View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by UPCommons. Portal del coneixement obert de la UPC

Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Assessment strategies for resource sharing networks and ad hoc systems

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Núria León Anglès

DIRECTOR: Roc Messeguer Pallarès

DATE: December 14 th 2011

Title: Strategies assessment for resource sharing networks and ad hoc systems

Author: Núria León Anglès

Director: Roc Messeguer Pallarès

Date: December 14 th 2011

Overview

Traditionally, the formal modelling of systems has been done by using mathematical expression. Actually the current growing capacity of computers provides new tools to support the process of decision making in various disciplines and areas.

Nowadays, computer simulation has become an essential part of system modelling. By definition a computer simulation is an attempt to model almost all the imaginable real-life or hypothetical situation on a computer so that it can be studied to see how the system works. It is a tool to virtually investigate the behaviour of the system under study.

This is exactly the aim of this master thesis, examining the conduct of the system that is wanted to study using data mining methods.

In essence, that system is a distributed network with resource sharing, implanted in order to solve the insufficiency of CPU resources due to the constant increasing demand. The simulation tool is capable of running various cooperation games, strategies and topologies in a fully distributed environment in order to know what the relationship between these elements is.

Furthermore, the procedures used to handle the resultant information are data mining algorithms which combine tools from statistics and artificial intelligence with database management.

Then, the first part focuses on the study of the simulator tool; the parameters used and output results. The second task is to investigate the data mining methods and tools used to the implementation. The next objective is to adapt the simulator results to the analysis tools and execute them. And finally analyze the results in other to check the effectiveness to see the behaviour of the system.

.

Títol: Evaluació d’estrategies per compartir recursos en xarxes y sistemes adhoc

Autor: Núria León Anglès

Director: Roc Messeguer Pallarès

Data: 24 de Desembre de 2011

Resum

Tradicionalment, el modelat de sistemes es realitzava mitjançant l'ús d’expressions matemàtiques. Actualment, gracies al recent creixement de la capacitat dels ordinadors y els estudis realitzats han aparegut noves eines per donar suport al hora de realitzar la presa de decisions en quasi totes les disciplines.

Avui dia, la simulació per ordinador s'ha convertit en una part essencial del modelatge de sistemes. Per definició, una simulació és un intent de modelatge de gairebé totes les situacions imaginables de la vida real, perquè puguin ser estudiades. És una eina per investigar virtualment el comportament del sistema sota estudi.

Aquest és exactament el propòsit d'aquest projecte, examinar la conducta del sistema que es vol estudiar amb mètodes de mineria de dades.

En aquest cas, aquest sistema és una xarxa distribuïda amb intercanvi de recursos, implantat per tal de resoldre la insuficiència dels recursos de CPU, causada pel constant augment de la demanda de les aplicacions actuals.

A més, els procediments utilitzats per tractar la informació resultant del simulador son algorismes de mineria de dades que combinen eines d’estadística i d’intel·ligència artificial amb la gestió de bases de dades.

Així dons, la primera part del projecte es centra en l'estudi del simulador, els paràmetres utilitzats i els resultats de sortida d’aquest. La segona tasca consisteix a investigar els mètodes de mineria de dades i estudiar les eines utilitzades per a la posar-les en pràctica. El següent pas seria l'adaptació dels resultats del simulador a les eines d'anàlisi i la pròpia execució de les probes. Finalment, analitzar els resultats per tal comprovat l'efectivitat d’aquestes en per veure el comportament del sistema.

INDEX

INTRODUCTION 1

CHAPTER 1. PROJECT DESCRIPTION 3

1.1. The simulator 3

1.2. Simulator output data 4

1.2.1. The CPU 4

1.2.2. Cooperation 5

1.2.3. Clustering coefficient 5

1.3. Objectives 5

1.3.1. Knowledge objectives 6

1.3.2. Principal objectives 6

CHAPTER 2. BASIC CONCEPS 7

2.1. Data mining theory 7

2.2. Data mining methods 8

2.2.1. Regression 8

2.2.2. Feature selection 8

2.2.3. Classification 8

2.2.4. Clustering 9

2.3. Network topologies 9

2.3.1. Barabási-Albert (BA) model 11

2.3.2. Small-world 12

2.3.3. Torus 12

2.3.4. Waxman 13

CHAPTER 3. TOOLS 15

3.1. WEKA 15

3.1.1. The Graphical User Interface 15

3.1.2. The Command-line 17

3.2. Weka input datasets 18

CHAPTER 4. EXPERIMENTAL RESULTS 19

4.1. Data pre-processing 19

4.2. Attribute selection 26

4.2.1. Native 26

4.2.2. Meta-classifier 29

4.2.3. Attribute selection results 31

4.3. Regression 32

4.3.1. Lineal Regression 33
4.3.2. Tree M5P 34
4.4. Clustering 37
4.4.1. Distribution-based clustering 37
4.4.2. Centroid-based clustering 42
4.4.3. Clustering results 45
CHAPTER 5. CONCLUSIONS 47
5.1. Data mining conclusions 47
5.2. Network parameters conclusions 48
5.3. Personal conclusions 48
5.4. Study of environmental effects 49
BIBLIOGRAPHY 51

To my family, friends and i2cat colleagues whose
encouragement and constant support have
allow me to finish this master thesis.

But especially to Alberto Lopez
for stay there always.
.

FIGURE INDEX

Figure 2.1. Basic topologies: (a) Bus, (b) Ring, (c) Star, (d) Extended star, (e) Tree , (f) Mesh 10
Figure 2.2. Random network (a) and scale-free network (b). 12
Figure 2.3. Torus topology representation 13
Figure 2.4. Waxman representation 14
Figure 3.1 Weka GUI chooser 16
Figure 3.2. Weka explorer 17
Figure 4.1. Attributes topologies and Number of nodes representation 21
Figure 4.2. Attributes percentage of Mobiles and CPU of each Node representation 22
Figure 4.3. Attributes links of each node and cluster coefficient representation 22
Figure 4.4 Attributes queries of each node and satisfied queries representation 23
Figure 4.5. Attributes cooperation and CPU of neighbours representation 23
Figure 4.6. Attributes num of neighbours that are Desktops and type of nodes representation 24
Figure 4.7. Non discretized Cooperation 25
Figure 4.8. Discretized Cooperation 25
Figure 4.9. Score assigned to each attribute using native execution way 29
Figure 4.10. Score assigned to each attribute using meta-classifier execution way 31
Figure 4.11. Comparison between native and meta-classifier execution ways. 32
Figure 4.12. Linear Regression predicted cooperation and Cooperation 33
Figure 4.13. Linear Regression predicted cooperation vs. Cooperation Linear Regression 34
Figure 4.14 Test1 tree M5P Predicted cooperation vs. Cooperation 35
Figure 4.15 Test 1 Tree M5P results of Predicted cooperation vs. Cooperation 35
Figure 4.16. Complete tree M5P Predicted cooperation vs. Cooperation 36
Figure 4.17 Complete data Tree M5P predicted cooperation vs. Cooperation . 36
Figure 4.18. K-means separated data 38
Figure 4.19. EM-clustering results of CPU slots vs. Number of links of each node 39
Figure 4.20. EM-clustering results of of neighbours CPU slots vs. Number of neighbours belonging to the type desktops 39
Figure 4.21. EM-clustering results of Cooperation coefficient vs. Cluster classification 40
Figure 4.22. EM-clustering results of Topology vs. number of nodes 41
Figure 4.23. EM-clustering results of Cluster coefficient vs. queries 41
Figure 4.24. K-means separates data into Voronoi-cells, which assumes equal-sized clusters 43
Figure 4.25. Results M-means of variables instance number, topology number of nodes, percentage of mobiles, number of CPU Link 44
Figure 4.26. All attributes distributed by cluster colours 45

TABLE INDEX

Table 4.1. Total number of attributes and the corresponding description 21
Table 4.2. InfoGainAttributeEval results 27
Table 4.3. GainRatioAttributeEval results 27
Table 4.4. ReliefFAtributeEval results 28
Table 4.5. CsfSubsetEval results 28
Table 4.6. Results ReliefFAtributeEval 30
Table 4.7. Results CsfSubsetEval 30
Table 4.8. Tree M5P Simulation summary 37

INTRODUCTION

Traditionally, the formal modelling of systems has been done by using mathematical expression, which attempts to find analytical solutions enabling the prediction of the behaviour of the system from a set of parameters and initial conditions.

The growing capacity of computers and recent research in the computer science field provides new tools to support the process of decision making in various disciplines and areas.

Nowadays, computer simulation has become a useful part of modelling many natural systems in physics, chemistry, biology, human systems, economics as well as in engineering, helping to understand the operation of those systems. By definition a computer simulation is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. Changing variables in the simulation, predictions may be made about the behaviour of the system. It is a tool to virtually investigate the behaviour of the a system under study.

This is exactly the purpose of this master thesis, examining the conduct of the system under study using data mining methods. In this case that system is a distributed network with resource sharing, considering the existing problems of these applications, such as, the incentive of cooperation the topology, the information management and so one.

The purpose of this simulator [1] is to solve the insufficiency of resources due to the constant increasing demand, by using resource sharing across distributed networks; in this case the resource piece that can be remotely accessed from another computer will be the CPU slots. The simulation tool is capable of running various games, strategies and topologies in a fully distributed environment in order to know what the relationship between these elements and cooperation is.

Furthermore the procedures used to handle the resultant information are data mining algorithms which combine tools from statistics and artificial intelligence with database management.

The first chapter of this master thesis explains the project description talking about the simulator, the results provided by these one and the detailed objectives of this master thesis divided in to learning and principal objectives.

The second chapter focuses in to explain in basic words the theoretical concepts necessaries along this project and indispensable to be introduced for the correct understanding of the work done. In it are detailed the methods of data mining and the basic notions of the networks used by the simulator.

The thirds chapter talks about the tool used for the realization of this master thesis called Weka, her functionalities and applications. Moreover are introduced the inputs formats that it requires.

Then in the fourth chapter are presented the results of the experiment realised, starting for of the pre-processing stage and continuing showing the results of attribute selection, regression and clustering.

Finally, the extracted conclusion of the project realization is commented. It is followed of the environmental impact that may result in the project. To conclude this chapter the personal conclusions are shown.

CHAPTER 1. PROJECT DESCRIPTION

The context of this project is the simulation tool developed at the Master Thesis of the UPC (Technical University of Catalonia) entitled “Design and implementation of a simulator to explore cooperation in distributed environments” [1] developed by Davide Vega.

This simulator emulates a resource sharing scenario over a distributed network, considering the incentive of cooperation, the topology, node connectivity and the information management. In order to find conclusions to solve the insufficiency of computer resources due to the constant increasing demand, because the current applications needs, by using resource sharing across distributed networks. Concretely this simulator has been developed to study the effect of topology and incentive methods over cooperation on distributed systems games.

This project deals with the results obtained by the mentioned simulator, these information is processed and analyzed in order to obtain more statistics and conclusions, which can help in the future to improve the relations and the cooperation in a diversified distribution network.

The next sections will be focus on explaining the simulator purpose and the objectives of this master thesis in a more detailed point of view.

1.1. The simulator

The simulator [2] is a tool designed to evaluate cooperation over different games, strategies, topologies and extract statistics about several interesting parameters. The program imitates real distributed applications configuring different initial conditions in order to study the system behaviour, the discovery of the others net nodes and the quantity of resources that any node needs to share or demand.

Usually, the simulator played a Prisoner’s Dilemma game, using a Tit-for-tat [3] strategy on a distributed scenario. In order to simplify the simulator, in the game are not take into account the physical effects like network problems and all the effects not related with the cooperation between nodes. Moreover during the game the network topologies and the node placemen would not change.

These experiments are performed over a discrete scenario with 250 iterations. Verification experiments performed by Vega [1] have demonstrated that this number of cycles is enough to extract significant statistical conclusions after discarding the first 50 transitory iterations.

The simulation process starts by loading a synthetically created topology graph and by setting up the variables representing the environmental conditions. Then, the simulations are executed with only one independent variable, ensuring the results only reflect the impact of such parameter. Later the results are collected after running a considerably high number of simulations.

1.2. Simulator output data

The simulator output data is divided into different text files and sorted by number of nodes, topology identifier, percentage of mobiles used, the game strategy, which in this case will be Tit-for-tat [3], and degree of connectivity that is the number of links per node. In addition to these archives there are also the topology files that will give the information of placement and interconnection of nodes. Seven basic variables are obtained after processing the data, which we work with and combined them to understand the relation between them and the system's behaviour.

The basic resultant parameters are the maximum number of CPUs of each device depending on the type, which can be a mobile or desktop, the number of requests issued by each node and the number of these requests satisfied by the others, the clustering coefficient of each node, the number of links per node and finally the coefficient of cooperation that will be the most important parameter to consider and on which revolved our statistics and predictions. In order to find a method to increase the cooperation coefficient in real applications.

Further details focusing on the CPU concept used in the simulator and the cluster cooperation coefficients is shown in the next section.

1.2.1. The CPU

The CPU sharing game implemented pretend to replicate what happens in the real world into a resource sharing scenario. On a real scenario, every machine has a variable quantity of CPUs with different features (number of cores, frequency or hyper-threading/processing capabilities are some examples). When a node does not have the minimum necessary resources to perform a task, it requests a portion of neighbour's node resource to help it finish a task. After negotiation, some other nodes decides to cooperate, and others to decline. In any case, the requesting node knows how much resources have achieved and if is enough to perform the task; but if not, free all the pre-allocated resources. Finally, when the task is performed the nodes leave free all resources that were using.

In order to simplify the strategy, the game only focuses on CPU resource. Transform CPUs properties into a fixed value for each node, a natural number that represents their maximum available CPU that, in a certain moment can be used by it or temporally transferred to other node of the topology.

1.2.2. Cooperation

The evolution of cooperation [4] is one of emerging fields in the recent research of large-scale distributed networks. This kind of networks has no centralized control of behaviour of the nodes. For that reason, social policy methods have to be implemented to try to increase the overall cooperation by prioritizing the cooperation of the users rather than individual interests of each one.

There are many incentivation methods and the election of one or another has to be studied for each case. Some of them can encourage cooperation, and other may be able to opt for the motivation of competition. The topology can be used to develop the cooperation too.

