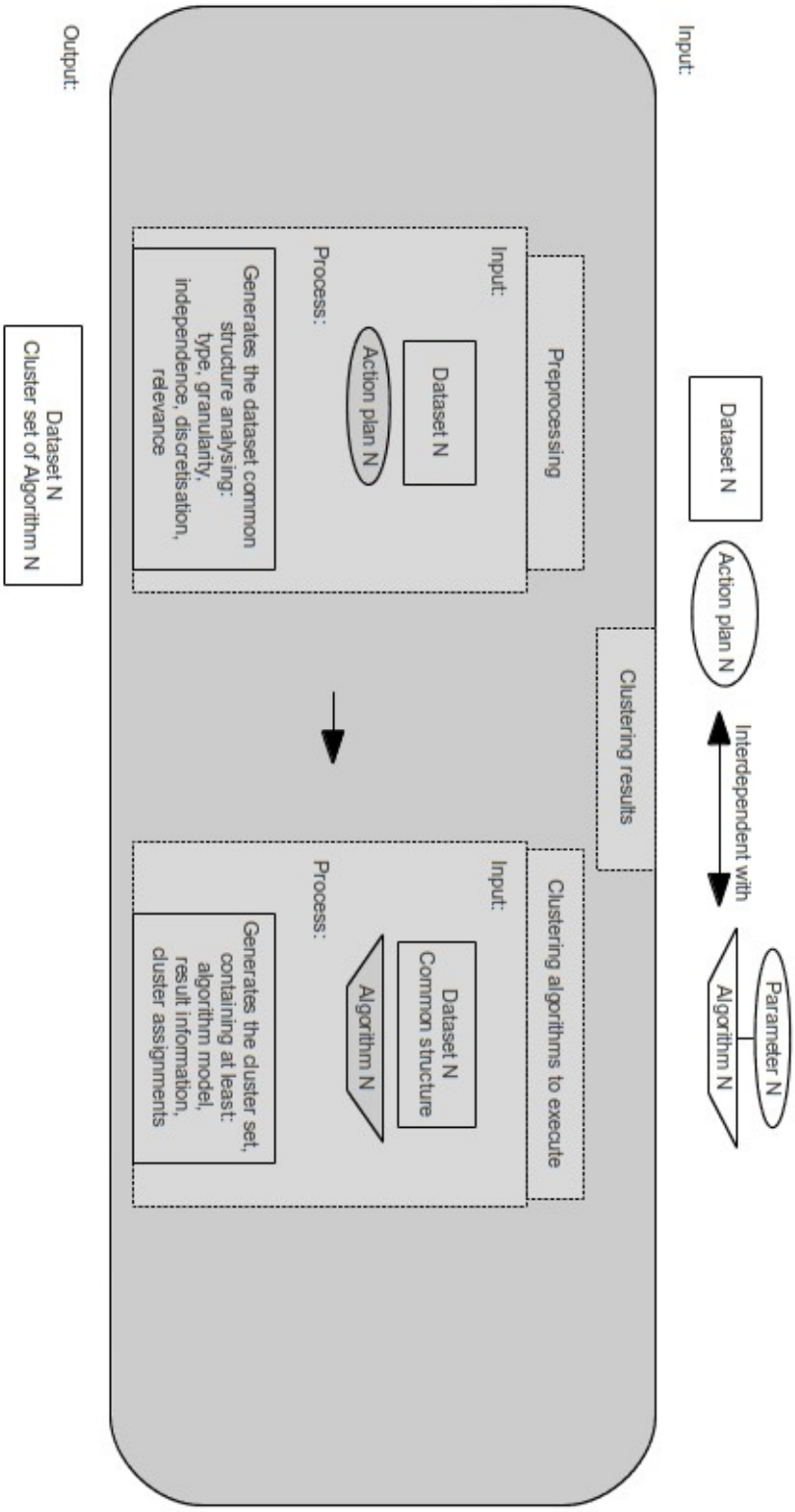

Figure 3.2: Clustering results step of the optimal clustering model selection methodology.

Table 3.1: Travelling salesmen management company dataset.

Nation	Salesman name	Salesman identifier	Month	Mileage	Sales	Favourite colour
Wales	Trevor	11	January	4,123	40,454.87 £	Red
Wales	Gwyn	12	January	3,987	73,263.00 £	Blue
Scotland	Niall	21	January	5,010	19,438.23 £	Green
Scotland	Macbeatha	22	January	4,564	70,430.99 £	Yellow
England	Alan	31	January	3,026	34,065.01 £	Black
England	John	32	January	4,794	20,145.45 £	White

When working with these instances, it is important to define and understand the meaning and semantics of each of the attributes of the instances. Table 3.2 describes each of them in our example.

Table 3.2: Dataset attributes description.

Attribute	Description
Nation	Nation of Britain in which the salesman operates
Salesman name	Name of the travelling salesman
Salesman identifier	Salesman identification code
Month	Reference to the month of the year for the instance
Mileage	Travelled distance in miles for the instance month
Sales	Achieved sales in pounds for the instance moth
Favourite colour	Salesmen's favourite colour

These attributes present different types and compose a feature vector of measures that represents the data instances properties [Jai10]. The type can range from strings, integers, reals, booleans, dates to nominal, for instance. It is important, as it will determine the possibility of applying certain clustering algorithms or others. Table 3.3 shows the attribute types of the example dataset.

Moreover, each attribute will show its own granularity. It can be unitary, when the attribute cannot be split in new attributes, or compound, when the attribute may be divided. In some cases the information about the attribute can be truncated by simplifying it.

Date is a classic example of a compound attribute. Normally, it is composed of year, month, day, hour, minute, second and time zone. If it is relevant to the problem, the attribute could be asunder in new unitary attributes: year, month,

Table 3.3: Dataset attributes type.

Attribute	Type
Nation	Nominal, String
Salesman name	String
Salesman identifier	Integer
Month	Nominal, String or Integer (could be represented in both ways)
Mileage	Numeric, Integer
Sales	Numeric, Real
Favourite colour	String

hour, minute, second and time zone. The decision of unbinding the data or not will depend on the problem and the desired results.

On the other hand, numeric real values can be an example of an attribute truncation. For instance, $\Pi(pi) = 3,14159265358979323846...$ can be truncated to four decimals $3,1416$. In the case of string valued attributes, they can be truncated by cutting the string from a certain char until the end of the chain. For example, the string *"user-name@deusto.es"* can be truncated to *"user-name"*. This action allows to extract the user-name from the email attribute.

In the example of the travelling salesmen, all the attributes have a unitary granularity as indicated in Table 3.4. Mileage and sales could be truncated in order to simplify and generalise their values.

Table 3.4: Dataset attributes granularity.

Attribute	Granularity
Nation	Unitary
Salesman name	Unitary
Salesman identifier	Unitary
Month	Unitary
Mileage	Unitary, can be truncated
Sales	Unitary, can be truncated
Favourite colour	Unitary

Attributes can also be interdependent and redundant. The same information can be represented completely or partially in more than one attribute. An attribute

will be independent if, and only if, there are no other attributes that represent the same information or part of it. The removal of duplicated and non useful attributes optimises the clustering results because data representation is augmented [Ber06].

There may be cases in which it is difficult to determine if an attribute is independent or not. The analysis of this property requires a deep understanding of the dataset (i.e. expert knowledge).

Further, repeated attributes are an example of a redundant attribute. This situation may happen in case there are data redundancies caused by a bad dataset design or to ensure that the recorded values are correct if they are recorded from different inputs.

For instance, an interdependent attribute example is the price of a kilogram of rice (3 £ per kilogram) and the price of a rice package (6 £ per package). These attributes are interdependent because the rice package has a certain amount of rice measured in kilograms, two kilograms in this case.

In the aforementioned example, the attribute nation, month, mileage and sales attributes are independent and there are no redundant attributes. Nevertheless, salesmen name and salesmen identifier are interdependent attributes, since both can unequivocally identify a salesman. See Table 3.5 for a short summary.

Table 3.5: Dataset attributes independence.

Attribute	Independence
Nation	Independent
Salesman name	Interdependent with Salesman identifier
Salesman identifier	Interdependent with Salesman name
Month	Independent
Mileage	Independent
Sales	Independent
Favourite colour	Independent

Moreover, attributes can be discretised in order to introduce ranges in the information. This ability is especially useful to reduce the data dimensionality and simplify the possible values of an attribute. If the dataset is especially big, the discretisation will reduce the computational cost of the clustering algorithm execution [DKS95].

The following is an example of attribute discretisation: given an attribute with possible numeric values from one to a hundred, a tentative discretisation ranges from one to ten, from eleven to twenty, from twenty-one to thirty and so on.

Table 3.6 shows the attributes of our dataset and their discretisation capabilities.

Table 3.6: Dataset attributes discretisation.

Attribute	Discretisation
Nation	Undiscretisable
Salesman name	Discretisable
Salesman identifier	Discretisable
Month	Undiscretisable
Mileage	Discretisable
Sales	Discretisable
Favourite colour	Undiscretisable

The relevance of an attribute is specified by its importance in algorithm executions or clustering results. If an attribute is redundant or contains interdependent information it may not be relevant and shall be removed. There may be other reasons to consider that an attribute is not relevant. Attribute information may not be related to the dataset at all, it may not be significant or it could contain debugging information. Irrelevant attributes shall be removed as they do not provide the problem with any facts.

In our example, as shown in Table 3.7, all the attributes except for favourite colour are relevant. The information about the favourite colour attribute is not relevant because it does not provide with any useful information to the problem resolution.

At this point, we analysed the different properties of the dataset attributes. Now, we are ready to trace an action plan that will define all the data pre-processing step, including one step per attribute characteristic.

In the case of the given travelling salesmen management company dataset example:

- **Description:** helps to understand the meaning and semantics of the attribute.
- **Type:** helps to determine other characteristics.

Table 3.7: Dataset attributes relevance.

Attribute	Relevance
Nation	Relevant
Salesman name	Relevant
Salesman identifier	Relevant
Month	Relevant
Mileage	Relevant
Sales	Relevant
Favourite colour	Irrelevant

- **Granularity:** since all the attributes are unitary, they are not going to be split. Further, mileage and sales can be truncated; we are going to truncate the sales attribute and remove the decimals in order to simplify the values. Finally, we will not truncate the mileage attribute in order to maintain data representativeness.

- **Independence:** independent attributes (nation, month, mileage and sales) are kept. As salesman name and salesman identifier are interdependent, we only need one of them. Therefore, we are going to keep salesman name because its type (string) is more representative than the type of salesman identifier (integer).

- **Discretisation:** Only some of the attributes (salesman name, salesman identifier, mileage and sales) can be discretised. Still, we will not discretise the instances because there are not many instances with different values and the range of the discretisation would not be representative.

- **Relevance:** Favourite colour is irrelevant and, thus, it is not required. The remaining attributes are relevant and are maintained in the dataset.

After pre-processing the source dataset, we obtain the dataset represented in Table 3.8.

Once the data are pre-processed, we need to select the clustering algorithms to execute over these data. These algorithms were described in great detail in Section 2.2.

Table 3.8: Travelling salesmen management company dataset after the pre-processing step.

Nation	Salesman name	Month	Mileage	Sales
Wales	Trevor	January	4,123	40,454 £
Wales	Gwyn	January	3,987	73,263 £
Scotland	Niall	January	5,010	19,438 £
Scotland	Macbeatha	January	4,564	70,430 £
England	Alan	January	3,026	34,065 £
England	John	January	4,794	20,145 £

Depending on the input dataset characteristics, some clustering algorithms could be applied and some others could not. The merge between the clustering algorithms capabilities and the datasets attributes characteristics will tell us if we can apply a certain algorithm or not.

In the example of the travelling salesmen management company, and due to the nature of the dataset, among the studied algorithms we can only execute the following ones:

- Cobweb
- DBScan
- EM
- FarthestFirst
- SimpleKMeans

Although we could carry on all five clustering algorithms, and for the sake of simplicity, from this point on we are only going to use the SimpleKMeans algorithm.

At the same time, each of the algorithms requires parameterisation options to tune their performance up. These parameters are algorithm-specific, differing from each other. Furthermore, similar parameters in different algorithms may generate completely different clustering results.

As aforementioned, in our example we are only going to execute the SimpleKMeans algorithm. Table 3.9 describes the different clustering algorithm execution parameter configurations that are applied to the travelling salesmen management company. We are going to compute three different model executions with different configuration parameters (please see Table 2.1 for a definition of the parameterisation options of this and other algorithms).

Table 3.9: SimpleKMeans execution parameters for the travelling salesmen management company dataset.

Model ID	displayStdDevs	distanceFunction	dontReplaceMissingValues	fastDistanceCalc	maxIterations	numClusters	preserveInstancesOrder	seed
SKM-1	True	EuclideanDistance	True	False	500	2	False	10
SKM-2	True	EuclideanDistance	True	False	500	3	False	10
SKM-3	True	EuclideanDistance	True	False	500	4	False	10

The clustering execution results will generate three separate results grouped in a cluster set. Firstly, it provides a model that represents dataset behaviour (stored as serialised data). Secondly, result information that details cluster readable output. And finally, cluster assignments to the dataset (adds a new column to the dataset indicating the cluster).

We are going to illustrate the clustering execution results with one example of the travelling salesmen management company. Specifically, we are going to represent the results of the Model identifier SimpleKMeans-1 (described in Table 3.9).

Listing 3.1 shows the clustering results output information for the Model ID SKM-1 execution over the dataset.

```
=== Run information ===
Scheme: weka.clusterers.SimpleKMeans -V -M -N 2 -A
   "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Relation: Travelling salesman management company dataset after
   the pre-processing step
Instances: 6
Attributes: 5 (Nation, Salesman name, Month, Mileage, Sales)
Test mode: evaluate on training data
=== Model and evaluation on training set ===
kMeans
======
Number of iterations: 3
Within cluster sum of squared errors: 7.1024397357207
Cluster centroids:
                          Cluster#
Attribute Full Data 0 1
                    (6) (3) (3)
==========================================================
Nation Wales Scotland Wales
  Wales 2 ( 33%) 0 ( 0%) 2 ( 66%)
  Scotland 2 ( 33%) 2 ( 66%) 0 ( 0%)
  England 2 ( 33%) 1 ( 33%) 1 ( 33%)
Salesman name Trevor Niall Trevor
  Trevor 1 ( 16%) 0 ( 0%) 1 ( 33%)
  Gwyn 1 ( 16%) 0 ( 0%) 1 ( 33%)
  Niall 1 ( 16%) 1 ( 33%) 0 ( 0%)
  Macbeatha 1 ( 16%) 1 ( 33%) 0 ( 0%)
  Alan 1 ( 16%) 0 ( 0%) 1 ( 33%)
  John 1 ( 16%) 1 ( 33%) 0 ( 0%)
Month January January January
  January 6 (100%) 3 (100%) 3 (100%)
Mileage 4250.6667 4789.3333 3712
               +/-714.927 +/-223.0366 +/-597.9724
Sales 42965.8333 36671 49260.6667
            +/-23799.5054 +/-29238.2886 +/-21030.6641
Clustered Instances
0 3 ( 50%)
1 3 ( 50%)
```

Listing 3.1: SimpleKMeans Model ID SKM-1 clustering results output information

The model that embodies dataset behaviour is represented as serialised data and may not be printed with understandable characters.

Cluster assignments to the dataset are shown in Table 3.10. This information corresponds to the original dataset, to which a new attribute (cluster assignment) was added. This attribute indicates the cluster memberships of an instance.

Table 3.10: Travelling salesmen management company dataset cluster assignments.

Instance number	Nation	Salesman name	Month	Mileage	Sales	Cluster assignment
0	Wales	Trevor	January	4,123	40,454	cluster1
1	Wales	Gwyn	January	3,987	73,263	cluster1
2	Scotland	Niall	January	5,010	19,438	cluster0
3	Scotland	Macbeatha	January	4,564	70,430	cluster0
4	England	Alan	January	3,026	34,065	cluster1
5	England	John	January	4,794	20,145	cluster0

3.2.2 Best cluster set selection per dataset

The next function in our optimal clustering set selection methodology chooses the best cluster set for each dataset. It needs four new different inputs: the base attribute, a metric, a constraint and a criterion. Please note that this function is also fed with the output of the clustering result function, that is, the cluster set of each algorithm parameter configuration and dataset combination. Figure 3.3 shows in detail the best cluster set selection per dataset step of the optimal clustering model selection methodology.

These cluster sets will be processed according to a base attribute, which gives us the basis for comparison. The definition of a common base attribute for all the cluster sets is especially important when working with different algorithms since each one outputs different information. See Table 2.2 for further details on each algorithms' output. We need to select a base attribute that must be present in all the cluster sets of the previous step.

There could also be more than one base attribute. However, in that case, we would also need extra metrics, constraints and criteria. The attributes may be combined in many ways to generate new parameters.

Figure 3.3: Best cluster set selection per dataset step of the optimal clustering model selection methodology.

