



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI MATEMATICA

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

Interactive Subgroup Discovery

Relatore: Prof.ssa Francesca ROSSI

Correlatore: Dr. Arno KNOBBE

Laureando:

Andrea IMPARATO

Anno Accademico 2012/2013

Dedicated to the Knowledge Discovery

Abstract

Subgroup discovery is a Knowledge Discovery task that aims at finding subgroups of a population with high generality and distributional unusualness. In many cases the user could be interested in discovering and comparing subgroups that are difficult to find with the standard beam search. In this thesis we developed a novel method to explore the hypothesis space with the use of contingency tables analysis and the interaction between the user and the subgroup discovery system. To test the effectiveness of this approach we created several use cases on two different datasets.

Contents

1	Introduction	6
2	Background	8
2.1	Data Mining	8
2.2	Rules Mining	9
2.3	Linear Regression Models	11
2.4	Bayesian networks	13
3	Subgroup Discovery and Exceptional Model Mining	16
3.1	Problem Formalization	17
3.2	Searching through the hypothesis space	18
3.2.1	Generating refinements	19
3.2.2	Depth First Search	19
3.2.3	Beam Search	19
3.2.4	Cover-based beam selection	21
3.3	Quality measures	22
3.3.1	Weighted Relative Accuracy	22
3.3.2	Z-score	22
3.3.3	Exceptional Model mining Quality Measures	24
3.4	ROC Space	25
3.4.1	Isometrics	26
3.4.2	AUC	28
3.4.3	Evaluation of the mining process	28
3.4.4	Nominal value set and Numeric Intervals	29

4	Statistical background	32
4.1	Null hypothesis, statistical independence and association-correlation	32
4.2	Contingency Table analysis	33
4.3	Yule's Q Coefficient	35
4.4	Phi Coefficient	35
4.5	Numerical properties of phi and yule's q coefficient	35
4.6	Chi squared test	38
4.7	P-value	38
4.8	Fisher's Exact test	39
4.9	Bonferroni procedure	39
4.10	Coefficients Examples	40
5	Interactive Subgroup Discovery	42
5.1	Interactive Data mining systems	42
5.2	Influencing the beam search	43
5.3	Interactive beam search	45
5.3.1	Like	47
5.3.2	Dislike	47
5.3.3	Opposite	47
5.3.4	Combining the qualities	48
5.3.5	Subgroup Investigation	48
5.3.6	Subgroup Focus	49
5.3.7	Validation	49
5.4	About coefficients indefiniteness	49
6	Cortana	54
6.1	Background and features	54
6.2	Loading Data and setting the search parameters	55
6.2.1	Dataset section	56
6.2.2	Target Concept section	57
6.2.3	The search conditions	58
6.2.4	The search strategy	59
6.2.5	The results	60

6.2.6	Results window buttons	60
6.2.7	ROC Curve	61
6.3	Interactive implementation design	63
6.4	Interacting with the data	67
6.4.1	Dislike of the intent	68
6.4.2	Info Window	69
6.4.3	History and filtering	70
6.4.4	File menu	71
6.4.5	Results Buttons	72
6.4.6	Results Visualisation	72
6.4.7	Charts	74
6.4.8	About Cortana's coding practice	76
7	Results	78
7.1	General knowledge, local knowledge and the analysis approach	78
7.2	UCI datasets	80
7.3	Adults Dataset	80
7.4	Mammals Dataset	95
7.5	About running time of the algorithm	99
8	Conclusions and future work	100
	Glossary	104
A	Coefficients Generator	113
B	Relevant code	120

Chapter 1

Introduction

In this thesis we will discuss a method to integrate the user's knowledge with the Subgroup Discovery process, a data mining technique to discover subgroups that show interesting differences in distribution in one of their attribute (the target attribute), compared to the whole dataset.

The reasons behind this work were two. First of all, we wanted to allow the user to exploit the power of the beam search, the algorithm that searches the subgroups through the hypothesis space. Second, we wanted to change the whole subgroup discovery process, which usually is considered a *black box* process where the user is unable to interact with the procedure, to an interactive process where the user can choose which regions of the hypothesis space to explore and which subgroups to focus on.

In the first part of the thesis, we will discuss the basic concepts behind data mining and subgroup discovery, we will not focus on a particular subject. Our aim is instead to underline the motivations for subgroup discovery and its extension, the exceptional model mining. In the second part, we will present our interactive subgroup discovery method and how we integrated it in Cortana, a Subgroup Discovery tool developed by the data mining group at Leiden University. In the final part, we will present the results obtained and we will discuss the analysed datasets.

Chapter 2

Background

2.1 Data Mining

Data mining is defined as the process of discovering patterns in data. The process must be automatic or (more usually) semiautomatic. The patterns discovered must be meaningful in that they lead to some advantage, usually an economic one. The data is invariably present in substantial quantities. [17]

The data analysed by data mining algorithms could be either in multiple tables (multi relational data mining) or a singular table. In this thesis we will not analyse the multi relational data mining, the reader can just remember that every multi relational database can be transformed into a singular table with multiple join operations. In this way the space occupied by the data increases exponentially with the number of joint operations, however the algorithms can be defined more easily if they do not have to take into account the relations within a multi relational database. The tables for data mining algorithms are usually prepared in Attribute-Relation File Format format, to see more information about it take a look at the glossary. After applying an algorithm on the data, the output that this algorithm generates is a single or multiple *patterns*, they are called also **structural patterns** because they reflect the structural descriptions of the information. There are different types of patterns and they differ basically on how much they are understandable by

the user.

The first type are *black box* patterns: their innards are incomprehensible by a human, their structure is really complex and usually the building process of the model is really slow due to this complexity. Bayesian and neural networks for example belong to this kind of patterns.

The other type are the *transparent box* patterns, patterns with understandable nature, rules for example belong to this type. Rules are sort of if/then clauses: if conjunction_of_attributes then class where the attributes are the columns (or attributes of the items) of the table and class is one of this attribute, see the next section for further informations.

After we discovered the data patterns we have to use them to extract the desired new knowledge.

We have two possible procedures:

- predictive induction: we use the patterns to make predictions of new examples. The patterns represent a model that we fit upon the whole dataset and we use it to predict the new outputs.
- descriptive induction: we use the patterns to get a description of the data and we focus on relations between the attributes. Here we do not want to maximise the accuracy of the patterns but the process it is meant to be **exploratory** and aims to give an overview of regions of the data. Usually it is the user that have to understand the patterns extracted and to decide where the patterns are new knowledge or already known knowledge. Association rules are an example of this kind of patterns.

Before we go deep with the topic of this document, we want to explain some background concepts that can be useful to better understand our work.

2.2 Rules Mining

Rules, as we said in the previous section, are patterns in the form of if/then clauses. In literature we can find two main types of rules: *classification rules*

and *association rules*. The former are used in the prediction context, and they focus on the accuracy of the prediction of a new example and only on a single target class, the latter are used to discover strong association between multiple attributes and multiple class attributes.

More formally, a Rule consists in two parts: the head and the body.

$$\text{Body} \rightarrow \text{Head}$$

The body consists in conjunctions of conditions made by attribute-value pairs and head is a class assignment. The conditions on attribute-value pairs depends on the type of the attribute and the bias of the algorithm: if the attribute it is nominal we find *equal* operator, if it is numeric we can find *greater/less equal than*, sometimes we can also find the *not equal* operator. Therefore the rules become:

$$X_1 \text{ op}_1 \text{ value}_1 \wedge X_2 \text{ op}_2 \text{ value}_2 \wedge \dots \wedge X_n \text{ op}_n \text{ value}_n \rightarrow \text{Head}$$

Where X_i are attributes, op_i are the operators ($=, \leq, \geq$), and value_i are numeric or nominal values

The rules extracted from the dataset should be interesting and not known by the user and possibly comes in small numbers, only the interesting ones should be presented to the user. This problem is really important with rules discovery approach because this kind of algorithms tends to generate a huge amount of rules and discarding the redundant ones make the output more understandable for the user. More generally these are good properties for rules [14]:

- conciseness: the rules should contain few attribute-value pairs so they can be understood easier by the user
- generality/coverage: the rules should cover relative large subset of the dataset so it can be considered general
- reliability: a rule is reliable if it can be applied to a high percentage of applicable cases

- peculiarity: a rule is peculiar if it is different from the whole set of the rules
- diversity: a rule it is diverse if the elements that belong to the rule are different from each other
- novelty: a rule is novel if it wasn't known a priori by the user so it represents new knowledge
- surprisingness: a rule is surprising if it contradicts the prior knowledge of the user
- utility: a rule is useful if it helps the user to reach his goals
- actionability/applicability: a rule is actionable if it can be applied to a specific domain

As we can see, most of these properties are really subjective (except conciseness and generality) so is up to the user to evaluate the rules.

To reach good rules and not present so many of them to the user but only the best of them and non-spurious rules we can use some quality measures to rank the rules and to prune the hypothesis space. The most used algorithm for classification rules is the CN-2 algorithm [3]. However the problem with the approach *divide and conquer* and the filtering criteria as the significance test on identifying significant deviations from random classification that this algorithm uses, does not necessarily avoid overfitting [20].

2.3 Linear Regression Models

Linear regression models are stable models in statistics when all the attributes and the class are numeric (numeric prediction).

The idea behind this method is to express the class as a linear combination of the attributes, with predetermined weights:

$$x = w_0 + w_1a_1 + w_2a_2 + \dots + w_k a_k$$

where x is the class, a_1, a_2, \dots, a_k are the attribute values and w_0, w_1, \dots, w_k are weights.

For the first example in the training set, the predicted class, becomes:

$$w_0 a_0^{(1)} + w_1 a_1^{(1)} + w_2 a_2^{(1)} + \dots + w_k a_k^{(1)} = \sum_{j=0}^k w_j a_j^{(1)}$$

We can use all the instances in the training set to predict the new example's class. To get good predictions we can minimise the square of the errors of the predicted class and the actual class of the example:

$$\min \sum_{i=1}^n (x^{(i)} - \sum_{j=0}^k w_j a_j^i)^2$$

where $x^{(i)}$ is the class value of the i -th example and $w_j a_j^i$ are the weights associated with its attributes.

After we calculated the weights using the training data, the predicted class of a new example is just the output of the linear combination of the weights.

This model is really simple and with complex and variate data it cannot be used, more complex models are needed, like for example SVMs or neural networks.

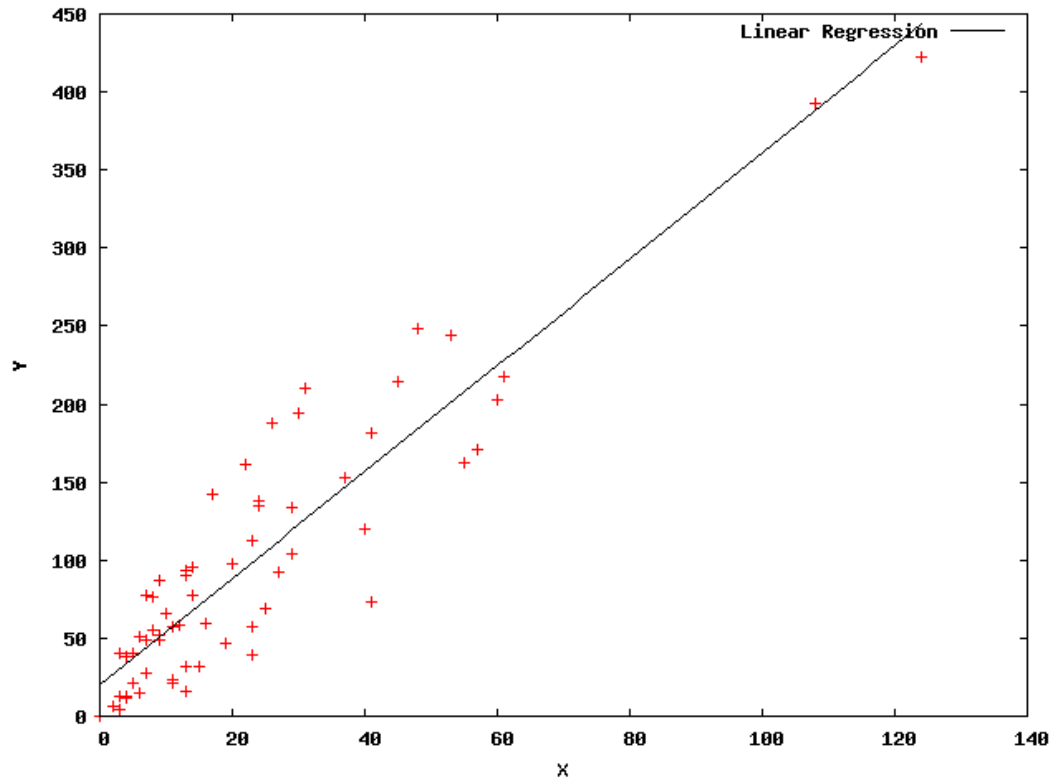


Figure 2.1: A linear regression Model

2.4 Bayesian networks

Bayesian networks are *directed acyclic graph* (or DAG), networks of nodes, one for each attribute of the dataset, connected by directed edges without cycles. They represent a complete probability distribution in a compact way using probability theory to reduce the size of the whole table. There are several algorithms to infer probability given evidence and also to build bayesian networks given a datasets. Their foundations are described in [19]. A possible way to fit the a bayesian model could be using an hill-climbing approach [18]. Bayesian networks are used to build models on *multi-label* datasets.

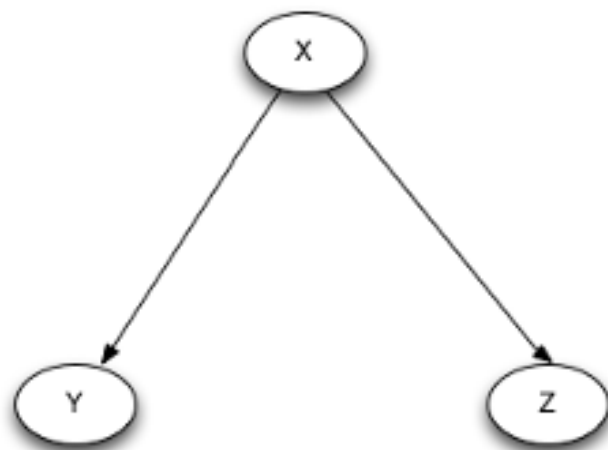


Figure 2.2: A naive bayes classifier with three variables, X influences both Y and Z, the simplest bayesian network.

Chapter 3

Subgroup Discovery and Exceptional Model Mining

The field of subgroup discovery (SD) is concerned with the discovery of subsets of the data, where the target attribute(s) show an interesting difference in distribution, compared to that of the entire dataset [34]

The target attribute is one of the columns that describes the data table, so when this target is fixed we want to discover all the interesting subgroups respect to this column. In Exception Model Mining (EMM), the powerful extension of SD, this concept is extended to sets (two or more) of target attributes: we fit a model on these and we want to find models that are different respect to the model build upon the remaining data.

The purpose of subgroup discovery is to generate *actionable* knowledge, that is knowledge understandable and useful for the user. The subgroups are expressed in form of rules and the length of the rules is generally restricted, generating rules easy to interpret [29]. The subgroup discovery, as we will see later, is more like a framework of techniques rather than a single technique to discover interesting subgroups of data. In fact is built in such a way that can be adapted to the objective of the discovery without changing the general infrastructure that stands behind. Basically this means that the way to search in the hypothesis space remains the same, what changes is the evalua-

tion of the current explored state, that is based upon on which target we have set. So this setting of the search parameters makes the subgroup discovery a supervised learning algorithm moreover it belongs to descriptive induction algorithms due to its nature to discover interesting local patterns of the data rather than building a global predictive model. Its applications are different for some examples see [23], [21].