The main real problem in distributed networks, takes place when the users consumes more than their can provide to other ones, this phenomenon is called the free-riding effect, and a user that perform this effect are called free-rider. That free-rider does not pay the cost of what they consume and that fact is, although they do not realize, more harmful than beneficial. The resultant net behavior will be worse than the expected one and the system become unsustainable.

1.2.3. Clustering coefficient

A clustering coefficient [5] is a measure which tries to evaluate how the nodes in a graph tend to cluster together. Evidence suggests that in most real-world networks, and in particular social networks too, nodes tend to create tightly knit groups characterised by a relatively high density of ties

In real-world networks, this probability tends to be greater than the average possibility of a randomly established tie between two nodes [7].

Two versions of this measure exists, the global and the local. The global version was designed to give an overall indication of the clustering in the network, whereas the local gives an indication of the embeddedness of single nodes.

1.3. Objectives

The objectives of this Project are divided in two groups. The first group of objectives contains the ones related to the topic knowledge and the learning process. The second group considers the main objectives which refers to the application of the gained knowledge and the realization of the principal project goals.

1.3.1. Knowledge objectives

- Study and understand the simulator results, all the parameters, topologies and the information needed to work with this resulting information.

- Study data mining methods, which parts are useful and its possibilities to take statistics from the simulator data.

- Learn to use the data mining tools, in this case Weka software, and all its execution ways and applications in this project.

- Study the way to provide automation to the process of pre-processing information, taking into account the current data formats.

- Search and learn the scripting languages that would be useful and efficient for our tasks.

1.3.2. Principal objectives

- Process the simulator output data to be possible to work with.

- Apply the learned concepts about attribute selection to study the importance and the influence of each variable in front of each others. Try to understand the influence of the topology or the cluster coefficient over the cooperation coefficient.

- Do predictions with stored data, by applying data mining regressions methods. Try to predict the cooperation coefficient behaviour, in order to use those predictions, in case that a new device might want to be placed at the best point in the network in the most efficient way.

- Apply clustering methods to the data in order to check the relations and the relations that exist behind each variable, and try to know the clustering criteria of each one.

- Extract the relation between nodes placement and connectivity with their achieved results, studying the topologies effects.

- Know how we can make grow the mobiles cooperation coefficient taking into account the relation between all the variables.

CHAPTER 2. BASIC CONCEPS

In order to follow this project satisfactorily it is necessary to introduce some basic concepts. First, are presented a summary about the algorithms used for process the information provided for the simulator. These methods are included in data mining theory besides functionalities that data mining provide to the users.

Finally, basic network topologies are described, with particular emphasis on the distributions used in the realization of this master thesis.

2.1. Data mining theory

Data mining [8], also called knowledge discovery in databases is the process of finding interesting and useful patterns and relationships in large volumes of data. Combining tools from statistics and artificial intelligence (such machine learning) with database management to analyze large digital data collections. Data mining is widely used in business (insurance, banking, retail), science research (astronomy, medicine), and government security (detection of criminals and terrorists).

Further, it could be divided into two types:

- Directed data mining, when you are trying to predict a particular data point like, for example, the sales price of a house given information about other houses for sale in the neighbourhood.

- Undirected data mining, when you are trying to create groups of data, or find patterns in existing data.

Modern data mining started in the mid-1990s [9], as the current computing capabilities, and the cost of calculation and storage finally reached a level where it was possible for companies to do it herself.

But as is known, the term data mining is referent to dozens of techniques and procedures used to examine and transform data. For this project are been studied only the ones that we have considered that could accomplish our purposes.

2.2. Data mining methods

Depending on the outcome that we want to obtain should be applied a different model. Some tools will allow to analyze our data in a comfortable way; others, helps to make predictions or decisions. In the next sections will be explained the methods used in more detail.

2.2.1. Regression

In statistics [10], regression includes any techniques for modelling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. More specifically, regression analysis helps oneself to understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed.

To sum up, the regression model is then used to predict the result of an unknown dependent variable resulting in a numerical output, given the values of the independent variables.

2.2.2. Feature selection

Feature selection [11], is the process of selecting a subset of original features according to certain criteria, is an important and frequently used dimensionality reduction technique for data mining [12] [13] [14]. It reduces the number of features, removes irrelevant, redundant, or noisy data, and brings the immediate effects for applications such a speeding up a data mining algorithm, and improving mining performance such as predictive accuracy and result comprehensibility.

Feature selection algorithms typically are classified into two categories: feature ranking and subset selection. Feature ranking ranks the features by a metric and eliminates all features that do not achieve an adequate score. Subset selection searches the set of possible features for the optimal subset.

2.2.3. Classification

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. A classification task begins with a data set in which the class assignments are known.

For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period of time. In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments, and so on. Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case.

2.2.4. Clustering

Cluster analysis or clustering is the task of organizing objects into groups whose members are similar in some way.

Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields. Cluster analysis itself is not an algorithm but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them.

2.3. Network topologies

In this thesis, Network topology is considered the pattern of interconnections of various elements in a computer network. The main elements are nodes and links. The node is a connection point that is attached to a network, and is capable of sending, receiving, or forwarding information over a communications channel. That channel is known as link which is the means of connecting one location to another for the purpose of transmitting and receiving information.

Network topologies may be physical or logical. Physical topology refers to the physical design of a network including the devices, location and cable installation. Logical topology refers to how data is actually transferred in a network as opposed or not to its physical design. In general physical topology relates to a core network whereas logical topology relates to basic network.

Topology can be understood as the shape or structure of a network. This shape does not necessarily correspond to the actual physical design of the devices on the computer network.

Essentially there are six different common topologies [15]: Bus, Ring, Star, Extended Star, Hierarchical or also know as Tree, and Mesh. The Figure 2.1 illustrates the different types of topologies just mentioned.

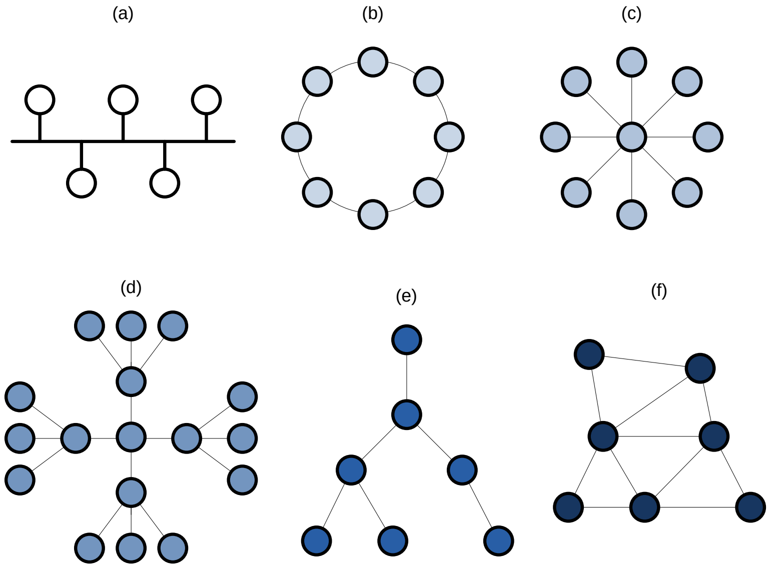


Figure 2.1. Basic topologies: (a) Bus, (b) Ring, (c) Star, (d) Extended star, (e) Tree , (f) Mesh

Starting for the first one, is easy to notice that all devices on the Bus Topology are connected using a single cable. The Bus Topology is less common these days. In fact, this topology is commonly used to network computers via coaxial cable.

On the other hand, The Ring Topology is a very interesting topology indeed. It is providing a collision-free and redundant networking environment where each node is connected only with two others, creating a closed circle. So, communication between nodes depends directly of the distance in term of number of hops among them.

The Star Topology works by connecting each node to a central device. This central connection allows us to have a fully functioning network even when other devices fail. The only real threat to this topology is that if the central device goes down, so does the entire network. The Extended Star Topology is a bit more advanced. Instead of connecting all devices to a central unit, we have sub-central devices added to the mix.

The Tree Topology has a node at top level of the hierarchy, connected to one or more nodes that are one level lower in the hierarchy, while each of the second level nodes will also have one or more other nodes that are one level lower in the hierarchy. The advantage of this topology is that access to information is very ordered, and although not all information must pass through the central node. In contraposition, the management of add, reallocate and remove the non-edge nodes is more complicated.

In mesh networks the nodes are connected with one or other without follow any pattern; it is possible that exist a high grade of redundancy. There are several types of this topology, where the distribution and connectivity of the nodes provided special properties to the network. Mesh topology is can be considered random when the probability of a node is connected to each of others is equal.

But in real world commonly networks are normally not so clearly labelled and simple such the ones described, most social, biological, and technological networks display substantial non-trivial topological features, with patterns of connection between their elements that are neither purely regular nor purely random. Taking to purely random a graph obtained by starting with a set of *n* vertices and adding edges between them at random. These real networks are called complex network.

Such features include a heavy tail in the degree distribution, a high clustering coefficient, assortativity or disassortativity among vertices, community structure, and hierarchical structure.

Once introduced the main types of networks, it is the time to go into more detail on the topologies used in this thesis. In next sections these ones are been explained in more detail.

2.3.1. Barabási-Albert (BA) model

The Barabási-Albert (BA) model [16] is an algorithm for generating random scale-free networks [17] using a preferential attachment mechanism.

The most notable characteristic in a scale-free network is the relative commonness of vertices with a degree that greatly exceeds the average. The highest-degree nodes are often called "hubs", and are thought to serve specific purposes in their networks, although this depends greatly on the domain.

In Figure 2.2 are shown the node distribution of a random network and scale-free network. In the scale-free network, the larger hubs are highlighted.

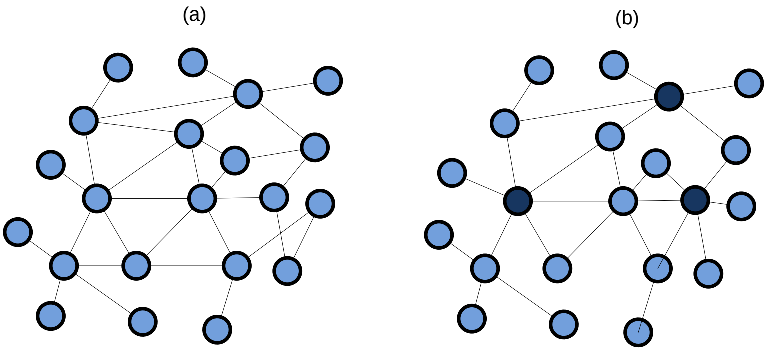


Figure 2.2. Random network (a) and scale-free network (b).

Scale-free networks are widely observed in natural and man-made systems, including the Internet, the World Wide Web, citation networks, and some social networks.

2.3.2. Small-world

In mathematics, physics and sociology, a small-world network is a type of mathematical graph in which most nodes are not neighbours of one another, but most nodes can be reached from every other by a small number of hops or steps.

Specifically, a small-world network is defined to be a network where the typical distance *L* between two randomly chosen nodes (the number of steps required) grows proportionally to the logarithm of the number of nodes *N* in the network, that is [22] :

$$L \propto \log N \tag{2.1}$$

In the context of a social network, this results in the small world phenomenon of strangers being linked by a mutual acquaintance. Many empirical graphs are well-modelled by small-world networks. For example, social networks, the connectivity of the Internet and plenty more all exhibit small-world network characteristics.

2.3.3. Torus

A grid network is a kind of computer network consisting of a number of computer systems connected in a grid topology.

In a regular grid topology, each node in the network is connected with two neighbours along one or more dimensions. If the network is one-dimensional, and the chain of nodes is connected to form a circular loop, the resulting topology is known as a ring. In general, when an *n*-dimensional grid network is connected circularly in more than one dimension, the resulting network topology is a torus, and the network is called "toroidal", as can be seen on Figure 2.3.

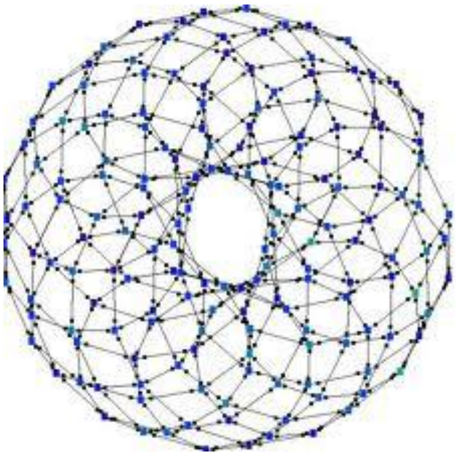


Figure 2.3. Torus topology representation

Conceptually, the torus topology can be considered to consist of a number of rings in different dimensions, viz. X_j, rings in the j-th dimension. The links connecting the nodes, and hence the corresponding rings as well, can be either unidirectional or bidirectional.

In geometry, a torus [18] is a surface of revolution generated by revolving a circle in three dimensional space about an axis coplanar with the circle. In most contexts it is assumed that the axis does not touch the circle, in this case the surface has a ring shape and is called a ring torus or simply torus if the ring shape is implicit.

2.3.4. Waxman

Waxman graphs [19] are a popular class of random graphs used for modelling the Internet topology, especially for the intra-domain part. When used for network modelling purposes their connectedness properties are particularly relevant, both for the characteristics of the realized graph and for the generation time.

In a Waxman graph, the nodes are uniformly distributed over a rectangular area, and links are added between the nodes through a random mechanism, where the probability that two nodes are directly connected decreases exponentially as their Euclidean distance increases.

The Figure 2.4 shows the representation of a Waxman network representation.

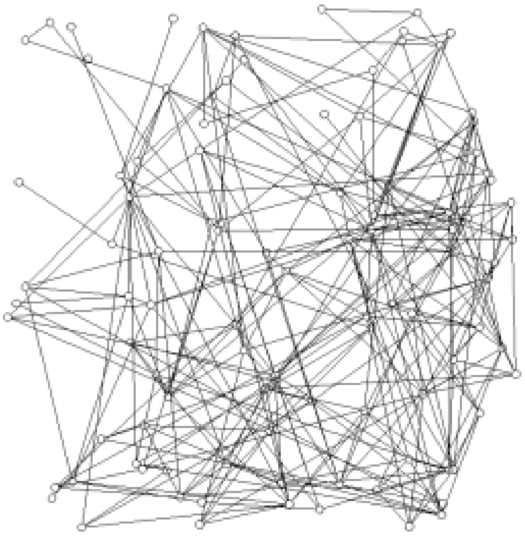


Figure 2.4. Waxman representation

In the original formulation by Waxman, such graphs have a pre-determined number N of nodes, which are uniformly distributed over a rectangular coordinate grid [20]; this means that every node has integer coordinates, while we consider the more general case, where a node can lie anywhere within the rectangular area. The probability that a direct link exists between the generic nodes u and v is related to the Euclidean distance d(u,v) between them by the expression

$$P(\{u,v\}) = \beta e^{-\frac{d(u,v)}{L\alpha}} \tag{2.2}$$

Where L is the maximum distance between two nodes, and α and β are two parameters in the (0,1] range. Larger values of β result in graphs with higher link densities, while small values of α increase the density of short links relative to longer ones.