In our travelling salesmen management company, for instance, we could use the mileage or the cluster assignment attributes. For the sake of simplicity, we are only going to use one attribute, cluster assignments.

The metric defines the fitness function for measuring. Given the base attribute, it defines how to apply it. It will give us a quantification of cluster set performance.

In the given example, the metric could be the average mileage between the salesmen or the number of instances per cluster which we used. Table 3.11 shows what the cluster sets look like with the metric of number of instances per cluster.

Table 3.11: Travelling salesmen management company number of instances per cluster metric.

Model ID \ Cluster	Cluster0	Cluster1	Cluster2	Cluster3
1	3	3		
2	2	3	1	
3	2	1	1	2

The metric could also include constrains that cluster sets must accomplish. They are base attribute-dependant: if a cluster set does not meet the requirements of the constraint, it will be removed and its data subsequently omitted.

In the given example, we could define a minimum average mileage of 5,000 of miles or a minimum number of two instances per cluster assignment. For the sake of simplicity, however, we are not going to apply any constraints to this example.

The final input is the criterion. It defines how to interpret the metric values and gives us a ranking standard to finally select the best cluster sets for a given dataset.

In our example, if the base attribute was the mileage, we could define the criterion to rank the mileage from the lowest to the highest. Nevertheless, we are applying the number of instances per cluster metric. Table 3.12 shows the travelling salesmen dataset by using the criterion that looks for the biggest difference of number of instances between any two clusters.

In summary, this best cluster set selection per dataset for the given base attribute (cluster assignments), metric (number of instances per cluster), constraint (not defined) and criterion (biggest difference of number of instances between any

Table 3.12: Travelling salesmen management company clustering results for the criterion that looks for the biggest difference of number of instances between any two clusters by clustering algorithm.

Algorithm / Model ID	SimpleKMeans
1	0
2	2
3	1

two clusters) will output the cluster set with Model ID SKM-2. It has the best representativeness for the given dataset, giving the greatest overall likeness of the behaviour.

If we had executed different clustering algorithms, we would have obtained one best cluster set per dataset. Nevertheless, if the example had had more datasets, we would have obtained one best cluster set for each of the datasets. Having more than one dataset-associated best cluster set, we could perform a first *global* validation.

This checkout will give us a first validation of the obtained results. It would compare the relative metric values for each of the datasets best cluster set. The values must be relative because each dataset may have a different number of instances. This fact will unveil potential deviations of the results among the datasets.

In our example, the validation would use the number of instances per cluster and calculate the values of equivalence in percentage, as the absolute number of instances could be different from dataset to dataset.

We could also find out the biggest value of each cluster and the smallest one, and calculate the difference to ease the detection of relevant deviations. For further information, we could calculate other indicators such as mean, median and standard deviation, which would assist us interpreting whether the selected best cluster models are relevant or not.

3.2.3 Operation with the global best cluster set

This function will help us obtain data-independent knowledge applicable to the overall problem. Figure 3.4 shows in detail the best cluster set selection per dataset

step of the optimal clustering model selection methodology.

It receives all the datasets best cluster sets as input data that were globally validated. The characteristics of these cluster sets have interdependency with the operation attribute, or attributes, to be analysed. These attributes define which feature we analyse in order to extract knowledge.

In our example, we could address sales attribute as the operation attribute but miles, nation, or a combination of any of them would also do. Table 3.13 shows the selected dataset best cluster set, corresponding to Model ID SKM-2, analysed by nation attribute.

Table 3.13: Travelling salesmen management company selected dataset best cluster set analysed by nation attribute.

Cluster label / Nation Attribute	Cluster0	Cluster1	Cluster2
Wales	0	1	1
Scotland	2	0	0
England	0	2	0

The operation type will define which business concern we are facing. It will determine the type of information we want to extract. Further, it will be related to the operation attribute, as it determines the information for analysis.

In our example, the operation type could be commercial efficiency or the salesmen's nationality. As a start, we focus on the origin of the salesmen.

At this point, we have one table that represents each dataset best cluster set analysed by operation. Now, we must check the similarity between the cluster sets of each dataset. As the number of instances of each dataset may differ, we must transform the aforementioned table from the absolute number of instances to a relative percentage of membership. Next, we should calculate the maximum difference between percentages in each dataset for the defined operation attribute(s). In this way, we are able to validate the representativeness of the cluster set for cluster-attribute(s).

If the difference is below an acceptable risk defined by the similarity threshold (which is an *a priori* externally given value) we would remove those dataset best

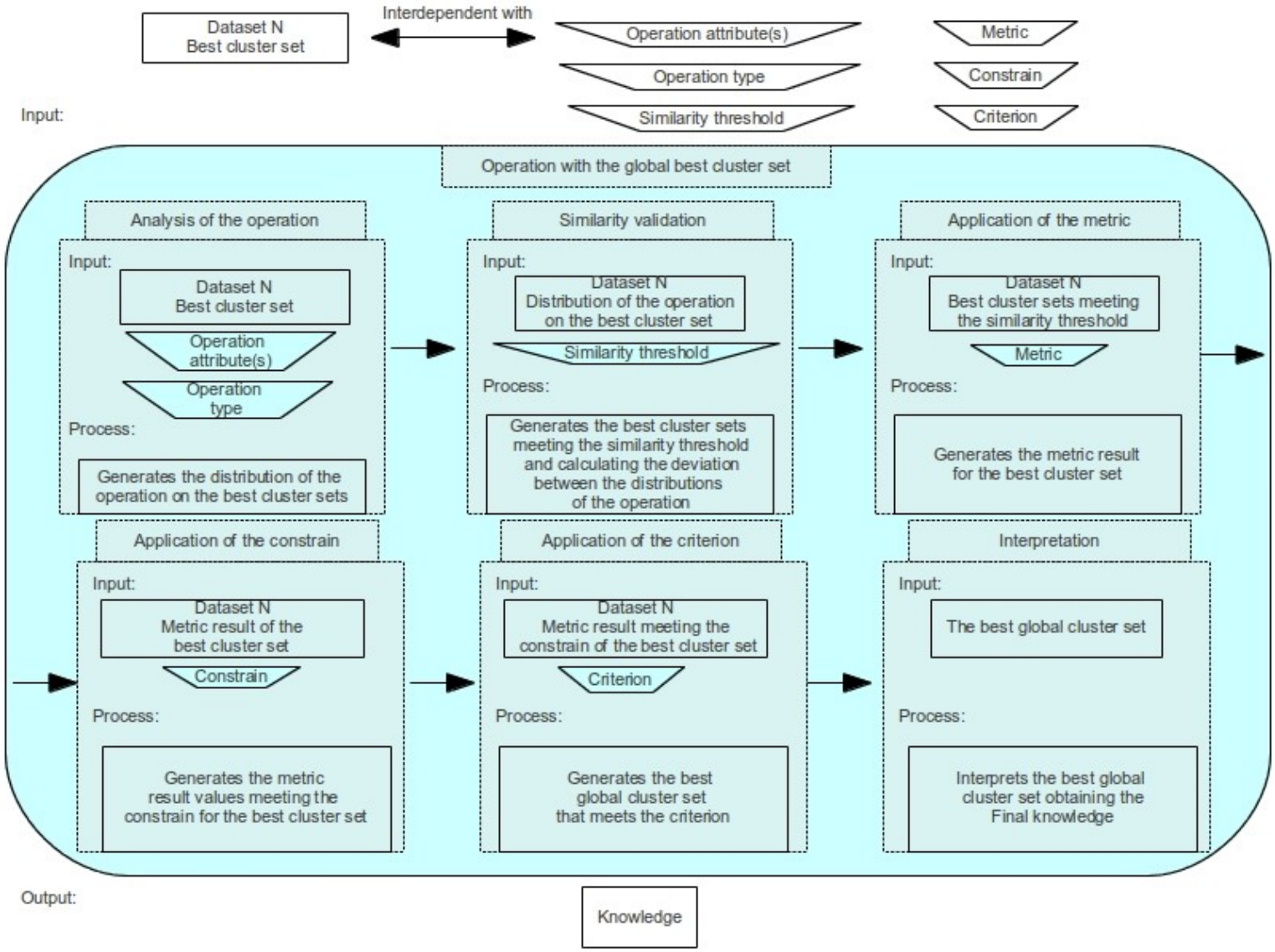

Figure 3.4: Operation with the global best cluster set step of the optimal clustering model selection methodology.

cluster sets since the knowledge extracted from these cluster set would not be representative of all the datasets. In case they are under the similarity threshold (i.e. they are similar), we will accept that the best-cluster sets are indeed the best for each dataset.

In our example, Table 3.14 shows the selected dataset best cluster set analysed by nation attribute, in percentage. Worth mentioning, if we had had more datasets, we would have an extra table for each additional dataset.

Table 3.14: Travelling salesmen management company selected dataset best cluster set analysed by nation attribute, in percentage.

Nation Attribute \ Cluster label	Cluster0	Cluster1	Cluster2
Wales	0	16.66	16.66
Scotland	33.33	0	0
England	0	33.33	0
Total	33.33	50	16.66

Finally, we need to reintroduce the same metric, constraint and criterion that were used in Section 3.2.2.

We would find the best global cluster set of the problem by using just the cluster models that passed the similarity threshold and by applying again the metric, constraint and criterion.

This cluster set will be the one that best represents the overall problem behaviour and from which we can extract the most relevant knowledge.

In our example, if we had had more than one dataset, we would have calculated the maximum deviations between the dataset best cluster sets. As we are only working with one dataset, we will consider this cluster set as our best global cluster set.

Now, this knowledge can be exported to all the tested datasets, algorithm and configuration parameters.

In our example, we could affirm that the clusters with label Cluster0 are Scottish, and those with label Cluster2 are from Wales.

The logic of validation allows us to move between the two limits of dogmatism and skepticism.

Paul Ricoeur

CHAPTER

4

Validation of the methodology

All clustering algorithms in the literature follow different strategies focused on the extraction of a certain kind of knowledge. In this way, their results are usually disparate. This diversity is cased by choosing the optimal clustering algorithm for a definite purpose and having each model tailored to a certain goal. Indeed, model comparison is one of the classic clustering challenges.

This chapter is devoted to tailor and apply the optimal clustering model selection methodology introduced in Chapter 3 and shows how to select the optimal clustering model in a given real problem with the goal on improving decision support systems.

We will first introduce a real-world scenario using a VoIP use-case. We apply the optimal clustering model selection methodology to discover valuable information about the behaviour of VoIP services (more accurately, calls over the *Internet Protocol (IP)* network). This use-case will help us to validate the methodology and apply all its steps.

4.1 VoIP use-case experiment

In this use-case, we have obtained two main contributions. First, we have accomplished the selection of the best clustering algorithm from a given set. We have introduced the use of a problem-independent metric and an algorithm-independent criterion to rank the obtained cluster distributions accordingly. Second, we have clustered the *Call/(Charging) Detail Record (CDR)* data of a VoIP network and obtained the targeted information to perform an optimal physical planning. We have tackled this challenge by comparing well-known clustering algorithms and by selecting the best cluster set following our methodology. Furthermore, we have applied it to analyse the call distribution, helping in this way to asses the trunk-lines infrastructure regarding its use and to improve it, if needed.

4.1.1 About the VoIP

The general banner of *VoIP services* groups a number of technologies aiming at enabling voice communication over an IP network. Services range from a simple call between two users to more complex practices including multiple-user teleconference, call transferring, call-center functionalities, and so on [Goo02].

The data about user identifier, call duration, receivers, call time and type, channels, and so forth. is recorded and can be processed by data mining techniques to obtain valuable knowledge such as *call sinks* (i.e. very frequently called users), *call niche* (i.e. very frequently calling users), diverse calling trends (e.g. call type concentration or peak hours), and so on.

Many applications may take advantage from this information. For example, *congestion prediction* tries to alleviate situations in which the resources are saturated. Depending on the user types and their behaviour, companies may address different *marketing strategies*. In the same way, *network planning* intends to design and organise infrastructures in an optimal way to adequately respond to the system's requirements. We focus on this latter. Specifically, we have access to the CDRs and our objective in this work is to classify the different types of users according to their calling behaviour. We presume this information (e.g. peak hours or regular calling trends) may help us optimise trunk-lines design infrastructure.

Regarding the development of support systems VoIP networks and the use of clustering algorithms, there have been several applications, such as traffic balancing [Rou10], or selecting the most suitable pricing scheme with the provider [FDL09] but they rely on simple techniques not related to knowledge discovery (for instance, the use of a rule engine to detect fraud in VoIP calls [RAPB10b])

4.1.1.1 Optimal clustering model selection in VoIP services

In this use-case, we will apply the methodology introduced in the Chapter 3 to a specific VoIP service. The knowledge that we may obtain from the VoIP datasets are relevant to different applications, such as the ones aforementioned in Section 4.1.1. Nevertheless, among these applications, this use-case focuses on the peculiarities of network planning.

When defining the optimal architecture of telecommunication infrastructure, network planning has to find a suitable trade-off between cost and needs (e.g. service quality or service availability). Inappropriate design leads to bottle-necks, service failures, congestion in rush-hours, and, generally, low customer satisfaction.

Network planning may be accomplished *a priori*, before infrastructures deployment, or *on the fly*, in order to improve them. To this end, we may focus on channels (i.e. used physical trunk-line), call length, call source and destination, context (e.g. land-line, mobile, international) or call time (i.e. time of the call).

Among the attributes of our dataset, which we will analyse in detail in Section 4.1.2, we focus on call behaviour over the day. From the point of view of the service client (say the corporation that contracts a VoIP service), this information is crucial to devise the optimal infrastructure in operation time (i.e. on the fly).

The diverse clustered groups arising from this analysis will show unique trends regarding their activity period, so the network administrator will be able to accordingly improve the performance, for instance by implementing traffic balancing techniques [Rou10], optimising the use of the existing trunk-lines [EEV08] or selecting the most suitable pricing scheme with the provider [FDL09].

From the point of view of the service provider, this knowledge may also be useful to improve revenue management, prevent fraud through the analysis of anomalous deviations in the client behaviour and plan infrastructure re-dimensioning

to answer the needs of clients' QoS or better aim at marketing strategies (e.g. customer-specific campaigns to increase client loyalty) [AY09].

4.1.1.2 VoIP dataset origin

Our experiment studies the behaviour of VoIP users in a certain corporation. More specifically, we use the accounting records generated by a VoIP *Private Branch Exchange* (PBX) with more than 1,300 users as our experimental dataset. A PBX is a telephone exchange that serves a particular corporation; all CDRs belong to the users of this corporation. PBXs are used to connect the internal telephones of a corporation and to connect to the *Public Switched Telephone Network* (PSTN) by means of trunk lines (i.e. bunch of lines shared by many clients). They can include modems, fax machines, and other telephone devices.

Among the possibilities that VoIP PBXs offers, we focus on the CDRs logs. These records are produced when two or more participants communicate over a partial or complete Internet-based voice connection. The call is normally initiated by one of the participants (i.e., the call initiator) and is received by one or more participants (i.e., call recipients). The call can be IP to IP, PSTN to IP, IP to PSTN, mobile to IP or any other possible combination. In each case, the resulting CDR reflects the nature of the call and provides all its details.

We obtained data from the *"Asterisk"* PBX database records of a corporation. Asterisk is a VoIP PBX based on free software [VMSM05]. It allows linked telephones to make regular calls and to connect to other telephone services, including the PSTN and other VoIP services.

We decided to use these data source because it represents a medium-sized corporation with numerous types of users and terminals. To address data privacy concerns, all confidential information was made anonymous, guaranteeing the rights of users under compliance with the existing laws (under the Spanish jurisdiction) on data protection [Org99].

4.1.2 Clustering results in VoIP use-case

As introduced in Section 3.2.1, this phase of the methodology is fed with three different inputs. First, the source datasets, containing the data from which we wish

to obtain the knowledge. Second, the action plans to transform and tune the datasets to optimise the data processing. And, last but not least, the clustering algorithms to execute over these datasets.

Our experiment studies the behaviour of VoIP users in a certain corporation. The source dataset we work with corresponds to all the CDRs recorded in a full year (up to 700,000 entries) from a PBX of a medium-size corporation. Due to our limited computational capabilities, and with the intention of validating the experiment, we split the original dataset in three sub-datasets. The first dataset represents the full month of October containing 48,849 records, the second dataset represents the days between the first and the seventh of October (a subset of the first sample) containing 12,039 records and the third dataset contains the days between the first and the seventh of November containing 9,959 records. The split has no lack of representativeness as the dataset shows that the users' behaviour is similar from week to week [KKZ09], since it will be proved hereafter.

Table 4.1 summarises the splitting of the original dataset in different datasets. It details the represented period of time and the number of records contained. These datasets are the ones that were used in this use-case.

Table 4.1: Datasets of the VoIP use-case.