3.1 Problem Formalization

As we said the tuples to be analysed are described by a set of attributes A which consists of k *description attributes* D and l *model (or target) attributes* M ($k \geq 1$ and $l \geq 1$). In other words, we assume a supervised setting, with at least a single target attribute M_1 (in the case of classical SD), but possibly multiple attributes M_1, \dots, M_l (in the case of EMM). Each attribute D_i (resp. M_i) has a domain of possible values $Dom(D_i)$ (resp. $Dom(M_i)$). Our dataset S is now a bag of tuples t over the set of attributes $A = D_1, \dots, D_k, M_1, \dots, M_l$. We use x^D resp. x^M to denote the projection of x onto its description resp. model attributes, e.g. $t^D = \pi_D(t)$ in case of a tuple, or $S^M = \pi_M(S)$ in case of a bag of tuples. Equivalently for individual attributes, e.g. $S^{M_i} = \pi_{M_i}(S)$. In subgroup discovery the most important concept is trivially the concept of *subgroup*. A subgroup S is a bag of tuples $G \subseteq S$, the *size* of the subgroup is $|G|$ and it is also defined by the *description* and the *cover*:

- description: is an indicator function s , as a function of description attributes D . $s : (Dom(D_1) \times \dots \times Dom(D_k)) \mapsto \{0, 1\}$
- cover: $G_s = \{t \in S \mid s(t^D) = 1\}$, the table's rows that belong to the subgroup

A subgroup description is a pattern, consisting of a conjunction of conditions on the description attributes, e.g. $D_x = true \wedge D_y \leq 3.14$ so the descriptions have the same semantics as rules (section 2.2), therefore the same evaluation measurers can be used.

Given a subgroup G , we would like to know how interesting it is, looking only at its model (or target) data G^M . We quantify this with a quality measure. A quality measure is a function $\phi : G^M \mapsto R$ that assigns a numeric value to a subgroup $G^M \subseteq S^M$, with G^M the set of all possible subsets of S^M . It is common to call the coverage of the subgroup, the tuples that it covers the **extent** of the subgroup and its description the **intent**.

Finally it is interesting also to define the subgroup G *complement* that is the subgroup that covers the tuples in $S - \{G\}$ where S is the whole dataset.

3.2 Searching through the hypothesis space

After defining what is a subgroup and how to evaluate if it is interesting or not, now we have to define how to search these subgroups through the hypothesis space made by all the possible interesting subgroups. Basically there are two main approaches to do that, the *complete* approach and the *heuristic* approach. We are interested in finding the *top-ranking* subgroups according to a quality measure ϕ that determines the level of interestingness in terms of unusual distribution of the target attribute(s) M .

To proceed with this search the better approach is to explore the space with a top-down strategy. The search space is traversed by starting with simple descriptions and refining these along the way, from general to specific. For this a refinement operator that specialises subgroup descriptions is needed. Given the empty subgroup description, the refinement operator generates all descriptions consisting of a single condition. Given any subgroup description X , consisting of $|X|$ conditions, it generates all allowed descriptions of size $|X| + 1$ containing X . These are the refinements of X . A minimum coverage threshold (mincov) is used to ensure that a subgroup covers at least a certain number of tuples. A maximum depth (maxdepth) parameter imposes a maximum on the number of conditions a description may contain.

3.2.1 Generating refinements

Besides the parameters that we mentioned above for the beam search, what it remains to explain is how generate the refinements for the descriptions of the subgroups. These refinements are generated from the domain of the attributes of the data table. If the current attribute is nominal or binary we generate *all* the possible values and we consider them, otherwise if the attribute is numeric an exhaustive evaluation is mostly unfeasible therefore we proceed with an *on the fly discretisation* dividing it in equal bins and for every extreme of the bin we use a set of numeric operators such as $=, \geq, \leq$.

3.2.2 Depth First Search

When exhaustive search is possible, depth-first search is commonly used. This is often the case with moderately sized nominal datasets with a single target. Whenever possible, (anti-)monotone properties of the quality measure are used to prune parts of the search space. When this is not possible, so-called optimistic estimates can be used to restrict the search space. An optimistic estimate function computes the highest possible quality that any refinement of a subgroup could give. If this upper bound is lower than the quality of the current k^{th} subgroup, this branch of the search space can be safely ignored [15].

3.2.3 Beam Search

When exhaustive search is not feasible, beam search is the widely accepted heuristic alternative. It also uses a levelwise top-down strategy and the same refinement operator, but it explores only part of the search space. It is the best approach to **exploit** and **explore** the hypothesis space. The basic algorithm is shown below. On each level, the w highest ranking subgroups with respect to the quality measure are selected for the beam. Candidate subgroups for the next level are generated from individual subgroups b using the refinement operator (GenerateRefinements), while respecting the mincov parameter. The initial candidate set is generated from the empty subgroup description. Select-Beam selects the w highest ranking $c \in \text{Cands}$ (with respect to ϕ) to form the

beam for the next level.

Algorithm 1 Beam Search

Input: A dataset S , a quality measure ϕ and parameters k as quality minimum, w as beam width, $mincov$ and $maxdepth$.

Output: R , an approximation of the top- k subgroups G_k .

$R \leftarrow \emptyset, B \leftarrow \{\emptyset\}, depth = 1$

while $depth \leq maxdepth$ **do**

$Cands \leftarrow \emptyset$

for all $b \in Beam$ **do**

$Cands \leftarrow Cands \cup GenerateRefinements(b, mincov)$

for all $c \in Cands$ **do**

$UpdateTopK(R, k, c, \phi(c))$

$Beam \leftarrow SelectBeam(Cands, w, \phi)$

$depth \leftarrow depth + 1$

return R

3.2.4 Cover-based beam selection

The problem with the beam search approach is that could introduce high redundancy in the results in fact only the quality measure and the mincov parameters are the filtering methods to exclude the subgroups and the possibility for the subgroups to overlap introduce even more redundancy. Anyway an approach an approach to deal with this problem has been found [34]. It is based on **multiplicative weighted sequential covering** and it consists to perform the beam search as usual but for every level of the beam search it introduces another quality measure that give *diversity* to the patterns of the beam, that is selecting multiple parts of it to avoid the redundancy. For each level of the beam search is based on computing a quality measure for each subgroups in the beam aiming to minimise the overlap between them:

$$\Omega(G, Beam) = \frac{1}{|G|} \sum_{t \in G} \alpha^c(t, Beam)$$

where $\alpha \in]0, 1]$ is the weight parameter. The less often tuples in subgroup G are already covered by subgroups in the beam, the larger the score. If the cover contains only previously uncovered tuples, $\Omega(G, Beam) = 1$. In w iterations, w subgroups are selected for inclusion in the beam. In each iteration, the subgroup that maximises $\Omega(G, Beam) \cdot \phi(G)$ is selected. The first selected subgroup is always the one with the highest quality, since the beam is empty and $\Omega(G, Beam) = 1$ for all G. After that, the Ω -scores for the remaining Cands are updated each iteration

3.3 Quality measures

After we have defined how to traverse the hypothesis space, we have to define some quality measure that can tell us if a subgroup has to be kept in the beam or discarded. In Subgroup Discovery there several types of quality measure each one can be used respect to the target attribute of the process involved. The most used one is the **Weighted Relative Accuracy**, it is used for binary targets and is very common due to its good quality for permitting the beam search to exploit and explore the search space in a good way. Another common measure is the **z-score** that is used for numeric attributes, it is basically how much a number is distant from the mean of the distribution of the whole target. In exceptional model mining instead are used other quality measures that evaluate the model considered.

3.3.1 Weighted Relative Accuracy

Weighted Relative Accuracy (WRAcc) is a well-known SD quality measure for datasets with one binary target attribute. Let 1^G (resp 1^S) denote the fraction of ones in the target attribute, within the subgroup (resp entire dataset). Weighted Relative Accuracy is then defined as

$$\phi_{WRAcc}(G) = \frac{|G|}{|S|}(1^G - 1^S)$$

It is a good trade-off between the coverage of the subgroup and the accuracy of the target of interest, avoiding subgroups with small coverage or low accuracy and in this way maximising the generality of the subgroup with high accuracy. There are also approach to deal with multiple binaries target [10].

3.3.2 Z-score

The standard score is the signed number of standard deviations for a number that is above or below the mean. Thus, a positive standard score represents a number above the mean, while a negative standard score represents a number

below the mean. It is a dimensionless quantity obtained by subtracting the population mean from an individual raw score and then dividing the difference by the population standard deviation.

$$Z = \frac{X - \mu}{\sigma}$$

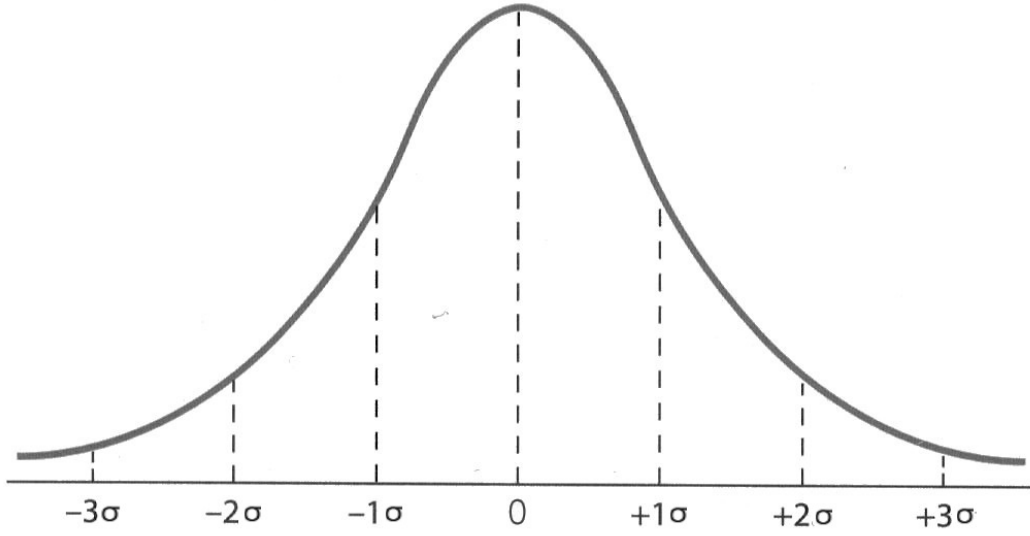


Figure 3.1: Z-score graph representation.

However in a subgroup discovery process we are not interested in the z-score of just one individual, but in the z-score of a whole subgroup's target attribute [28], which is a set of individuals. The z-score for a subgroup can be calculated by the *standardized* version of the z-score [22].

Given a subgroup G with the target attribute's mean μ_G , size $|G|$, the whole dataset target attribute mean μ_S and standard deviation σ_S its z-score is:

$$\phi_z(G) = \frac{\mu_G - \mu_S}{\frac{\sigma_S}{\sqrt{|G|}}} = \frac{\sqrt{|G|}(\mu_G - \mu_S)}{\sigma_S}$$

3.3.3 Exceptional Model mining Quality Measures

As we said before, with the subgroup discovery framework we can also search for models that are in some way *exceptional* from the whole dataset. Here we will discuss about two models: regression models and bayesian networks models. The first, as we said in the previous section, are linear models used to predict the outcome of a numeric variables given another variable, the second are models that can be fitted on multiple binary attributes. The proceeding for searching these models are the same of standard SD process but instead only considering the quality measure on the target attribute, first we build the model on the target attributes and then we evaluate the subgroup with the quality measure.

For regression models we use the difference between the slope of the regression model on the subgroup G and the regression model on its complement \bar{G} and then we define a statistic test for it [5]. For the calculation of the slope we use the least squares estimate:

$$\hat{b} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

An unbiased estimator for the variance of \hat{b} is given by:

$$s^2 = \frac{\sum \hat{e}^2}{(m-2) \sum (x_i - \bar{x})^2}$$

and then we define our test statistic:

$$t' = \frac{\hat{b}_G - \hat{b}_{\bar{G}}}{\sqrt{s_G^2 + s_{\bar{G}}^2}}$$

The degree of freedom of the t-statistic is defined as:

$$df = \frac{(s_G^2 + s_{\bar{G}}^2)^2}{\frac{s_G^4}{n-2} + \frac{s_{\bar{G}}^4}{n-2}} \quad (3.1)$$

The quality measure $\phi \in [0, 1]$ is defined as one minus the p-value computed on the basis of a t distribution with degrees of freedom given by 3.1.

For bayesian networks we use another quality measure based on the *edit distance* between two bayesian networks. The edit distance between two given graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ is the minimal number of edges we need to add to, remove from and reverse in E_1 to obtain E_2 . To select subgroups that can split well the dataset in two sets, to favour balanced splits over skewed splits we introduce also the entropy measure:

$$\phi_{weed} = \sqrt{\phi_{ent}(p)} \cdot \phi_{ed}(p)$$

Where ϕ_{ed} is the edit distance, p is the bayesian network considered and ϕ_{ent} is the entropy of the graph given by:

$$\phi_{ent} = -\frac{n}{N} \log\left(\frac{n}{N}\right) - \frac{N-n}{N} \log\left(\frac{N-n}{N}\right)$$

. where \log is the binary logarithm, n is the number of nodes in the bayesian network considered and N is the number of nodes in the complement of the subgroup. This measure anyway does not take into account the probability distribution of the variables but just the structure of the bayesian network.

3.4 ROC Space

The ROC (Receiver Operating Characteristics) space can be used to evaluate the performance of a subgroups mining process with a binary target attribute. This space has been used during the last years for evaluating classifier performance and it can be used for evaluating subgroups too if we see their descriptions as a binary classifier. Every point in the space is the evaluation measure in the ROC space of the subgroup and its coordinates are its *False positive rate* and *True positive rate*.

Before describing the ROC Space we wish to introduce some terminologies and notations:

- true positive: correct true example classified (n_{tp})
- true negative: correct negative example classified (n_{tn})
- false positive: negative example classified as true (n_{fp})
- false negative: true example classified as negative (n_{fn})

With these numbers we can then calculate:

- positives: $n_{tp} + n_{fp}$
- negatives $n_{fn} + n_{tn}$
- **true positive rate**: $t_{pr} = \frac{n_{tp}}{n_{tp} + n_{fn}}$
- **false positive rate**: $f_{pr} = \frac{n_{fp}}{n_{fp} + n_{tn}}$ or $1 - \text{true positive rate}$
- **accuracy**: $acc = \frac{n_{tp} + n_{tn}}{n_{tp} + n_{tn} + n_{fp} + n_{fn}} = t_{pr} \frac{n_{tp} + n_{fn}}{n} + (1 - f_{pr}) \frac{n_{fp} + n_{tn}}{n}$
 is the proportion of correct classifications in the test and $n = n_{tp} + n_{tn} + n_{fp} + n_{fn}$ is the total number of examples

After plotting the points in this space we can find the convex hull that contains them. The points on the convex hull form the **ROC curve**. This curve basically contains the best classifiers in terms of the accuracy in respect to some ratio of positive examples to negative ones of the dataset. Classifiers below the convex hull are always sub-optimal. The subgroups on the diagonal instead are considered random classifier and the ones under the diagonal are even worse.

3.4.1 Isometrics

Given the accuracy we can re-arrange the term and obtaining:

$$tpr = fpr \left(\frac{n_{fp} + n_{tn}}{n_{tp} + n_{fn}} \right) + \frac{1}{n_{tp} + n_{fn}} (acc \cdot n - (n_{fp} + n_{tn}))$$

a line with slope $\frac{n_{fp} + n_{tn}}{n_{tp} + n_{fn}}$ in the (fpr, tpr) plane. All the points in the line have the same accuracy acc.