The suitability for this purpose has been recognized also in the wider context of Internet topology modelling, in particular since its distance-dependent model of link formation among routers appears to describe remarkably well the real world [21].

CHAPTER 3. TOOLS

This chapter describes the tools involved in the realization of this project. In this case, the principal tool is Weka which provides a lot of functionalities, execution ways, different algorithms and methods. The first part focuses on the description of the graphical user interface and the functionalities of the program, such as processing, visualization and the implementation of statistics in text files. Secondly are explained an alternative execution way of this tool. Finally the required input format of data is explained.

3.1. WEKA

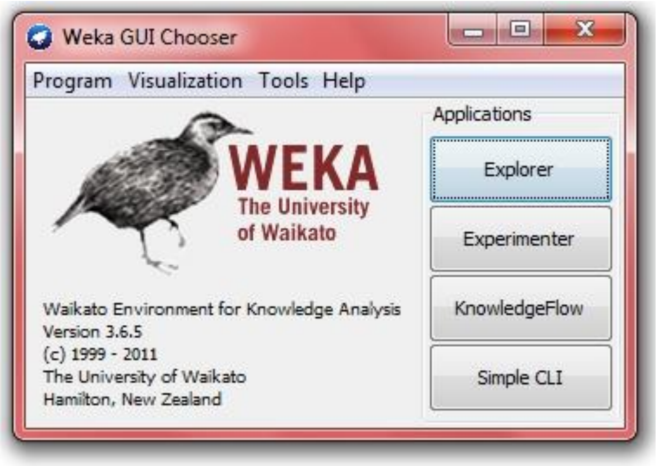
Waikato Environment for Knowledge Analysis (WEKA) [23], is a very popular open source software written in Java and developed at the University of Waikato, New Zealand, in 1997. It is available under the GNU General Public License.

Contains a graphical user interface (GUI) very useful for interacting with data files and represent de results in an intuitive form like curves or graphics. Unfortunately the GUI uses much more memory and moreover some functionalities are not available.

Thus for initial experiments the included graphical user interface is quite sufficient, but for in-depth usage the command line interface is recommended. It also has a general API, so you can embed WEKA, like any other library, in your own applications.

3.1.1. The Graphical User Interface

For having an easiest access to the main Weka’s functionalities the graphical user interfaces can be used. Its workbench contains a collection of algorithms for data analysis and predictive modelling together with a compilation of visualization tools. In Figure 3.1 can be seen the principal panel of the Weka Gui.



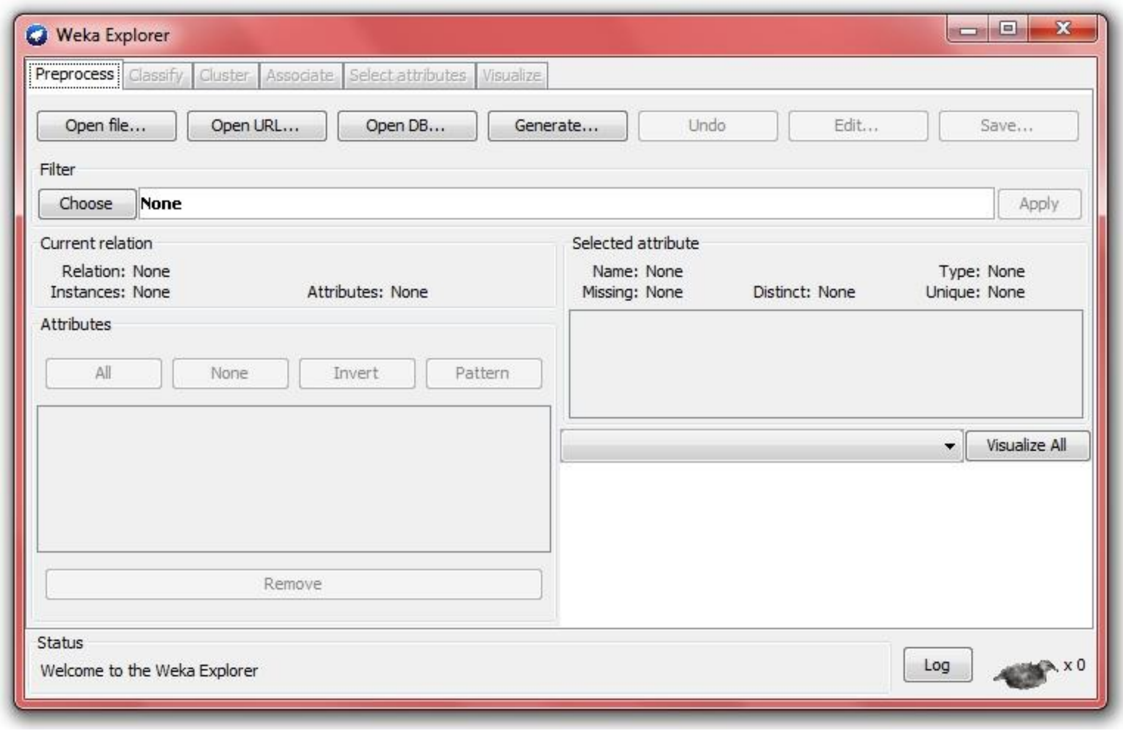
Waikato Environment for Knowledge Analysis
Version 3.6.5
(c) 1999 - 2011
The University of Waikato
Hamilton, New Zealand

Figure 3.1 Weka GUI chooser

The WEKA interface features several panels providing access to the main components:

- The “*Preprocess”* panel has facilities for importing data from files in different formats and for pre-processing this data using filtering algorithms. These filters can be used to transform the data and make it possible to delete instances and attributes according to specific criteria.
- The “*Classify”* panel enables the user to apply classification and regression algorithms, both called indiscriminately *classifiers* in Weka.
- The “*Associate”* panel provides access to association rule learners that attempt to identify all important interrelationships between attributes in the data.
- The “*Cluster”* panel gives access to the clustering techniques in Weka.
- The “*Select attributes”* panel provides algorithms for identifying the most predictive attributes in a dataset.
- The “*Visualize”* panel shows a scatter plot matrix, where individual scatter plots can be selected and enlarged, and analyzed further using various selection operators.

In Figure 3.2 Can be seen the Weka interface, on top are panels to select that give access to the others components.



None

Relation: None
Instances: None
Attributes: None

Name: None
Type: None
Missing: None
Distinct: None
Unique: None

Welcome to the Weka Explorer

Figure 3.2. Weka explorer

3.1.2. The Command-line

Using the command-line can be used all the functionalities of Weka. Furthermore, using this way of execution it is possible to increase the maximum heap size for your java engine. Usually the default setting of 16 to 64MB is too small and could get out memory errors. But an possible inconvenient it is that has to be specified explicitly set CLASSPATH via the -cp command line option, it can result cumbersome to use.

Cross-validation has to be used if one only has a single dataset and wants to get a reasonable realistic evaluation. Setting the number of folds equal to the number of rows in the dataset will give one leave-one-out cross-validation (LOOCV).

3.2. Weka input datasets

In the machine learning context the concept of datasets is a very basic idea, which is equivalent a two dimension table. Each row is called Instance and consists of a number of attributes (columns) any of which can be:

- *Nominal*: one of a predefined list of values
- *Numeric*: means a real or integer number, integer and real types are treated as numeric.
- *String*: an arbitrary long list of characters, enclosed in ”double quotes”

Typically the external representation of an Instances class is an ARFF (Attribute-Relation File Format) file is an ASCII text file, which consists of a header describing the attribute types and the data as comma-separated list. But Weka can read files in a variety of formats in addition to WEKA’s ARFF format, also accepts CSV format, C4.5 format, or serialized Instances format.

By default, the last attribute is considered the class or target variable, i.e. the attribute which should be predicted as a function of all other attributes. Have to be considered, that every data mining method can need the attribute class in a different format.

CHAPTER 4. EXPERIMENTAL RESULTS

In this chapter are presented the results obtained during the execution of this master thesis.

First are introduced the pre-processing phase, talking about the necessity to adapt the output of the network simulator.

The next section explains what happens once the data has been processed. In it can be seen the results obtained executing the Weka tool using methods of attribute selection, used to know the importance of each variable. Later are detailed the regression process in which are made predictions in order to prognosticate the behaviour of a variable. At last are shown the results obtained in the clustering classification phase where has been search relations between the variables.

4.1. Data pre-processing

As explained in Chapter 1, the simulator output data has a particular format and is not compatible with the required format for input data in Weka (explained in chapter 3).

Originally the simulator results are stored in a large number of text files neatly classified, one for simulation, in which have been used a different number of nodes, and also varies the topology and the percentage of mobiles in the network.

The simulator in addition, permit to change the strategy of cooperation, but in this project had been worked with Tit-for-tat [3], because the purpose is to find ways to increase the cooperation whatever the strategy chosen. Moreover, the connectivity degree of each node has been fixed, choosing a maximum of 6 links per node. Besides to the data files mentioned above, topology's information is recorded in another file set that contain the node interconnection, specifying which nodes are neighbours.

Should be noticed, that the results obtained from the network simulator have too much data to be direct processed, have redundant information and the format is not compatible with the Weka input requirements.

Ones the simulator outputs had been studied and understand all of this information had to be classified tidied and format matched. So, these files have been read and combined implementing various scripts, developed in Perl language [24]. Perl was chosen because it is considered very effective and quick to implement for processing text files.

Seven basic variables are obtained after processing the original data. The basic resultant parameters are the maximum number of CPU slots of each device depending on the type. This type is another variable and can be a mobile (lower than 3 CPU slots) or desktop (greater than 3 slots). Other two variables are the number of requests issued by each node and the number of these requests satisfied by the others. The clustering coefficient of each node and the number of links per node are repressed too. Finally the cooperation coefficient can be found, which is considered the most important parameter to be studied.

Moreover, realizing some calculations with the variables have been obtained two more attributes. The first one is the number of neighbours corresponding to desktop type, and the other one is the number of CPU slots that every nearby nodes have.

Up to this point have been obtained apparently completely valid 9 variables, which saved in different format files compatible with Weka, some of these archives have 4096 lines, which as explained in section 3.2 corresponds to 4096 instances in Weka, and also have 9 attributes. This file apparently seems to have a too much information and initially are unknown what effects will be produced to Weka, so it must to be checked.

After doing that the results show that Weka works properly and is able to process more volume of data. For that reason the present data files has been joined as much as possible testing if Weka is able to work properly in every junction. Finally had been proved that Weka is capable to process the number of data stored in only one large file, but obviously the test needs more time and computer resources to be executed.

This final complete file contains data from a lot of simulations containing different topologies, number of nodes and the percentage of mobiles used. The idea is use as much information as possible to be possible to understand the relationship between the entire variables and get more conclusions considering the system’s behaviour.

At this last phase, had been added three more attributes. These variables are the topology used for the simulation (see section 2.3), the number of nodes and the percentage of mobiles allowed in the network. The resultant file contains 91728 instances (rows), 12 attributes (columns) and has the properly format. So to sum up the final dataset are has the attributes shown in Table 4.1.

Num	Attribute	Possible Values	Description
1	maxCPU	Numerical	Maxim number of CPU slots of the node
2	Queries	Numerical	Number of queries realized by the node
3	Links	Numerical	Number of connections between neighbours
4	ClusterCoef	Numerical	Cluster Coefficient
5	SatisfiedQueries	Numerical	Number of queries that had been satisfied
6	Type	Desktop or Mobile	Type of dispositive
7	Cooperation	Numerical	Grade of cooperation
9	#NeighborsDesktop	Numerical	Number of neighbour desktop of the node
10	NeighborsCPU	Numerical	Addition of neighbour CPU
8	Topology	Barabasi, SmallWorld, Torus or Waxman	Type of topology
11	perMobiles	20, 40, 60 or 80	Percentage of mobiles in the simulation
12	#Nodes	125, 512,1000 or 4096	Number of nodes in the simulation

Table 4.1. Total number of attributes and the corresponding description

The figures below show the representation of all the attributes. In all the figures the four colours represent the four types of topologies that exist in the datasets. The number of nodes used for the simulations is 125, 512, 1000 or 4096 represented in Figure 4.1. as well the topologies mentioned. Not all the attributes are numerical, some of them are nominal such topology and type.

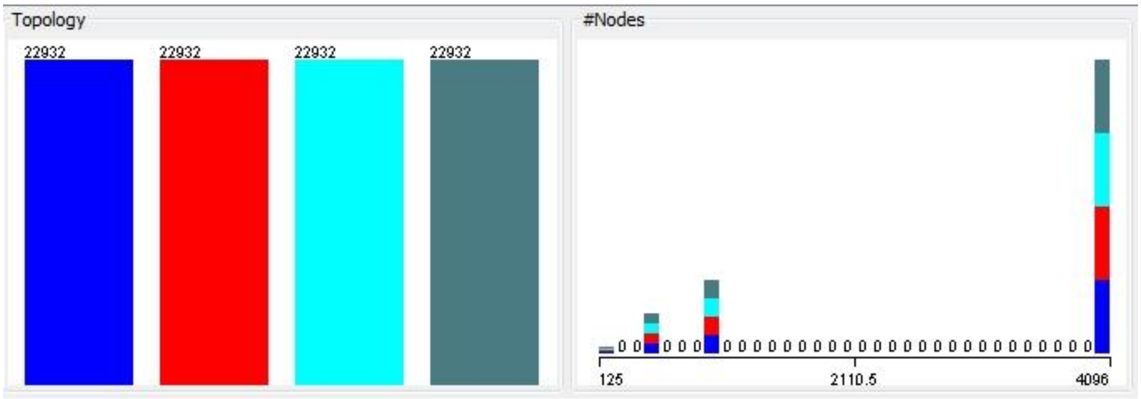


Figure 4.1. Attributes topologies and Number of nodes representation

In Figure 4.2 are represented the percentage of mobiles used for the simulations, which are 20%, 40%, 60% or 80% in respect of all network devices as can be seen all the topologies are simulated for the 4 possible values of this variable. On the other hand, in the right can be seen the number of CPU slots, the lower values are the mobile devices and the higher are the desktop devices.