Dataset identifier	Represented period of time	Number of records
VoIP-1	Full month of October	48,849
VoIP-2	From October 1 to October 7	12,039
VoIP-3	From November 1 to November 7	9,959

All the datasets used in this use-case share the same properties. Therefore, the designed action plan will be unique and applicable to all of them. Next, we study the different properties of them in order to layout the final action plan.

These datasets have various attributes whose meaning and semantics are analysed in Table 4.2.

Table 4.3 shows the attribute types of the VoIP use-case datasets. As we mentioned in Section 3.2.1, the attribute types will be determined by the possibility of using one clustering algorithm or another one, depending on their capabilities. The dataset attributes are date and time, strings and integers.

Table 4.2: VoIP use-case dataset attribute description.

Attribute	Description
calldate	When a CDR is recorded
clid	The Caller*ID using text
src	The Caller*ID using numbers
dst	The destination extension
dcontext	The destination context
channel	The channel used
dstchannel	The destination channel (if appropriate)
lastapp	The last executed application by the PBX (if appropriate)
lastdata	Data arguments of the last executed application
duration	The total time in system in seconds from dial to hangup
billsec	The total time the call is active in seconds from answer to hangup
disposition	The resolution of the call (Answered, No Answer, Busy or Failed)
amaflags	What flags to use (i.e., Documentation, Billing, Ignore and so on)
accountcode	Which account number to use
user field	A user-defined field

Table 4.3: VoIP use-case dataset attribute type.

Attribute	Type
calldate	Datetime
clid	Varchar(80)
src	Varchar(80)
dst	Varchar(80)
dcontext	Varchar(80)
channel	Varchar(80)
dstchannel	Varchar(80)
lastapp	Varchar(80)
lastdata	Varchar(80)
duration	Int(11)
billsec	Int(11)
disposition	Varchar(45)
amaflags	Int(11)
accountcode	Varchar(20)
user field	Varchar(255)

Next, we focus on the granularity of the attributes. The granularity can be unitary or compound, and in some cases it can be truncated. Table 4.4 represents the attribute granularity for this use-case. Most of the attributes are unitary, one is compound, and several can be truncated.

Table 4.4: VoIP use-case dataset attribute granularity.

Attribute	Granularity
calldate	Compound
clid	Unitary
src	Unitary
dst	Unitary
dcontext	Unitary
channel	Unitary, can be truncated
dstchannel	Unitary, can be truncated
lastapp	Unitary
lastdata	Unitary, can be truncated
duration	Unitary
billsec	Unitary
disposition	Unitary
amaflags	Unitary
accountcode	Unitary
user field	Unitary

Table 4.5 is a summary of the datasets analysed by the attributes' independence. Some of the attributes are not clearly independent or interdependent; hence, they are considered independent.

Table 4.6 shows the attributes of this use-case and their discretisation capabilities. It is worth mentioning that the possibility of discretising an attribute or not, in this case, is determined by the attribute type and granularity.

In our use-case, Table 4.7 shows which attributes are relevant, and which ones are not, to our experiment. Irrelevant attributes do not provide us with any data that could potentially turn into knowledge, as it is clearly not related with the problem.

After analysing all the attributes of this use-case datasets, we design an action plan by defining the pre-processing steps.

The action plan for this use-case consists on one step per attribute characteristic as indicated below:

Table 4.5: VoIP use-case dataset attribute independence.

Attribute	Independence
calldate	Independent
clid	Interdependent with src
src	Interdependent with clid
dst	Independent
dcontext	Not clear. Considered independent
channel	Not clear. Considered independent
dstchannel	Not clear. Considered independent
lastapp	Not clear. Considered independent
lastdata	Not clear. Considered independent
duration	Interdependent with billsec
billsec	Interdependent with duration
disposition	Independent
amaflags	Independent
accountcode	Not clear. Considered independent
user field	Independent

Table 4.6: VoIP use-case dataset attribute discretisation.

Attribute	Discretisation
calldate	Discretisable
clid	Undiscretisable
src	Undiscretisable
dst	Undiscretisable
dcontext	Undiscretisable
channel	Undiscretisable
dstchannel	Undiscretisable
lastapp	Undiscretisable
lastdata	Undiscretisable
duration	Discretisable
billsec	Discretisable
disposition	Undiscretisable
amaflags	Undiscretisable
accountcode	Undiscretisable
user field	Undiscretisable

Table 4.7: VoIP use-case dataset attribute relevance.

Attribute	Relevance
calldate	Relevant
clid	Relevant
src	Relevant
dst	Relevant
dcontext	Relevant
channel	Relevant
dstchannel	Relevant
lastapp	Relevant
lastdata	Relevant
duration	Relevant
billsec	Relevant
disposition	Relevant
amaflags	Irrelevant
accountcode	Irrelevant
user field	Irrelevant

- **Description:** indicates the semantic and meaning of the attributes, and in this use-case help us determine other characteristics.
- **Type:** defines the type of the attributes, and, as the previous one, it help us determine other characteristics.
- **Granularity:** in this use-case, most of the attributes are unitary and do not require further transformations. Thought, four of the attributes need to be considered in the action plan as they are compound or can be truncated.
  - calldate: indicates when a CDR is recorded and includes a Datetime type that contains all the timing information. We split the attribute in six new unitary attributes (year, month, day, hour, minute, and second) in order to enhance the access to data value and, hence, attribute representativeness.
  - channel: indicates the channel of the service in a Varchar(80) format. In this attribute, all the instances contain a representative section whose limit is delimited by a *"-"* character. The remaining data are generated

aleatory as an identification stamp and it is not relevant to our experiment and we are going to ignore it.

- dstchannel: indicates the destination channel of the service in a Varchar(80) format. In this attribute, all the instances contain a representative section whose limit is delimited by a "-" character. The remaining data are generated aleatory as a identification stamp and it is not relevant to our experiment and we are going to ignore it.
- lastdata: indicates the last executed application in the PBX recorded in a Varchar(80) format. This attribute could be truncated, however, we are not going to do it as it would generate attribute independence problems without any other benefit.

- **Independence:** in this case, four attributes need to be considered in the action plan as they present interdependency. Furthermore, in some attributes it is hard to determine if they have interdependencies, hence, they are considered independent. This option is less restricted than the other and put us in the safe side. The remaining attributes are independent and are ready to be used.

  - clid and src are interdependent: the values of these attributes may not be identical in content but they refer to the same entity; one of them uses text and the other one numbers. Attribute clid is not used, src is used to represent both, instead.
  - duration and billsec are interdependent: in this case, the interdependency of this attributes is not as strong as in the previous case and their relation hides potentially useful information because some algorithms can not detect the relation itself. Therefore, we consider that creating a new attribute by combining these two will enrich the representativeness of the dataset. This new attribute is calculated by subtracting the billsec to the duration. We name it answerTime and it represents the time lapse between the connection establishment and its answer. Further, we also maintain duration and billsec themselves.

- **Discretisation:** only calldate, duration and billsec can be discretisated. Despite having this option, we do not to discretise the instances because we wanted to maintain data representation as high as possible.

- **Relevance:** amaflags, accountcode and user field are not relevant to our use-case as they do not contain any useful data and are removed. The remaining attributes are relevant to the experiment and are the ones from which we will extract the knowledge.

In this way, we ended up with the following attributes indicated in Table 4.8 for all of the datasets (VoIP-1, VoIP-2 and VoIP-3).

Table 4.8: VoIP use-case dataset attribute description after the pre-processing step.

Attribute	Description
year	Year details of when the CDR is generated
month	Month details of when the CDR is generated
day	Day details of when the CDR is generated
hour	Hour details of when the CDR is generated
minute	Minute details of when the CDR is generated
second	Second details of when the CDR is generated
src	The Caller*ID using numbers
dst	The destination extension
dcontext	The destination context
channel	The channel used
dstchannel	The destination channel (if appropriate)
lastapp	The last executed application by the PBX (if appropriate)
lastdata	Data arguments of the last executed application
duration	The total time in system in seconds from dial to hangup
billsec	The total time the call is active in seconds from answer to hangup
answerTime	Created from $duration - billsec$
disposition	The resolution of the call (Answered, No Anwser, Busy or Failed)

As we introduced in Section 3.2.1, clustering algorithms can be parametrised in many different ways. All these different configuration options are known as *generative mechanisms*, which is a research area itself [MBTP04]. In our experiment, we decided to use the following generative mechanism options:

- Use of different clustering algorithms.
- Use of different initialisation options for each of the algorithms.
- Same subset of objects, instances, for each algorithm.
- Same subset of features, attributes, for each algorithm.
- No use of projection to subspaces.

Due to the nature of the datasets, we apply Cobweb, DBSCAN, EM, FF, OPTICS, and, *Simple K-Means (SKM)*. These algorithms are the ones, from our initial selection, that can handle and process our datasets as introduced in Section 2.2.3 (none of the others is able to handle these data).

The algorithms has been tested with different initialisation values widely used in the literature, as summarised in Table 4.9. Each model defines the required parameters for each algorithm and output a different result. Refer to Table 2.1 for a full definition of the parameterisation capabilities of each algorithm.

With these algorithms, their corresponding configuration parameters and the use-case datasets, after a period of computational processing we obtain one cluster set per model (algorithm and specific parameters combinations). These cluster sets, as introduced previously, contain three different elements:

- A model that represents dataset behaviour.
- Result information that details cluster readable output.
- Cluster assignments to dataset instances.

In order to keep the confidentiality of the dataset, we could not represent all the results of all the models. Nevertheless, the next steps of the methodology will use these results and enlighten us with useful knowledge.

Table 4.9: Algorithm execution parameter values by clustering algorithm in the VoIP use-case.

Cobweb		DBScan		EM			FF		OPTICS			SKM	
Model ID	seed	Model ID	minPoints	Model ID	numClusters	maxIterations	Model ID	numClusters	Model ID	minPoints	.	Model ID	numClusters
Cobweb1	42	DBScan1	6	EM1	1	100	FF1	2	OPTICS1	6		SimpleKMeans1	2
Cobweb2	10	DBScan2	60	EM2	1	1000	FF2	3	OPTICS2	60		SimpleKMeans2	3
		DBScan3	30	EM3	2	100	FF3	4	OPTICS3	30		SimpleKMeans3	4
		DBScan4	15	EM4	3	100	FF4	5	OPTICS4	15		SimpleKMeans4	5
		DBScan5	10	EM5	4	100	FF5	6	OPTICS5	10		SimpleKMeans5	6
		DBScan6	120	EM6	5	100	FF6	7	OPTICS6	120		SimpleKMeans6	7
				EM7	6	100	FF7	8				SimpleKMeans7	8
				EM8	7	100	FF8	9				SimpleKMeans8	9
				EM9	8	100	FF9	10				SimpleKMeans9	10
				EM10	9	100	FF10	11				SimpleKMeans10	11
				EM11	10	100	FF11	12				SimpleKMeans11	12
				EM12	11	100	FF12	13				SimpleKMeans12	13
				EM13	12	100	FF13	14				SimpleKMeans13	14
				EM14	13	100	FF14	15				SimpleKMeans14	15
				EM15	14	100							
				EM16	15	100							

Common execution parameters by algorithm

acuity	1.0	database_Type	SequentialDatabase	debug	False	seed	1	databaseOutput	.	displayStdDevs	True
cutoff	0.0028209479177387815	database_distanceType	EuclidianDataObject	displayModelInOldFormat	False			database_Type	SequentialDatabase	distanceFunction	EuclideanDistance
saveInstanceData	True	Epsilon	0,9	minStdDev	1.0E6			database_distanceType	EuclidianDataObject	dontReplaceMissingValues	True
				seed	100			epsilon	0,9	maxIterations	500
								showGUI	True	preserveInstancesOrder	False
								writeOPTICSresults	True	seed	10

4.1.3 Best cluster set selection per dataset in VoIP services

The next step in our methodology aims at finding the best cluster set per dataset. That is, which cluster set best fits the behaviour of the dataset for a certain problem. As we will define later on, the main problem is defining what the best behaviour itself is.

This function uses four different inputs: the cluster sets per dataset generated in the previous function, a metric that enables the comparison of these cluster sets, a constrain to discriminate non-relevant data, and a criterion to determine the best cluster set for the dataset.

The metric defines how to measure the fitness of the results and allows us to compare the results of cluster set performance in a quantitative and qualitative way. Nevertheless, in order to find a metric, we need to define a base attribute first.

Studying the output of the applied algorithms (e.g. cluster centroids, number of clusters, clustered instances or number of clusters) that were presented in Table 2.2, we only find a common item to all of them. This item is the *clustered instances distribution*. It measures how the instances of the dataset are distributed in the N clusters of each model, that is, how many instances there are in each group. We use this meta-data as the base attribute for our metric since choosing an attribute of the datasets would have been problem-specific.

Hence, for this use-case, the clustered instances distribution is one of the possible base attributes to use. We decide to use it because it does not require further calculations that would increase the complexity of this step.

At this point, we have a value that can help us comparing the clustered algorithm results objectively and determine which one is the best for a certain criterion. Now, we purge the results from those cluster sets not meeting the minimal constrains defined in the problem.

In our problem, we define the following constrains:

- All the instances of each dataset must have a cluster assignment. That is, there can not be unclustered instances.
- The clustering algorithm execution time must be less than a week given the computational and resources constrains we have.

After applying these constrains, we reduce the amount of representative cluster sets. In this way, for the VoIP-1, VoIP-2 and VoIP-3 datasets we remove the cluster sets of the following algorithms:

- **Cobweb:** the algorithms executions took more than the predefined time and were not considered.
- **DBScan:** most of the cluster sets output many unclustered instances.
- **OPTICS:** the algorithms executions could not assign a cluster to the instances. All the instances were unclustered.

Once all the cluster sets of all the datasets that meet the constrain are ready, we can apply a criterion over the metric that will help us picking up the best cluster set that represents the whole problem. It is worth defining the *"best cluster set"*: it will choose the model that best suits the instance distribution we are looking for.

In our problem domain the goal is extracting atypical behaviour. Thus, the best cluster set should contain a big cluster representing normal behaviour, and (possibly) many small ones grouping the atypical values. In a similar way, we apply the *kurtosis risk* method to distinguish between normal behaviour (in a statistical sense) and the abnormal one [MH04]. This method uses the kurtosis that measures, precisely, how *"heavy"* the atypical observations are.

The kurtosis is a measure of the *"peakedness"* of the probability distribution of a real-value random variable. A higher kurtosis means that a bigger part of the variance is the result of infrequent extreme deviations, as opposed to frequent modestly sized deviations [Mar70]. More formally, it measures whether the tail distribution is bigger than the normal one, and is defined as follows:

$$K(x) = \frac{\mu_4}{\sigma^4} - 3, \tag{4.1}$$

where μ_4 denotes the 4^{th} central moment about the mean and σ is the standard deviation.

In our context, taking the number of instances per cluster as the probability distribution to be studied (the base attribute), the kurtosis risk method helps us selecting the most suitable cluster set under these terms (the metric), since the

bigger the kurtosis is, the bigger the difference between the central cluster and the rest.

Generally, if we can devise a theoretical target clusterisation model, we will be able to select the best cluster set by simply measuring the difference of the actual cluster set with the theoretical one. In order to prevent the overfitting that might arise in the result, we may apply methods from the *model selection theory* such as AIC [Aka70] or *Structural Risk Minimization (SRM)* [VC74].

The selection of one criterion or another one should be based on the operation we want to perform with the selected model. In this experiment, the selection criterion will help us detecting which model best represents the sample with the highest differences between instances (i.e. the highest kurtosis). Namely, in which model the instances of the dataset (say, user phone calls) group more differently, so we can appreciate bigger differences in users' behaviour. This fact will ease the task of finding well-defined distinct user groups.

Now, we analyse the metric values for each dataset in each algorithm execution results generated by applying the metric to the base attribute. Table 4.10 represents the VoIP-1 dataset models' results for the criterion analysed by clustering algorithm. Similarly, Table 4.11 and Table 4.12 show the results of the best local cluster model in dataset VoIP-2 and VoIP-3, respectively.

Each dataset presents a different distribution of instances between clusters. This distribution represents the best clustering algorithm model according to our criterion (the one with the highest difference between instances).

We select the models with the biggest difference between clusters (biggest, positive kurtosis value) in each dataset and name the best of the three the best global cluster distribution (one best distribution per datatset). The best local models are highlighted in the results tables; they all correspond to the Farther First algorithm execution with 14 clusters (Model ID FF-14), and the global best was the VoIP1-FF14 of dataset 1. Table 4.13 illustrates this distribution by representing the number of instances per cluster of the best models in each dataset (i.e. the winning FF14 in each case).

Please note that although the best results are the last executions of FF, one could argue that VoIP1-FF15 might obtain even better results, following the rising trend of the FF series. First, we have an external budget constraint limiting the number

Table 4.10: VoIP-1 dataset cluster set results for the analysed metric (Kurtosis) by clustering algorithm model.