The slope of the line helps to select the best classifier respect to the ratio of positive and negative examples in the dataset, in fact if we suspect that there are the same number of positive to negative examples in the training set $\frac{n_{fp} + n_{tn}}{n_{tp} + n_{fn}} = 1$ so the iso-accuracy line with slope equals to 1 that intersects the roc convex hull is the best for the datasets.

Optimizing accuracy gives equal weight to covering a single positive example and excluding a single negative example. There are cases where this choice is arbitrary, for example when misclassification costs are not known in advance or when the samples of the two classes are not representative. This is the case with WRAcc where we give equal weight to increasing the true positive rate or to decreasing the false positive rate. The main difference with accuracy is that the isometrics are parallel to the diagonal, which reflects that we now give equal weight to increasing the true positive rate or to decreasing the false positive rate. This is the best approach for the subgroup discovery process, its explorative nature can be maximised indeed.

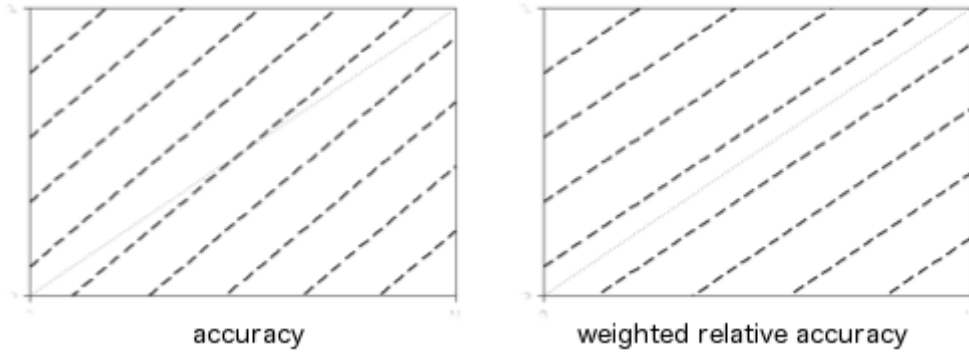


Figure 3.2: Isometrics lines for Accuracy and WRAcc.

3.4.2 AUC

Another interesting measure that we can associate to a ROC space is the *Area under the ROC curve* or AUC¹ that ranges between 0 and 1. This measure assesses the classification in terms of separation of the classes.

- all the positives before the negatives: $AUC = 1$
- random ordering: $AUC = 0.5$
- all the negatives before the positives: $AUC = 0$

The AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

Other interesting coordinates in the ROC Space can be expressed by their coordinates (FPr,TPr):

- (1,0): perfect subgroup, the target attribute is true for all the examples
- (0,0): empty subgroup
- (0,1): perfect negative subgroup
- (1,1): subgroup that covers the entire database

3.4.3 Evaluation of the mining process

In Figure 3.3 a roc curve from a subgroups mining process is plotted. As we can see there is a huge number of points under the roc curve each one representing a subgroups discovered by the process. This huge number is due to the high redundancy of the results discovered by the beam search (the overlapping of

¹ To compute the AUC just find the convex hull and compute the sum of the area of the triangles with the vertexes in the origin and on the roc curve. Given the points on the curve the formula is $\sum_{i=-1}^n ((FP(i+1) - FP(i)) * (TP(i) + TP(i+1))) / 2$ where FP(i) stands for the false positive rate of the i point and TP the true positive rate.

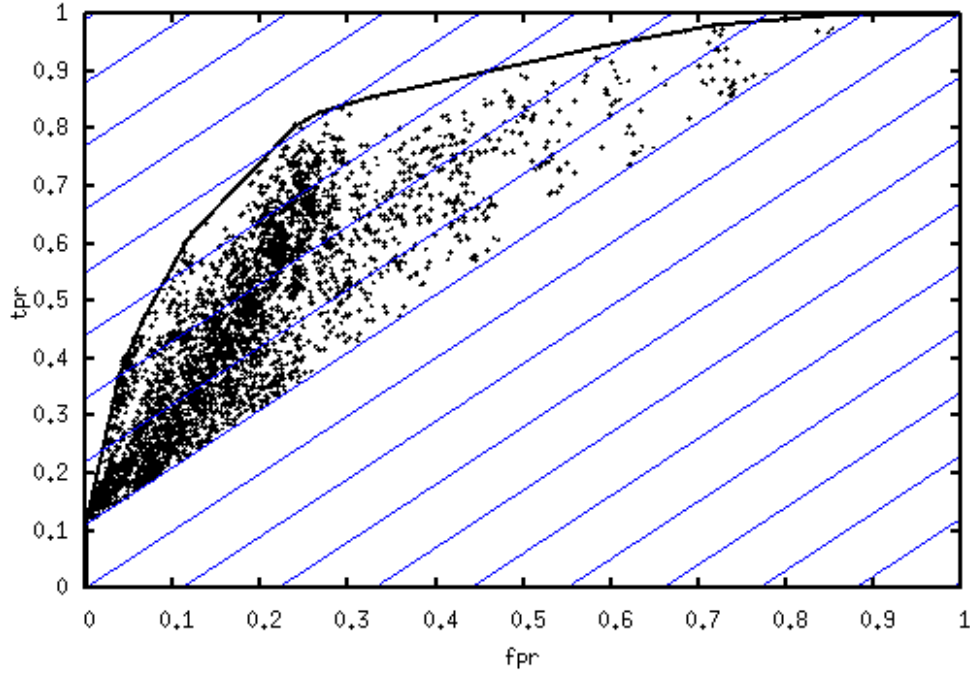


Figure 3.3: Roc Curve with WRAcc isometrics lines in blue.

the subgroups is the main cause). The best thing for the mining process should be to take in the results only the subgroups under the spike of the curve and discarding the ones below a certain region and possibly get different pattern descriptions to obtain different descriptions of the local patterns but as we said before the standard beam search cannot do this. An approach to resolve this has been described in section 3.2.4 but in this document we will see another approach that is based upon the help of the user. The evaluation of the roc curve will be remain a central process anyway.

3.4.4 Nominal value set and Numeric Intervals

It is also possible to get richer descriptions of the subgroups [26]. This method basically extends the pattern language for the subgroups descriptions using numeric intervals rather than the numeric operators for the numeric attributes

and set of nominal values for the nominal attributes. All the process has been made with linear complexity and only for the Weighted relative accuracy. This can lead to obtain higher quality subgroups. The main idea under this work is to consider only refinements of the subgroups that lie on a convex hull in ROC space, thus significantly narrowing down the search space.

Chapter 4

Statistical background

In this chapter we wish to underline some backgrounds on statistics, useful to understand how the user can interact with the data.

4.1 Null hypothesis, statistical independence and association-correlation

When statisticians have to analyse two statistical variables they first proceed to formulate two hypothesis about them:

- the **null hypothesis** H_0 : there is no relationship between the variables
- the **alternative hypothesis** H_1 : there is some kind of relationship between the variables

After formulating these hypotheses they use some kind of *statistic test* to *accept or reject* the null hypothesis. The type of the test depends on the type of the variable and which type of data the variable represents. If the null hypothesis will be accepted (the test will be negative) we can say that there are no relations between the two variables, they are **independent** from each other, otherwise if the null hypothesis will be rejected we can confirm that there is some kind of relationship and then we have to proceed the investigation using

some analysis procedure based up on the type of variables and the data. The relationship type between these variables could be of **association** or **correlation**. Association generally refers to any kind of relationship and correlation can be considered as a special case of association, where the relationship between the variables is linear in nature so correlation can be considered a stricter relation between the variables than association. Usually they are both described by a number that summarises the strength of the relationship. Something that is generally misunderstood is that association or correlation do not mean *causation*, causation has to be evaluated by other experiments. In our research we will deal only with *binary variables* and we will focus only on methods to analyse this kind of statistical variables.

4.2 Contingency Table analysis

The term contingency table was first used in [27]. The contingency table for binary variables are called *contingency table 2x2* because they have only four entries. The degree of freedom of these tables equals to one ($df = (rows - 1) * (columns - 1)$).

	0	1	
0	a	b	n_1
1	c	d	n_2
	n_3	n_4	N

Table 4.1: Contingency table 2x2.

Contingency tables are used to display the frequency distribution of the variables. If X stands for the first variable and Y for the second than the letters inside the contingency are:

- a: number of zeros in the same positions both in X and Y
- b: the same as a but with 0 in X and 1 in Y
- c: 0 in X , 1 in Y
- d: 1 in X and 0 in Y
- n_1, n_2, n_3, n_4 : sum of $a + b, c + d, a + c, b + d$ respectively. They are also called *marginal frequencies*
- N : sum of $n_1 + n_2 + n_3 + n_4$, called also *numerosity of the population*

Contingency tables are useful to describe in a visual way the distribution of two statistics variables. In the statistics literature we can find different coefficients related to them.

A coefficient is just a numeric measure that reflects some aspects between the variables. There are different types of coefficients each one with specific numerical properties [2].

For our research we will investigate the *Yule's Q* coefficient and the *Phi* coefficient that have been well described in [35]. With these coefficients and some other tests, a statistician can test if the null hypothesis is true or not. This process is called *non-parametric statistics* because there is no a priori assumption on the characteristics of the data or population taken into exam.

Historically these types of coefficients has been used by the scientists to compare different species of animals, their properties, such as shape, fur, presence of wings or not etc. were described by a binary variable and then the scientists with the support of the coefficients could do similarity analysis between the species and conclude if the species were similar and then build some sort of family of species.

4.3 Yule's Q Coefficient

$$YulesQ_{coeff} = \frac{(ad - bc)}{(ad + bc)}$$

Yule's Q coefficient is an association coefficient that means that reaches high values even if the relationship is not completely linear. It is the ratio between where the two variables are concordant and the total number of values of the variables. It takes range between -1 and 1.

4.4 Phi Coefficient

$$Phi_{coeff} = \frac{(ad - bc)}{\sqrt{(n_1 \cdot n_2 \cdot n_3 \cdot n_4)}}$$

The phi coefficient is the binary version of the well famous Pearson's coefficient's coefficient [31]. It is a correlation coefficient, in fact it increases/decreases only if there is correlation between two variables that means if there is only linear association. See [38] for a review on the literature on phi coefficient and some of its modifications. It is the ratio between where the two variables are concordant and the geometric mean of the marginal frequencies. It also takes range between -1 and 1.

4.5 Numerical properties of phi and yule's q coefficient

It could be interesting to see how these two coefficients take range between -1 and 1, their minimum and maximum values respectively. First of all we can say that they are both zero if the two binary variables share zero covariance, thus they are independent. Probability theory tells us that two binary variables satisfy statistical independence if the odds ratio equals unity, that is

$$\frac{ad}{bc} = 1$$

The odds ratio is defined as the ratio of the odds of an event occurring in one group (a/b) to the odds of it occurring in another group (c/d). An odds ratio of 1 indicates that the condition or event under study is equally likely in both groups. An odds ratio greater than 1 indicates that the condition or event is more likely in the first group. That formula above gives us the numerator of the two coefficients that is $ad - bc$. So, when this value is zero the whole coefficient goes to zero and the two binary variables are independent. To view instead how the coefficients take range between -1 and 1 it could be easier to do it in a visual way.

Below we report a graphical visualisation of the comparison between a random binary vector and 100 other vectors with linear increasing differences:

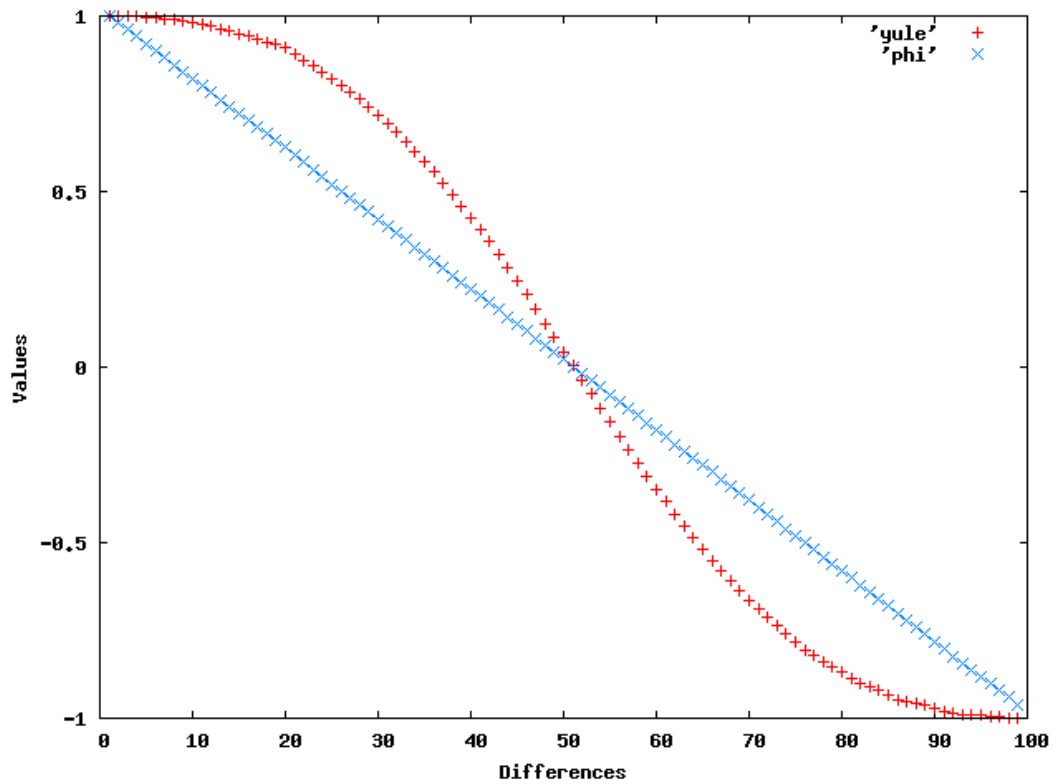


Figure 4.1: Coefficients ranges of Phi and Yule's Q coefficients.

As we can see, the phi coefficient is linear in nature, the Yule's Q instead is smoother.

Finally it could be interesting to see how statisticians usually set thresholds to determine if there is weak, moderate or high correlation/association. Due to the symmetric nature of the coefficients it is easy to see:

- $-1 \leq coefficient < -0.8$: **strong negative** association/correlation
- $-0.8 \leq coefficient < -0.5$: **high negative** association/correlation
- $-0.5 \leq coefficient < 0$: **weak negative** association/correlation
- $0 \leq coefficient < 0.5$: **weak positive** association/correlation
- $0.5 \leq coefficient < 0.8$: **high positive** association/correlation
- $0.8 \leq coefficient \leq 1$: **strong positive** association/correlation

4.6 Chi squared test

A chi-squared test, also referred to as chi-square test or χ^2 test is a statistical test for testing independence between two statistical variables. The computation of this test with a 2x2 contingency table is:

$$\chi^2 = \frac{N(ad - bc)^2}{(n_1 \cdot n_2 \cdot n_3 \cdot n_4)}$$

This test returns a number that defines the independence between the two variables. With this number we can also calculate the related p-value that is the probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. To calculate this p-value given the χ^2 test we can use the algorithm that is linear in the number of degree of freedom of the associated contingency table [6]. Usually the common p-value used for the passing of the test is 0.05, if it is greater than this the test fails and we can say that the differences between the contingency tables are only a matter of chance. It is reasonable to do a χ^2 test after the computation of the coefficients mentioned above to confirm or not the results from them.

4.7 P-value

Below we report the table for the *critical values* for χ^2 for binary variables.

p-value	0.25	0.20	0.15	0.10	0.05	0.025	0.02	0.01	0.005	0.0025	0.001	0.0005
χ^2	1.32	1.64	2.07	2.71	3.84	5.02	5.41	6.63	7.88	9.14	10.83	12.12

Table 4.2: Critical chi-squared values.