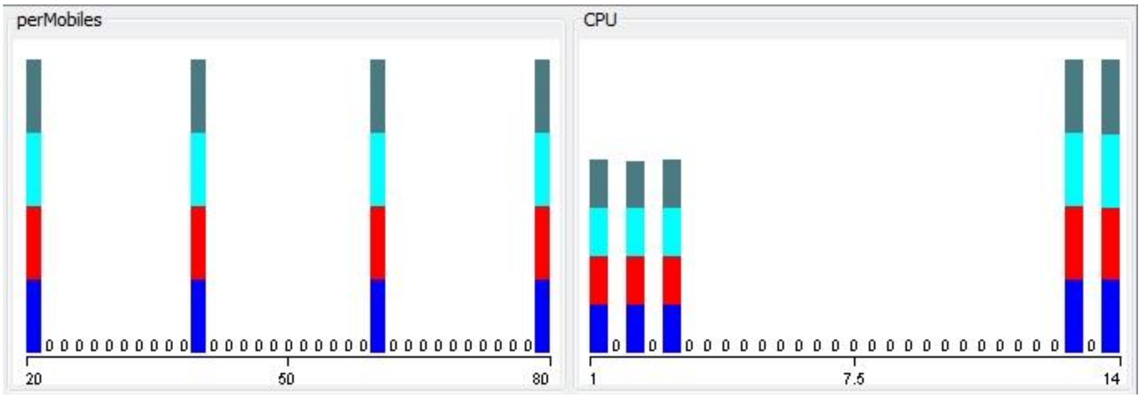


Figure 4.2. Attributes percentage of Mobiles and CPU of each Node representation

In Figure 4.3 the left chart introduce the number of links that has every node, and the right chart shows the cluster coefficient of each node, note that the torus topology (see section 2.3.3) has a constant cluster coefficient equal to one.

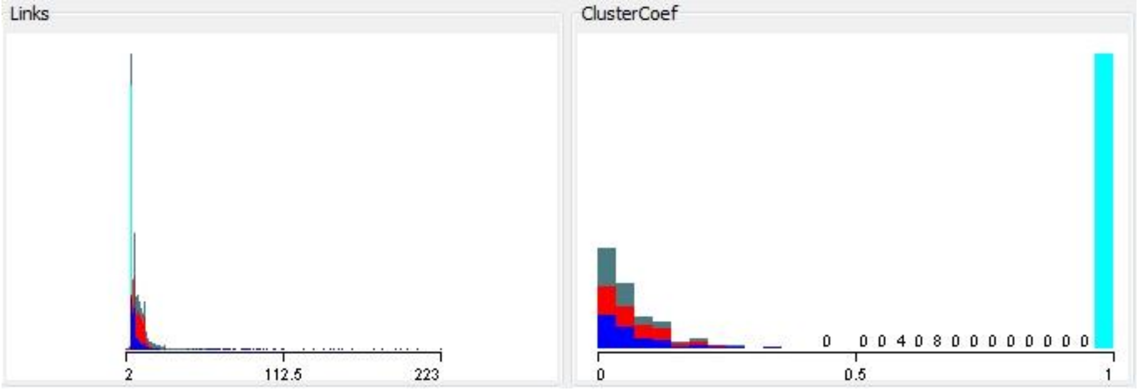


Figure 4.3. Attributes links of each node and cluster coefficient representation

The Figure 4.4 shows the graphical representation of the queries realized and satisfied in return for each node.

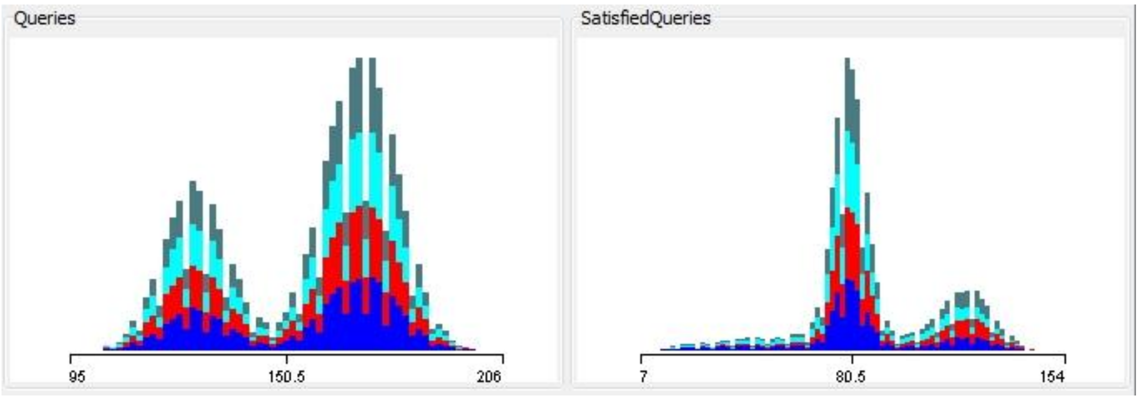


Figure 4.4 Attributes queries of each node and satisfied queries representation

As mentioned before the most important parameter is the cooperation coefficient, which is represented in Figure 4.5, and in the right is shown the graphic of the CPU slots number, resulting from the addition of the CPU slots of each node neighbours.

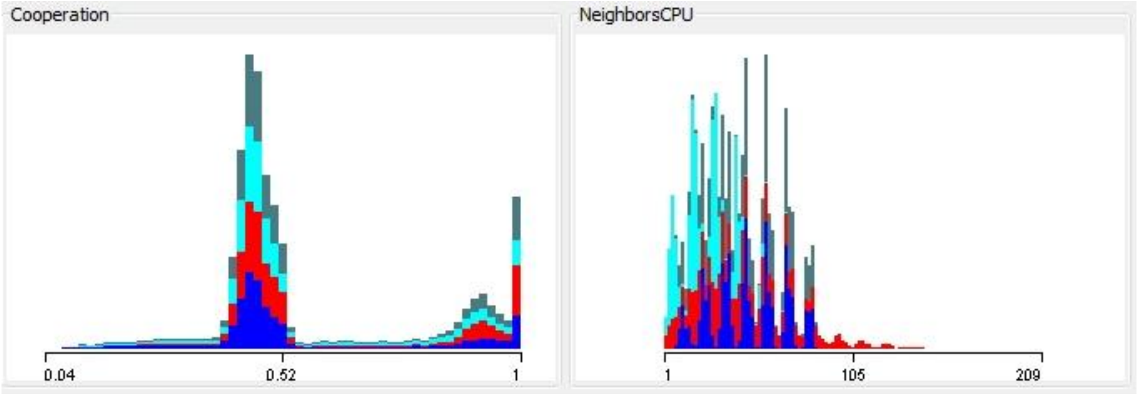


Figure 4.5. Attributes cooperation and CPU of neighbours representation

Finally the Figure 4.6, present in the left for each node the number of neighbours that are desktop and in the right the type of each node, that as has been mentioned can be mobile or desktop type.

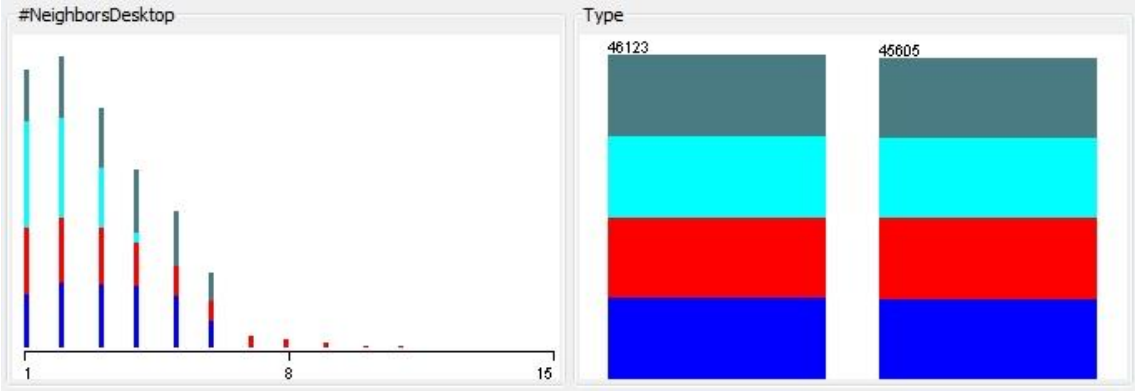


Figure 4.6. Attributes num of neighbours that are Desktops and type of nodes representation

Once the data are adapted and studied, is the moment to begin to work with. But as can be deduced these files have too much results and surely would have redundant information that would be not beneficial for testing. For that reasons and other ones, it is necessary to process the data and find ways to select only the useful information.

The correct way to perform this is using methods allowed the Weka workbench, this ones are explained in more detail in section 4.2.

The methods that allow to do this, require that the attribute format, of considered as the class, must to be nominal. In our case the variable that has this function is the cooperation and is a numerical one. For that reason is necessary to discretize the variable. Weka allows to discretize the attributes implement a concrete filter.

Is very important to note, that the discretization frequency has to be always constant. In other case the shape of the variable would be totally different than the expected one.

The figures below represent the same variable. Besides it is important to note, that the first one (Figure 4.8) is the original numerical attribute and the second one (Figure 4.7) had been discretized, as can be seen both conserve the same shape.

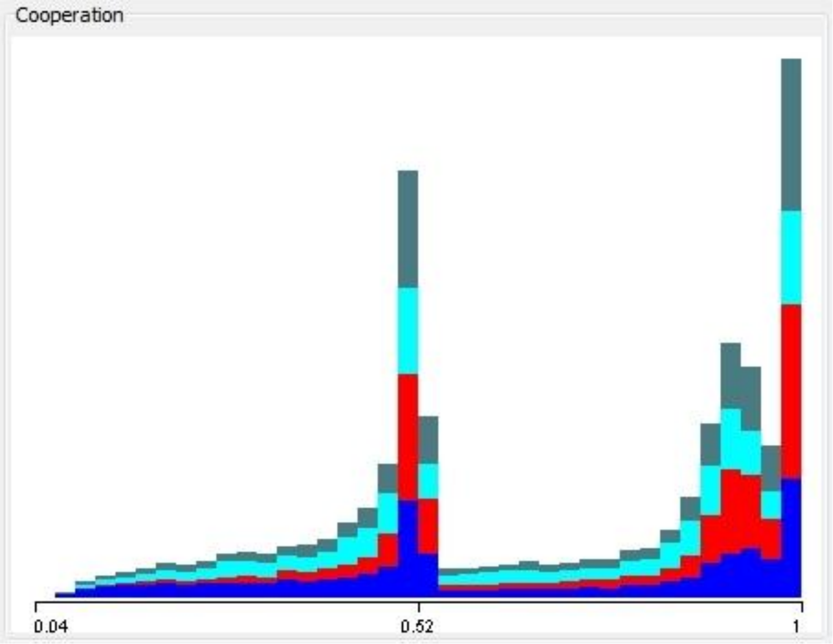


Figure 4.7. Non discretized Cooperation

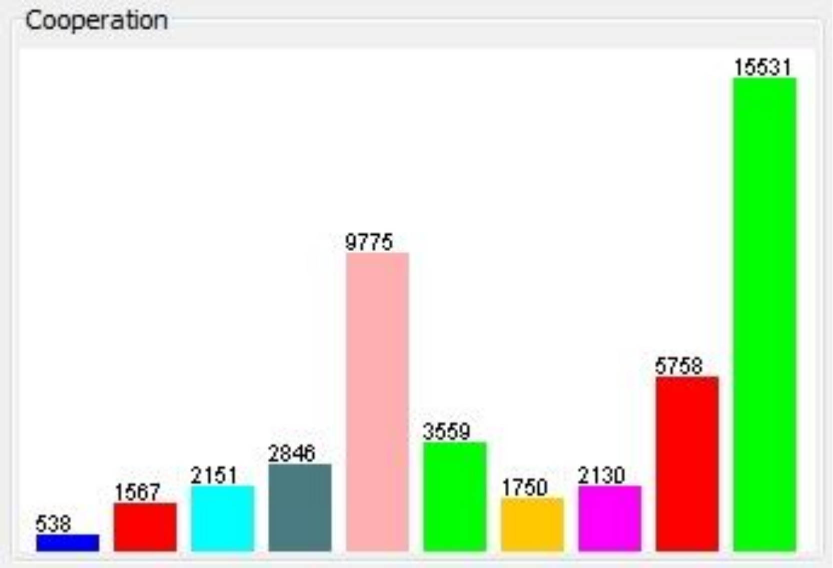


Figure 4.8. Discretized Cooperation

4.2. Attribute selection

As have been mentioned before, the data have to be as less as possible redundant information to obtain the best possible results. This is exactly the purpose of the Weka section dedicated to attribute selection, explained in section 2.2.2.

Exist a large number or attribute selection methods, for that project have been studied many of them, in this section only will be shown the most remarkable ones.

It is important to note that not everything is possible from the GUI, and in this case, the methods that are used, which will be explain in more detail in the next section, have had to be executed in a command-line-based environment.

In Weka, you have three options of performing attribute selection from command-line:

1) Low-level API usage: Using the native approach, with the attribute selection classes directly.
2) Using a meta-classifier: for performing attribute selection using a classifier next to attribute evaluator.
3) Using a filter: for pre-processing the data and save only the selected information.

For this Project have been used the first and the second options, which now will be explained.

4.2.1. Native

This method prints in the console only the ordering of the attributes or retrieve the indices of the selected attributes instead of outputting the reduced data.

Furthermore using the attribute selection classes directly is possible to obtain some additional useful information, like number of subsets with best merit, ranked output with merit per attribute.

The attribute selection classes are located in the “weka.attributeSelection” package.

To determine which attributes are the relevant ones have been used a large number of attribute selection methods. At the next sections that methods will be explained in addition of its results.

4.2.1.1. *InfoGainAttributeEval*

The method InfoGainAttributeEval evaluates the worth of an attribute by measuring the information gain with respect to the class. Note that the attribute class has to be numeric and situated at the end of the attribute list for properly operation.