Model ID	EM	Model ID	FF	Model ID	SKM
VoIP-1 + EM-1	1	VoIP-1 + FF1	1	VoIP-1 + SKM1	1
VoIP-1 + EM-2	1	VoIP-1 + FF2	1.5	VoIP-1 + SKM2	1.5
VoIP-1 + EM-3	1	VoIP-1 + FF3	2.306566	VoIP-1 + SKM3	1.389741
VoIP-1 + EM-4	1.5	VoIP-1 + FF4	3.169039	VoIP-1 + SKM4	1.459541
VoIP-1 + EM-5	2.074323	VoIP-1 + FF5	4.063443	VoIP-1 + SKM5	2.038804
VoIP-1 + EM-6	1.296791	VoIP-1 + FF6	4.999138	VoIP-1 + SKM6	3.110695
VoIP-1 + EM-7	2.385292	VoIP-1 + FF7	5.681268	VoIP-1 + SKM7	3.226012
VoIP-1 + EM-8	2.372355	VoIP-1 + FF8	6.545245	VoIP-1 + SKM8	3.510349
VoIP-1 + EM-9	2.029568	VoIP-1 + FF9	7.4802	VoIP-1 + SKM9	**3.856364**
VoIP-1 + EM-10	**5.437175**	VoIP-1 + FF10	8.314917	VoIP-1 + SKM10	1.941173
VoIP-1 + EM-11	2.155751	VoIP-1 + FF11	8.173458	VoIP-1 + SKM11	2.379827
VoIP-1 + EM-12	5.387637	VoIP-1 + FF12	8.836912	VoIP-1 + SKM12	2.251627
VoIP-1 + EM-13	4.678836	VoIP-1 + FF13	9.456464	VoIP-1 + SKM13	1.952906
VoIP-1 + EM-14	2.429541	VoIP-1 + FF14	**10.10474**	VoIP-1 + SKM14	1.801146
VoIP-1 + EM-15	1.482438				
VoIP-1 + EM-16	1.711231				

Table 4.11: VoIP-2 dataset cluster set results for the analysed metric (Kurtosis) by clustering algorithm model.

Model ID	EM	Model ID	FF	Model ID	SKM
VoIP-2 + EM-1	NaN	VoIP-2 + FF1	1	VoIP-2 + SKM1	1
VoIP-2 + EM-2	NaN	VoIP-2 + FF2	1.5	VoIP-2 + SKM2	1.5
VoIP-2 + EM-3	1	VoIP-2 + FF3	2.228944	VoIP-2 + SKM3	2.276468
VoIP-2 + EM-4	1.5	VoIP-2 + FF4	3.03195	VoIP-2 + SKM4	2.470783
VoIP-2 + EM-5	2.041038	VoIP-2 + FF5	3.892921	VoIP-2 + SKM5	3.170883
VoIP-2 + EM-6	1.364078	VoIP-2 + FF6	4.503747	VoIP-2 + SKM6	**3.370352**
VoIP-2 + EM-7	1.700368	VoIP-2 + FF7	5.015808	VoIP-2 + SKM7	3.063124
VoIP-2 + EM-8	2.417772	VoIP-2 + FF8	5.876241	VoIP-2 + SKM8	1.850531
VoIP-2 + EM-9	2.677382	VoIP-2 + FF9	6.533938	VoIP-2 + SKM9	1.581167
VoIP-2 + EM-10	1.61199	VoIP-2 + FF10	6.890177	VoIP-2 + SKM10	1.651486
VoIP-2 + EM-11	2.72228	VoIP-2 + FF11	7.711404	VoIP-2 + SKM11	2.151566
VoIP-2 + EM-12	**3.800068**	VoIP-2 + FF12	7.700427	VoIP-2 + SKM12	1.56243
VoIP-2 + EM-13	2.891272	VoIP-2 + FF13	7.974276	VoIP-2 + SKM13	1.999453
VoIP-2 + EM-14	3.096368	VoIP-2 + FF14	**8.20937**	VoIP-2 + SKM14	2.325274
VoIP-2 + EM-15	2.198781				
VoIP-2 + EM-16	2.361976				

Table 4.12: VoIP-3 dataset cluster set results for the analysed metric (Kurtosis) by clustering algorithm model.

Model ID	EM	Model ID	FF	Model ID	SKM
VoIP-3 + EM-1	NaN	VoIP-3 + FF1	1	VoIP-3 + SKM1	1
VoIP-3 + EM-2	NaN	VoIP-3 + FF2	1.5	VoIP-3 + SKM2	1.5
VoIP-3 + EM-3	1	VoIP-3 + FF3	2.314202	VoIP-3 + SKM3	1.302842
VoIP-3 + EM-4	1.5	VoIP-3 + FF4	3.184467	VoIP-3 + SKM4	1.330627
VoIP-3 + EM-5	2.086875	VoIP-3 + FF5	4.079792	VoIP-3 + SKM5	1.134665
VoIP-3 + EM-6	1.278842	VoIP-3 + FF6	5.026794	VoIP-3 + SKM6	1.207678
VoIP-3 + EM-7	1.605531	VoIP-3 + FF7	5.79056	VoIP-3 + SKM7	1.389523
VoIP-3 + EM-8	3.024639	VoIP-3 + FF8	6.677856	VoIP-3 + SKM8	1.846808
VoIP-3 + EM-9	2.818391	VoIP-3 + FF9	7.5965	VoIP-3 + SKM9	3.398993
VoIP-3 + EM-10	2.920922	VoIP-3 + FF10	8.530448	VoIP-3 + SKM10	3.31325
VoIP-3 + EM-11	2.555686	VoIP-3 + FF11	9.462954	VoIP-3 + SKM11	3.183834
VoIP-3 + EM-12	2.016453	VoIP-3 + FF12	8.863714	VoIP-3 + SKM12	1.744464
VoIP-3 + EM-13	2.917483	VoIP-3 + FF13	9.351196	VoIP-3 + SKM13	3.562417
VoIP-3 + EM-14	2.732966	VoIP-3 + FF14	**9.842789**	VoIP-3 + SKM14	**5.536112**
VoIP-3 + EM-15	2.8506				
VoIP-3 + EM-16	**3.744771**				

Table 4.13: Best local model clustered instance distribution per cluster for each dataset.

Cluster label	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
VoIP-1	17,253	982	119	1,965	2,701	2,425	1,621	4,927	2,522	2,023	890	5,816	2,347	1,539	1,719
VoIP-2	3,685	6	193	680	281	112	1,364	1,091	478	205	825	608	1,308	529	674
VoIP-3	3,296	222	402	374	202	563	505	1,016	383	494	468	480	1,194	213	147

of trunk lines up to 14 groups (say 14 lines). Second, we execute FF again (with $numClusters = 36$ and $seed = 1$) and, applying the same base attribute, metric and criterion, we obtain an absolute 9.724404 kurtosis value. Hence, the result is worse than the best model and the general results in this case seem not to degenerate (losing it representativeness).

In summary, this best cluster set selection per dataset for the given base attribute (cluster assignments), metric (kurtosis), constraints (all instances assigned and limited algorithm execution time) and criterion (biggest kurtosis) choices cluster set with Model identifier VoIP-1+FF14 for the first dataset, VoIP-2+FF14 for the second one and VoIP-3+FF14 for the third. These cluster sets show the best representativeness for their dataset, giving the greatest overall likeness of the behaviour (it best represents the problem).

At this point, we already have the best cluster sets for the criterion proposed in the three different datasets. In order to validate our model, we must compare the results of the best models between them. In this way, we will show that the representation of the cluster sets is homogeneous.

Table 4.14 details the validation of the results by comparing the instances distribution per cluster for each dataset in percentage. In order to validate them we have to analyse each cluster's instance distribution in Table 4.13, also studying the difference of the results among the datasets. To this end, we took the biggest value of each cluster, and the smallest one and calculated the difference among clusters by obtaining the highest difference between the results of each dataset.

This value (i.e. the difference) gives us a reference to measure the similarity between the best models of each dataset. In this value is smaller than a certain threshold, the results can be considered valid. In this experiment, we set this value at a 10%, as it represents a maximum difference of the ten percent of the total amount of instances in at least one of the clusters of the datasets.

In order to get further validation information, we also calculated the average absolute deviation of values from their mean *(AVEDEV)*, the median of a set of numbers *(MEDIAN)*, and the sample standard deviation of the arguments *(STDEV)*. Besides, the maximum differences of these functions are smaller than a relative 1%, portraying the big similarity between models' behaviour.

As the validation is under the defined threshold, we can consider that the models behave similarly enough and that they represent similar behaviour; hence, the best clustering model selection methodology proven right. In conclusion, these results indicate that any of the obtained best models can represent the rest of the datasets behaviour with an acceptable performance.

Table 4.14: Validation of the results by comparing the instances distribution per cluster for each dataset, in percentages.

Dataset \ Cluster label	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	AVEDEV	MEDIAN	STDEV
VoIP-1	35.32	2.01	0.24	4.02	5.53	4.96	3.32	10.09	5.16	4.14	1.82	11.91	4.8	3.15	3.52	4.97	4.14	8.47
VoIP-2	30.61	0.05	1.6	5.65	2.33	0.93	11.33	9.06	3.97	1.7	6.85	5.05	10.86	4.39	5.6	4.72	5.05	7.47
VoIP-3	33.1	2.23	4.04	3.76	2.03	5.65	5.07	10.2	3.85	4.96	4.7	4.82	11.99	2.14	1.48	4.7	4.7	7.86
Maximun	35.32	2.23	4.04	5.65	5.53	5.65	11.33	10.2	5.16	4.96	6.85	11.91	11.99	4.39	5.6	4.97	5.05	8.47
Minimun	30.61	0.05	0.24	3.76	2.03	0.93	3.32	9.06	3.85	1.7	1.82	4.82	4.8	2.14	1.48	4.7	4.14	7.47
Difference	4.71	2.18	3.79	1.89	3.5	4.72	8.01	1.14	1.32	3.26	5.03	7.09	7.18	2.26	4.12	0.27	0.91	0.99

4.1.4 Operation with the global best cluster set in VoIP services

The last step in our methodology aims at illustrating how to make use of the results obtained in the previous phases. They will allow us to obtain data-independent knowledge that may be used in the overall problem.

The operation receive seven different inputs: the best cluster set for each of the datasets obtained in the previous function, one or more operation attributes that define the environment of our analysis, the operation type determining the problem faced, a similarity threshold validating the global best cluster set for the current specific problem, a metric, a set of constraint's and a number of criteria that may be required to re-validate the results if the similarity threshold is not exceed.

We know from Section 4.1.3 that the best cluster sets of each dataset are the following:

- VoIP-1 + FF14 for the first dataset.
- VoIP-2 + FF14 for the second dataset.
- VoIP-3 + FF14 for the third dataset.

And that the global best cluster set corresponds to VoIP-1 + FF14. Nevertheless, we are also going to check if this cluster set represents the whole problem for this specific case.

In order to extract knowledge from these cluster sets, we need to define which operation attribute we are using. That is, which of the cluster sets attribute are we analysing. Specifically, in this use-case we are going to analyse the distribution of the service usage hour as it has been proved to be one of the most representative attributes for network planning [KKZ09] (please note that it shows when the CDR is generated).

This attribute is related to the operation type we planned for this use-case: we want to extract knowledge about the time calls are made.

Now, we will introduce a set of tables showing the instance distribution for this operation attribute and type. Table 4.15 represents instance distribution percentages among the clusters analysed by the attribute hour for the VoIP-1 dataset, table 4.16 for the VoIP-2 dataset and table 4.17 for the VoIP-3 dataset.

Table 4.15: Instance distribution percentage among the clusters (C) analysed by the attribute (A) hour for the VoIP-1 dataset.

(A)/(C)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	AVEDEV	MEDIAN	STDEV
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0.03	0	0	0	0.05	0	0	0	0	0	0	0	0	0.02	0	0.01	0	0.02
2	0.03	0	0	0	0.1	0	0	0	0	0	0	0	0	0.03	0	0.02	0	0.03
3	0.05	0	0	0	0.21	0	0	0	0	0	0	0	0	0.01	0	0.03	0	0.05
4	0.1	0	0	0	0.04	0	0	0	0	0	0	0	0	0.02	0	0.02	0	0.03
5	0	0	0	0	0.05	0	0	0	0	0	0	0	0	0.01	0	0.01	0	0.01
6	0.11	0	0	0.02	0.01	0	0	0.01	0	0	0	0	0	0.03	0	0.02	0	0.03
7	0.1	0	0	0.04	0.11	0.02	0	0.03	0.01	0	0	0.01	0	0.03	0.01	0.03	0.01	0.03
8	0.88	0.07	0	0.2	0.28	0.14	0.16	0.43	0.08	0.08	0.06	0.41	0.08	0.16	0.18	0.15	0.16	0.22
9	3.85	0.26	0.07	0.35	0.69	0.46	0.5	1.05	0.71	0.36	0.13	1.38	0.36	0.35	0.46	0.54	0.46	0.93
10	4.47	0.21	0.03	0.46	0.64	0.55	0.5	1.13	0.78	0.44	0.25	1.64	0.55	0.3	0.48	0.63	0.5	1.08
11	4.91	0.29	0.06	0.57	0.57	0.67	0.57	1.32	0.8	0.47	0.23	1.66	0.61	0.4	0.52	0.69	0.57	1.18
12	4.95	0.35	0.03	0.6	0.58	0.67	0.42	1.18	0.73	0.52	0.24	1.66	0.64	0.36	0.53	0.68	0.58	1.19
13	3.06	0.12	0.01	0.35	0.41	0.34	0.15	0.77	0.48	0.44	0.18	1.04	0.38	0.23	0.23	0.43	0.35	0.74
14	1.34	0.07	0	0.23	0.29	0.32	0.11	0.51	0.27	0.23	0.11	0.46	0.18	0.18	0.15	0.19	0.23	0.32
15	2.63	0.15	0.01	0.24	0.47	0.51	0.27	0.84	0.39	0.39	0.1	0.89	0.29	0.28	0.25	0.38	0.29	0.63
16	3.15	0.19	0.03	0.25	0.36	0.46	0.29	0.98	0.35	0.4	0.12	1.04	0.54	0.22	0.27	0.46	0.35	0.77
17	2.8	0.2	0.01	0.29	0.33	0.41	0.27	0.92	0.28	0.38	0.15	0.96	0.55	0.23	0.22	0.41	0.29	0.68
18	1.33	0.06	0.01	0.2	0.13	0.21	0.05	0.42	0.21	0.21	0.1	0.47	0.28	0.09	0.12	0.2	0.2	0.32
19	0.51	0.02	0	0.09	0.08	0.14	0.01	0.22	0.05	0.1	0.04	0.18	0.13	0.07	0.07	0.08	0.08	0.13
20	0.36	0.01	0	0.1	0.07	0.03	0.01	0.12	0.02	0.05	0.09	0.1	0.08	0.05	0.02	0.05	0.05	0.09
21	0.31	0	0	0.03	0.06	0.02	0.01	0.05	0	0.02	0.01	0	0.05	0.05	0.01	0.04	0.02	0.08
22	0.33	0	0	0.02	0.01	0.01	0	0.09	0	0.04	0	0	0.07	0.02	0	0.05	0.01	0.08
23	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SUM	35.32	2.01	0.24	4.02	5.53	4.96	3.32	10.09	5.16	4.14	1.82	11.91	4.8	3.15	3.52	4.97	4.14	8.47

Table 4.16: Instance distribution percentage among the clusters (C) analysed by the attribute (A) hour for the VoIP-2 dataset.