Given a result from a χ^2 test, the tester can look up in the table and if the result from the test is between the one reported in the cell, the equivalent p-value is reported in the cell above.

4.8 Fisher's Exact test

In the statistical literature we can find that if we do not have a large enough population we cannot apply the chi-squared test, instead we should use another approach, the Fisher's exact test that is indeed an exact method [9].

$$\text{p-value} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!}$$

4.9 Bonferroni procedure

Bonferroni procedure is a statistical procedure to obtain a singular p-value from multiple p-values coming from different experiments. Let be $p_{(1)}, p_{(2)}, \dots, p_{(m)}$ the p-values from m tests sorted in increasing order. The p-value p_B for the combined tests is:

$$p_B = m \cdot p_{(1)}$$

This formula means that you can test the hypothesis at some pre-determined level α , then you can reject the null hypothesis if at least one of the m tests has a p-value less than α/m . [32]

4.10 Coefficients Examples

In this section we will show some coefficients example values for some binary variables to get a general idea on how calculate them. For the example we will use five binary variables with $N = 12$. We will compare the first of them (a) with the others with 25%, 50%, 75%, 100% of differences between them.

The binary vectors a, b, c, d are:

0	0	0	1	1
0	0	0	1	1
1	0	1	0	0
0	0	1	1	1
1	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	1	0	0	0
0	0	1	1	1
1	1	1	1	0
0	0	0	0	1
0	0	0	0	1

(a) Binary variables

And their contingency tables $(a, a), (a, b), (a, c), (a, d)$:

	0	1		0	1		0	1		0	1		0	1	
0	7	0	7	0	6	2	8	0	4	3	7	0	2	4	6
1	0	5	5	1	1	3	4	1	3	2	5	1	4	2	6
	7	5	12		7	5	12		7	5	12		6	6	12

(a) Contingency tables

The resulting coefficients values are reported in the table below:

Variables	Yule's Q	Phi coefficient	Fisher's test (p-value)
(a,a)	1	1	0.0013
(a,b)	0.8	0.47	0.1515
(a,c)	-0.05	-0.2	0.6894
(a,d)	-0.6	-0.33	0.2835
(a,e)	-1	-1	0.0013

Table 4.3: Coefficients values.

As we can see the range of the coefficients are between -1 and 1 with the Yule's Q becoming more sparse than the phi coefficient that takes more values between the bounds than in the "centre" of the range. However only the first and the last comparison are statistically relevant as we can see from their p-value, basically because the associated contingency tables are too much similar.

Chapter 5

Interactive Subgroup Discovery

In this chapter we will describe how to implement an interactive Subgroup Discovery system. The features that this system should have and how the user can interact with it. Finally we will investigate how to influence the search of the subgroups with the user preferences.

5.1 Interactive Data mining systems

The main two goals of an interactive data mining systems are integrating the user's background knowledge to **improve the quality** of the patterns extracted by the data mining system and possibly to get **new knowledge** coming out from this interaction [37]. This can be seen as a process that loops until the user is satisfied. In this process the user is involved for the pattern creation. At the end of the process the user finally can get a visualisation of the patterns extracted by the algorithm to judge the quality of the knowledge extracted and then decide to continue the loop or ending it. The overall quality of the process can be calculated by the efficiency of the communication between the user and the system, basically how much the user can understand the knowledge extracted and how easily he can influence new results. Figure 5.1 reports a scheme for the process. The main part is still acted by the data mining system but now the user through some sort of interaction interacts with it. After this interaction the system applies some algorithm to the data

and get the patterns from it. Finally the user get a visualisation of the patterns. The process now can be continued or can be stopped.

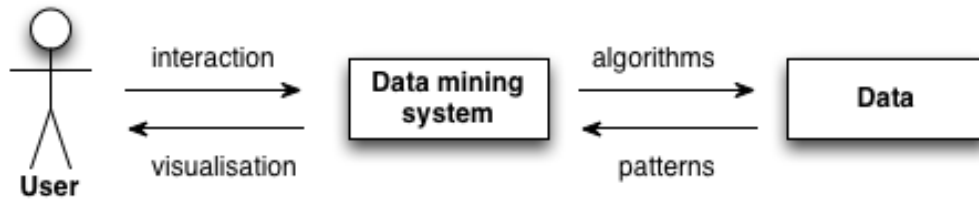


Figure 5.1: Interactive data mining systems.

5.2 Influencing the beam search

After we have defined what is an interactive data mining system, we have to develop one for the subgroup discovery task. The main purpose of the system will be reducing the hypothesis space to select part of it of interest for the user and possibly extract other knowledge that is not possible to discover with the standard process. All the process will use the statistics techniques that we described before and the user's input. We will not take into consideration the exhaustively approach (see deep search first) but we rather focus only on the beam search and we will use only the weighted relative accuracy as reference for quality measure but as we will see later this approach can be used with every **symmetric** quality measure for subgroup discovery.

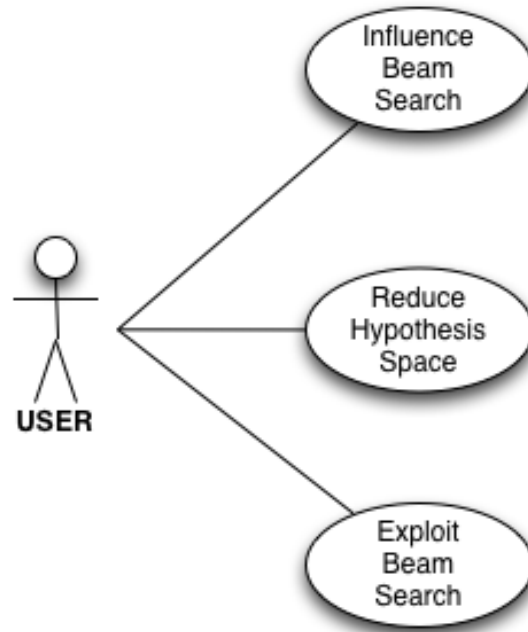


Figure 5.2: Use case for interacting with the beam search.

The user's objective can be summarised as, given a set of subgroups which he is interested on, he should be able to discover **similar** subgroups to them, **removing** similar subgroups from the beam and searching for **opposite** subgroups to the given ones. These concepts of course are intended for the rows in the table that the subgroups cover. After that, it is desirable also to exploit the power of the beam search to **explore exhaustively** part of the space and filtering irrelevant subgroups with the quality measure maybe manipulating the intent of the subgroups or expanding only some subgroups in the result rather than the others (investigation of a subgroup) or even every subgroup (search deeper).

The user's activity diagram (Figure 5.3) describes the main user's activities. First of all he will set the search parameters for the subgroups mining process and then run a first search with these. Because of the nature of the beam search to introduce a lot of redundancy we advice the user to choose a depth search between 1 or 2, depending on the dataset and then proceed to the interaction process. After the first run search the user can interact with the results in several ways, basically manipulating the intents of the subgroups in the result or manipulating the beam search through their extents (we will see later how actually influence it). In the intents manipulating we find the disabling of the attributes of the table that means that the entire attribute in the data table will be ignored for the searching process and the dislikes of the intents of the subgroups, this will be analysed in the implementation design. The focus on the subgroup in some way is also an intent manipulation, in fact it consists in "brute-forcing" the subgroup description and search for every value of its attribute: it is possible that some description could have been discarded in the previous level of the beam search but maybe they can be still interesting in the user perspective. Another possible way to manipulate the intent of the subgroup is to "investigate it" that is expanding its description with the next level description, so for example if the length of the description is 1, all the refinements will be generated and the corresponding subgroups with depth equals to 2 will be evaluated. After this there is the extent manipulating that consists in search for similar (like), different (dislike), opposite (opposite), subgroups extents. In the final stage of the process it is possible to get graphical visualisation of result of the interaction on the results and then get a statistical validation of the process. After that it will be possible to filter the results with a filtering method based on the evaluation of thresholds for the quality and the coefficients values.

5.3 Interactive beam search

Below we report the pseudo-code for our interactive beam search implementation. It is analogous to the standard beam search except two important things.

The first one is reduction of the attributes domain by the dislikes of the subgroups intents and the expansion of the candidate set with the investigated subgroups and the second one is how the subgroup quality measure is evaluated.

The new quality for the candidate subgroup has to be **balanced** between the original quality and the quality from the interactive evaluation, we do not want to prefer one or another. The first step to do so is to have the qualities to have the same ranges. For the Weighted Relative Accuracy it is easy to do so, it is just a matter to divide it by its maximum. The maximum is reached when a subgroup contains all the 1s in the dataset and its complement only zeros, if G_1 is this subgroup:

$$WRAcc_{max} = \frac{|G_1|}{|S|}(1^{G_1} - 1^S)$$

$1^S = 0$, $|G_1| = N$ where N number of the 1s in the dataset and $|S| = K$ where K is the number of zeros.

After defining the $WRAcc_{max}$ quality, we can obtain a quality with range between -1 and 1 simply dividing it with its maximum:

$$WRAcc_{new} = \frac{WRAcc}{WRAcc_{max}}$$

With this new quality we should find a sort of linear combination with the user input to discover subgroups closer to the user's interest. This linear combination consists in two parts. In the first one, for each set of interest (like, dislike, opposite) we calculate a new single quality, then we combine these qualities with the original one to get the **new score**. The main concept behind this procedure is using the contingency table analysis between the user's input subgroups and the candidate subgroups. Every subgroup can be effectively seen as **binary vectors** with length equal to the number of rows in the data table and every value x_i of this binary vector is 1 if the subgroup covers the i row in the table and 0 vice versa.

5.3.1 Like

For subgroups that the user likes, as we said before, we have to search for subgroups that more or less cover the same rows in the table. To do that we can use the Phi Coefficient described in Section 4.4. The new quality for the subgroups like will be the sum of all the candidate subgroups that have a **positive correlation** (the coefficient has to be greater than 0) with the ones liked by the user calculated using the phi coefficient between the candidate and the liked subgroups. This can easily exclude the ones that are not correlated or correlated in a negative way. Using a correlation coefficient as the phi coefficient we can focus better the beam search on subgroups that are similar in a linear way to the input ones basically because the nature of the coefficient permits to reach high values only with linear associations.

5.3.2 Dislike

For subgroups that the user dislikes, we have to remove them from the beam because he does not want to find in the results subgroups that cover those rows. Again we use the phi coefficient but rather than summing the results we subtract the result **reducing** the original quality measure if there is a positive correlation.

5.3.3 Opposite

To search for opposite subgroups that are subgroups that have binary vectors with *inverted* indexes (where is 1 there is 0 and vice versa) we use the Yule's Q coefficient when there is **negative association** between the candidate and the input opposite subgroups. Using an association coefficient as Yule's Q rather than a correlation coefficient like the phi coefficient permits to get a more variate result thus exploring a larger hypothesis space.

5.3.4 Combining the qualities

We will call each of the qualities calculated *like_quality*, *dislike_quality* and *opposite_quality* respectively. After calculating these qualities we have to find a method to combine them. The method is simple: we have just to increase or decrease the original quality of the subgroup. To do that, after obtaining the original quality measure range between -1 and 1 we add the *like_quality* divided by the number of the liked subgroups, we subtract *dislike_quality* divided by the number of the disliked subgroups and finally we subtract *opposite_quality* divided by the number of the opposite subgroups. The divisions of the quality measure are needed to keep all the quality measure in range -1 and 1 thus obtaining a balanced evaluation of the subgroup between the input of the user and the original quality considered for the beam.

The final quality of the candidate subgroup therefore can be summarised as a **linear combination** with all the variables in range between -1 and 1:

$$\phi(c) = \text{quality}(c) + \text{like_quality}(c) - \text{dislike_quality}(c) + \text{opposite_quality}(c)$$

As the reader can image this approach can work only if the quality measure taken into account is symmetric otherwise this method is impassable. Because both the WRAcc and the z-score are symmetric measures and they are the most used in the subgroup discovery field we consider it a good results. However it could be still interesting to search for a similar approach for Exceptional Model Mining.

5.3.5 Subgroup Investigation

Finally we wish to describe how to investigate a subgroup. Basically when the user wants to expand one subgroup or even all the subgroup in the results of a search process we could apply a simple strategy. If the attribute that we are considering is nominal we proceed as the standard procedure does thus considering every values of the attribute and we evaluate them as usual, otherwise if the attribute is numeric we could set a threshold indicating if the domain

of the attribute is smaller than the threshold we consider every values with the usual numeric operators $=, \geq, \leq$. The results from the investigation can be evaluated with the same interactive quality measure of the results, therefore when we expand a subgroup its quality will be evaluated in the same way.

5.3.6 Subgroup Focus

The focus on the subgroup, as we said will evaluate every values for every attributes in its description. The process is similar to the investigation but here we will evaluate every values also for numeric attributes.

5.3.7 Validation

After the beam search is finished reporting the results, it could be also interesting going through a validation of the results in a statistical point of view. This can be done obtaining the p-value from the χ^2 with the input subgroups and the subgroup in the results. If the subgroup is not larger enough as we said in the previous section 4.6 we can use the Fisher's exact test. If the user has interacted with one or more subgroups from the results we can apply the Bonferroni procedure obtaining a single p-value from the various comparisons. The validation process is crucial because as we will see the user will be able to see how his interaction has spread on the data and giving to him a validation of the process is significant to obtain good results.

5.4 About coefficients indefiniteness

Someone could argue and be worried that during the calculation some coefficients could be undefined due to a division by zero wasting the beam search but this is not possible because the coefficients will not never divided by zero or will be equal to $\frac{0}{0}$ simply because there will be never a computation of them with an empty subgroup. Only with an empty binary vector there are problems with the use of coefficients. In fact without an empty subgroup all the coefficients denominators $ad + bc$ for Yule's Q and $\sqrt{((n_1 \cdot n_2 \cdot n_3 \cdot n_4))}$ for

the Phi coefficient will never be equal to 0 because for the first denominator either ad is equal to 0 or bc and never both of them and for the Phi Coefficient all the marginal frequencies are always greater than 0 because there will be always at least 1 zero in a or b or c or d if the subgroups are not empty so the values are always greeter or equal to 1.