The Table 4.2. the extracted results executing InfoGainAttributeEval, the results are represented from most important to lower.

Average relevance	Attribute
0.629	CPU
0.553	Queries
0.23	NeighborsCPU
0.189	perMobiles
0.074	#NeighborsDesktop
0.033	Topology
0.014	Links
0.005	ClusterCoef
0	#Nodes

Table 4.2. InfoGainAttributeEval results

4.2.1.2. *GainRatioAttributeEval*

This method evaluates the worth of an attribute by measuring the gain ratio with respect to the class. As the previous case, the class attribute has to be numeric and situated in last position of the classes list to be succeeded executed.

Table 4.3. GainRatioAttributeEval resultslisted the attributes ranked by importance using algorithm GainRatioAttributeEval.

Average relevance	Attribute
0.397	CPU
0.263	Queries
0.103	perMobiles
0.067	NeighborsCPU
0.034	#NeighborsDesktop
0.016	Topology
0.009	Links
0.003	ClusterCoef
0	#Nodes

Table 4.3. GainRatioAttributeEval results

4.2.1.3. ReliefFAttributeEval

ReliefFAttributeEval evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. In this case, can operate on both discrete and continuous class data and the results are shown in Table 4.4. ReliefFAtributeEval

Average relevance	Attribute
0.168	CPU
0.079	Queries
0.069	perMobiles
0.039	#NeighborsDesktop
0.032	NeighborsCPU
0.005	Topology
0.004	Links
0.003	#Nodes
0.001	ClusterCoef

Table 4.4. ReliefFAtributeEval results

4.2.1.4. CfsSubsetEval

The method CfsSubsetEval evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.

Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.

Table 4.5 present the attributes but this time are not ranked, instead of this are presented the four more important al the top of the table.

Number of folds (%) attribute	Attribute
10(100 %)	CPU
10(100 %)	Queries
10(100 %)	perMobiles
10(100 %)	Topology
0(0 %)	#Nodes
0(0 %)	Links
0(0 %)	ClusterCoef
0(0 %)	NeighborsCPU2
0(0 %)	#NeighborsDesktop

Table 4.5. CsfSubsetEval results

4.2.1.5. *Results comparison for native execution*

Ones all algorithms are applied, a count have been made in order to choose which are the main attributes. In the next Figure 4.9 are resumed the attribute ranking, can be seen the times that every variable have been choose by the previous methods.

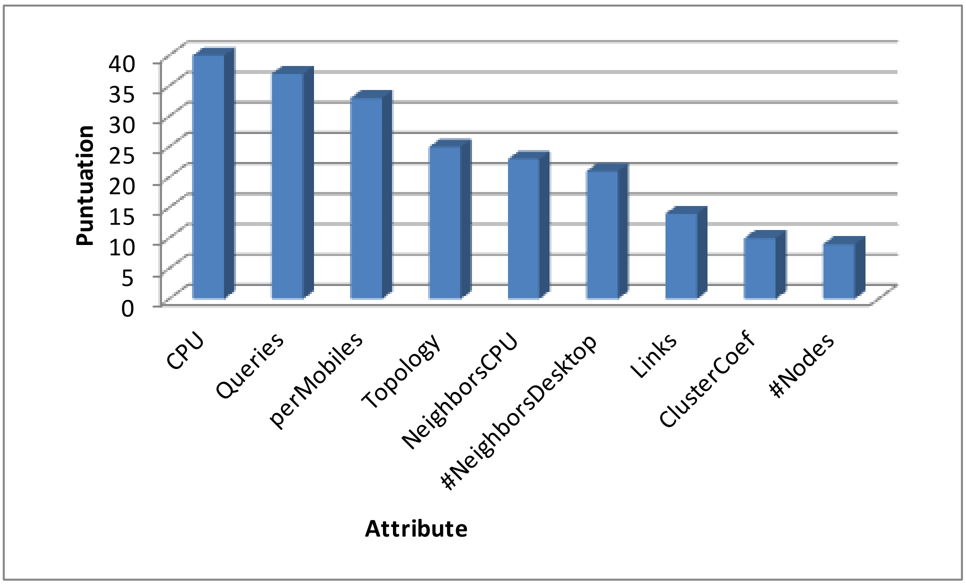


Figure 4.9. Score assigned to each attribute using native execution way

The four principal attributes using the native execution way are CPU, Queries, percentage of mobiles used and the topology. Then, these can be seen very clearly as the most influential attributes is this set of results.

On the other hand, are expect that in this data set exist another set of attributes considered expendable and that can be clearly superseded. This group of variables are the number of nodes in the network, the clustering coefficient and the number of links of each node.

4.2.2. Meta-classifier

The meta-classifier called as AttributeSelectedClassifier; uses a search algorithm to perform the attribute selection and a base-classifier to train on the reduced data.

This makes the attribute selection process completely transparent and the base classifier receives only the reduced dataset.

The meta-classifier execution way and the methods GainRatioAttributeEval and InfoGainAttributeEval used in the native way are not compatible because this methods needs the class attribute in nominal format and the classificators used require numerical class. For that reason is impossible to compare these methods in this execution way with the other used above.

4.2.2.1. ReliefFAttributeEval

As showed in the native way on Table 4.6 is possible to see the different variables ordered by importance.

Average relevance	Attribute
0.1775461	#NeighborsDesktop
0.0110234	NeighborsCPU
0.0000162	CPU
0	Topology
-0.0000139	Links
-0.0000304	perMobiles
-0.0009328	#Nodes
-0.0034105	ClusterCoef
-0.0046187	Queries

Table 4.6. Results ReliefFAtributeEval

4.2.2.2. CfsSubsetEval

This time the four chose variables are the ones showed on Table 4.7. As we can see the CPU are not chosen instead of the native way and the cluster coefficient is considered as important.

Number of folds (%) attribute	Attribute
10(100 %)	Topology
10(100 %)	Queries
10(100 %)	perMobiles
10(100 %)	ClusterCoef
0(0 %)	#Nodes
0(0 %)	Links
0(0 %)	CPU
0(0 %)	NeighborsCPU2
0(0 %)	#NeighborsDesktop

Table 4.7. Results CsfSubsetEval

4.2.2.3. Results comparison of meta-classifier execution

Comparing all the results obtained using meta-classifier execution way the principal variables are topology, percentage of mobiles used, queries and instead of CPU this time can find the cluster coefficient (see Figure 4.10)., instead of which are selected in the native execution way.

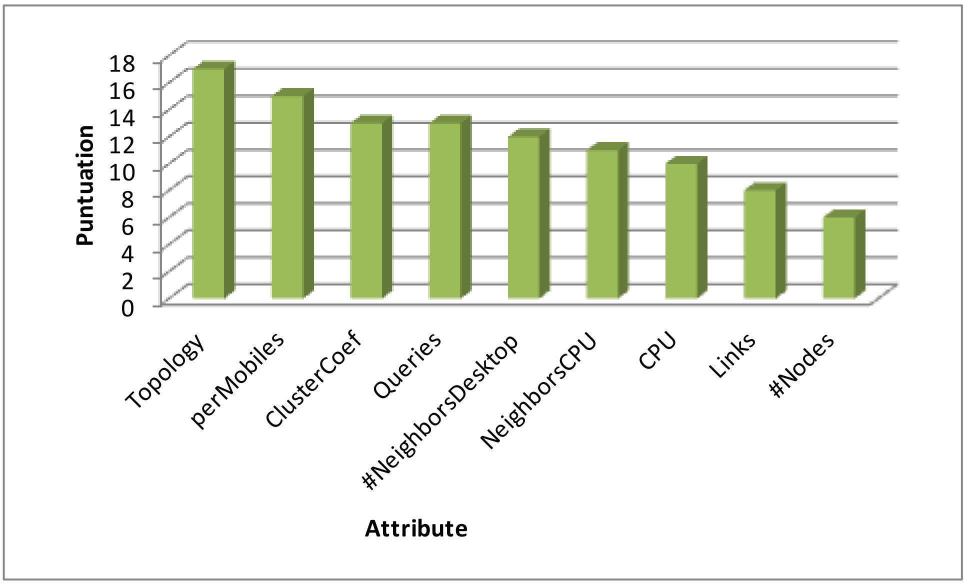


Figure 4.10. Score assigned to each attribute using meta-classifier execution way

4.2.3. Attribute selection results

Thus, changing the mode of execution in broad terms the results are almost equal, in term of chosen attributes. As can be seen in the chart below (Figure 4.11), the main attributes are the topology, the Queries and the Percentage of mobiles used.

The only attributes that comparing the two methods of execution does not considered as important are the CPU slots, in the case of the native method, and cluster coefficient, in the case of meta-classifier. But the number of nodes and links matched as unimportant.

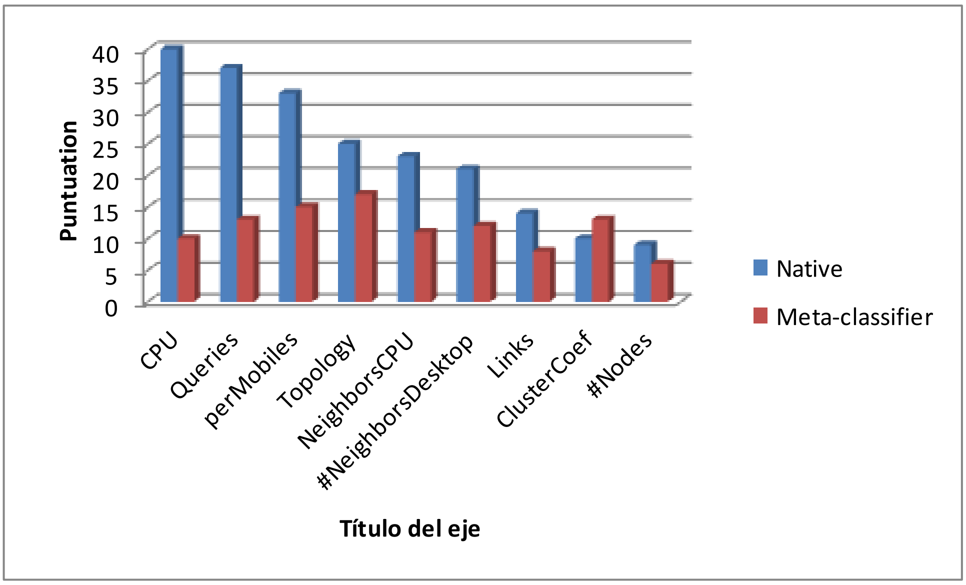


Figure 4.11. Comparison between native and meta-classifier execution ways

A priori the results obtained in the execution of attribute selection methods are considered both valid and logic, but at this point it is very difficult to say which are the most accurate. For that reason in the posteriors experiments are used the two set of attributes in order to choose the one which shows the best results.

Also after the execution of the algorithms for attributes selection was observed that several of the variables analyzed did not provide information and others were redundant. As the case of the data referred to the type desktop instances, had been notice that the variable that really provided useful information are the mobile instances. For that, reason has been made the rest of the tests using only the instances of mobiles devices.

4.3. Regression

As have been explained in section 2.2.1, the regression is a technique to predict the behaviour of an unknown dependent variable using the values of the independent variables obtaining the numerical valour that this must to take.

Doing the experiments has been noticed that the results are more favourable as more data has been used. In other words, when has been used the attributes obtained applying attribute selection the prediction of the cooperation coefficient has not been the expected one. On the other hand, when applied almost the complete useful data, depreciating only the direct redundant attributes, the results has been very optimistic.

The cluster methods that are considered that are most appropriate for our purpose are linear regression and a decision tree called M5P.

4.3.1. Lineal Regression

In linear regression, data are modelled using linear functions, and unknown model parameters are estimated from the data.

First have been used regression experiments with only the variables obtained after the attribute selection, but as can be seen in Figure 2.1 the resultant predicted coefficient cooperation, does not seems the original one.

First of all the attributes used in Figure 4.12 are:

- Topology
- perMobiles
- CPU
- Queries
- Cooperation

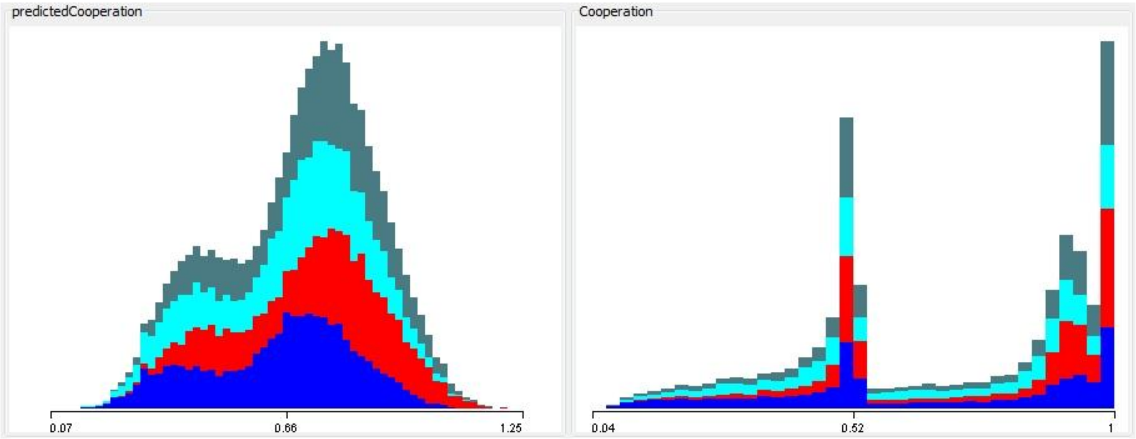
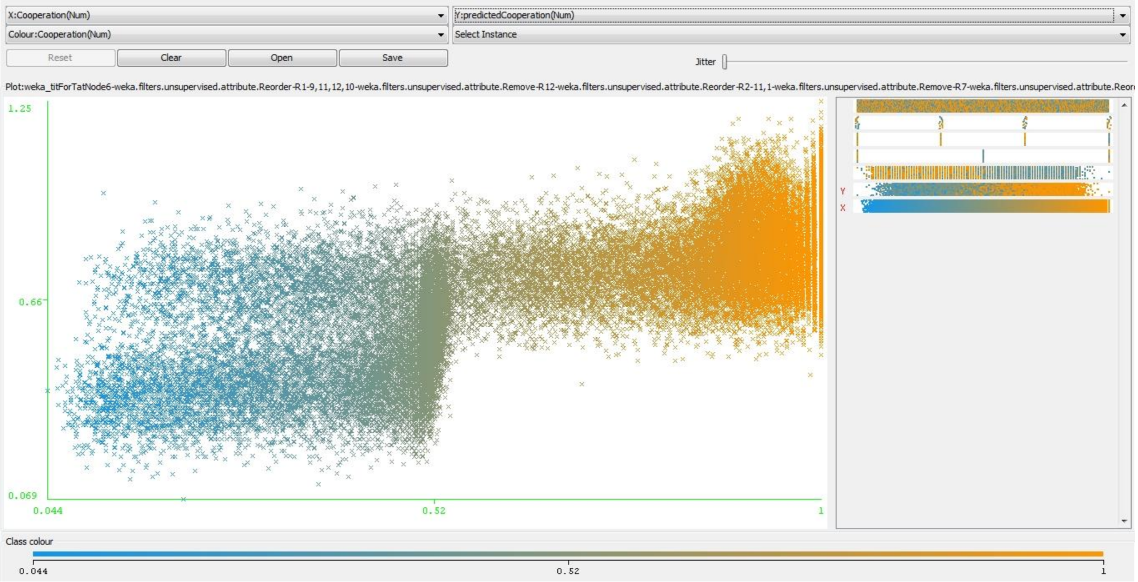


Figure 4.12. Linear Regression predicted cooperation and Cooperation

Representing the results in a different way can be seen more clearly, as shown in Figure 4.13, the distribution should be linear but it is not the case.