(A)/(C)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	AVEDEV	MEDIAN	STDEV
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.03
2	0.07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.02
3	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.04	0	0.08
4	0.37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0	0.1
5	0.06	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0.01	0	0.02
6	0.09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.02
7	0.17	0	0.01	0	0.01	0	0	0.07	0	0	0.02	0.01	0.02	0	0.02	0.03	0.01	0.05
8	0.91	0	0.07	0.13	0.02	0.02	0.22	0.36	0.07	0.06	0.1	0.19	0.22	0.1	0.11	0.14	0.1	0.23
9	3.16	0.01	0.12	0.76	0.26	0.08	1.56	0.89	0.44	0.17	0.61	0.57	1.23	0.37	0.61	0.53	0.57	0.8
10	3.42	0.01	0.17	0.86	0.27	0.09	1.45	1.15	0.51	0.22	0.81	0.62	1.25	0.37	0.82	0.55	0.62	0.85
11	4.63	0.01	0.23	0.93	0.33	0.12	1.66	1.45	0.52	0.39	0.88	0.76	1.33	0.59	0.92	0.69	0.76	1.12
12	3.79	0	0.17	0.69	0.35	0.19	1.64	1.25	0.37	0.2	0.73	0.82	1.43	0.68	0.66	0.62	0.68	0.94
13	3.16	0	0.16	0.47	0.24	0.07	0.96	0.62	0.27	0.17	0.67	0.32	0.69	0.39	0.57	0.42	0.39	0.76
14	1.58	0	0.08	0.25	0.02	0	0.42	0.43	0.14	0.11	0.29	0.27	0.58	0.17	0.14	0.24	0.17	0.39
15	1.98	0.01	0.09	0.42	0.15	0.07	0.89	0.68	0.3	0.15	0.6	0.37	0.98	0.48	0.48	0.34	0.42	0.5
16	2.26	0.02	0.21	0.48	0.25	0.16	0.91	0.75	0.35	0.07	0.76	0.44	1.16	0.37	0.48	0.39	0.44	0.56
17	2.18	0	0.09	0.3	0.25	0.12	0.99	0.7	0.32	0.07	0.65	0.38	1.11	0.53	0.44	0.39	0.38	0.56
18	1.07	0	0.07	0.25	0.16	0.02	0.42	0.39	0.2	0.05	0.39	0.12	0.37	0.25	0.27	0.17	0.25	0.26
19	0.54	0	0.04	0.1	0.02	0	0.14	0.21	0.12	0.02	0.11	0.07	0.2	0.06	0.04	0.09	0.07	0.14
20	0.18	0	0.05	0.01	0	0	0.07	0.07	0.11	0.02	0.12	0.03	0.07	0	0.02	0.04	0.03	0.05
21	0.37	0	0.02	0	0	0	0	0.02	0.05	0.01	0.07	0.06	0.07	0.02	0	0.05	0.02	0.09
22	0.21	0	0	0.02	0	0	0	0.01	0.21	0.01	0.03	0.01	0.14	0.01	0	0.06	0.01	0.08
23	0	0	0	0	0	0	0	0.02	0	0	0.01	0	0.01	0	0	0	0	0.01
SUM	30.61	0.05	1.6	5.65	2.33	0.93	11.33	9.06	3.97	1.7	6.85	5.05	10.86	4.39	5.6	4.72	5.05	7.47

Table 4.17: Instance distribution percentage among the clusters (C) analysed by the attribute (A) hour for the VoIP-3 dataset.

(A)/(C)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	AVEDEV	MEDIAN	STDEV
0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0
1	0.03	0	0	0	0	0	0	0	0	0	0	0.08	0.01	0	0	0.01	0	0.02
2	0.03	0	0	0	0	0	0	0	0	0	0	0.28	0	0	0	0.04	0	0.07
3	0.04	0	0	0	0	0	0	0	0	0	0	0.06	0.04	0	0	0.01	0	0.02
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0.02	0	0	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0.01
6	0.01	0	0	0	0	0	0	0	0	0.01	0	0.06	0	0	0	0.01	0	0.02
7	0.16	0	0	0	0	0	0	0.04	0	0.05	0.07	0.09	0	0	0	0.04	0	0.05
8	1.04	0.05	0.12	0.03	0.02	0.09	0.12	0.4	0.01	0.15	0.29	0.24	0.27	0.02	0	0.17	0.12	0.27
9	3.49	0.16	0.35	0.25	0.16	0.26	0.58	0.94	0.25	0.51	0.39	0.52	1.39	0.18	0.11	0.52	0.35	0.86
10	4.7	0.29	0.69	0.49	0.23	0.6	0.68	1.41	0.48	0.43	0.74	0.59	1.64	0.38	0.16	0.67	0.59	1.13
11	4.77	0.23	0.34	0.43	0.2	0.74	0.71	1.42	0.67	0.6	0.54	0.53	1.41	0.53	0.06	0.66	0.54	1.14
12	4.23	0.15	0.89	0.48	0.18	0.57	0.78	1.57	0.67	0.64	0.52	0.45	1.84	0.38	0.12	0.66	0.57	1.04
13	2.19	0.13	0.32	0.38	0.13	0.52	0.32	0.76	0.27	0.47	0.25	0.42	0.94	0.21	0.12	0.32	0.32	0.52
14	1.43	0.07	0.17	0.06	0.2	0.26	0.1	0.3	0.01	0.43	0.34	0.23	0.52	0.05	0.13	0.21	0.2	0.35
15	2.69	0.14	0.29	0.3	0.16	0.52	0.41	0.82	0.32	0.4	0.35	0.42	1.01	0.12	0.08	0.39	0.35	0.65
16	3.4	0.17	0.28	0.41	0.27	0.55	0.43	1.02	0.54	0.33	0.35	0.22	1.14	0.1	0.12	0.49	0.35	0.83
17	2.95	0.12	0.27	0.41	0.16	0.52	0.51	0.79	0.55	0.33	0.33	0.25	1	0.09	0.12	0.41	0.33	0.71
18	1.09	0.13	0.16	0.3	0.16	0.38	0.19	0.56	0.05	0.21	0.24	0.19	0.23	0.04	0.07	0.17	0.19	0.26
19	0.34	0.04	0.08	0.09	0.08	0.27	0.05	0.08	0	0.22	0.16	0.07	0.31	0.03	0.04	0.09	0.08	0.11
20	0.18	0.2	0.06	0.1	0.01	0.09	0.06	0.05	0.01	0.06	0.04	0.08	0.06	0	0.15	0.05	0.06	0.06
21	0.17	0.1	0	0.01	0.06	0.08	0.06	0.02	0	0.1	0.04	0	0.15	0	0.1	0.05	0.06	0.06
22	0.12	0.23	0	0	0	0.17	0.05	0.01	0	0	0.01	0.01	0.01	0	0.09	0.06	0.01	0.07
23	0	0.01	0	0	0	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0
SUM	33.1	2.23	4.04	3.76	2.03	5.65	5.07	10.2	3.85	4.96	4.7	4.82	11.99	2.14	1.48	4.7	4.7	7.86

The next step, as depicted in Table 4.18, studies the maximum difference between the percentage among the clusters analysed by the attribute hour between all the datasets.

Table 4.18: Maximum difference of the percentage among the clusters (C) analysed by the attribute (A) hour between all the datasets.

(A)/(C)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	AVEDEV	MEDIAN	STDEV
0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0
1	0.11	0	0	0	0.05	0	0	0	0	0	0	0.08	0.01	0.02	0	0.03	0	0.03
2	0.07	0	0	0	0.1	0	0	0	0	0	0	0.28	0	0.03	0	0.05	0	0.08
3	0.3	0	0	0	0.21	0	0	0	0	0	0	0.06	0.04	0.01	0	0.06	0	0.09
4	0.37	0	0	0	0.04	0	0	0	0	0	0	0	0	0.02	0	0.05	0	0.1
5	0.06	0	0	0	0.05	0	0	0	0	0	0.02	0	0.01	0.01	0	0.01	0	0.02
6	0.11	0	0	0.02	0.01	0	0	0.01	0	0.01	0	0.06	0	0.03	0	0.02	0	0.03
7	0.17	0	0.01	0.04	0.11	0.02	0	0.07	0.01	0.05	0.07	0.09	0.02	0.03	0.02	0.04	0.03	0.05
8	1.04	0.07	0.12	0.2	0.28	0.14	0.22	0.43	0.08	0.15	0.29	0.41	0.27	0.16	0.18	0.15	0.2	0.24
9	3.85	0.26	0.35	0.76	0.69	0.46	1.56	1.05	0.71	0.51	0.61	1.38	1.39	0.37	0.61	0.58	0.69	0.89
10	4.7	0.29	0.69	0.86	0.64	0.6	1.45	1.41	0.78	0.44	0.81	1.64	1.64	0.38	0.82	0.68	0.81	1.08
11	4.91	0.29	0.34	0.93	0.57	0.74	1.66	1.45	0.8	0.6	0.88	1.66	1.41	0.59	0.92	0.69	0.88	1.12
12	4.95	0.35	0.89	0.69	0.58	0.67	1.64	1.57	0.73	0.64	0.73	1.66	1.84	0.68	0.66	0.74	0.73	1.14
13	3.16	0.13	0.32	0.47	0.41	0.52	0.96	0.77	0.48	0.47	0.67	1.04	0.94	0.39	0.57	0.41	0.52	0.71
14	1.58	0.07	0.17	0.25	0.29	0.32	0.42	0.51	0.27	0.43	0.34	0.46	0.58	0.18	0.15	0.21	0.32	0.36
15	2.69	0.15	0.29	0.42	0.47	0.52	0.89	0.84	0.39	0.4	0.6	0.89	1.01	0.48	0.48	0.38	0.48	0.6
16	3.4	0.19	0.28	0.48	0.36	0.55	0.91	1.02	0.54	0.4	0.76	1.04	1.16	0.37	0.48	0.47	0.54	0.78
17	2.95	0.2	0.27	0.41	0.33	0.52	0.99	0.92	0.55	0.38	0.65	0.96	1.11	0.53	0.44	0.43	0.53	0.67
18	1.33	0.13	0.16	0.3	0.16	0.38	0.42	0.56	0.21	0.21	0.39	0.47	0.37	0.25	0.27	0.17	0.3	0.29
19	0.54	0.04	0.08	0.1	0.08	0.27	0.14	0.22	0.12	0.22	0.16	0.18	0.31	0.07	0.07	0.09	0.14	0.13
20	0.36	0.2	0.06	0.1	0.07	0.09	0.07	0.12	0.11	0.06	0.12	0.1	0.08	0.05	0.15	0.05	0.1	0.08
21	0.37	0.1	0.02	0.03	0.06	0.08	0.06	0.05	0.05	0.1	0.07	0.06	0.15	0.05	0.1	0.05	0.06	0.08
22	0.33	0.23	0	0.02	0.01	0.17	0.05	0.09	0.21	0.04	0.03	0.01	0.14	0.02	0.09	0.08	0.05	0.1
23	0.01	0.01	0	0	0	0.01	0	0.02	0	0	0.01	0	0.01	0	0	0.01	0	0.01
SUM	37.36	2.73	4.07	6.06	5.56	6.08	11.44	11.12	6.03	5.13	7.22	12.54	12.48	4.74	6.03	5.17	6.06	8.39

Similarly, as introduced in Table 4.19, we also calculate the minimum difference between the percentage among the clusters analysed by the attribute hour between all the datasets.

Table 4.20 presents the difference between instance distribution percentage among the clusters analysed by the attribute hour, representing the relative likelihood that a call shows to belong to a certain cluster in each dataset. If the operation with the model was based on other attributes, this deviation could also be analysed for them.

In other words, these data defines how well the instances are distributed in the moment the event of the VoIP service is performed. Furthermore, the highest deviation represents a relative 1.27% of difference, meaning that the representativeness of this attribute (hour) has a very small error (i.e. is very representative).

Table 4.19: Minimum difference of the percentage among the clusters (C) analysed by the attribute (A) hour between all the datasets.

(A)/(C)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	AVEDEV	MEDIAN	STDEV
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01
2	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01
3	0.04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0.1	0	0	0	0	0	0	0.03	0	0	0	0.01	0	0	0	0.01	0	0.03
8	0.88	0	0	0.03	0.02	0.02	0.12	0.36	0.01	0.06	0.06	0.19	0.08	0.02	0	0.14	0.03	0.23
9	3.16	0.01	0.07	0.25	0.16	0.08	0.5	0.89	0.25	0.17	0.13	0.52	0.36	0.18	0.11	0.43	0.18	0.78
10	3.42	0.01	0.03	0.46	0.23	0.09	0.5	1.13	0.48	0.22	0.25	0.59	0.55	0.3	0.16	0.46	0.3	0.84
11	4.63	0.01	0.06	0.43	0.2	0.12	0.57	1.32	0.52	0.39	0.23	0.53	0.61	0.4	0.06	0.62	0.4	1.14
12	3.79	0	0.03	0.48	0.18	0.19	0.42	1.18	0.37	0.2	0.24	0.45	0.64	0.36	0.12	0.52	0.36	0.93
13	2.19	0	0.01	0.35	0.13	0.07	0.15	0.62	0.27	0.17	0.18	0.32	0.38	0.21	0.12	0.29	0.18	0.53
14	1.34	0	0	0.06	0.02	0	0.1	0.3	0.01	0.11	0.11	0.23	0.18	0.05	0.13	0.18	0.1	0.33
15	1.98	0.01	0.01	0.24	0.15	0.07	0.27	0.68	0.3	0.15	0.1	0.37	0.29	0.12	0.08	0.28	0.15	0.49
16	2.26	0.02	0.03	0.25	0.25	0.16	0.29	0.75	0.35	0.07	0.12	0.22	0.54	0.1	0.12	0.33	0.22	0.56
17	2.18	0	0.01	0.29	0.16	0.12	0.27	0.7	0.28	0.07	0.15	0.25	0.55	0.09	0.12	0.32	0.16	0.54
18	1.07	0	0.01	0.2	0.13	0.02	0.05	0.39	0.05	0.05	0.1	0.12	0.23	0.04	0.07	0.16	0.07	0.27
19	0.34	0	0	0.09	0.02	0	0.01	0.08	0	0.02	0.04	0.07	0.13	0.03	0.04	0.06	0.03	0.09
20	0.18	0	0	0.01	0	0	0.01	0.05	0.01	0.02	0.04	0.03	0.06	0	0.02	0.03	0.01	0.05
21	0.17	0	0	0	0	0	0	0.02	0	0.01	0.01	0	0.05	0	0	0.03	0	0.04
22	0.12	0	0	0	0	0	0	0.01	0	0	0	0	0.01	0	0	0.01	0	0.03
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SUM	30.61	0.05	0.24	3.76	2.03	0.93	3.32	9.06	3.85	1.7	1.82	4.82	4.8	2.14	1.48	4.06	2.14	7.52

Table 4.20: Difference of the percentage among the clusters (C) analysed by the attribute (A) hour between all the datasets.

(A)/(C)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	AVEDEV	MEDIAN	STDEV
0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0
1	0.08	0	0	0	0.05	0	0	0	0	0	0	0.08	0.01	0.02	0	0.02	0	0.03
2	0.04	0	0	0	0.1	0	0	0	0	0	0	0.28	0	0.03	0	0.04	0	0.07
3	0.26	0	0	0	0.21	0	0	0	0	0	0	0.06	0.04	0.01	0	0.05	0	0.08
4	0.37	0	0	0	0.04	0	0	0	0	0	0	0	0	0.02	0	0.05	0	0.1
5	0.05	0	0	0	0.05	0	0	0	0	0	0.02	0	0.01	0.01	0	0.01	0	0.02
6	0.1	0	0	0.02	0.01	0	0	0.01	0	0.01	0	0.06	0	0.03	0	0.02	0	0.03
7	0.08	0	0.01	0.04	0.11	0.02	0	0.03	0.01	0.05	0.07	0.08	0.02	0.03	0.02	0.03	0.03	0.03
8	0.16	0.07	0.12	0.17	0.26	0.12	0.1	0.07	0.07	0.09	0.23	0.22	0.19	0.14	0.18	0.05	0.14	0.06
9	0.69	0.25	0.29	0.5	0.53	0.38	1.06	0.16	0.46	0.35	0.48	0.86	1.02	0.19	0.5	0.21	0.48	0.28
10	1.28	0.28	0.66	0.4	0.41	0.51	0.95	0.27	0.3	0.22	0.57	1.04	1.09	0.08	0.66	0.29	0.51	0.36
11	0.27	0.29	0.29	0.5	0.37	0.63	1.09	0.13	0.28	0.21	0.65	1.13	0.79	0.19	0.86	0.28	0.37	0.33
12	1.16	0.35	0.87	0.21	0.4	0.48	1.22	0.38	0.36	0.44	0.49	1.21	1.19	0.32	0.54	0.33	0.48	0.37
13	0.97	0.13	0.32	0.12	0.28	0.45	0.82	0.14	0.21	0.31	0.49	0.73	0.56	0.18	0.45	0.21	0.32	0.26
14	0.24	0.07	0.17	0.19	0.26	0.32	0.31	0.2	0.26	0.32	0.23	0.23	0.4	0.13	0.02	0.08	0.23	0.1
15	0.71	0.14	0.28	0.17	0.32	0.46	0.62	0.16	0.09	0.25	0.5	0.52	0.72	0.36	0.4	0.17	0.36	0.21
16	1.14	0.18	0.25	0.24	0.11	0.39	0.62	0.28	0.19	0.32	0.63	0.82	0.63	0.27	0.36	0.23	0.32	0.29
17	0.77	0.2	0.26	0.12	0.17	0.41	0.72	0.22	0.27	0.32	0.49	0.71	0.56	0.44	0.32	0.17	0.32	0.21
18	0.26	0.13	0.15	0.1	0.03	0.36	0.37	0.17	0.16	0.16	0.29	0.35	0.13	0.21	0.2	0.08	0.17	0.1
19	0.2	0.04	0.08	0.01	0.06	0.27	0.13	0.14	0.12	0.2	0.12	0.11	0.18	0.04	0.03	0.06	0.12	0.07
20	0.18	0.2	0.06	0.09	0.07	0.09	0.06	0.07	0.1	0.04	0.08	0.06	0.02	0.05	0.13	0.04	0.07	0.05
21	0.19	0.1	0.02	0.03	0.06	0.08	0.06	0.03	0.05	0.09	0.06	0.06	0.1	0.05	0.1	0.03	0.06	0.04
22	0.2	0.23	0	0.02	0.01	0.17	0.05	0.08	0.21	0.04	0.03	0.01	0.13	0.02	0.09	0.07	0.05	0.08
23	0.01	0.01	0	0	0	0.01	0	0.02	0	0	0.01	0	0.01	0	0	0.01	0	0.01
MAX DIFFERENCE	1.28	0.35	0.87	0.5	0.53	0.63	1.22	0.38	0.46	0.44	0.65	1.21	1.19	0.44	0.86	0.3	0.63	0.34

Finally, as the similarity threshold provides valid results, the metric, constraint and criterion are not required this time.