Algorithm 2 Interactive Beam Search

Input: A dataset S , a quality measure ϕ and parameters k as quality minimum, w as beam width, $mincov$ and $maxdepth$ **likes as vector of subgroups, dislikes_intent as vector of subgroups, opposite as vector of Subgroups, dislikes_intents as a vector of descriptions, investigate as vector of subgroups, focus as vector of subgroups**

Output: R , an approximation of the top- k subgroups G_k respect to user's interest and quality measure.

```
 $R \leftarrow \emptyset, B \leftarrow \{\emptyset\}, depth = 1$ 
reduceDomain(dislikes_intents)
while  $depth \leq maxdepth$  do
     $Cands \leftarrow \emptyset$ 
    for all  $b \in Beam$  do
         $Cands \leftarrow Cands \cup GenerateRefinements(b, mincov)$ 
    for all  $i \in investigate$  do
         $Cands \leftarrow Cands \cup GenerateRefinements(i, mincov)$ 
    for all  $f \in focus$  do
         $Cands \leftarrow Cands \cup Focus(f, mincov)$ 
    for all  $c \in Cands$  do
        for all  $l \in likes$  do
            if  $PhiCoefficient(c, l) > 0$  then
                 $like\_quality \leftarrow like\_quality + PhiCoefficient(c, l)$ 
        for all  $d \in dislikes$  do
            if  $PhiCoefficient(c, d) > 0$  then
                 $dislike\_quality \leftarrow dislike\_quality - PhiCoefficient(c, d)$ 
        for all  $o \in opposites$  do
            if  $YulesQ(c, o) < 0$  then
                 $opposite\_quality \leftarrow opposite\_quality - YulesQ(c, o)$ 
         $\phi(c) \leftarrow \phi(c) / max\phi(c)$ 
         $\phi(c) \leftarrow \phi(c) + like\_quality / |likes|$ 
         $\phi(c) \leftarrow \phi(c) - dislike\_quality / |dislike\_quality|$ 
         $\phi(c) \leftarrow \phi(c) - opposite\_quality / |opposite|$ 
         $UpdateTopK(R, k, c, \phi(c))$ 
     $Beam \leftarrow SelectBeam(Cands, w, \phi)$ 
     $depth \leftarrow depth + 1$ 
return  $R$ 
```

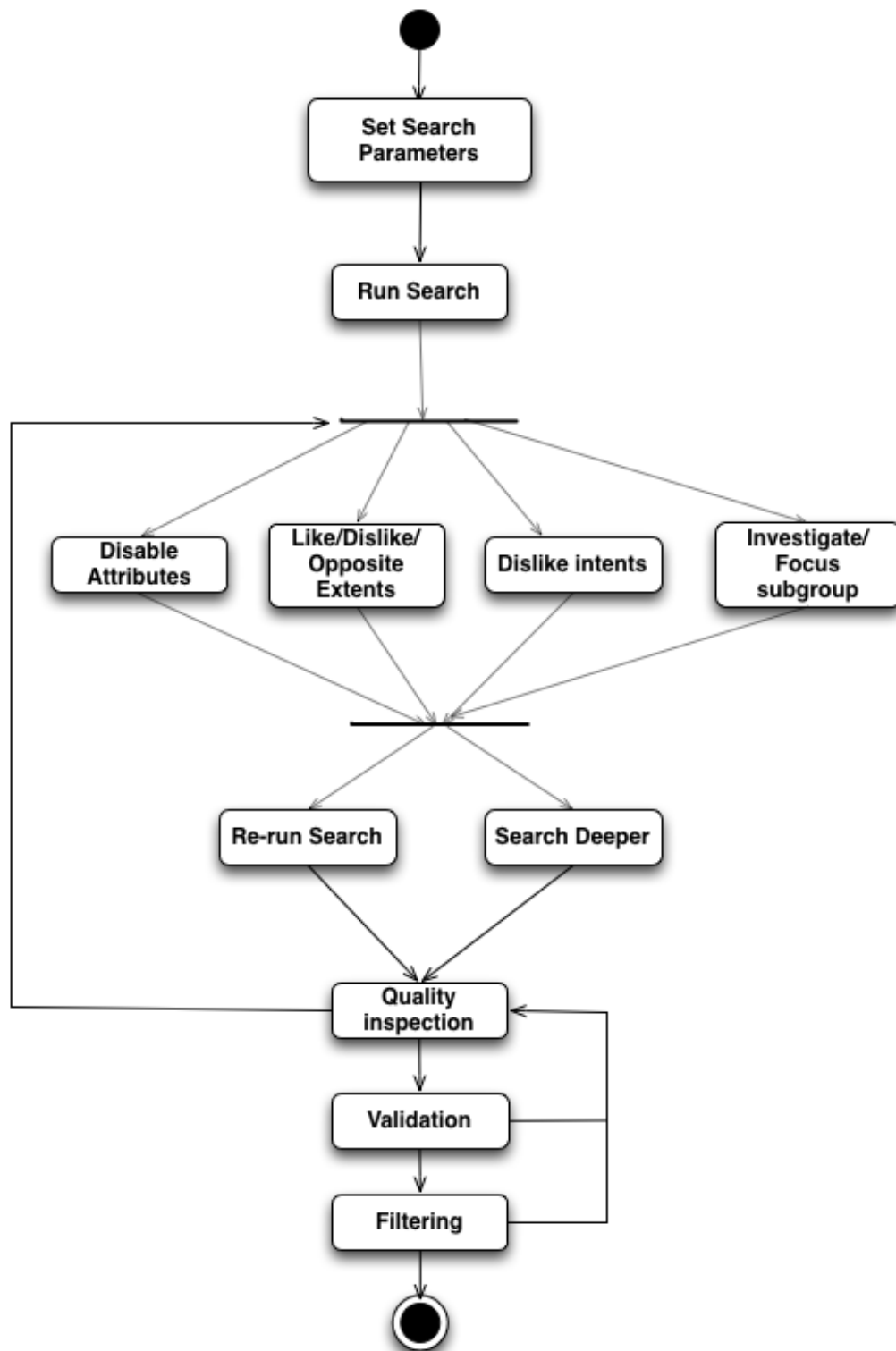


Figure 5.3: User activity diagram.

Chapter 6

Cortana

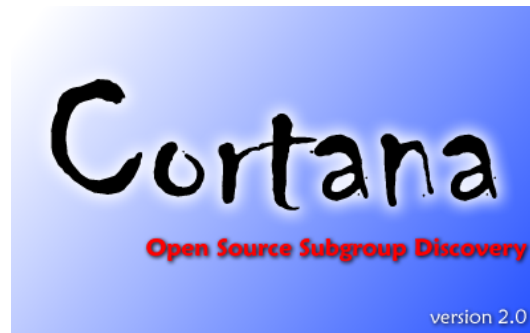


Figure 6.1: Cortana logo.

In this chapter we will describe our interactive implementation. In the first section of the chapter we will introduce the reader to Cortana, the data mining tool where we developed the implementation, its main feature and objectives. After that we will describe the interactive section for Cortana.

6.1 Background and features

Cortana is a Data Mining tool for discovering local patterns in data. Cortana features a generic Subgroup Discovery algorithm that can be configured in many ways, in order to implement various forms of local pattern discovery. The tool can deal with a range of data types, both for the input attributes as

well as the target attributes, including nominal, numeric and binary. A unique feature of Cortana is its ability to deal with a range of Subgroup Discovery settings, determined by the type and number of target attributes. Where regular SD algorithms only consider a single target attribute, nominal or sometimes numeric, Cortana is able to deal with targets consisting of multiple attributes for Exceptional Model Mining [11]. Cortana has been written in Java by the members of the LIACS Data Mining Group [12] at Leiden University. Its main role is to provide a common platform for the research group to make experiments and implementing new algorithms that come from their studies. The authors used several libraries to help the tool's developing but most of its code has been written from scratch.

6.2 Loading Data and setting the search parameters

The Cortana's main window consists of 4 main sections. The sections are: the **dataset**, the **search conditions**, the **search strategy** and the **target concept** sections.

Under the main window there are four buttons:

Interactive: it starts the interactive process. This is where all the features described in this thesis for the interactive subgroup discovery can be used and evaluated

Subgroup Discovery: it starts the standard subgroup discovery process

Cross-validate: it performs a *cross-validation* of the subgroup discovery process

Compute Threshold: it computes a threshold for the quality measure, the algorithm is described in [36]

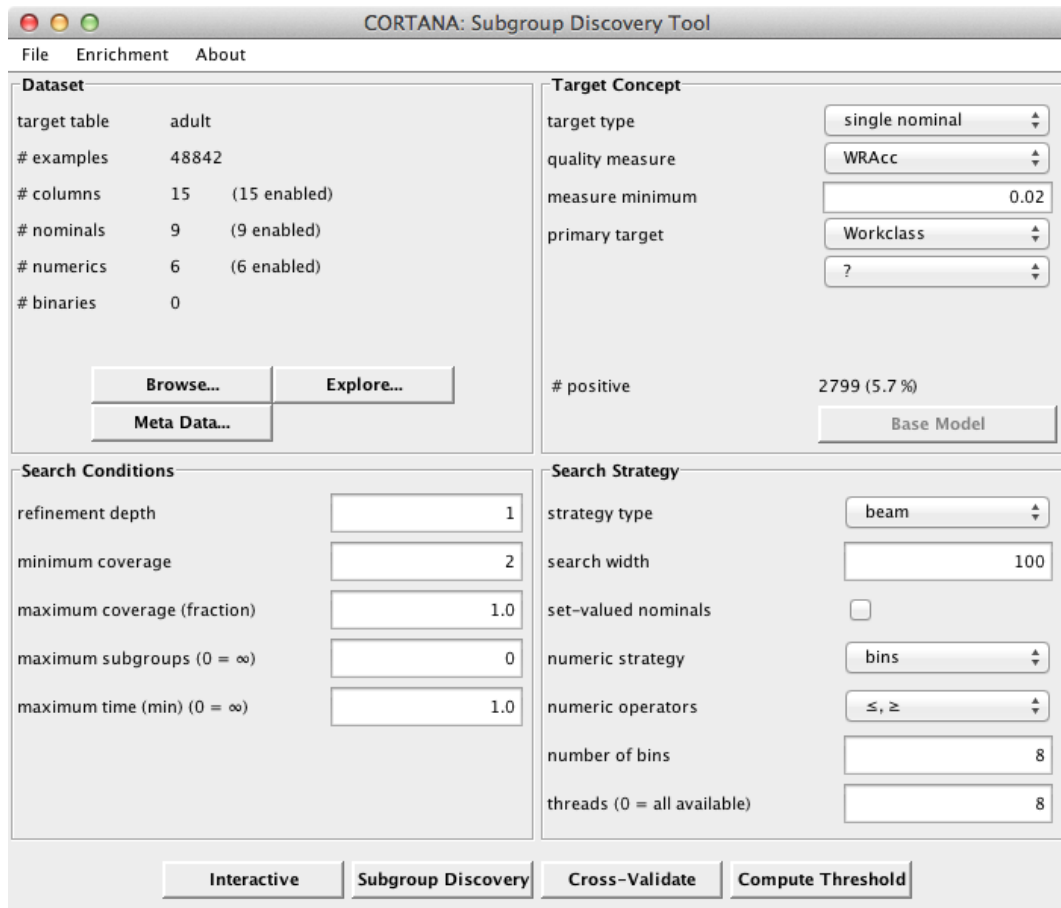


Figure 6.2: Cortana’s main window.

6.2.1 Dataset section

In the dataset section the main properties of the dataset are displayed. These properties are the target table name, the dataset number of examples, the number of column of the data table and the number of nominals, numeric, binaries attributes. There are also 3 different buttons. The **Browse** button will open up another window displaying the rows of the data table, the **Explore** button will pop up another window displaying several histograms about the data and finally the **Meta Data** button permits the user to see the type of the attributes and eventually change their types.

Dataset		
target table	adult	
# examples	48842	
# columns	15	(15 enabled)
# nominals	9	(9 enabled)
# numerics	6	(6 enabled)
# binaries	0	

Browse...

Explore...

Meta Data...

Figure 6.3: Dataset details.

6.2.2 Target Concept section

Target Concept	
target type	single nominal
quality measure	WRAcc
measure minimum	0.02
primary target	Class
	> 50K
# positive	11687 (23.9 %)
Base Model	

Figure 6.4: Target concept section.

In the target concept section the user can select the target type of the target column this includes single nominal, single numeric for the singular targets but it includes also the settings for the exceptional model mining such as double regressions, double correlations and multi-label (bayesian networks) settings. For each target type it will show also some measure on the whole dataset: for

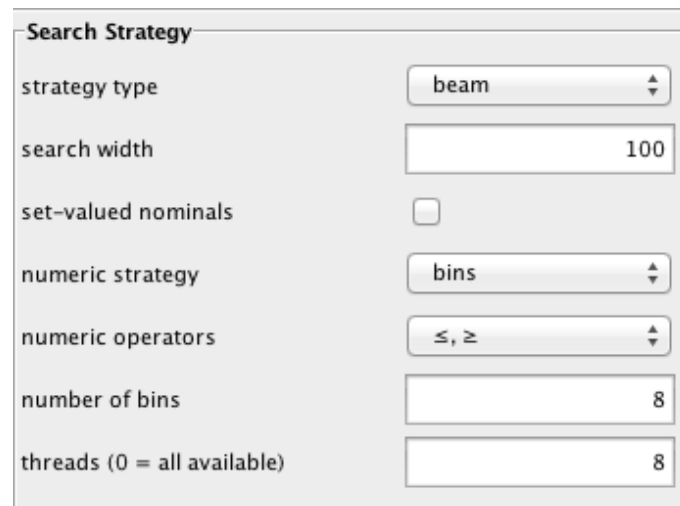
single nominal targets the number of positive, for single numeric targets the average of the target, for regressions the correlation between the values and for the multi-label setting it shows the bayesian network fitted on the data.

6.2.3 The search conditions

Search Conditions	
refinement depth	3
minimum coverage	10
maximum coverage (fraction)	1.0
maximum subgroups (0 = ∞)	0
maximum time (min) (0 = ∞)	1.0

Figure 6.5: Search conditions.

In the search strategy section the user can set several parameters for the exploration of the hypothesis space. The first one is the **strategy type**, it includes the beam search, the cover-based beam search, the depth-first search and the breadth first search. The next parameter is the **search width**, that is of course the width of the beam search, how many subgroups refinements can be extended in the next level. Set-valued nominals is a checkbox for using the algorithms described in [26]. Numeric operators are the operators to use with the bins numeric options, they can be set as \leq , \geq , only \leq , only \geq , \leq , \geq and $=$ and finally only $=$. Number of bins are the number of equal size bins for the numeric attributes and threads are the number of concurrent threads that can be spawned for the search, lowering the total time for the search.



The image shows a 'Search Strategy' dialog box with the following settings:

Parameter	Value
strategy type	beam
search width	100
set-valued nominals	<input type="checkbox"/>
numeric strategy	bins
numeric operators	\leq, \geq
number of bins	8
threads (0 = all available)	8

Figure 6.6: Search strategy.

6.2.4 The search strategy

The search conditions are the other configurations remaining for the beam search. The **refinement depth** is the number of levels for the beam search, the **minimum coverage** is the minimum coverage for the subgroup that has to take to be considered, the **maximum coverage** is the fraction of rows that can cover the subgroup, the **maximum subgroups** are the maximum number of subgroups that can be displayed in the results and the **maximum time** is the maximum time that the beam search can take for the whole process.

6.2.5 The results

After defining the search parameters the search's results are displayed in the **results window**. Here, for each subgroups are displayed the id of the subgroup (its position compared to the quality of the others subgroups), the depth of its description, its coverage, its quality, the probability of the target attribute to be true within the subgroup, the number of positives examples covered by the subgroup and finally its description's conditions. The p-value column is initially blank but after pressing the button Gaussian p-values it is been displayed the number coming from the procedures described in [36]. Basically it is a statistical validation of the whole Subgroup Discovery process.