X:Cooperation(Num)

Y:predictedCooperation(Num)

Colour:Cooperation(Num)

Select Instance

Plot:weka_titForTatNode6-weka.filters.unsupervised.attribute.Reorder-R1-9,11,12,10-weka.filters.unsupervised.attribute.Remove-R12-weka.filters.unsupervised.attribute.Reorder-R2-11,1-weka.filters.unsupervised.attribute.Remove-R7-weka.filters.unsupervised.attribute.Reor

Figure 4.13. Linear Regression predicted cooperation vs. Cooperation Linear Regression

It is obvious that using that set of data and method the results are not the expected ones.

4.3.2. Tree M5P

This method is based on the original algorithm M5 which was invented by R. Quinlan and Yong Wang made improvements.

It is a structured regression is build on the assumption that the functional dependency is not constant in the whole domain, but can be approximated as such on smaller subdomanis. The most attractive advantages is that by dividing the function being induced into linear patches, providing a representation that is reproducible and easy comprehensible by practitioner. To run this algorithm the class attribute must be numeric.

Figure 4.14 shows the results using the same variables that in the previous section, which are:

- Topology
- perMobiles
- CPU
- Queries
- Cooperation

The results obtained are slightly better than obtained executing the linear regression (see Figure 4.14), this is obvious because this is a more accurate algorism. But instead of this the results are still not as expected.

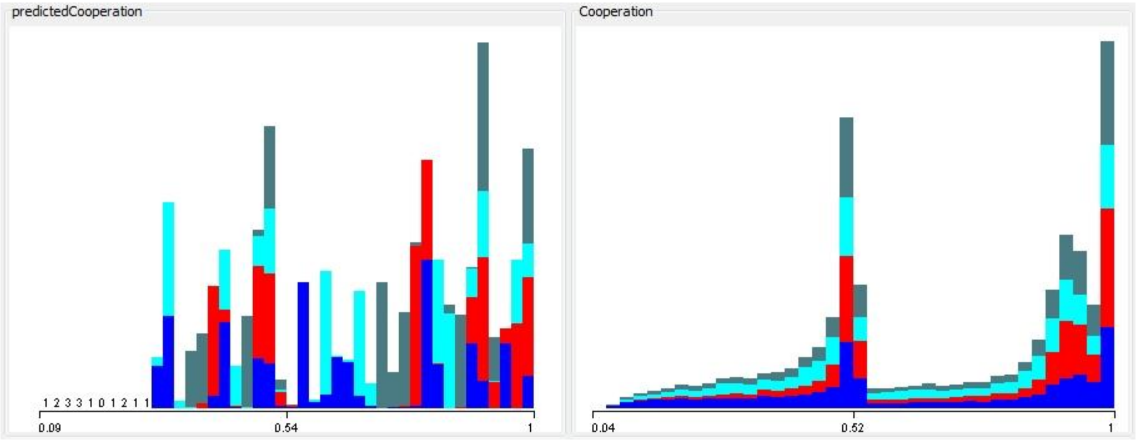


Figure 4.14 Test1 tree M5P Predicted cooperation vs. Cooperation

In Figure 4.15 the representation is still very far to become a linear distribution, that is not what are desired.

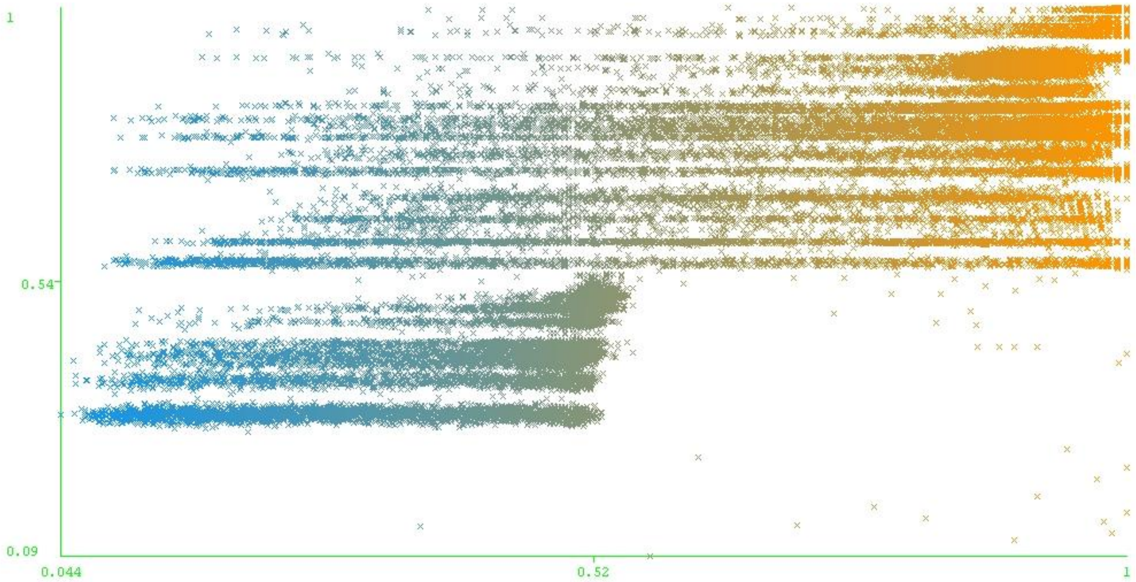


Figure 4.15 Test 1 Tree M5P results of Predicted cooperation vs. Cooperation

But instead, if do not use the attributes chosen in the selection of attributes and using the same prediction algorithm only removes the essential attributes leaving the following ones, then the attribute list is :

- Topology
- #Nodes
- perMobiles
- CPU
- Links
- ClusterCoef
- Queries
- NeighborsCPU
- #NeighborsDesktop
- Cooperation

The results are very favourable and the predict cooperation is very similar to the original cooperation, as can be seen in Figure 4.16.

In Figure 4.17 the distribution is not entirely linear but is certainly much closer than the other ones approximations. And considering that not all the instances have been classified, because the algorithm has been considered them less important, the results are very favourable.

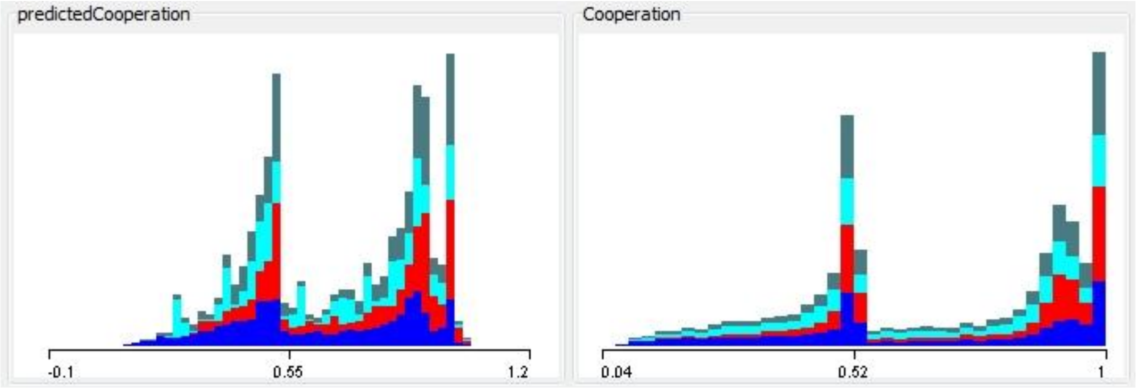


Figure 4.16. Complete tree M5P Predicted cooperation vs. Cooperation

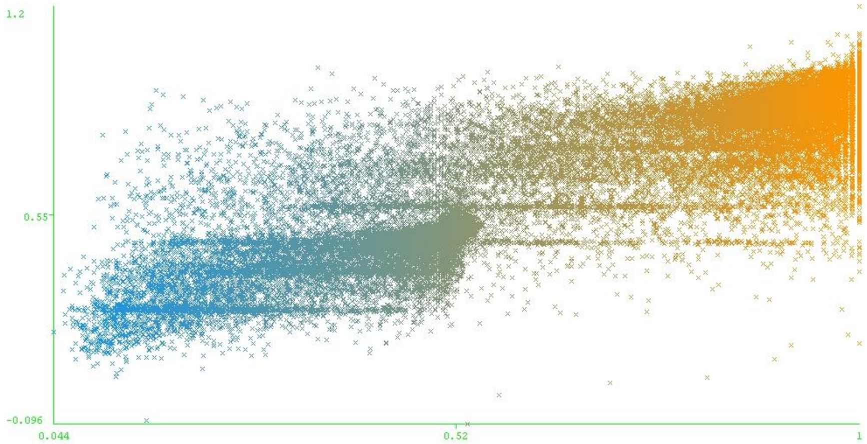


Figure 4.17 Complete data Tree M5P predicted cooperation vs. Cooperation

To be more precise showing the Figure 4.16 and Figure 4.17. is essential to give the values extract from the execution, showed in Table 4.8. Tree M5P Simulation summary which can be seen the error and grade of correlation between the variables. The correlation coefficient is a mathematical measure to show how much one variable can expected to be influenced by changes in another. The second value is the root mean square error which is a statistical measure of the magnitude of a varying quantity. And the last one is the relative absolute error which takes the total absolute error and normalizes it by dividing by the total absolute error of the predictor.

Summary	
Correlation coefficient	0.8888
Root mean squared error	0.0755
Relative absolute error	0.1185

Table 4.8. Tree M5P Simulation summary

4.4. Clustering

The appropriate clustering algorithm and parameter settings depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery that involves try and failure. It will often be necessary to modify pre-processing and parameters until the result achieves the desired properties.

4.4.1. Distribution-based clustering

The most prominent method is known as expectation-maximization algorithm, or to short EM-clustering. Here, the data set is usually modelled with a fixed number of Gaussian distributions, to avoid overfitting, that are initialized randomly and whose parameters are iteratively optimized to fit better to the data set.

This will converge to a local optimum, so multiple runs may produce different results. In order to obtain a hard clustering, objects are often then assigned to the Gaussian distribution they most likely belong to, for soft clustering this is not necessary.

Distribution-based clustering is a semantically strong method, as it not only provides you with clusters, but also produces complex models for the clusters that can also capture correlation and dependence of attributes. However, using these algorithms puts an extra burden on the user: to choose appropriate data models to optimize, and for many real data sets, there may be no mathematical model available the algorithm is able to optimize.

In Figure 4.18 are represented an example of data classified in three clusters using K-means, assuming equal-sized clusters.

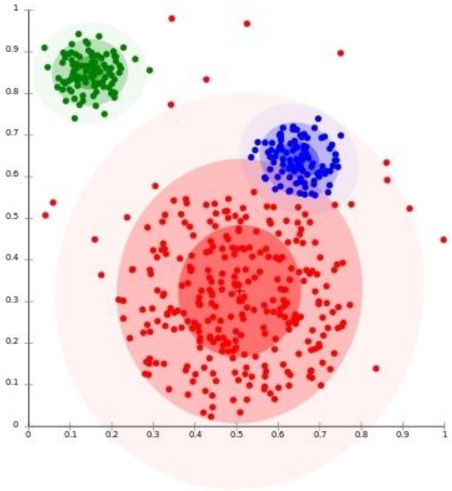


Figure 4.18. K-means separated data

4.4.1.1. *EM-clustering results*

EM assigns a probability distribution to each instance which indicates the probability of it belonging to each of the clusters. EM can decide how many clusters to create by cross validation, or you may specify a priori how many clusters to generate.

All the figures of this section shows the clusters that Weka has done each one represented in a different colour, can be seen how has been distributed the data in each cluster, taking notices the colours distribution.

Figure 4.19 represents the distributions of the CPU slots and the number of links into the different clusters painted in different colours. As can be seen only exist the values of 1, 2 or 3 CPU slots this is because this tests are used only the mobiles devices.

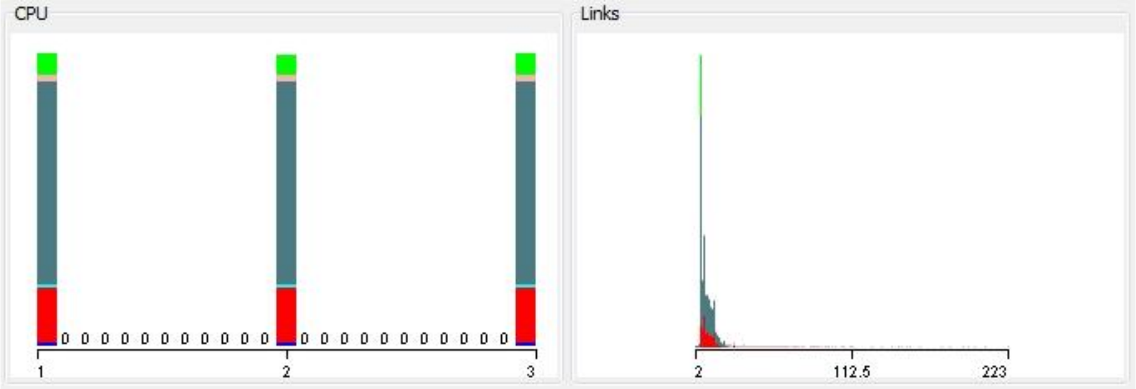


Figure 4.19. EM-clustering results of CPU slots vs. Number of links of each node

Figure 4.20 shows the number of CPU slots of the neighbours and the number of neighbours that are desktop devices, the relation between the clusters and these variables is clearly represented but difficult to know the reason of this classification.