With this hour attribute dependence distribution data, we already can determine which clusters best and worst represents each hour attribute value, as represented in Table 4.21. This information gives an internal understanding of how the clustering of the data behaves for this attribute, deciphering the internal logic of the model.

Table 4.22 summarises which hour attribute values suit each cluster best. Each cluster has a set of hours that are more representative; this is exactly the information we are looking for in order to support the improvement of the corporation infrastructure.

Specifically, we can accomplish this goal by splitting the network traffic in fourteen VoIP channels or trunk lines, designing each of them to have the best performance in the hour(s) in which they are most demanded. This decision is especially relevant if the corporation needs to sign up trunk-lines usage contracts with telecommunication providers. The information in Table 4.22 could be essential to reach an optimal agreement in the negotiation of the pricing scheme of the service with the VoIP provider, as it could ensure significant savings. Furthermore, if we consider the point of view of the service provider, this information is also useful to organise the infrastructure that enables VoIP communication traffic. In this way, the provider will also optimise its networks by obtaining important savings and by improving service quality.

For instance, cluster number zero and number five have a big condensation of calls that took place at any hour. They do not have a significant call time. This fact is useful to rearrange the call traffic of these clusters through a flat-rate tariff.

Other clusters such as number one, two, six or ten have just one representative hour. This means that these clusters represents the calls that have taken place in those particular hours. This peculiarity allow us to route the calls that took place in these clusters through trunk-lines that have a time-hour based pricing scheme, improving in this way the billing cost of this service.

The remaining clusters represents more than one hour, meaning that the calls that took place in these hours tend to group in these clusters. Again, we can arrange changes in the service support systems to improve the service.

Next, we are going to propose a network planning improvement that based on small changes on the pricing schemes will imply compelling ameliorations. These changes can be negotiated with the service provider, and can involve significant savings in the use of the service. It is worth mentioning that the analysis could be based on other aspects besides the pricing schemes, but given the nature of this use case, we consider that this is the most representative area.

Thus, based on the information of Table 4.14, we can notice that the percentages of the instances of the cluster number five are below a six percent, and as seen in Table 4.22, this cluster number five does not represent any calling hour in particular.

Therefore, we consider that the trunk-line that the cluster number five represents can be removed and the corresponding cost reduced. Table 4.23 represents the maximum possible savings through the application of this change to the support systems. The final saving is the difference between the current situation cost and the cost after applying the change to the support system. In this case, the cost is equivalent to the cost of maintaining the removed trunk-line (represented in X).

We could also propose an improvement to the service analysing only the data obtained from the Table 4.22. If we consider that the use case corporation has its office hours from 08:00 to 19:00, we can discriminate the clusters that best represent the hours that do not take place in the office hours. In this way, the clusters numbers one, two, six, nine, ten, eleven, and twelve do not contain office hours. On the other hand, the clusters three, four, five, seven, eight, thirteen, and fourteen contain office hours.

Again, if we assume that we can rearrange the pricing schemes of the service, we can select a time-based pricing scheme in which the calls are charged according to the time of call.

The corporation can negotiate a call pricing scheme for some of the trunk-lines for those calls that do not take place in office hours. If we consider the maximum and minimum distribution percentages of instances for each cluster that Table 4.14 indicates we can estimate the maximum possible savings.

Considering the maximum instances distribution in the clusters that do not take place in office hours, we calculated the relevance of these clusters. Table 4.24

shows this maximum percentage of the instances of the calls that do not take place in office hours.

Table 4.25 represents this maximum possible savings through the change on the pricing scheme of certain trunk-lines.

We propose the change of the pricing scheme on those trunk-lines that are tentatively used for the calls that do not take place in office hours. With the proposed change, the trunk lines with a not in office hour pricing scheme are seven (one per cluster). The new cost is $7X + 7Y$, i.e. the cost per instance per number of trunk lines. The savings are the difference between the original costs and the savings, that result in seven times the absolute difference between the regular pricing scheme and the not in office hour pricing scheme.

The possible changes to the support systems are wide. More interpretations of the results and changes in the support systems could be done in order to improve the services. Nevertheless, as mentioned above, the introduced scenario is the most representative one for the present use case.

Table 4.21: Clusters that best and worst represent each hour attribute value.

Hour attribute value	Minimum value (excluding zero)	Corresponding cluster	Maximum value	Corresponding cluster
0	0.002	4	0.01	11
1	0.01	12	0.0803	11
2	0.002	7	0.2812	11
3	0.002	6	0.2589	0
4	0.0225	13	0.3738	0
5	0.002	11	0.0541	0
6	0.002	10	0.1005	0
7	0.002	1	0.1105	4
8	0.0657	8	0.2624	4
9	0.1634	7	1.0641	6
10	0.0827	13	1.2771	0
11	0.1291	7	1.128	11
12	0.2074	3	1.2187	6
13	0.1151	3	0.9674	0
14	0.0169	14	0.4013	12
15	0.0879	8	0.7214	12
16	0.1091	4	1.1446	0
17	0.121	3	0.7675	0
18	0.0296	4	0.3703	6
19	0.0093	3	0.2711	5
20	0.0155	12	0.2008	1
21	0.0249	2	0.1948	0
22	0.0059	11	0.2309	1
23	0.002	9	0.0249	7

Table 4.22: Summary of the calling hour attribute and the clusters which best represented each value.

Cluster	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Hour		7	21	12	0		3	2	8	23	6	5	1	4	14
				13	16			9	15			22	20	10	
				17	18			11							
				19											

Table 4.23: Maximum possible savings through the removal of a trunk-line.

Analysis \ Scenario	Current situation	Change to the support systems
Number of trunk-lines	14	13
Cost per channel	X	X
Cost	$14X$	$13X$
Saving	$14X - 13X$	

Table 4.24: Maximum instances distribution in the clusters that do not take place in office hours.

Cluster(s)	Percentage
In office hours	53,31%
Not in office hours	71,9%

Table 4.25: Maximum possible savings through the change on the pricing scheme of certain trunk-lines.

Analysis \ Scenario	Current situation	Change on the pricing scheme on certain trunk-lines
Trunk lines with a regular pricing scheme	14	7
Trunk lines with a not in office hour pricing scheme	0	7
Cost per instance for a regular pricing scheme	X	X
Cost per instance for a not in office hour pricing scheme	0	Y
Cost	$14X$	$7X + 7Y$
Saving	$14X - (7X + 7Y)$	
Saving for $X = 10$ and $Y = 9$ monetary units	7	

4.2 Validation summary

This chapter has introduced a VoIP use case to validate the proposed methodology. We decided to use this Internet service for its relevance and exponential growth.

More accurately, our goal generates the optimal representation of the service behaviour to improve the support systems. The methodology detailed in Chapter 3 guided us through the validation experiments.

First, we have introduced the dataset origin and describe its characteristics. With this characterisation, we have executed an action plan that adapted the original dataset to our validation requirements. The representation of the service have been gathered by means of clustering algorithms executions. These executions had their configuration parameters that have been also detailed and justified.

Second, we have analysed the results of the previous step for each of the dataset obtaining the best cluster set selection which has the optimal representation of the service. Specifically, we have applied certain constrains to determine with clustering results where correct, and above them, we have used a metric to compare the cluster sets. This metric has been based on the kurtosis and provided as the results to select the best cluster set for each dataset according to the defined criterion.

Last but not least, we have operated with the global best cluster for our use case. Using the results of the previous steps, we have calculated which model best represents the global problem. Using this best model, we have analysed one of the most representative attributes of the dataset, the hour. Furthermore, we have proposed improvements to the VoIP service support systems making changes in the network planning. These changes has been given with an estimation of the possible maximum savings, improving the service and accomplishing the validation of the methodology.

Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.

Winston Churchill

CHAPTER

5

Conclusions

In this Chapter, we first summarise the results achieved in this dissertation and corroborate the hypothesis of the dissertation Second, we list the achieved publications of the author until the date of dissertation. Third, we discuss and review the results in greater detail and, finally, conclude outlining the avenues of future work.

5.1 Contributions and hypothesis corroboration

This dissertation has produced the following results:

- We have presented a standard research vocabulary in the area of Internet Economics. This vocabulary establishes the state of the art in this area. Furthermore, we introduce a taxonomy that vertebrates all the concepts related to Internet Economics.

- We have developed a methodology to obtain optimal models representing the behaviour of any service of Internet Economics. This representation was deployed after acquiring various service data and using several clustering algorithms.

- We have designed a set of metrics to measure the output results of these clustering algorithms. This strategy gives us the ability to compare clustering algorithms results. Further, we have detailed a set of constrains and criteria to select the best possible results for a given problem.

- We have validated the proposed methodology by using a real-world use-case scenario. The results of this use-case allow us to suggest improvements to enhance the corresponding service support systems.

At this point, and after having accomplished this contributions we have been able to fulfil the specific objectives outlined in Section 1.2.

- To establish the state of the art in Internet Economics.

- To establish a taxonomy to identify the concepts of the services of the Internet Economics.

- To design and implement a data acquisition process that allows experimentation with any set of service data.

- To design and implement a procedure for the analysis of the results of clustering algorithms.

- To apply clustering algorithms to obtain model results.

- To design and implement a set of metrics to measure the results of clustering algorithms.

- To design and implement a set of criteria to select the best model, given the requirements of a given problem.

- To test and validate the proposed methodology.

- To design and implement experiments by using optimal models to improve support systems.

With the achievement of all the objectives of the dissertation, we consider that we have developed a method that extracts knowledge from Internet Economics service data. Further, this knowledge is represented as models that help us improve the support systems of these services, and hence, we can affirm that the initial hypothesis is accomplished.

5.1.1 Relevant publications

The relevance of outreaching the results of our research goes far beyond the *Publish or Perish* proverb. This proverb refers to the pressure to publish work constantly to further or sustain a career in academia despite being a single-minded focus [Nei08].

Aside from disseminating our work, we gathered feedback from the referees of the different committees, reviewers and editors. These comments guided and helped us through the conducted research.

Up to the writing of this dissertation, the following is the list of the most relevant publications that are related to this dissertation, and that were accepted in international conferences, book chapters, and journal articles:

- Ruiz-Agundez, I.; Penya, Y. K. & Bringas, P. G. (2011), Network planning of a VoIP-capable PBX. The use of data profiling techniques for an efficient network planning, in 'Proceedings of the 2011 International Conference on Data Communication Networking (DCNET) 2011', SciTePress, Seville, Spain, pp. 85-88.

- Ruiz-Agundez, I.; Penya, Y. K. & Bringas, P. G. (2011), A Flexible Accounting Model for Cloud Computing, in 'Proceedings of the 2011 Service Research & Innovation Institute (SRII) Global Conference', IEEE, California, USA, pp. 277-284.

- Ruiz-Agundez, I.; Penya, Y. K. & Bringas, P. G. (2010), Information Security Theory and Practices: Security and Privacy of Pervasive Systems and Smart Devices. Lecture Notes in Computer Science 6033, Springer, chapter Fraud Detection for Voice over IP Services on Next-Generation Networks, pp. 199-212. Also published as conference proceedings.

- Ruiz-Agundez, I.; Penya, Y. K. & Bringas, P. G. (2010), A Taxonomy of the Future Internet Accounting Process, in 'Proceedings of ADVCOMP 2010 : The Fourth International Conference on Advanced Engineering Computing and Applications in Sciences', IARIA, Florence, Italy, pp. 111-117.

- Ruiz-Agundez, I.; Penya, Y. K. & Bringas, P. G. (2010), Tarificacion flexible de servicios en Internet, in 'Proceedings of the XX Jornadas de Telecom I+D', Telefonica, Valladolid, Spain.

- Ruiz-Agundez, I.; Penya, Y. K. & Bringas, P. G. (2010), Fraud Detection for Voice over IP Services on Next-Generation Networks, in P. Samarati et al., ed.,'Proceedings of the 4th Workshop in Information Security Theory and Practices (WISTP 2010)', Springer, Passau, Germany, pp. 199-212.

- Ruiz-Agundez, I.; Bringas, P. G. & Penya, Y. K. (2010), Addressing the Billing Needs for the Internet of Services and Things, in 'Proceedings of the IADIS International Conference on Information Systems', IADIS, Porto, Portugal, pp. 579-582.

- Ruiz-Agundez, I. & Penya, Y. K. (2009), Case Study Handbook. 30 Real-World Service Provider and Vendor Solutions Using TM Forum Best Practices & Frameworks, TeleManagement Forum, chapter Fraud Detection For Next-Generation Network, pp. 19.

- Ruiz-Agundez, I. (2009), 'Edits of: Service Specification - Voice over IP (VoIP)', TeleManagement Forum.

In addition, other publications were also achieved:

- Ruiz-Agundez, I. (2011), 'Collaborative writing with Google Docs', TUGboat, The Communications of the TeX Users Group 32(3). In press.

- Guidi, M.; Ruiz-Agundez, I. & Canga-Sanchez, I. (2011), Computational Social Networks: Mining and Visualization, Springer, chapter Knowledge Mining from the Twitter Social Network. The case of Barack Obama. In press.

- Ruiz-Agundez, I. & Nieves, J. (2011), 'EURion konstelazioa eguneroko billeteetan', Elhuyar zientzia eta teknologia 272, 50-53.

- Penya, Y. K.; Ruiz-Agundez, I. & Bringas, P. G.. Skrobanek, P., ed., (2011), Intrusion Detection Systems, InTech, chapter Integral Misuse and Anomaly Detection and Prevention System, pp. 155-172.

- Ruiz-Agundez, I.; Canga-Sanchez, I. & Guidi, M. (2011), Barack Obamaren edo beste norbaiten tweet-etatik mamia ateraz, in 'Informatikari Euskaldunen VIII. Bilkura (IEB2011: Gizarte-sareak)', Udako Euskal Unibertsitatea, Donostia/San Sebastian, Basque Country.

- Nieves, J.; Ruiz-Agundez, I. & Bringas, P. G. (2010), Recognizing Banknote Patterns for Protecting Economic Transactions, in A M. Tjoa & R.R. Wagner, ed.,'Proceedings of the Twenty-First International workshop on Database and Expert Systems Applications', IEEE , Bilbao, Basque Country, pp. 247-249.

- Ruiz-Agundez, I.; Penya, Y. K. & Bringas, P. G. (2010), Optimal Bayesian Network Design for Efficient Intrusion Detection, in 'Proceedings of the 3rd International Conference on Human System Interaction (HSI 2010)', IEEE, Rzeszow, Poland, pp. 444-451.

- Lequerica, I.; Ruiz-Agundez, I. & et al. (2010), Vehicular Networks and Services , eMOV AEI, 8, 9, 11.

Please, note that a complete and up to date list of the author publications can be accessed at the University of Deusto author's personal web space [1].

5.2 Discussion

At this point it is time to consider all the work done and to do some self-criticism of the results. One of the hardest problems was to find a balance between the procedural complexity and the potential of extracting knowledge. Below, there is some discussion about the different aspects of our dissertation.

[1] http://paginaspersonales.deusto.es/igor.ira/

In Section 2.1 we introduced a taxonomy of the Internet Economics process. We detailed all the functions involved in it, looking at all the relationships between them. We believe that the presented taxonomy contributes to the learning, training and assessing in the present area because it gives an integrated vision of the process. Since it defines a common vocabulary, it is also useful for the definition of the economic requirements among different actors. This taxonomy was defined following a methodology that ensures its quality and the maintainability of the knowledge to potential changes. Furthermore, we have proposed a unified and controlled vocabulary that can be used in any operation related to Internet Economics.

In Section 2.2 we detailed the clustering algorithms and their potential. Given the existing data mining techniques, it is obvious that we may have used others to extract different types of knowledge. In our case, clustering algorithms allow us to build usage profiles; that is, grouping the instance data of our datasets. It is true that there are many other clustering algorithms in the literature and that we could have used them to compare the results. The incremental clustering method applied is, however, the only one that generates an explicit knowledge representation model that describes clustering in such a way that it can be easily visualised and understood [WF05]. Therefore, we decided to use those ones, since they are broadly used and represent a good set of algorithms that, if required, could be extended in the future.