Nr.	Depth	Coverage	Quality	Probability	Positives	p-Value	Conditions
1	3	10341	0.102031	0.111155	9416	-	Age >= 23.0 AND Marital status = 'Married-civ-spouse' AND Education num >= 9.0
2	3	19573	0.101911	0.493588	9661	-	Age >= 23.0 AND Marital status = 'Married-civ-spouse' AND Education num >= 9.0
3	3	17467	0.101254	0.522414	9125	-	Marital status = 'Married-civ-spouse' AND Education num >= 9.0 AND Age >= 30.0
4	3	17630	0.101213	0.519682	9162	-	Education num >= 8.0 AND Marital status = 'Married-civ-spouse' AND Age >= 30.0
5	2	19794	0.100972	0.488431	9668	-	Marital status = 'Married-civ-spouse' AND Education num >= 9.0
6	2	19993	0.100836	0.48562	9709	-	Education num >= 8.0 AND Marital status = 'Married-civ-spouse'
7	3	18512	0.100148	0.503511	9321	-	Hours per week >= 28.0 AND Marital status = 'Married-civ-spouse' AND Education num >= 9.0
8	2	20664	0.097774	0.470383	9720	-	Age >= 28.0 AND Marital status = 'Married-civ-spouse'
9	3	17455	0.09771	0.51269	8949	-	Marital status = 'Married-civ-spouse' AND Hours per week >= 36.0 AND Education num >= 9.0
10	3	16406	0.097648	0.529989	8695	-	Age >= 32.0 AND Marital status = 'Married-civ-spouse' AND Education num >= 9.0
11	3	17364	0.097234	0.512785	8904	-	Marital status = 'Married-civ-spouse' AND Education num >= 9.0 AND Hours per week >= 37.0
12	3	17540	0.09717	0.509863	8943	-	Education num >= 8.0 AND Marital status = 'Married-civ-spouse' AND Hours per week >= 37.0
13	2	19703	0.096319	0.478049	9419	-	Marital status = 'Married-civ-spouse' AND Age >= 30.0
14	2	22992	0.095999	0.451521	9975	-	Age >= 23.0 AND Marital status = 'Married-civ-spouse'
15	3	17080	0.095882	0.513466	8770	-	Marital status = 'Married-civ-spouse' AND Education num >= 9.0 AND Hours per week >= 40.0
16	3	18360	0.095876	0.494336	9076	-	Hours per week >= 28.0 AND Marital status = 'Married-civ-spouse' AND Age >= 30.0
17	3	17255	0.095823	0.510519	8809	-	Education num >= 8.0 AND Marital status = 'Married-civ-spouse' AND Hours per week >= 40.0
18	3	15835	0.095471	0.533754	8452	-	Education num >= 9.0 AND Age >= 33.0 AND Marital status = 'Married-civ-spouse'
19	3	15979	0.095441	0.531009	8485	-	Education num >= 8.0 AND Age >= 33.0 AND Marital status = 'Married-civ-spouse'
20	3	18138	0.09506	0.495259	8983	-	Age >= 28.0 AND Marital status = 'Married-civ-spouse' AND Hours per week >= 36.0
21	2	20867	0.094916	0.461446	9629	-	Hours per week >= 28.0 AND Marital status = 'Married-civ-spouse'
22	1	22379	0.094777	0.446133	9984	-	Marital status = 'Married-civ-spouse'
23	3	19402	0.093883	0.475621	9228	-	Age >= 23.0 AND Marital status = 'Married-civ-spouse' AND Hours per week >= 36.0
24	3	17281	0.093546	0.503675	8704	-	Marital status = 'Married-civ-spouse' AND Age >= 30.0 AND Hours per week >= 36.0
25	3	17808	0.093442	0.495564	8825	-	Marital status = 'Married-civ-spouse' AND Education num >= 9.0 AND Race = 'White'
26	3	18035	0.093435	0.49232	8879	-	Marital status = 'Married-civ-spouse' AND Education num >= 9.0 AND Native country = 'United-States'
27	3	15221	0.093421	0.539058	8205	-	Marital status = 'Married-civ-spouse' AND Education num >= 9.0 AND Age >= 34.0
28	3	15359	0.093421	0.536363	8238	-	Education num >= 8.0 AND Marital status = 'Married-civ-spouse' AND Age >= 34.0
29	3	18191	0.093387	0.490023	8914	-	Education num >= 8.0 AND Marital status = 'Married-civ-spouse' AND Native country = 'United-States'
30	3	17975	0.093319	0.492851	8859	-	Education num >= 8.0 AND Marital status = 'Married-civ-spouse' AND Race = 'White'
31	3	17760	0.093288	0.495833	8806	-	Age >= 28.0 AND Marital status = 'Married-civ-spouse' AND Hours per week >= 40.0
32	2	18521	0.093183	0.485017	8983	-	Age >= 32.0 AND Marital status = 'Married-civ-spouse'
33	2	19613	0.093034	0.470963	9237	-	Marital status = 'Married-civ-spouse' AND Hours per week >= 36.0
34	3	17618	0.092693	0.496254	8743	-	Marital status = 'Married-civ-spouse' AND Age <= 58.0 AND Education num >= 9.0
35	3	18235	0.092517	0.487085	8882	-	Age >= 28.0 AND Marital status = 'Married-civ-spouse' AND Age <= 59.0
36	3	16703	0.092467	0.509669	8513	-	Age >= 32.0 AND Hours per week >= 35.0 AND Marital status = 'Married-civ-spouse'
37	3	18991	0.092191	0.476384	9047	-	Age >= 23.0 AND Marital status = 'Married-civ-spouse' AND Hours per week >= 40.0
38	3	16918	0.091823	0.504374	8533	-	Marital status = 'Married-civ-spouse' AND Age >= 30.0 AND Hours per week >= 40.0
39	3	18483	0.09167	0.481524	8900	-	Age >= 28.0 AND Marital status = 'Married-civ-spouse' AND Native country = 'United-States'
40	3	17320	0.091574	0.497517	8617	-	Marital status = 'Married-civ-spouse' AND Education num >= 9.0 AND Fnlwgt >= 77698.0
41	3	17494	0.09154	0.494855	8657	-	Education num >= 8.0 AND Marital status = 'Married-civ-spouse' AND Fnlwgt >= 77651.0
42	3	16317	0.09141	0.512901	8369	-	Age >= 28.0 AND Relationship = 'Husband' AND Education num >= 9.0

Figure 6.7: Cortana results.

6.2.6 Results window buttons

Under the results window there are several buttons:

Browse: it permits to see the tuples covered by the subgroup

Delete: it deletes the subgroup from the result

Pattern Team: it reduces the number of subgroups using the procedure described in [1], basically maximising the Joint Entropy of the results

Roc: it shows the ROC Curve for the current result set

Save: it saves on file the results

Print: it connects to the printer for printing the results

Gaussian p-values: already described in the section above

Regression tests: it does regression tests for the qualities on the subgroup, another approach than the Gaussian one

Empirical p-value: another approach for the statistical validation

Close: it closes the window

6.2.7 ROC Curve

In Figure 6.8 is reported the ROC Curve window generated with Cortana. On the top of the window have been reported the subgroups on the ROC Curve with their ids, FPR (false positive rate) and TPR (true positive rate). The isometrics line are displayed with green, white and red colours from good, random and bad accuracy and same WRAcc, On top of the window are displayed the subgroups that form the ROC Curve. On the bottom of the window there are two buttons, **close** and **gnuplot**. Close will simply close the window and gnuplot will create a gnuplot script to make graphs like the one in Figure 3.3. The numbers on the top-left and bottom-right corners represent the maximum and minimum values that the quality measure can reach.

6.3 Interactive implementation design

Due to the lack of the access to development version of Cortana (basically we did not want to break possible current working) we designed a new complete package for the interactive implementation. We developed three java packages: *interactive*, *interactive.gui* and *interactive.interfaces*.

The interactive package contains the classes that manage the software logic. Basically we inherited the original classes and overridden their methods and developed new methods where required. Even if this strategy could lead to generate duplicated code with the original code we thought that with this approach it could be easier to divide the code from the development version so including it in a future version of Cortana (after the development of the interactive part) it could be easier and therefore not possible to generate inconsistencies between the standard beam search and the interactive search.

In the *interactive.gui* package we developed the graphical user interface. We use the swing java framework to write the classes.

Finally in the *interactive.interfaces* package we wrote different interfaces that could be used as bridge between the interactive part and the original one so some classes of the original packages can work with the interactive part. In the *interactive.test* package we developed several test classes for testing the interactive logic of Cortana, such as the functions for the coefficients calculation and the new beam search and also some classes to test the gui interactions. We adopted a simple naming convention for the classes names: it starts with the name of the inherited class followed by the name *Interactive* and for the interfaces we used *Abstract* followed by the name of the interface. Below we report simple UML classes schemes for each package and a brief description for each class.

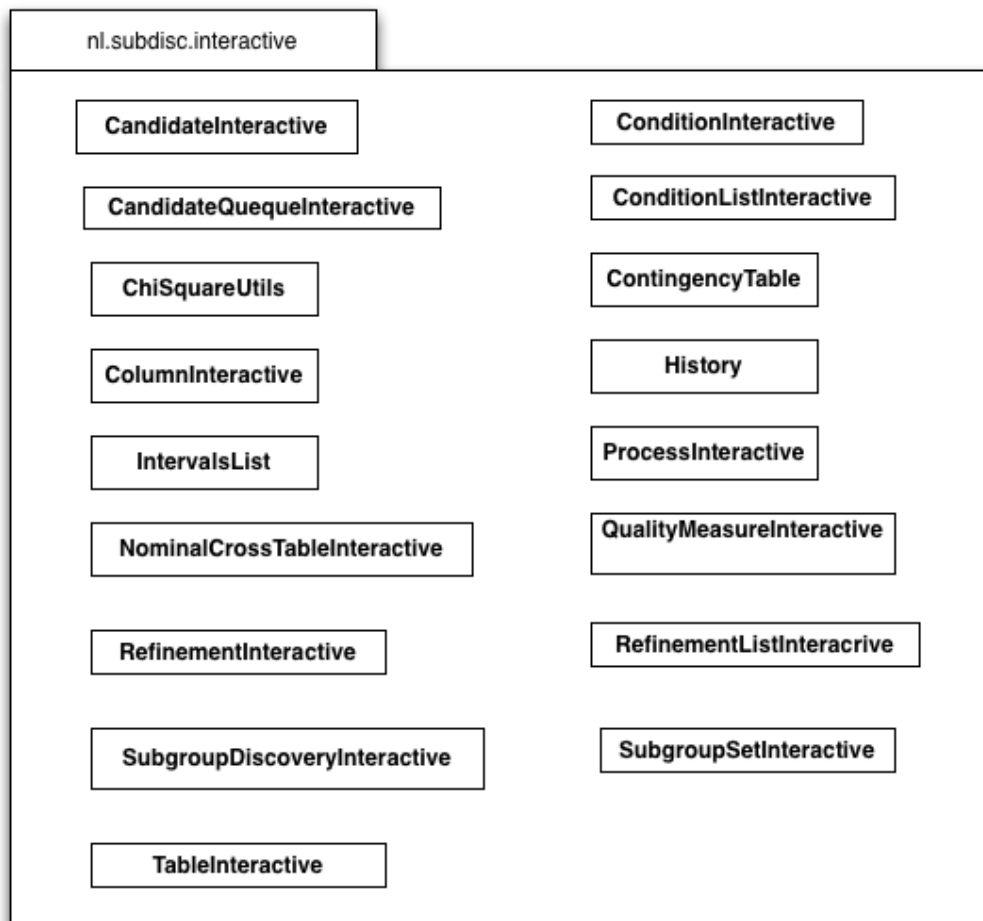


Figure 6.9: UML classes diagram.

CandidateInteractive: it represents a candidate subgroup in the beam

CandidateQueueInteractive: it represents a queue of candidates

ChiSquareUtils: utility class for computing the χ^2 test

ColumnInteractive: a column in the data table. Its domain is reduced when the user interacts with the subgroups intents

IntervalsList: a class to compute the intersections of numeric intervals if in Cortana the numeric operator is set with Interavals

NominalCrossTableInteractive: class for set-valued nominal attributes

RefinementInteractive: a class that represents the a single refinement for a subgroup

SubgroupDiscoveryInteractive: the subgroup discovery process plus the interactive evaluations of the subgroups

TableInteractive: the data table

ConditionInteractive: a single condition in the subgroup's intent

ConditionListInteractive: the list of conditions in the subgroup's intent

ContingencyTable: the contingency table where we compute the coefficients

History: the history of the search

ProcessInteractive: the class that starts the subgroup discovery process

QualityMeasureInteractive: it computes the quality measure given a contingency table

RefinementListInteractive: a list of refinements for a subgroup

SubgroupSetInteractive: a set of subgroups representing the results of the subgroup discovery process

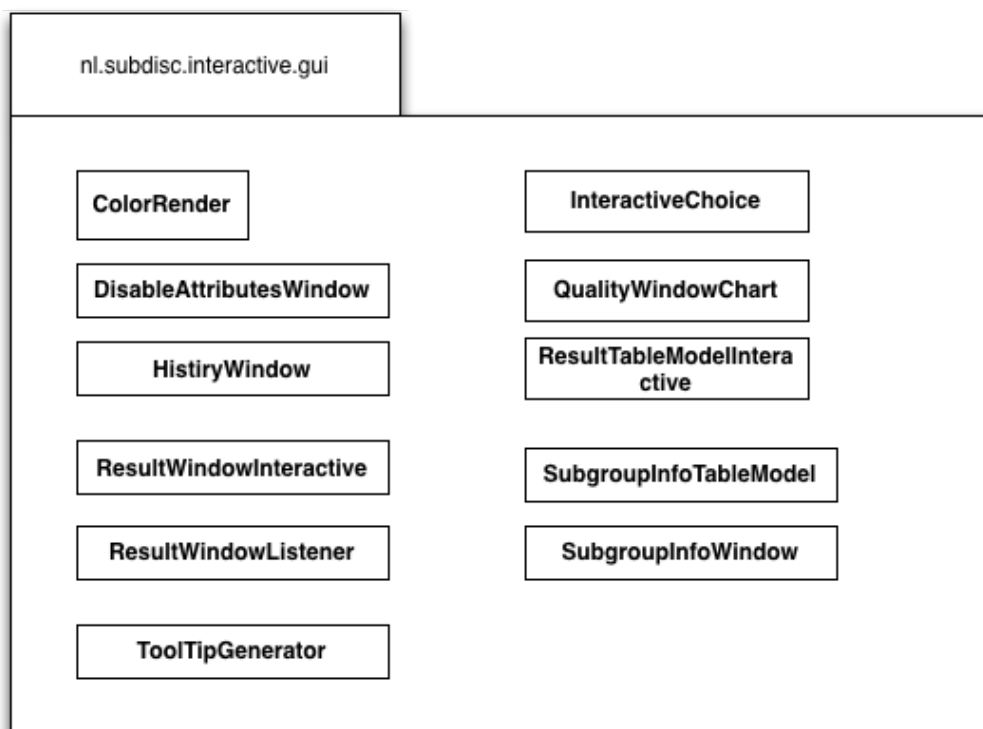


Figure 6.10: UML GUI classes diagram.