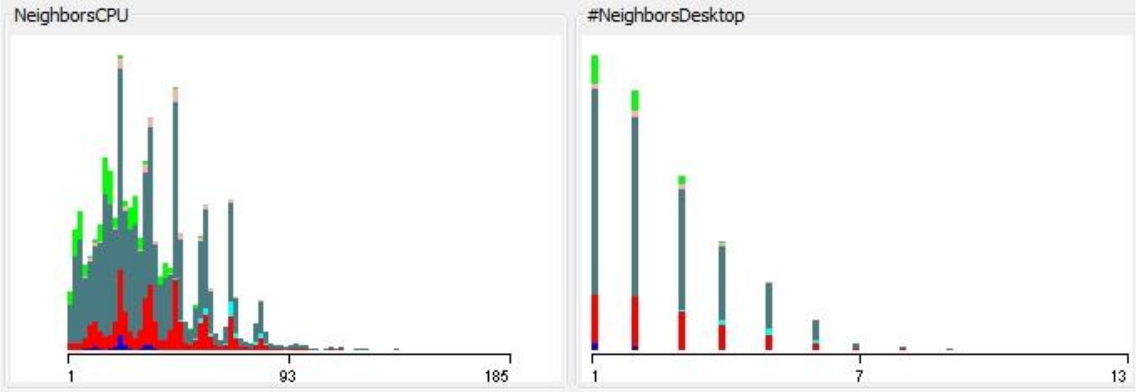


Figure 4.20. EM-clustering results of of neighbours CPU slots vs. Number of neighbours belonging to the type desktops

The main objective is to know the relation between the clusters realized and the cooperation coefficient.

As shows in Figure 4.21, as in other variables, is not obvious the reason of why each node has been included in which cluster. But studying the results has been concluded that the cooperation is excluded when the repartition has taking place. During the classification in clusters the algorism look at the biggest group of instances remarkable of each attribute and place it in the same cluster. Finally when all the attributes are classify the system look how the cooperation chart has been distributed and delete the less important clusters, choosing by a result the information of cooperation that are considered most important.

If are compared the graph to the right, that are the clusters resultants of the execution of Expectation-Maximization clustering algorithm and one to the left which is the variable of cooperation. It can be seen in different colours the section of the data that has been classified in each place and so the only conclusion than can be done is that it is not possible to extract useful information from this test.

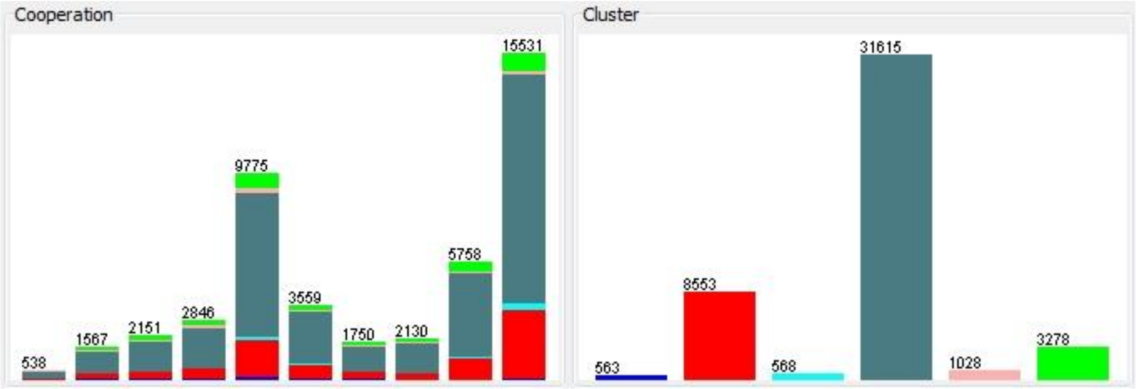


Figure 4.21. EM-clustering results of Cooperation coefficient vs. Cluster classification

This is corroborated by Figure 4.22 which shows the number of nodes where can be seen that one of the clusters is composed almost entirely by the instances that have the greatest number of nodes and this is not the criterion that is expected to be used for realize the cluster classification.

At the left top of the same figure (Figure 4.22) are represented all the instances painted with the colours of each cluster, as can be seen the number of instances of each cluster are not equal distributed, almost all the instances corresponds to the fourth cluster coloured in gray.

Furthermore note that the variable number of nodes is considered as the less important in the chapter 4.2. and now, this attribute consists almost entirely the cluster with more number of instances.

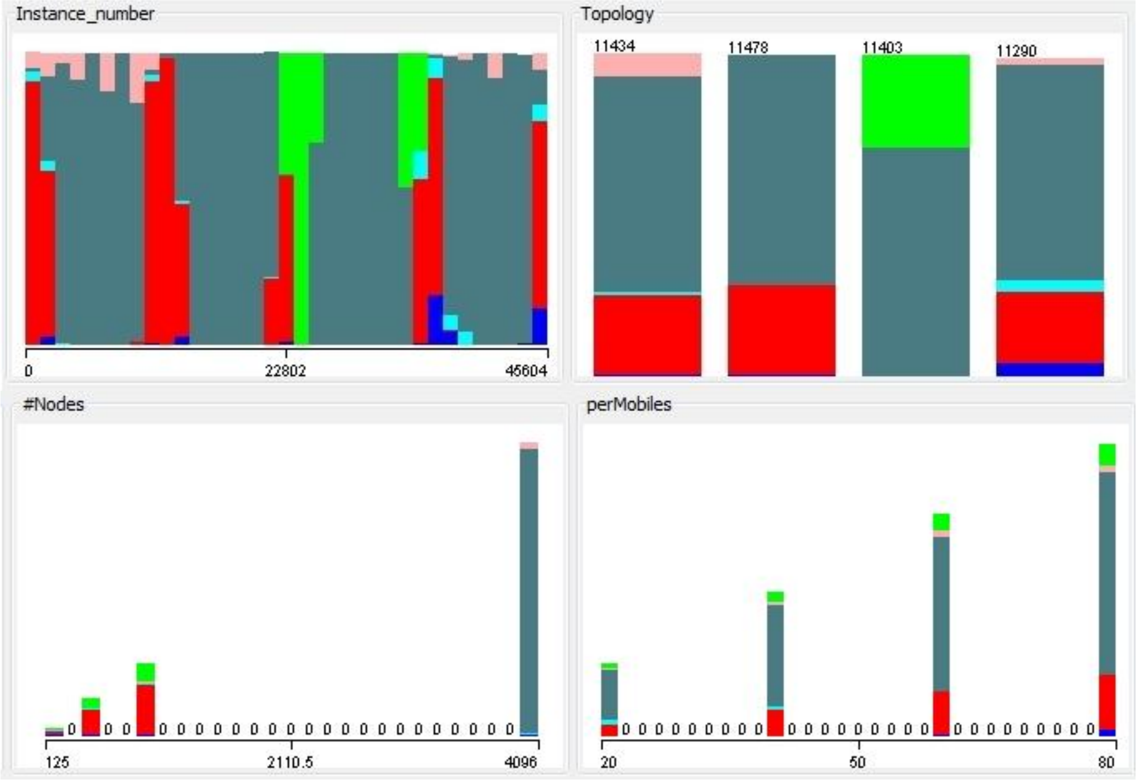


Figure 4.22. EM-clustering results of Topology vs. number of nodes

The graphic below (Figure 4.23) shows two more variables, where the instances than contain the lowest cluster coefficient are the red cluster. Referring to the queries note, that in the results shows that in each cluster have been classified the same number of instances in mean, for that reason has been detected that this parameter are not relevant for the rest ones, and if this had been excluded has no effects in the results.

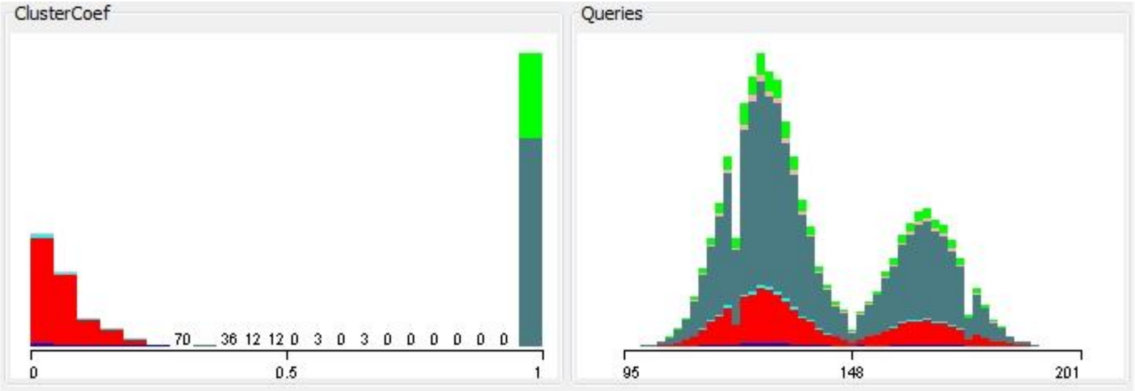


Figure 4.23. EM-clustering results of Cluster coefficient vs. queries

4.4.2. Centroid-based clustering

Other method to perform clustering classification is centroid-based clustering,in this method clusters are represented by a central vector, which must not necessarily be a member of the data set. When the number of clusters is fixed to k, *k*-means clustering gives a formal definition as an optimization problem, find the *k* cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

Variations of k-means often include such optimizations as choosing the best of multiple runs, but also restricting the centroids to members of the data set, choosing medians (k-medians clustering), choosing the initial centers less randomly (K-means++) or allowing a fuzzy cluster assignment (Fuzzy c-means).

One of the biggest drawbacks of these algorithms is that require the number of clusters, called “k”, to be specified in advance.
Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroids. This often leads to incorrectly cut borders in between of clusters.

K-means has a number of interesting theoretical properties. On one hand, it partitions the data space into a structure known as Voronoi diagram. On the other hand, it is conceptually close to nearest neighbour classification and as such popular in machine learning. Third, it can be seen as a variation of model based classification, and Lloyd's algorithm as a variation of the Expectation-maximization algorithm for this model discussed below.

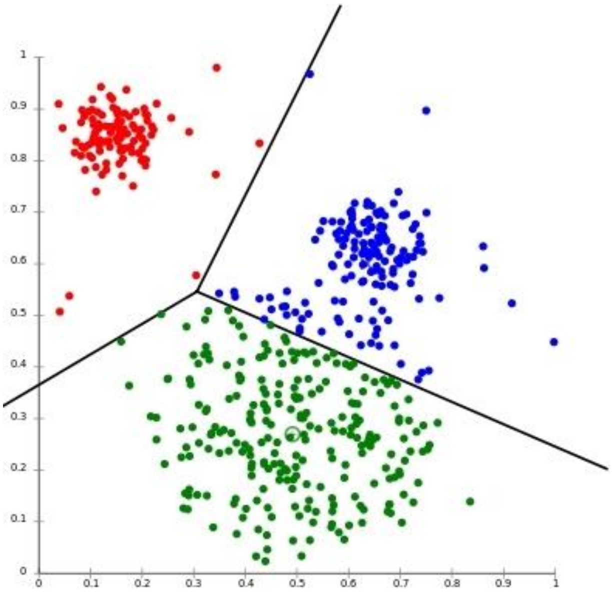


Figure 4.24. K-means separates data into Voronoi-cells, which assumes equal-sized clusters

4.4.2.1. M means results

X-Means is K-Means extended by an Improve-Structure part In this part of the algorithm the centers are attempted to be split in its region. The decision between the children of each center and itself is done comparing the BIC-values of the two structures.

As has been said in the description of this methods to run this algorithm is necessary to choose the number of clusters, in this case have been chosen the number of clusters that has resulted using the method EM in the previous section.

The results obtained using this method are almost the same as the obtained using the EM clustering method. In other words, data are distributed in the clusters tanking in to account the quantity of instances on each group, placing together all the instances that has 4096 node number as a variable for example. And this is not the criteria than are expected to use. As can be seen in Figure 4.25.