We also considered the possibility of developing a new algorithm that could be applied in any domain and could help us to cluster and extract knowledge in any problem. Besides, this enterprise is not feasible since the results of the tentative algorithm would still require a comparison with existing algorithms in order to validate the results and the requirement of such comparison would lead us to design a potentially similar methodology to the one introduced in this dissertation.

In Section 3.1, we suggested a knowledge discovery methodology that potentially might be applied to any domain. This methodology uses clustering algorithms to process the data and transform it into knowledge. The validation experiments showed that it is able to present knowledge in a highly comprehensible format. The possibility to accurately replicate this knowledge extraction methodology makes the proposed approach quite promising. It can be applied in other datasets of the same VoIP domain or on completely different areas.

Our optimal clustering model selection methodology could include some more functions or leave some of the presents aside. In any case, the general structure and logic would remain equal as the performed steps were designed to be as general and as simple as possible [Dom99].

The novelty of our approach lays on the fact of using a problem-dependent domain-independent metric that allows comparing the different clusterisation algorithms: the metric must be chosen according to the desired properties of the targeted clusters. Moreover, related work focuses extensively on meta-learning from meta-information but, in our approach, we lack of that meta-information. Further, we infer it from the algorithms themselves.

The objective of the methodology is to define a model selection strategy. It could be further refuted by performing experiments in other domains and not just in the VoIP use-case introduced in Section 4.1. The exploration of other domains is promising, as we address the problems by comparing a number of applicable clustering algorithms and then by sorting them according to a problem-dependent and dataset-independent metric. The novelty lays on the fact that the metrics are interchangeable, and selected with the help of an external problem expert (and not a dataset or domain expert). Then, the winning clustering algorithm model may be applied to a certain dataset of any domain. We do not intend to claim our model's universality, since we should back it mathematically by a mathematical proof (say theorem), which we deem does not exist, as happened in other areas [HP02]. Still, we have shown the suitability of our methodology in a concrete use-case but, we dare say, though even lacking of the mathematical proof, this model aspires and aims at being universal.

If the results of the methodology were below an acceptable threshold, the results would provide with strong evidence that the learning dataset is not a representative subset of the validation set. In this case, a more detailed study would be required because it would imply that the datasets are significantly different and that there is no clustering algorithm that will manage to determine the natural trend form of the data.

In order to validate the proposed methodology, we used different clustering algorithms to process our VoIP data records. The results of the experiment showed

that it is possible to conduct knowledge discovery by using clustering algorithms. Next, we are going to analyse some aspects of this use-case.

The use-case tried three subsets of a dataset. We could argue that there should be other completely different sources in order to obtain results from a different organisation. In fact, we only had access to the PBX of one corporation. The information that this PBX contains is confidential and its access implies complex administrative processes. We carried out the experiments with a subset of the biggest dataset as a sort of CV. Further, our goal was to represent the behaviour of the data for a specific time period and organisation, and analyse that behaviour for other time periods. The discovered knowledge, thus, can be generalised.

We could have also incremented the granularity of the dataset by splitting the date attribute by year, month and day. Nevertheless, since the dataset in the experiment corresponds to one single month, this criterion did not affect the results. Moreover, we did not study whether the attributes are interdependent. Thus, for this experiment, all of the attributes were handled as covariant; Bayesian-network-based algorithms [RAPB10c] may help in the study of such dependencies. Other attribute transformations (e.g. discretisation, relevance, and so on) could also be considered.

In all cases, the results of this methodology must be interpreted by an expert in the area. This expert should be responsible for extracting conclusions from the results [HBV01] in order to identify conclusions and, if appropriate, drive an action plan. This information can be used for decision-making in the appropriate domain area in order to develop improvement in the support systems.

In our specific use-case, we proposed an analysis in the network planning that could result into savings by obtaining a more convenient telephone call scheme. The change consisted of adopting a flat-rate pricing scheme for one of the telephone channels (say, trunk lines) for a set of calls that would be more expensive with a time-based pricing scheme. We contributed to improving the resource management and the service tariffing of the experiment's corporation. Furthermore, this methodology could be generalised for any corporation size as service usages can always be clustered.

Finally, it is worth highlighting that the most exciting conclusion of the proposed methodology is, in our opinion, the ability to produce very representative

results that could provide a high level of knowledge. The benefits and possible applications of such methodology include: user behaviour modelling, fraud detection [RAPB10a] through anomalous behaviour analysis, technical and economical implications (e.g. network planning, load balancing), and pricing scheme modelling (i.e. tariff planning) [FDL09]. We decided to initially focus on the optimisation of the infrastructure, as this was the most relevant matter for our dataset provider.

The metric (kurtosis of the cluster instance distribution) is well-motivated and seems to produce reasonable results on the VoIP data. It would be interesting to use other metrics and compare the results empirically. Nevertheless, it is almost impossible to mathematically determine an optimal metric even though it does exist theoretically as it would imply having a mathematical representation of the problem.

Each metric is problem-specific and obtains different results. Furthermore, each problem defines its own metric (which comprises problem features, definition of *"better"* and so on). Therefore, it is not possible to compare metrics empirically. What we can empirically test is whether the model is valid for other domain/metric combinations or not.

Furthermore, the metric may not fully apply to the initially given problem or in some cases; the problem already gives a metric. In contrast, in case there is no clear metric for the problem, we would be forced to apply diverse metrics and then choose the one that best fits the problem. In our use-case, the selection of the kurtosis seemed to be the most appropriate candidate metric since we are looking for a cluster distribution with one big cluster containing most of the relevant instances and several tiny clusters with scarce instances.

We also considered a metric by looking for the biggest difference between the two biggest clusters among all the models, tested it and obtained worse results than with the kurtosis.

5.3 Future Work

There are still some open issues regarding this dissertation domain. Next, we are going to overview the possibility of studying in more detail different aspects of it and the interrelation with other fields.

5. CONCLUSIONS

Concerning to the taxonomy of Internet Economics, we plan to perform a proof-of-concept of the presented taxonomy by implementing it in a real system. Furthermore, this taxonomy defines the pillars for the development of accounting related applications, such as fraud management systems [RAPB10a] or data mining [HYW06]. We also plan a validation of the proposed taxonomy, using both direct inspection and validation metrics [SK02] as well as updating the taxonomy itself if there are substantial changes in the area of Internet Economics. We are also considering a deeper analysis of each of the functions and comparing the interrelations among them [SLAB10].

Even more, the definition of this taxonomy provides us with the pillars to structure a service of Internet Economics. We aim at deploying a functional implementation of this taxonomy allowing the exploitation of services all over the value chain. Our first steps in this area have been the creation of a model that enables a flexible economisation of cloud computing services [RAPB11].

Mining data to extract valuable knowledge is a discipline with long tradition. When it comes to clustering techniques, there are many algorithms that could be applied in every case. We have shown that the key to enabling the comparison between all the candidate clustering algorithms lays on finding a common metric in their results. We have proved the soundness of this model by testing it in a real application with real data.

Although the experiments have been successful, our main concern is that this methodology still requires some knowledge of the problem domain in order to define the metric and the criterion to accomplish the ranking of the clustering algorithms (i.e. it is not totally automated). Future work will include strategies to alleviate this need and will deal with an scenario in which using the kurtosis is not suitable. We will also focus on the study of other less significant attributes (e.g. *dcontext*, *duration*) and on the extraction of further knowledge from the models to improve support systems of VoIP services.

In the same way, regarding the presented methodology, we are planning to use the generated cluster set models as training data for classification algorithms. These algorithms should allow us to predict the values of certain attributes. For example, in the VoIP use-case, we should be able to predict within a given probability a call's duration or the channel used to route the call given its source and destination. In

this way, we would go one step beyond the knowledge extraction and would be predicting new data in new scenarios. The combination of different algorithms and the creation of a meta-classifier for a domain specific problem is an open topic [NSP+09] [SN+10].

Similarly, the analysis of new attributes in the VoIP use-case scenario would lead to new knowledge. The study of these new attributes would lead to further service improvements, such as user experience, infrastructure management, revenue planning, or QoS.

Finally, we consider that our methodology could be applied to diverse areas such as the carbon black nano-aggregates [LdURS+11], email spam filtering [LSS+11], computer-packed executable filtering [UPSB11], web-page URL analysis [DCAB11], or defect prediction in high-precision foundry [SGP+].

Bibliography

[AAH00] B. Aboba, J. Arkko, and D. Harrington. RFC2975: Introduction to Accounting Management. *RFC Editor United States*, 2000. 15, 22, 24, 25, 27, 29

[ABKS99] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander. OPTICS: ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM, 1999. 44

[ACM+00] P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Rathsand, and M. C. Wittrock. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn and Bacon, abridged edition, 2000. 13, 15

[AF07] J. Abonyi and B. Feil. *Cluster analysis for data mining and system identification*. Birkhauser, 2007. 34

[AGGR98] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998. 38

[Aka70] H. Akaike. Statistical Prediction Information. *Annals of the Institute of Statistical Mathematics*, 22:203–217, 1970. 46, 94

[AKK03] V. Agarwal, N. Karnik, and A. Kumar. Metering and accounting for composite e-Services. In *Proc. 1st IEEE Intl Conf. on E-Commerce*, pages 35–39, 2003. 14

[ALA+97] B. Aboba, J. Lu, J. Alsop, J. Ding, and W. Wang. RFC2194: Review of Roaming Implementations. *RFC Editor United States*, 1997. 24

[ASM11] A. Albalate, D. Suendermann, and W. Minker. On cluster validation for detecting the number of clusters in a data set. *International Journal on Artificial Intelligence Tools*, 20(5):941–953, 2011. 54

[AY00] C.C. Aggarwal and P.S. Yu. *Finding generalized projected clusters in high dimensional spaces*, volume 29. ACM, 2000. 45

[AY09] A. Ayanso and R. Yoogalingam. Profiling Retail Web Site Functionalities and Conversion Rates: A Cluster Analysis. *International Journal of Electronic Commerce*, 14(1):79–114, 2009. 82

[AZ99] B. Aboba and G. Zorn. RFC2477: Criteria for Evaluating Roaming Protocols. *RFC Editor United States*, 1999. 24

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC2475: An Architecture for Differentiated Service. *RFC Editor United States*, 1998. 20

[BE84] J. C. Bezdek and R. Ehrlich. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984. 33

[BEH09] L. Burness, P. Eardley, and R. Hancock. The trilogy architecture for the future internet. *Towards the Future Internet*, page 79, 2009. 2

[Ber06] P. Berkhin. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, pages 25–71, 2006. 34, 65

[BGCSV08] P. B. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta. *Metalearning: Applications to data mining*. Springer-Verlag New York Inc, 2008. 58

[BHEG+02] A. Ben-Hur, A. Elisseeff, I. Guyon, et al. A stability based method for discovering structure in clustered data. In *Pacific symposium on biocomputing*, volume 7, pages 6–17, 2002. 55

[Bis06] C. M. Bishop. *Pattern recognition and machine learning*. Springer New York., 2006. 41

[BK99] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1):105–139, 1999. 33

[BMAPO05] B. Bella, Eloff M. A., J. H. P., and M. S. Olivier. Using the ipdr standard for ngn billing and fraud detection. In *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, 2005. 14

[BMR99] N. Brownlee, C. Mills, and G. Ruth. RFC2722: Traffic flow measurement: architecture. *RFC Editor United States*, 1999. 18

[BP98] J.C. Bezdek and N.R. Pal. Some new indexes of cluster validity. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 28(3):301–315, 1998. 54

[CAC10] S. Chaimontree, K. Atkinson, and F. Coenen. Best Clustering Configuration Metrics: Towards Multiagent Based Clustering. In *Advanced Data Mining and Applications: 6th International Conference, ADMA 2010, Chongqing, China, November 19-21, 2010, Proceedings*, page 48. Springer London, Limited, 2010. 54

[CCC08] H. L. Chen, K. T. Chuang, and M. S. Chen. On Data Labeling for Clustering Categorical Data. *IEEE Transactions on Knowledge and Data Engineering*, 20(11), 2008. 54

[CFK03] E. Cohen, A. Fiat, and H. Kaplan. Associative search in peer to peer networks: Harnessing latent semantics. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1261–1271. IEEE, 2003. 3

[CH05] H. C. Chang and C. C. Hsu. Using topic keyword clusters for automatic document clustering. *IEICE Transactions on Information and Systems*, E88D(8):1852–1860, AUG 2005. 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004), Tokyo, Japan, Oct 26-29, 2004. 31

[CH07] M. Choi and J. W. Hong. Towards management of next generation networks. *IEICE Trans Commun*, E90-B(11):3004–3014, 2007. 13

[Che10] C. Chen. *Handbook of pattern recognition and computer vision*. World Scientific, 2010. 53

[CNP06] G. Chicco, R. Napoli, and F. Piglione. Comparisons among clustering techniques for electricity customer classification. *IEEE Transactions On Power Systems*, 21(2):933–940, May 2006. IEEE Bologna PowerTech, Bologna, Italy, Jun 23-26, 2003. 31

[CP01] X. Chang and D. W. Petr. A survey of pricing for integrated service networks. *Computer communications*, 24(18):1808–1818, 2001. 25

[CPP+10] J. Castro, A. Punzon, G. J. Pierce, M. Marin, and E. Abad. Identification of metiers of the Northern Spanish coastal bottom pair trawl fleet by using the partitioning method CLARA. *Fisheries Research*, 102(1-2):184–190, February 2010. 33, 36

[CY00] R. Chau and C. H. Yeh. Intelligent data analysis for business decision support via fuzzy clustering. In Mohammadian, M, editor, *Advances In Intelligent Systems: Theory And Applications*, volume 59 of *Frontiers In Artificial Intelligence And Applications*, pages 45–52. IOS Press, 2000. International Conference on Advances in Intelligent Systems: Theory and Applications (AISTA 2000), Canberra, Australia, Feb 02-04, 2000. 31

[Das02] S. Dasgupta. Performance guarantees for hierarchical clustering. In *Computational Learning Theory*, pages 235–254. Springer, 2002. 42

[DB79] D.L. Davies and D.W. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979. 53, 54

[DCAB11] J. Devesa, X. Cantero, G. Alvarez, and P. G. Bringas. An efficient security solution for dealing with shortened url analysis. In *In Proceedings of the 8th International Workshop on Security in Information Systems (WOSIS) in conjunction with ICEIS 2011*, pages 70–79, Beijing (China), June 2011. 119

[Der06] L. Deri. Open source VoIP traffic monitoring. *SANE 2006*, 2006. 13

[DKS95] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Machine Learning-International Workshop Then Conference*, pages 194–202. Morgan Kaufmann Publishers, Inc., 1995. 65

[DLB09] G. Detal, D. Leroy, and O. Bonaventure. An adaptive three-party accounting protocol. In *Proceedings of the 5th international student workshop on Emerging networking experiments and technologies*, pages 3–4. ACM, 2009. 13

[DLR+77] A. P. Dempster, N. M. Laird, D. B. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 41

[Dom99] P. Domingos. The role of occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery*, 3(4):409–425, 1999. 115

[Dun73] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973. 42

[Dun74] J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104, 1974. 54

[EEV08] A. Estepa, R. Estepa, and J. Vozmediano. Traffic Trunk parameters for voice Transport over MPLS. *E-Business and Telecommunication Networks*, pages 199–210, 2008. 81

[EKSX96] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996. 40

[FDL09] M. Falkner, M. Devetsikiotis, and I. Lambadaris. An overview of pricing concepts for broadband IP networks. *Communications Surveys & Tutorials, IEEE*, 3(2):2–13, 2009. 81, 117

[Fis87] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2(2):139–172, 1987. 39

[FPSM92] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus. Knowledge discovery in databases: An overview. *Ai Magazine*, 13(3):57, 1992. 3

[FPSSU96] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in knowledge discovery and data mining*. MIT press, 1996. 52, 58

[GI10] F. Greselin and S. Ingrassia. Constrained monotone EM algorithms for mixtures of multivariate t distributions. *Statistics And Computing*, 20(1):9–22, JAN 2010. 33

[GLF89] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial intelligence*, 40(1-3):11–61, 1989. 39

[Goo02] B. Goode. Voice over internet protocol (VoIP). *Proceedings of the IEEE*, 90(9):1495–1517, 2002. 80

[GRS98] S. Guha, R. Rastogi, and K. Shim. Cure: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84. ACM, 1998. 39

[GRS00] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000. 33, 47

[GWR+06] T. Grotkjaer, O. Winther, B. Regenberg, J. Nielsen, and L. K. Hansen. Robust multi-scale clustering of large DNA microarray datasets with the consensus algorithm. *Bioinformatics*, 22(1):58–67, JAN 1 2006. 31

[HBV01] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001. 33, 34, 47, 52, 54, 58, 61, 116

[HG07a] A. Hinneburg and H. Gabriel. DENCLUE 2.0: Fast clustering based on kernel density estimation. In Berthold, MR and ShaweTaylor, J and Lavrac, N, editor, *Advances In Intelligent Data Analysis Vii, Proceedings*, volume 4723 of *Lecture Notes In Computer Science*, pages 70–80. Springer-Verlag Berlin, 2007. 7th International Symposium on Intelligent Data Analysis, Ljubljana, Slovenia, Sep 06-08, 2007. 33