ColorRender: a render for the gui that colours the new subgroups that were not present in the result before

DisableAttributesWindow: the window for disabling the attributes

HistoryWindow: the history window that shows the user's interactions

ResultWindowInteractive: the results window

ResultWindowListener: a window listener to clean the table from the domain manipulating

ToolTipGenerator: a tooltip generator to help the user to see the id of the subgroup in the charts

InteractiveChoice: a java enum representing if the user interacted with the subgroup to like, dislike it or to get opposite subgroups

ResultTableModel: the table model for the results

SubgroupInfoWindow: the info window for the subgroup

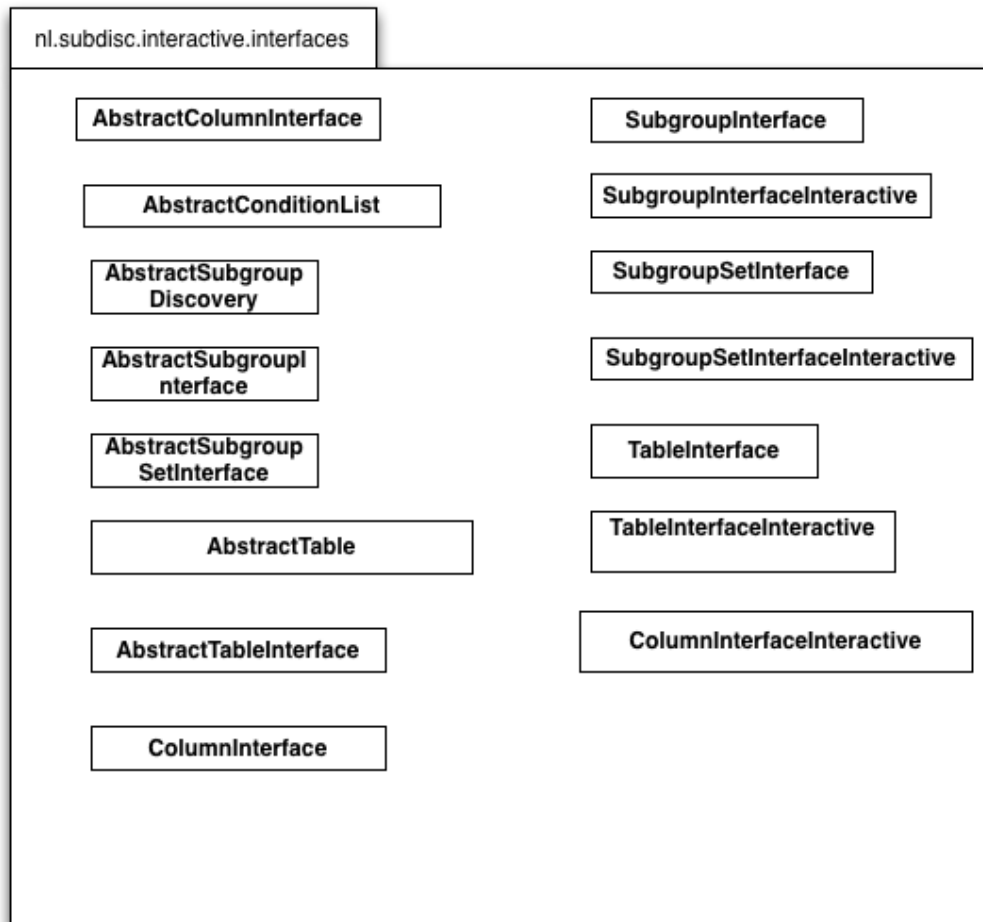


Figure 6.11: UML Interfaces diagram.

The name of the interface is explanatory, if there is an *abstract* suffix it means that the interface is on the top of the inheritance tree otherwise if there is only the *Interface* name that represents the Interface for the single class.

6.4 Interacting with the data

To interact with the beam search, as we said, the user could first run a subgroup discovery search with standard parameters, get some results from that and then interacting with the subgroups. To do that we advice to set the depth of the beam search at the most two because of the possible redundancy in the results, then proceed with an iterative approach exploring only the hypothesis space that is interesting.

To make this interaction we divided in two sections the results window, on the left there are the subgroups with their standard parameters plus the values from the coefficients evaluations (for the first run they are set at zero of course), on the right part instead there are several *checkboxes* representing the subgroup categories like, dislike, opposite and also the dislike of the intent of the subgroup and the investigation.

6.4.1 Dislike of the intent

When the user dislikes the intent of a subgroup a new window pops up with a checkbox for each condition that describes the intent of the subgroup. For each checkbox it presented to the user the opposite condition that will be used to restrict the domain of the corresponding attribute. For nominal values the condition will be *Condition!* = *Value* otherwise for numeric ones the condition will be the opposite of the numeric operator so if there is $\geq \rightarrow \leq$ and vice versa. After selecting which condition to dislike the corresponding attribute domain will be restricted correspondingly.

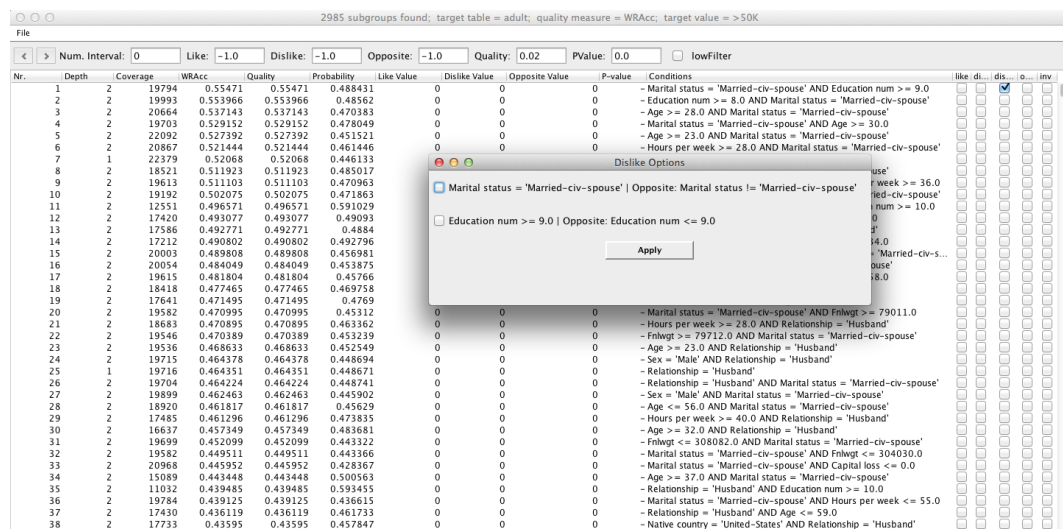


Figure 6.12: Dislike option window.

After restricting the domain, if the user will perform another search, the restriction of the domain will be displayed in the subgroup description making

easier for the user to see the ranges of the numeric attributes (Education num domain has range between $[-\infty, 9.0]$)

869	2	1484	0.053417	0.053417	0.559299	0 0 0 - Fnlwgt >= 79712.0 AND Workclass = 'Self-emp-inc'
870	2	8167	0.053404	0.053404	0.297416	0 0 0 - Sex = 'Male' AND Fnlwgt <= 117963.0
871	2	1502	0.053383	0.053383	0.55526	0 0 0 - Workclass = 'Self-emp-inc' AND Age <= 61.0
872	2	6554	0.053287	0.053287	0.311565	0 0 0 - Relationship = 'Husband' AND Education num >= 8.0 Education num(-inf, 9.0]

Figure 6.13: Bounds for numeric attributes on the subgroup's description.

6.4.2 Info Window

If the user double clicks a row on the result table an info window pops up with the information about the subgroup that were not be possible to display in the result window. Pressing on the focus button every value of every condition in the subgroup's description will be evaluated.

< > Num. Interval: 0															Like: -1.0		Dislike: -1.0		Opposite: -1.0		Quality: 0.02		PValue: 0.0		<input type="checkbox"/> lowFilter	
Nr.	Depth	Coverage	WRAcc	Quality	Probability	Like Value	Dislike Value	Opposite Value	P-value	Conditions	like	dis	o	inv												
1	2	19794	0.55471	0.55471	0.488431	0	0	0	0	- Marital status = 'Married-civ-spouse' AND Education num >= 9.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
2	2	19993	0.553966	0.553966	0.48562	0	0	0	0	- Education num >= 8.0 AND Marital status = 'Married-civ-spouse'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
3	2	20664	0.537143	0.537143	0.470383	0	0	0	0	- Age >= 28.0 AND Marital status = 'Married-civ-spouse'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
4	2	19703	0.529152	0.529152	0.478049	0	0	0	0	- Marital status = 'Married-civ-spouse' AND Age >= 30.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
5	2	22092	0.527392	0.527392	0.451521	0	0	0	0	- Age >= 23.0 AND Marital status = 'Married-civ-spouse'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
6	2	20867	0.521444	0.521444	0.461446																					
7	1	22379	0.52068	0.52068	0.446133																					
8	2	18521	0.511923	0.511923	0.485017																					
9	2	19613	0.511103	0.511103	0.470963																					
10	2	19192	0.502075	0.502075	0.471863																					
11	2	12551	0.496571	0.496571	0.591029																					
12	2	17420	0.493077	0.493077	0.490983																					
13	2	17586	0.492771	0.492771	0.4884																					
14	2	17212	0.490802	0.490802	0.492796																					
15	2	20003	0.489808	0.489808	0.456981																					
16	2	20054	0.484049	0.484049	0.453875																					
17	2	19615	0.481804	0.481804	0.45766																					
18	2	18418	0.477465	0.477465	0.469758																					
19	2	17641	0.471495	0.471495	0.4769																					
20	2	19582	0.470995	0.470995	0.45312																					
21	2	18683	0.470895	0.470895	0.463362																					
22	2	19546	0.470389	0.470389	0.453239																					
23	2	19536	0.468633	0.468633	0.452549	0	0	0	0	- Age >= 23.0 AND Relationship = 'husband'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
24	2	19715	0.464378	0.464378	0.448694	0	0	0	0	- Sex = 'Male' AND Relationship = 'husband'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
25	1	19716	0.464351	0.464351	0.448671	0	0	0	0	- Relationship = 'husband'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
26	2	19704	0.464224	0.464224	0.448741	0	0	0	0	- Relationship = 'husband' AND Marital status = 'Married-civ-spouse'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
27	2	19499	0.462463	0.462463	0.445902	0	0	0	0	- Sex = 'Male' AND Marital status = 'Married-civ-spouse'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
28	2	18920	0.461817	0.461817	0.45629	0	0	0	0	- Age <= 36.0 AND Marital status = 'Married-civ-spouse'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
29	2	17485	0.461296	0.461296	0.473835	0	0	0	0	- Hours per week >= 40.0 AND Relationship = 'husband'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
30	2	16637	0.457349	0.457349	0.483681	0	0	0	0	- Age >= 32.0 AND Relationship = 'husband'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
31	2	19699	0.452099	0.452099	0.443322	0	0	0	0	- Fnlwgt <= 308082.0 AND Marital status = 'Married-civ-spouse'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
32	2	19582	0.449511	0.449511	0.443366	0	0	0	0	- Marital status = 'Married-civ-spouse' AND Fnlwgt <= 304030.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
33	2	20968	0.445952	0.445952	0.428367	0	0	0	0	- Marital status = 'Married-civ-spouse' AND Capital loss <= 0.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
34	2	15089	0.443448	0.443448	0.500563	0	0	0	0	- Age >= 37.0 AND Marital status = 'Married-civ-spouse'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
35	2	11032	0.439485	0.439485	0.593455	0	0	0	0	- Relationship = 'husband' AND Education num >= 10.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
36	2	19784	0.439125	0.439125	0.436615	0	0	0	0	- Marital status = 'Married-civ-spouse' AND Hours per week <= 55.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
37	2	17430	0.436119	0.436119	0.461733	0	0	0	0	- Relationship = 'husband' AND Age <= 59.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												
38	2	17733	0.43595	0.43595	0.457847	0	0	0	0	- Native.GENR = 'United-States' AND Relationship = 'husband'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>												

Figure 6.14: Additional Information Window.

6.4.3 History and filtering

On top of the results window we added several text boxes to set quality filters. We will see in the next sections how to choose these values, we will use essentially a graph visualisation technique. Clicking on the lowFilter checkbox the thresholds will be considered as maximum thresholds otherwise it will be considered the minimum value that the subgroup has to have to remain in the results (except for p-value that is always considered as maximum threshold). The default value for the interaction qualities is -1, for the interaction quality measure is set to 0.02 that is the default value for Cortana.

Cortana remembers the user's interactions that the user made through the interactions so it is useful to report the subgroups with which the user interacted: this is the interaction's *history*. Because of that we added a sort of browser's chronological show order on the top-left corner with arrows that the user can click to show previous or forward results set.

Finally the first text box represents the minimum domain cardinality that a numeric attribute has to have to be explored exhaustively.

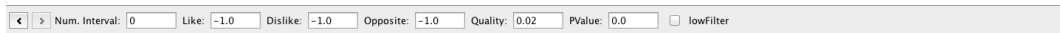


Figure 6.15: Filtering.

6.4.4 File menu

2985 subgroups found; target table = adult; quality measure = WRAcc; target value = >50K

File

History Interaction
Disable Attributes
Clear Values
Apply Filtering
Close

0 Like: -1.0 Dislike: -1.0 Opposite: -1.0 Quality: 0.02 PValue: 0.0 lowFilter

Nr.	Depth	Coverage	WRAcc	Quality	Probability	Like Value	Dislike Value	Opposite Value	P-value	Conditions	like	dis	o	inv
1	2	19794	0.55471	0.55471	0.488431	0	0	0	0	Marital status = 'Married-civ-spouse' AND Education num >= 9.0				
2	2	19993	0.553966	0.553966	0.48562	0	0	0	0	Education num >= 8.0 AND Marital status = 'Married-civ-spouse'				
3	2	20664	0.537143	0.537143	0.470383	0	0	0	0	Age >= 28.0 AND Marital status = 'Married-civ-spouse'				
4	2	19703	0.529152	0.529152	0.478049	0	0	0	0	Marital status = 'Married-civ-spouse' AND Age >= 30.0				
5	2	22092	0.527392	0.527392	0.451521	0	0	0	0	Age >= 23.0 AND Marital status = 'Married-civ-spouse'				
6	2	20867	0.521444	0.521444	0.461446	0	0	0	0	Hours per week >= 28.0 AND Marital status = 'Married-civ-spouse'				
7	1	22379	0.52068	0.52068	0.446133	0	0	0	0	Marital status = 'Married-civ-spouse'				
8	2	18521	0.511923	0.511923	0.485017	0	0	0	0	Age >= 22.0 AND Marital status = 'Married-civ-spouse'				
9	2	19613	0.511103	0.511103	0.470963	0	0	0	0	Marital status = 'Married-civ-spouse' AND Hours per week >= 36.0				
10	2	19192	0.502075	0.502075	0.471863	0	0	0	0	Hours per week >= 40.0 AND Marital status = 'Married-civ-spouse'				
11	2	12551	0.496571	0.496571	0.591029	0	0	0	0	Marital status = 'Married-civ-spouse' AND Education num >= 10.0				
12	2	17420	0.493077	0.493077	0.49093	0	0	0	0	Relationship = 'Husband' AND Education num >= 9.0				
13	2	17586	0.492771	0.492771	0.4884	0	0	0	0	Education num >= 8.0 AND Relationship = 'Husband'				
14	2	17212	0.490802	0.490802	0.492796	0	0	0	0	Marital status = 'Married-civ-spouse' AND Age >= 34.0				
15	2	20003	0.489808	0.489808	0.456981	0	0	0	0	Native country = 'United-States' AND Marital status = 'Married-civ-s...				
16	2	20054	0.484049	0.484049	0.453875	0	0	0	0	Race = 'White' AND Marital status = 'Married-civ-spouse'				
17	2	19615	0.481804	0.481804	0.45766	0	0	0	0	Marital status = 'Married-civ-spouse' AND Age <= 58.0				
18	2	18418	0.477465	0.477465	0.469758	0	0	0	0	Age >= 28.0 AND Relationship = 'Husband'				
19	2	17641	0.471495	0.471495	0.4769	0	0	0	0	Relationship = 'Husband' AND Age >= 30.0				
20	2	19582	0.470995	0.470995	0.45312	0	0	0	0	Marital status = 'Married-civ-spouse' AND Fnlwgt >= 79011.0				
21	2	18683	0.470895	0.470895	0.463362	0	0	0	0	Hours per week >= 28.0 AND Relationship = 'Husband'				
22	2	19546	0.470389	0.470389	0.453239	0	0	0	0	Fnlwgt >= 79712.0 AND Marital status = 'Married-civ-spouse'				
23	2	19536	0.468633	0.468633	0.452549	0	0	0	0	Age >= 23.0 AND Relationship = 'Husband'				
24	2	19715	0.464378	0.464378	0.448694	0	0	0	0	Sex = 'Male' AND Relationship = 'Husband'				
25	1	19716	0.464351	0.464351	0.448671	0	0	0	0	Relationship = 'Husband'				
26	2	19704	0.464224	0.464224	0.448741	0	0	0	0	Relationship = 'Husband' AND Marital status = 'Married-civ-spouse'				
27	2	19699	0.462461	0.462461	0.445902	0	0	0	0	Sex = 'Male' AND Marital status = 'Married-civ-spouse'				

Figure 6.16: File Menu.