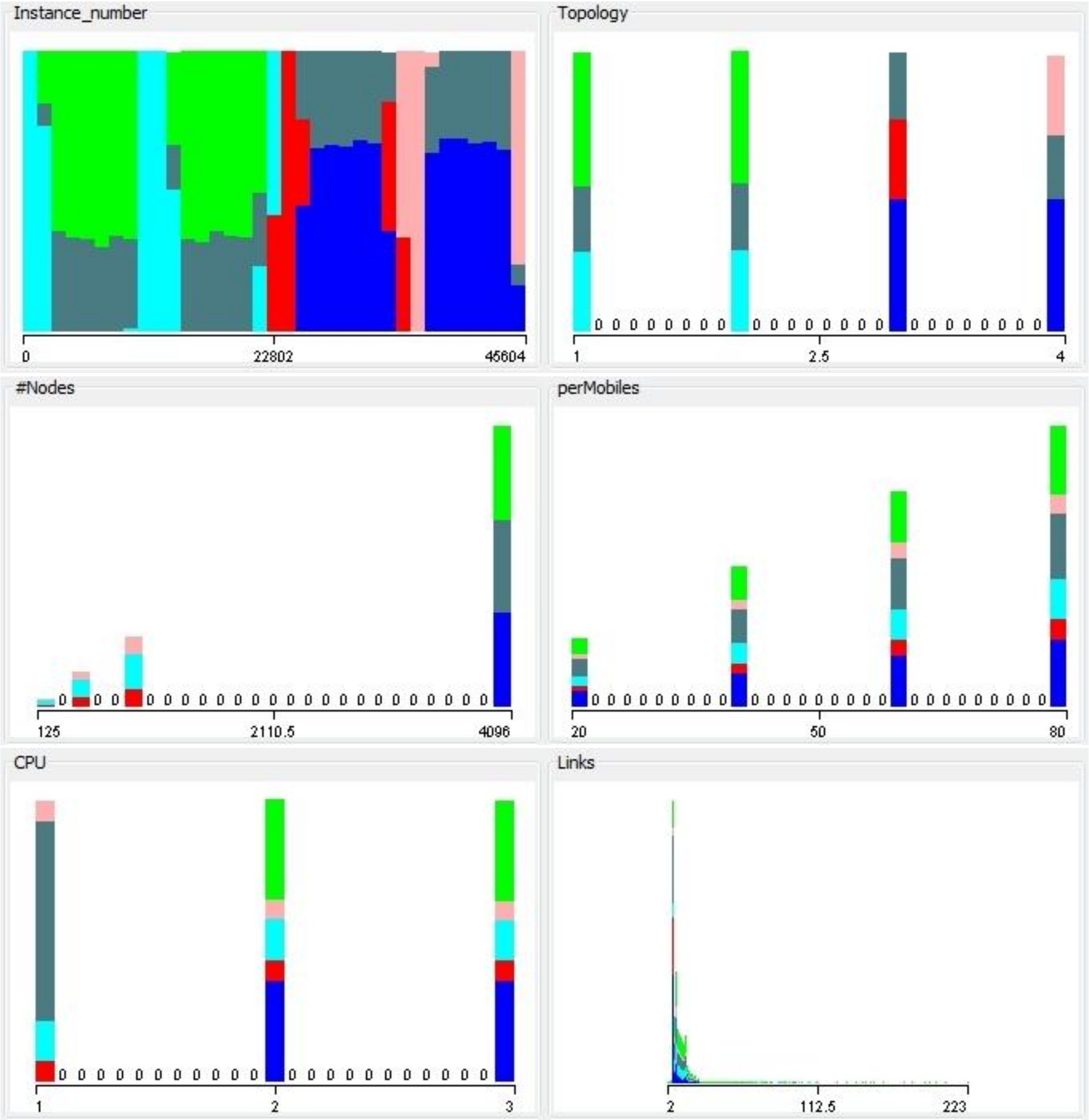


Figure 4.25. Results M-means of variables instance number, topology number of nodes, percentage of mobiles, number of CPU Link

Figure 4.26 depict that using this algorithm number of instances of each cluster are more equitable in number compared to the resulting using EM method. Even so, this fact does not bring great benefits to the showed results.

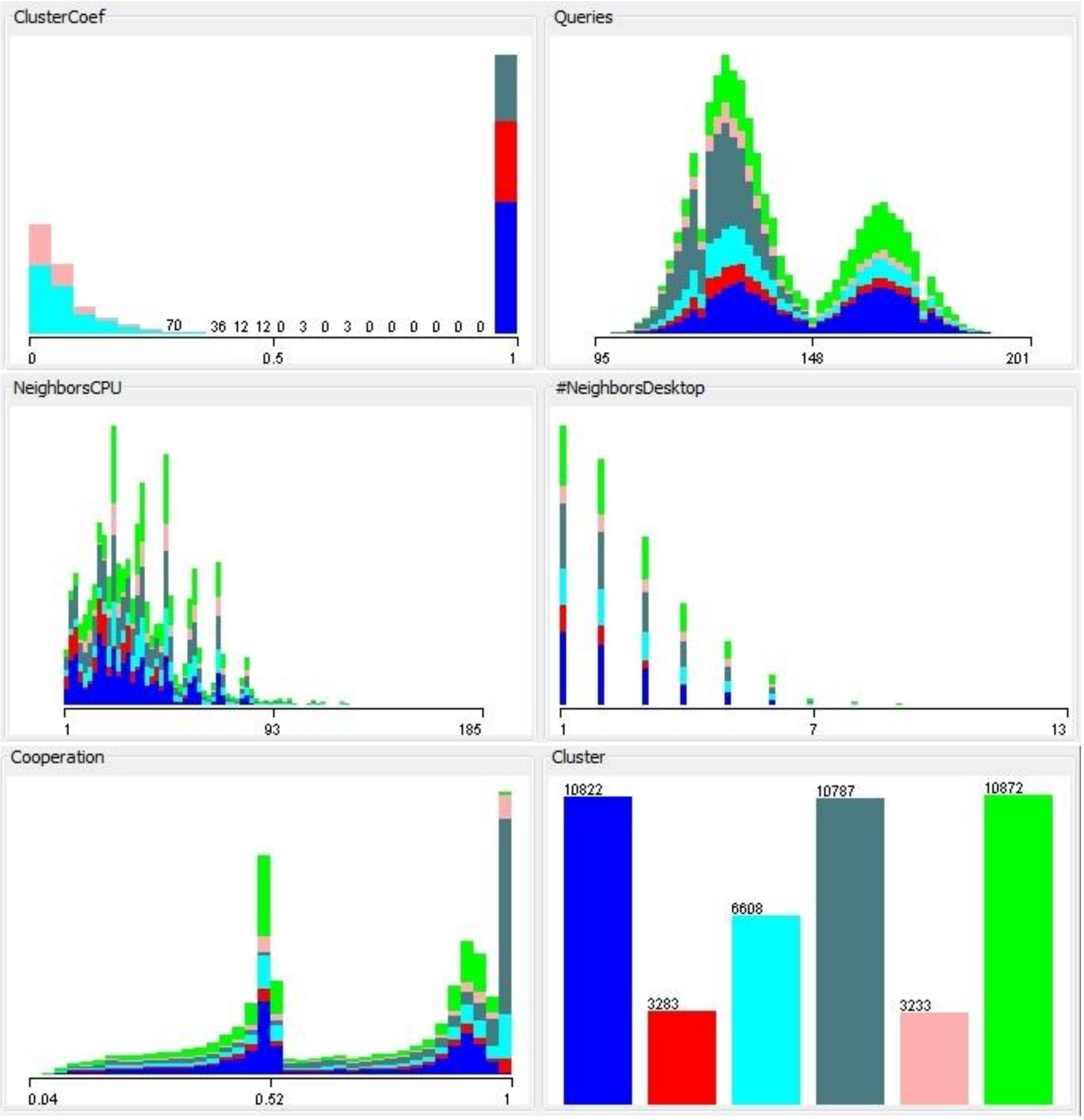


Figure 4.26. All attributes distributed by cluster colours

4.4.3. Clustering results

After executing the algorithm the results seem to show that first the algorithm classify the attributes, taking into account the most interesting features. Considering as most interesting the greatest number of instances. Then evaluate which attribute are most important, taking into account the cooperation variable, so do not miss valuable information on this variable. But this does not really allow us to observe any relationship between the various attributes existents.

CHAPTER 5. CONCLUSIONS

In this chapter finally are make a summary of the objectives achieved during the master thesis, the simulation conclusion and finally the personal conclusions of this master thesis where concepts, skills and lessons learned will be identified in order to evaluate the whole results. Finally, the environmental effects of this project have been argued.

5.1. Data mining conclusions

The results obtained by the data mining tool are interesting and useful but it must be analyzed carefully. They are not as fast as one would expect to interpret. The user must to have some basic knowledge of the algorithms used, in order to understand the variable behaviour. It is important to note that, not all the methods has the same difficulty to be understand, the results of some algorisms are easier to interpret in comparison to others.

During the realization of data mining experiments, many conclusions have been learned.

To begin, after running the attribute selection procedure are found that this really helps to understand the relationship between attributes and the importance of each one. Through this process can be eliminated the attributes are not greatly influence on the behaviour of others ones. But really, if we remove too many variables to take charge of information for the realization of the other tests, the results are not coherent with respect those expected. So we really need to run the data mining methods using the maximum possible attributes without being redundant.

On the other hand, using the regression methods it has been demonstrated that is possible to predict the behaviour of a variable having a considerable precision. But to do that is needed to run complex algorithms and handle a lot of information. This fact causes that the execution requires several hours to be finished.

In the case of clustering classification the results are more complex to analyze than in the other cases. It is not easy to understand how shows the clusters that have assign and the number of attributes that have been allocated to each group. So, it is necessary to interpret the empirically, identifying the possible trigger for the classification and studding the patterns shown in the outcomes.

5.2. Network parameters conclusions

Regardless of the effectiveness or problems caused by the tools used and whether or not it has been served our purpose. It is important to notice the conclusions extract by use these tools.

To begin, after the execution of the algorithms for attributes selection was observed that several of the variables analyzed did not provide information and others were redundant. As the case of the data referred to the type desktop instances, had been notice that the variable that really provided useful information are the mobile instances. For that, reason has been made the rest of the tests using only the instances of mobiles devices.

Furthermore, are concluded that the most influential attributes are CPU, Queries, percentage of mobiles used, topology and finally the cluster coefficient.

On the other hand, exist another set of attributes considered expendable and that theoretically could be superseded, these are the number of nodes in the network and the number of links of each node.

The attributes that are not consider either important neither superfluous are the, the number of neighbour desktops and the number of CPU slots of the nearby nodes.

It is remarkable talking about the clustering algorithms that the variable called queries has been classified the same number of instances in average for each cluster. For that reason, it has been detected that this parameter is not relevant and if excluded has no influence over the final results.

In conclusion, using this tool has been possible to observe relationships between the variables, as well as establish a range of importance of each parameter, and quite correctly predict the behaviour of one of them. But it is important to note that not all algorithms and method of data mining are valid for any type of data set. Most of them require a fairly lengthy process of study of both the data and search algorithms to find the best combination possible.

5.3. Personal conclusions

Personally, this project has enriched me both academically and professionally.
I have learned a lot about data mining I have seen an enormous number of methods, algorithms and the infinite possibilities of its applications. Furthermore I have learned to realize scripts in languages that I never had been used before.

The most important fact is that the obtained formation of this Thesis is not focused only in the knowledge that I have obtained about the tools that had been used. Also has centred to the availability of learning the way to perform a project of investigation and to discover about project management. Also I think that the redaction of this kind of reports helps to improve my skills in writing technique documents, which I consider also important for the development of my professional career.

In addition, that training is important to include the learning about how to find the correct information and how to learn about the obstacles found during the development of this Master Thesis.

5.4. Study of environmental effects

The realization of this master thesis project has more benefits for the environment than negatives aspects. Because is based on interpret the data obtained from a simulator tool which intends to emulate a complete network. That is instead of using a large number of devices of many types, taking into account the energy consumption and pollution that entails manufacturing these devices. Furthermore the simulation is performed using a single computer.

It is true that the utilization of one computer has an impact to the environment, but is obvious that this is a minor damage comparison the amount of waste that would occur if we do not use a simulator. Furthermore had been used faster processors which allow enhance efficiency while reducing energy use in a reduced time.

In addition, the purpose of this project is to understand the behaviour of a network with specific characteristics in other to find ways to improve efficiency of the system and consequently decrease the energy consumption.

BIBLIOGRAPHY

[1] Davide Vega, Esunly Medina, Roc Messeguer, Dolors Royo, Felix Freitag “Characterizing the Effects of Sharing Hardware Resources in Mobile Collaboration Scenarios”, Department of Computer Architecture Universitat Politècnica de CatalunyaBarcelona, Spain

[2] Davide Vega D’Aurelio “Design and implementation of a simulator to explore cooperation in distributed environments” June 17 th 2010

[3] Tit for tat (2011, 6 December) In: Wikipedia, the free encyclopedia [On line] Wikipedia article URL: ttp://en.wikipedia.org/wiki/Tit_for_tat

[4] Cooperation. (2011, 6 December) In: Wikipedia, the free encyclopedia [On line] Wikipedia article. URL: http://en.wikipedia.org/wiki/Co-operation_(evolution)

[5] Clustering coeficient. (2011, 8 December) In: Wikipedia, the free encyclopedia [On line] Wikipedia article URL: <http://en.wikipedia.org/wiki/Clustering_coefficient>

[6] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[7] Holland and Leinhardt, 1971; Watts and Strogatz, 1998

[8] Encyclopædia Britannica: http://www.britannica.com/EBchecked/topic/1056150/data-mining

[9] Data mining with WEKA, Part 1: Introduction and regression, Michael Abernethy, Freelance

[10] Programmer,Freelancer. Abailable at: http://www.ibm.com/developerworks/opensource/library/os-weka1/index.html

[11] Advancing Feature Selection Research: http://featureselection.asu.edu/featureselection_techreport.pdf

[12] H. Liu and H. Motoda. Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic Publishers, 1998.

[13] Guyon and A. Elissee_. An introduction to variable and feature selection. Journal of Machine Learning Research, 3:1157{1182, 2003.

[14] H. Liu and H. Motoda, editors. Computational Methods of Feature Selection. Chapman and Hall/CRC Press, 2007.

[15] A Guide to Network Topology:
http://learn-networking.com/network-design/a-guide-to-network-topology

[16] Réka, Albert; Barabasi, Albert. Statistical mechanics of complex networks. Reviews of modern physiscs. 2002, vol. 74. n. 1. pp. 47-97.

[17] Barabási, Albert-László; Bonabeau, Eric. Scale-Free Networks. Scientific American. 2003, vol. 288, no. 2, pp. 50-59.

[18] Torus. (2011, 6 December) In: Wikipedia, the free encyclopedia [On line] Wikipedia article. URL: http://en.wikipedia.org/wiki/Torus

[19] M. Naldi, "Connectivity of Waxman topology models",Universita` di Roma 'Tor Vergata', Dip. di Informatica Sistemi Produzione (DISP), Via del Politecnico 1, 00133 Roma, Italy. Accepted: 31 January 2005, URL:http://www.sciencedirect.com/science/article/pii/S0140366405000630

[20] B.M. Waxman, Routing of multipoint connections, IEEE Journal on Selected Areas in Communications 6 (9) (1988) 1617–1622.

[21] A. Lakhina, J.W. Byers, M. Crovella, I. Matta, On the geographic location of internet resources, IEEE Journal on Selected Areas in Communications 21 (6) (2003) 934–948.

[22] Duncan J. Watts & Steven H. Strogatz, April 1998, "Collective dynamics of 'small-world' networks" Department of Theoretical and Applied Mechanics, KimballHall, Cornell University, Ithaca, New York, USA http://www.nature.com/nature/journal/v393/n6684/full/393440a0.html

[23] G. Holmes; A. Donkin and I.H. Witten (1994). "Weka: A machine learning workbench". Proc Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia. URL:http://www.cs.waikato.ac.nz/~ml/publications/1994/Holmes-ANZIIS-WEKA.pdf. Retrieved 2007-06-25.

[24] Online Perl Documentation, URL: http://www.perl.org/docs.html