[HG07b] A. Hinneburg and H.H. Gabriel. Denclue 2.0: Fast clustering based on kernel density estimation. *Advances in Intelligent Data Analysis VII*, pages 70–80, 2007. 40

[HHH+09] T. Hoßfeld, D. Hausheer, F. Hecht, F. Lehrieder, S. Oechsner, I. Papafili, P. Racz, S. Soursos, D. Staehle, G.D. Stamoulis, et al. An economic traffic management approach to enable the triplewin for users, ISPs, and overlay providers. *Towards the Future Internet*, page 24, 2009. 2

[HK98] A. Hinneburg and D.A. Keim. *An efficient approach to clustering in large multimedia databases with noise*. Bibliothek der Universität Konstanz, 1998. 40

[HK99] A. Hinneburg and D.A. Keim. *Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering*. Citeseer, 1999. 44

[HK05] J. Handl and J. Knowles. Exploiting the trade-off—the benefits of multiple objectives in data clustering. In *Evolutionary Multi-Criterion Optimization*, pages 547–560. Springer, 2005. 55

[HKK05] J. Handl, J. Knowles, and D.B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201, 2005. 52

[HNF+09] D. Hausheer, P. Nikander, V. Fogliati, K. Wunstel, M.A. Callejo, S.R. Jorba, S. Spirou, L. Ladid, W. Kleinwachter, B. Stiller, et al. Future internet socio-economics–challenges and perspectives. *Towards the Future Internet*, page 1, 2009. 2

[Hoc01] T. A. Hoc. On the relaying capability of Next-Generation GSM cellular networks. *IEEE Personal Communications*, page 41, 2001. 24

[HP02] Y. C. Ho and D. L. Pepyne. Simple explanation of the no-free-lunch theorem and its implications. *Journal of Optimization Theory and Applications*, 115(3):549–570, 2002. 115

[HS85] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985. 42

[Hua97] Z. Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *Workshop on Research Issues on Data Mining and Knowledge Discovery*, 1997. 47

[HW79a] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28:100–108, 1979. 43

[HW79b] J.A. Hartigan and M.A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979. 53

[HYW06] S. Y. Hung, D. C. Yen, and H. Y. Wang. Applying data mining to telecom churn management. *Expert Systems with Applications*, 31(3):515–524, 2006. 118

[IAH08] L. F. Ibrahim and M. H. Al Harbi. Using Clustering Technique M-PAM in Mobile Network Planning. In Mastorakis, NE and Mladenov, V and Bojkovic, Z and Simian, D and Kartalopoulos, S and Varonides, A, editor, *Proceedings Of The 12Th Wseas International Conference On Computers , Pts 1-3 - New Aspects Of Computers* , Recent Advances in Computer Engineering, pages 868–873, 2008. 33

[Jai10] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. 52, 61, 63

[JD88] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. 1988. 34

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999. 34

[JR99] M. I. Jordan and S. Russell. Computational intelligence. *The MIT encyclopedia of the cognitive sciences*, 1999. 4

[KKA02] M. Koutsopoulou, A. Kaloxylos, and A. Alonistioti. Charging, accounting and billing as a sophisticated and reconfigurable discrete service for next generation mobile networks. In *IEEE Vehicular Technologhy Conference*, volume 4, pages 2342–2345, 2002. 13

[KKA+04] M. Koutsopoulou, A. Kaloxylos, A. Alonistioti, L. Merakos, and K. Kawamura. Charging, accounting and billing management schemes in mobile telecommunication networks and the internet. *IEEE Communications Surveys*, 6(1):50–58, 2004. 15, 22, 27, 29

[KKZ09] H. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1–58, 2009. 83, 99

[KL88] W.J. Krzanowski and YT Lai. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, pages 23–34, 1988. 54

[Koh89] T. Kohonen. *Self-Organization and Associative Memory*. Springer, 3rd edition, 1989. 45

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1137–1145, 1995. 41, 42

[Koh03] T. Kohonen. *The Handbook of Brain Theory and Neural Networks*, chapter Learning Vector Quantization, pages 631–634. MIT Press, 2nd edition, 2003. 44

[KP02] M. Kouadio and U. Pooch. A taxonomy and design considerations for Internet accounting. *ACM SIGCOMM Computer Communication Review*, 32(5):48, 2002. 11, 16, 25

[KSSW00] M. Karsten, J. Schmitt, B. Stiller, and L. Wolf. Charging for packet-switched network communication - motivation and overview. *Computer Communications*, 23:290–302, 2000. 13

[KSWS99] M. Karsten, J. Schmitt, L. Wolf, and R. Steinmetz. Cost and price calculation for internet integrated services. In *Proceedings of Kommunikation in Verteilten Systemen (KiVS'99*, pages 46–57. Springer, 1999. 25

[LDJ07] T. Li, C. Ding, and M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 577–582, Washington, DC, USA, 2007. IEEE Computer Society. 54

[LdURS+11] J. López-de Uralde, I. Ruiz, I. Santos, A. Zubillaga, P. Bringas, A. Okariz, and T. Guraya. Automatic morphological categorisation of carbon black nano-aggregates. In *Database and Expert Systems Applications*, pages 185–193. Springer, 2011. 119

[Lia08] T. W. Liao. Enterprise Data Mining: A Review and Research Directions. *Recent advances in data mining of enterprise data: algorithms and applications*, page 1, 2008. 52, 58

[Lov83] M. C. Lovell. Data Mining. *Review of Economics and Statistics*, 65(1):1–12, 1983. 31

[LSS+11] C. Laorden, B. Sanz, I. Santos, P. Galán-García, and P. G. Bringas. Collective classification for spam filtering. In *Computational Intelligence in Security for Information Systems*, volume 6694 of *Lecture Notes in Computer Science*, pages 1–8, 2011. 119

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate data. In *The fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967. 33

[Mar70] K. V. Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3):519, 1970. 93

[MB95] L. W. McKnight and J. P. Bailey. An introduction to internet economics. *The Journal of electronic publishing*, 1, January 1995. 2, 10

[MB97] L. W. McKnight and J. P. Bailey. *Internet Economics*. Mit Press Cambridge, Massachusetts, 1997. 10

[MBTP04] B. Minaei-Bidgoli, A. Topchy, and W. F. Punch. A comparison of resampling methods for clustering ensembles. In *International conference on Machine Learning*, pages 939–945, 2004. 89

[MH04] B. Mandelbrot and R. L. Hudson. *The (Mis)behavior of Markets: A Fractal View of Risk,Ruin, and Reward*. Basic Books, New York, 2004. 93

[MHR91] C. Mills, D. Hirsh, and G. R. Ruth. RFC1272: Internet Accounting: Background. *RFC Editor United States*, 1991. 13, 14, 16, 22

[Mic80] R. S. Michalski. Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. *International Journal of Policy Analysis and Information Systems*, 4:219–244, 1980. 39

[MRS10] C. D. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval. *Information Retrieval*, 13(2):192–195, 2010. 55

[MWS06] C. Morariu, M. Waldburger, and B. Stiller. An integrated accounting and charging architecture for mobile grids. In *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, pages 1–10, Oct. 2006. 13, 27

[MZX05] F. C. Meng, D. C. Zhan, and X. F. Xu. Business component identification of enterprise information system: A hierarchical clustering method. In *ICEBE 2005: IEEE International Conference On E-Business Engineering, Proceedings*, pages 473–480. IEEE Computer Soc, 2005. 31

[NC07] N. Nguyen and R. Caruana. Consensus clusterings. In *ICDM'07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, 2007. 54

[Nei08] U. S. Neill. Publish or perish, but at what cost? *The Journal of clinical investigation*, 118(7):2368, 2008. 111

[Non94] I. Nonaka. A dynamic theory of organizational knowledge creation. *Organization science*, pages 14–37, 1994. 3

[NSP+09] J. Nieves, I. Santos, Y. K. Penya, S. Rojas, M. Salazar, and P. G. Bringas. Mechanical properties prediction in high-precision foundry production. In *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on*, pages 31–36. IEEE, 2009. 119

[Org99] Ley Orgánica. 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. *Madrid: Boletín Oficial del Estado*, 1999. 82

[PM] D. Pelleg and A. Moore. X-means: Extending K-means with E cient Estimation of the Number of Clusters. In *Proceedings of the 17th International Conference on Machine Learning*, page 727. 46

[PR07] P. Krishna Prasad and C. P. Rangan. Privacy preserving BIRCH algorithm for clustering over arbitrarily partitioned Databases. In Alhajj, R and Gao, H and Li, X and Li, JZ and Zaiane, OR, editor, *Advanced Data Mining and Applications, Proceedings*, volume 4632 of *Lecture Notes In Artificial Intelligence*, pages 146–157. Harbin Inst Technol, Springer-Verlag Berlin, 2007. 3rd International Conference on Advanced Data Mining and Applications, Harbin, Peoples R China, Aug 06-08, 2007. 33

[Pro08] Interface Programme. *IPDR Service Specification. Design Guide*. TeleManagement Forum, September 2008. 13, 14, 16, 24

[Pro09] Interface Programme. *IPDR Business Solution Requirements*. TeleManagement Forum, May 2009. 20, 22

[PvBSP01] A. Pras, B. J. van Beijnum, R. Sprenkels, and R. Parhonyi. Internet accounting. *IEEE Communications Magazine*, 39(5):108–113, 2001. 16

[Pá05] R. Párhonyi. *Micro payment gateways*. PhD thesis, Twente University, 2005. 13, 15, 18, 31

[QGZ03] W. N. Qian, X. Q. Gong, and A. Y. Zhou. Clustering in very large databases based on distance and density. *Journal Of Computer Science And Technology*, 18(1):67–76, Jan 2003. 33

[Ran71] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, pages 846–850, 1971. 55

[RAPB10a] I. Ruiz-Agundez, Y. K. Penya, and P. G. Bringas. Fraud detection for voice over ip services on next-generation networks. In *Proceedings of the 4th Workshop in Information Security Theory and Practices (WISTP 2010)*, Passau, Germany, 12-14 April 2010. Springer. 117, 118

[RAPB10b] I. Ruiz-Agundez, Y. K. Penya, and P. G. Bringas. *Information Security Theory and Practices: Security and Privacy of Pervasive Systems and Smart Devices. Lecture Notes in Computer Science 6033*, chapter Fraud Detection for Voice over IP Services on Next-Generation Networks, pages 199–212. Springer, 2010. 81

[RAPB10c] I. Ruiz-Agundez, Y. K. Penya, and P. G. Bringas. Optimal bayesian network design for efficient intrusion detection. In *Proceedings of the 3rd International Conference on Human System Interaction (HSI 2010)*, Rzeszow, Poland, 13-15 May 2010. IEEE. 116

[RAPB11] I. Ruiz-Agundez, Y. K. Penya, and P. G. Bringas. A flexible accounting model for cloud computing. In *Proceedings of the 2011 Service Research & Innovation Institute (SRII) Global Conference*, pages 277–284, California, USA, 30 March - 2 April 2011. IEEE. 118

[RG75] L.R. Rabiner and B. Gold. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1, 1975. 45

[Rom04] C. Romesburg. *Cluster analysis for researchers*. Lulu. com, 2004. 55

[Rou87] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 54

[Rou10] M. Roughan. Robust Network Planning. *Guide to Reliable Internet Services and Applications*, pages 137–177, 2010. 81

[SCZ98] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 428–439. INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERS, 1998. 33, 45

[SFPW98a] B. Stiller, G. Fankhauser, B. Plattner, and N. Weiler. Charging and accounting for integrated internet services - state of the art, problems, and trends. In *Problems, and Trends; The Internet Summit (INET'98)*, pages 21–24, 1998. 20, 22, 27, 29, 30

[SFPW98b] B. Stiller, G. Fankhauser, B. Plattner, and N. Weiler. Pre-study on "Customer Care, Accounting, Charging, Billing, and Pricing". *Computer Engineering and Networks Laboratory TIK, ETH Zurich, Switzerland, Pre-study performed for the Swiss National Science Foundation within the "Competence Network for Applied Research in Electronic Commerce*, 1998. 10, 16, 24

[SFT02] N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 129–136. ACM, 2002. 46

[SGP+] M. Salazar, D. Gallego, Y. K. Penya, I. Santos, and P. G. Bringas. Fault-tolerant defect prediction in high-precision foundry. In *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, pages 373–378. IEEE. 119

[SGRF01] B. Stiller, J. Gerke, P. Reichl, and P. Flury. Management of differentiated services usage by the cumulus pricing scheme and a generic internet charging system. In *Proceedings of the Symposium on Integrated Network Management*, 2001. 18, 20, 22, 25, 27

[SK02] S. Spangler and J. Kreulen. Interactive methods for taxonomy editing and validation. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 665–668. ACM, 2002. 118

[SLAB10] B. Sanz, C. Laorden, G. Alvarez, and P. G. Bringas. A threat model approach to attacks and countermeasures in on-line social networks. In *In Proceedings of the 11th Reunion Española de Criptografía y Seguridad de la Información (RECSI), 7-10th September, Tarragona (Spain), in press.*, page in press, 2010. 118

[SN+10] I. Santos, J. Nieves, , Y. K. Penya, and P. G. Bringas. Towards noise and error reduction on foundry data gathering processes. In *Proceedings of the International Symposium on Industrial Electronics (ISIE)*, 2010. 1765–1770. 119

[Tak06] T. Takuji. Backend systems architectures in the age of the next generation network. *NEC Technical Journal*, 1(2):51–55, May 2006. 14

[TDG+09] G. Tselentis, J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zahariadis. Towards the future internet, 2009. 2

[TSK06] P. N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson Addison Wesley Boston, 2006. 31, 33

[Tur50] A.M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950. 4

[TWH01] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63:411–423, 2001. 53

[Uni08] European Union. The BLED declaration: Towards a european approach to the future internet, 3 2008. 13

[UPSB11] X. Ugarte-Pedrero, I. Santos, and P. G. Bringas. Boosting scalability in anomaly-based packed executable filtering. In *Proceedings of the 7th International Conference on Information Security and Cryptology (INSCRYPT)*, Beijing (China), 30 November-3 December 2011. 119

[VC74] V. Vapnik and A. Chervonenkis. Ordered risk minimization. *Automation and Remote Control*, 34:1226–1235, 1974. 94

[VMSM05] J. Van Meggelen, J. Smith, and L. Madsen. *Asterisk: the future of telephony*. O'Reilly Media, Inc., 2005. 82

[WB08] M. Whittaker and K. Breininger. Taxonomy development for knowledge management. In *World Library and Information Congress: 74th IFLA General Conference and Council*, August 2008. 15, 16

[WF05] I. H. Witten and E. Frank. *Data mining. Practical machine learning tools and techniques*. Elsevier, 2005. 15, 47, 114

[WG01] C. Welty and N. Guarino. Supporting ontological analysis of taxonomic relationships. *Data and knowledge engineering*, 39(1):51–74, 2001. 15

[WLH03] C. P. Wei, Y. H. Lee, and C. M. Hsu. Empirical comparison of fast partitioning-based clustering algorithms for large data sets. *Expert Systems With Applications*, 24(4):351–363, May 2003. 33, 37

[WQZ07] X. Wang, W. Qiu, and R.H. Zamar. Clues: A non-parametric clustering method based on local shrinking. *Computational Statistics & Data Analysis*, 52(1):286–298, 2007. 39

[WYM97] W. Wang, J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. In *Proceedings of the International Conference on Very Large Data Bases*, pages 186–195, 1997. 33

[XW+05] R. Xu, D. Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005. 34

[YGY02] Y. Yang, X. Guan, and J. You. CLOPE: a fast and effective clustering algorithm for transactional data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 682–687. ACM, 2002. 38

[ZRL96] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. In *ACM SIGMOD Record*, volume 25, pages 103–114. ACM, 1996. 36

[ZRM03] G. Zhang, B. Reuther, and P. Mueller. User Oriented IP Accounting in multi-user systems. In *Integrated network management VIII: managing it all: IFIP/IEEE Eighth International Symposium on Integrated Network Management (IM 2003), March 24-28, 2003, Colorado Springs, USA*, page 59. Kluwer Academic Pub, 2003. 22

[ZZC02] T. Zseby, S. Zander, and C. Carle. RFC3334: Policy-Based Accounting. *Internet RFCs*, 2002. 11, 20, 22, 25

Declaration

I herewith declare that I have produced this work without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This work has not previously been presented in identical or similar form to any examination board.

The dissertation work was conducted from 2008 to 2012 under the supervision of Pablo García Bringas and Yoseba Koldobika Peña Landaburu at the University of Deusto.

Bilbao,

This dissertation was finished writing in Bilbao on Wednesday 15 February, 2012

This page is intentionally left blank