On the top-left corner of the results window we added a file menu with different entries:

History Interaction: it reports to the user which interactions he made through the process

Disable attributes: it pops up a window with a checkbox for each table's attribute, the user can select which attribute disable

Clear values: it clear all the interactions

Apply Filtering: it applies the quality filtering made by the user

2985 subgroups found; target table = adult; quality measure = WRAcc; target value = >50K

File

< > Num. Interval: 0 Like: -1.0 Dislike: -1.0 Opposite: -1.0 Quality: 0.02 PValue: 0.0 lowFilter

Nr.	Depth	Coverage	WRAcc	Quality	Probability	Like Value	Dislike Value	Opposite Value	P-value	Conditions	like	dis	o	inv
1	2	19794	0.55471	0.55471	0.488431	0	0	0	0	Marital status = 'Married-civ-spouse' AND Education num >= 9.0				
2	2	19993	0.553966	0.553966	0.48562	0	0	0	0	Education num >= 8.0 AND Marital status = 'Married-civ-spouse'				
3	2	20664	0.537143	0.537143	0.470383	0	0	0	0	Age >= 28.0 AND Marital status = 'Married-civ-spouse'				
4	2	19703	0.529152	0.529152	0.478049	0	0	0	0	Marital status = 'Married-civ-spouse' AND Age >= 30.0				
5	2	22092	0.527392	0.527392	0.451521	0	0	0	0	Age >= 23.0 AND Marital status = 'Married-civ-spouse'				
6	2	20867	0.521444	0.521444	0.461446	0	0	0	0	Hours per week >= 28.0 AND Marital status = 'Married-civ-spouse'				
7	1	22379	0.52068	0.52068	0.446133	0	0	0	0	Marital status = 'Married-civ-spouse'				
8	2	18521	0.511923	0.511923	0.485017	0	0	0	0	Age >= 22.0 AND Marital status = 'Married-civ-spouse'				
9	2	19613	0.511103	0.511103	0.470963	0	0	0	0	Marital status = 'Married-civ-spouse' AND Hours per week >= 36.0				
10	2	19192	0.502075	0.502075	0.471863	0	0	0	0	Hours per week >= 40.0 AND Marital status = 'Married-civ-spouse'				
11	2	12551	0.496571	0.496571	0.591029	0	0	0	0	Marital status = 'Married-civ-spouse' AND Education num >= 10.0				
12	2	17420	0.493077	0.493077	0.49093	0	0	0	0	Relationship = 'Husband' AND Education num >= 9.0				
13	2	17586	0.492771	0.492771	0.4884	0	0	0	0	Education num >= 8.0 AND Relationship = 'Husband'				
14	2	17212	0.490802	0.490802	0.492796	0	0	0	0	Marital status = 'Married-civ-spouse' AND Age >= 34.0				
15	2	20003	0.489808	0.489808	0.456981	0	0	0	0	Native country = 'United-States' AND Marital status = 'Married-civ-s...				
16	2	20054	0.484049	0.484049	0.453875	0	0	0	0	Race = 'White' AND Marital status = 'Married-civ-spouse'				
17	2	19615	0.481804	0.481804	0.45766	0	0	0	0	Marital status = 'Married-civ-spouse' AND Age <= 58.0				
18	2	18418	0.477465	0.477465	0.469758	0	0	0	0	Age >= 28.0 AND Relationship = 'Husband'				
19	2	17641	0.471495	0.471495	0.4769	0	0	0	0	Relationship = 'Husband' AND Age >= 30.0				
20	2	19582	0.470995	0.470995	0.45312	0	0	0	0	Marital status = 'Married-civ-spouse' AND Fnlwgt >= 79011.0				
21	2	18683	0.470895	0.470895	0.463362	0	0	0	0	Hours per week >= 28.0 AND Relationship = 'Husband'				
22	2	19546	0.470389	0.470389	0.453239	0	0	0	0	Fnlwgt >= 79712.0 AND Marital status = 'Married-civ-spouse'				
23	2	19536	0.468633	0.468633	0.452549	0	0	0	0	Age >= 23.0 AND Relationship = 'Husband'				
24	2	19715	0.464378	0.464378	0.448694	0	0	0	0	Sex = 'Male' AND Relationship = 'Husband'				
25	1	19716	0.464351	0.464351	0.448671	0	0	0	0	Relationship = 'Husband'				
26	2	19704	0.464224	0.464224	0.448741	0	0	0	0	Relationship = 'Husband' AND Marital status = 'Married-civ-spouse'				
27	2	19699	0.462461	0.462461	0.445902	0	0	0	0	Sex = 'Male' AND Marital status = 'Married-civ-spouse'				

Attributes Window

Age Workclass Fnlwgt Education Education num per week >= 36.0 Married-civ-spouse Education num >= 10.0 Education num >= 9.0 Relationship Race Capital gain Relationship = 'Husband' Age >= 34.0 Relationship = 'Husband' AND Age >= 30.0 Marital status = 'Married-civ-spouse' AND Fnlwgt >= 79011.0 Hours per week >= 28.0 AND Relationship = 'Husband' Fnlwgt >= 79712.0 AND Marital status = 'Married-civ-spouse' Age >= 23.0 AND Relationship = 'Husband' Sex = 'Male' AND Relationship = 'Husband' Relationship = 'Husband' Relationship = 'Husband' AND Marital status = 'Married-civ-spouse' Sex = 'Male' AND Marital status = 'Married-civ-spouse'

Apply

Figure 6.17: Disable attributes window.

6.4.5 Results Buttons

On the bottom of the results window we added four different buttons:

Run Search: it runs a beam search with same depth with input the subgroups that the user chose

Quality Inspection: it opens up the graphs for the qualities

Search Deeper: it adds one to the depth of the search and expands all the subgroups in the result with the next condition

Compute P-values: it performs the computation of the p-value (with the χ^2 test is the coverage of the population is large enough, otherwise with the Fisher's exact test) between the subgroups in the results and the input subgroups. If the input subgroups are more than 1 it computes the Bonferroni procedure.



Figure 6.18: Interactive buttons.

6.4.6 Results Visualisation

For the results visualisation we added the possibility for the subgroups that in the previous search weren't present to be coloured with cyan colour. This can help the user to see which new subgroups entered in the results and to get a simple view on how much his interaction influenced them. The results are ordered by the original quality measure so the user can see easily which subgroup take higher value than the others so if it is relevant in terms of the standard subgroup discovery set.



Figure 6.19: Interaction with data and new subgroups showing

6.4.7 Charts

To help the user to see how the coefficients spread over the subgroups results we developed two types of charts. A *line chart* and a *box and whisker chart* [33]. We used the jfreechart [24] library to implement them. In the x-axis of the line graph there are the ids of the result subgroups and on the y-axis the values of their coefficients values. A line is drawn for each coefficients. Anyway other two lines are plotted: the interactive quality measure and the original quality measure. This is helpful to see how the coefficients take range and to see which region is more influenced by the interactions.

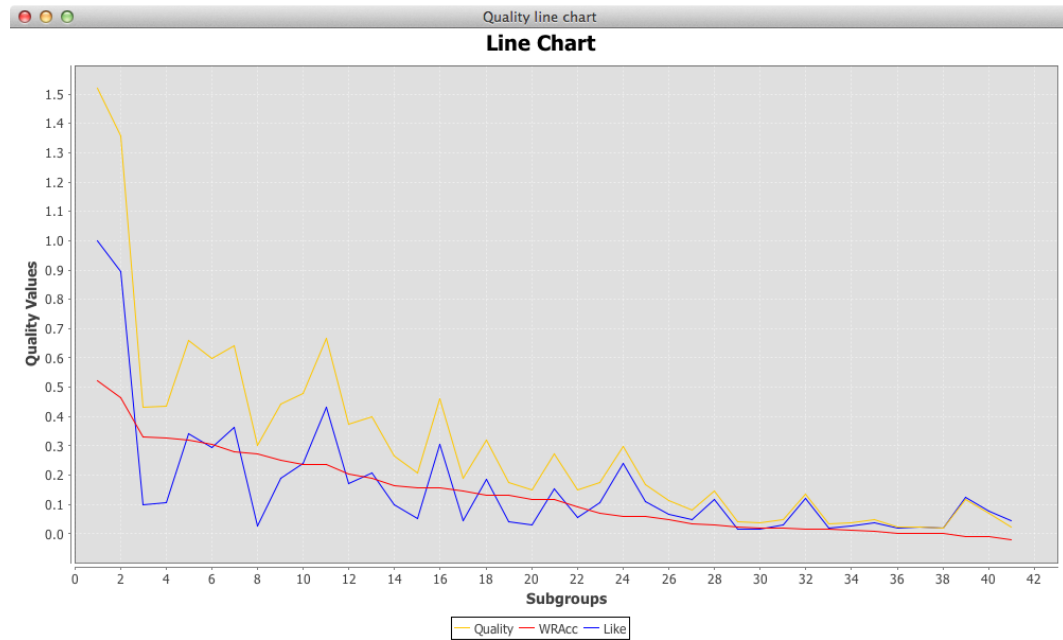


Figure 6.20: Line chart for qualities.

To help the user to select the thresholds for the filtering process we developed a box and whisker graph. In this graph the quality measures and the coefficients are plotted as box plots. A box plot is a particular plot that can be used in the descriptive statistics to draw in an intuitively way a numeric distribution. The method to draw a box plot is simple: we get a set of numbers, we order them in ascending order. After that we take the median value that is the number in the centre of the values list (if we have an even number of numbers

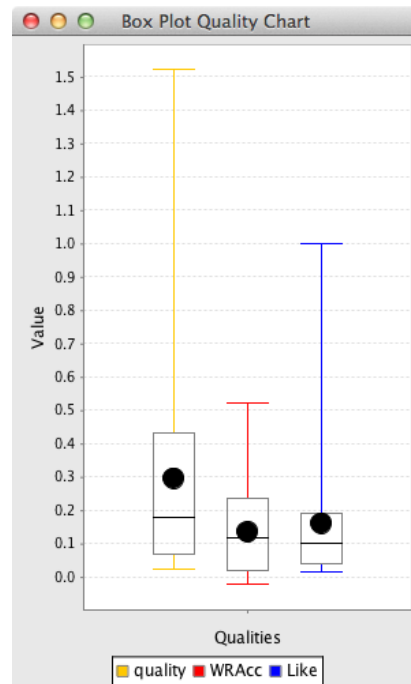


Figure 6.21: Box and whisker plot for qualities values.

we take the mean of the two centres). The median separate the list into two other sets, for each of them we take the median. These last two numbers are called *first quartile* and *third quartile* respectively.

After that we draw a box around the first quartile and the third quartile with a segment inside at the median level, then we draw the whisker from the minimum to the first quartile and from the third quartile and the maximum. After that we represent the mean value of the set with a black ball. There is a thing however that we have to consider. In the original version of the plot the values that are 1.5 times lower or greater than the first or the third quartile are considered outliers and plotted as circles on the plot and the values greater or lower than 2 times considered out of bounds and not plotted. Here we do not consider them and instead we take the whole distribution to plot because we need to see the whole distribution and we are not interested to find outliers but just to see how the values take range. The last thing to mention is that inside the box there are exactly the 50% of the total values, this will be useful

to choose the optimal threshold for filtering the qualities values and to get a general idea on the qualities and coefficients range.

6.4.8 About Cortana's coding practice

There is still something to mention about the original Cortana's code, that is the total lack of *design patterns* [8]. The project just for the logical part that manages the calculation of the different subgroup discovery strategies consists of **80 classes** and **30 classes** for the graphical interface. The first critic that someone could arise is that dividing this big number of classes in just two packages is too reductive, there should be more packages each one dividing the classes for their scope. The second thing is that without a clear design pattern for the whole class hierarchy the flow of the entire calculation is confusing, it proceeds through different classes making difficult to understand how is done and finally it produces *content coupling* and *low cohesion*. For this reason the inheritance process for the interactive part of Cortana has been challenging, without a clear structure inserting a new module for new features as the interaction process can be frustrating (in Cortana there is no use of java interfaces for example). Finally due to the computation intensive nature of subgroup discovery other languages with better performances like C++ or python should have been used rather than Java that has high overhead due to the interpretation of the bytecode by the Java Virtual Machine.

Chapter 7

Results

In this chapter we will report the results of the effectiveness of our approach. We made several use cases for two different datasets. Of course due to the really subjective aspect of the search we will try to impersonate at the best a domain expert that wants to explore the data and get new knowledge from that.

7.1 General knowledge, local knowledge and the analysis approach

We will divide the new generated knowledge into two main categories: the **general knowledge** and the **local knowledge**. The first knowledge will come from the interaction between the coefficients and the whole set of the beam search results, essentially we will take a look at the charts and the ROC Curve to get a general idea on how the coefficients interacted and how the binary target attribute is distributed along them. From this knowledge we will get the local knowledge that will come instead from the analysis on the singular subgroups and their coefficients values.

The search parameters are set to the Cortana's defaults:

- strategy type: beam search
- quality minimum: 0.02
- minimum coverage: 2
- beam width: 100
- numeric strategy: bins (8)
- numeric operators: \leq, \geq

The analysis will proceed as follows:

- we set the depth of the search to 1
- we interact with the results choosing some interesting subgroups. Eventually we disable attributes on which we are not interested on
- we get a validation of the results from the p-value calculation
- we filter the results basing on the distribution of the coefficients and their p-values
- we expand singular subgroups that could be still interesting setting the numeric threshold to 10000 to be sure to evaluate all the numeric attributes domain
- we get a new validation
- the results represent the local knowledge obtained
- we expand all the subgroups to populate the ROC space getting the general knowledge of the distribution of the target attribute and with the help of the charts we will try to find other local knowledge

7.2 UCI datasets

The first dataset is taken from the UCI repository [30]. The UCI repository is a datasets collection in ARFF format that can be used to test machine learning algorithms. There are a lot of them, every one can represent different aspects that can be used to test the algorithms, some datasets for example are only numeric or only nominal and some others have attributes both numeric and nominal.

7.3 Adults Dataset



The adults dataset represents a subset of population extracted from the 1994 american census database, it is a really good representation for a knowledge discovery in database process as subgroup discovery. The tuples in the set are described by several numeric and nominal attributes, for a general view on their ranges and

domains see Table 1 in the Appendix. Usually this dataset is used to build a predictive model for the target attribute $Class \geq 50K$ to predict when an individual earns more than 50 thousand dollars a year, we will use it for the subgroup discovery target as well.

Below we report another table that summarises some other aspects about the dataset (all the informations are automatically generated by Cortana in the main window):

nr of examples	48842
columns	15
nominals	9
numerics	6
binaries	0
positives	11687 (23.9%)

Table 7.1: Adults dataset

We will present three uses cases. The first one will be the easiest one because it will not contemplate too much interaction with the results, it is fast and easy. The second one will be a little bit difficult and the third will be the most difficult. On the other hand the difficult is not evaluated only on the number of interaction but also with the difficult to find in the dataset significant results. This is due essentially for the fact that if the user interact with small subgroups to find significant results we have to expand the subgroups to very high depth because the coverage of the compared subgroups has to be similar: if a subgroup is big and another is small even if the first contains the other there will be a lot of zeros in the compared subgroup binary variable making the subgroups not correlated. So, to prevent this, we search for other small subgroups going deeper with the research.

For all the use cases we disable the attribute **fnlwgt**. This attribute represents a score that the Population Division the Census Bureau give to an example in the dataset and it represents the number of individuals in the target population that the corresponding individual represents [7] therefore we consider this attribute irrelevant.

In the first use case we want to compare the civilian married population (subgroups with *Marital status*='civ-spouse') with the population not married. This comparison comes from the analysis of the roc space with the subgroups with *depth* = 1:

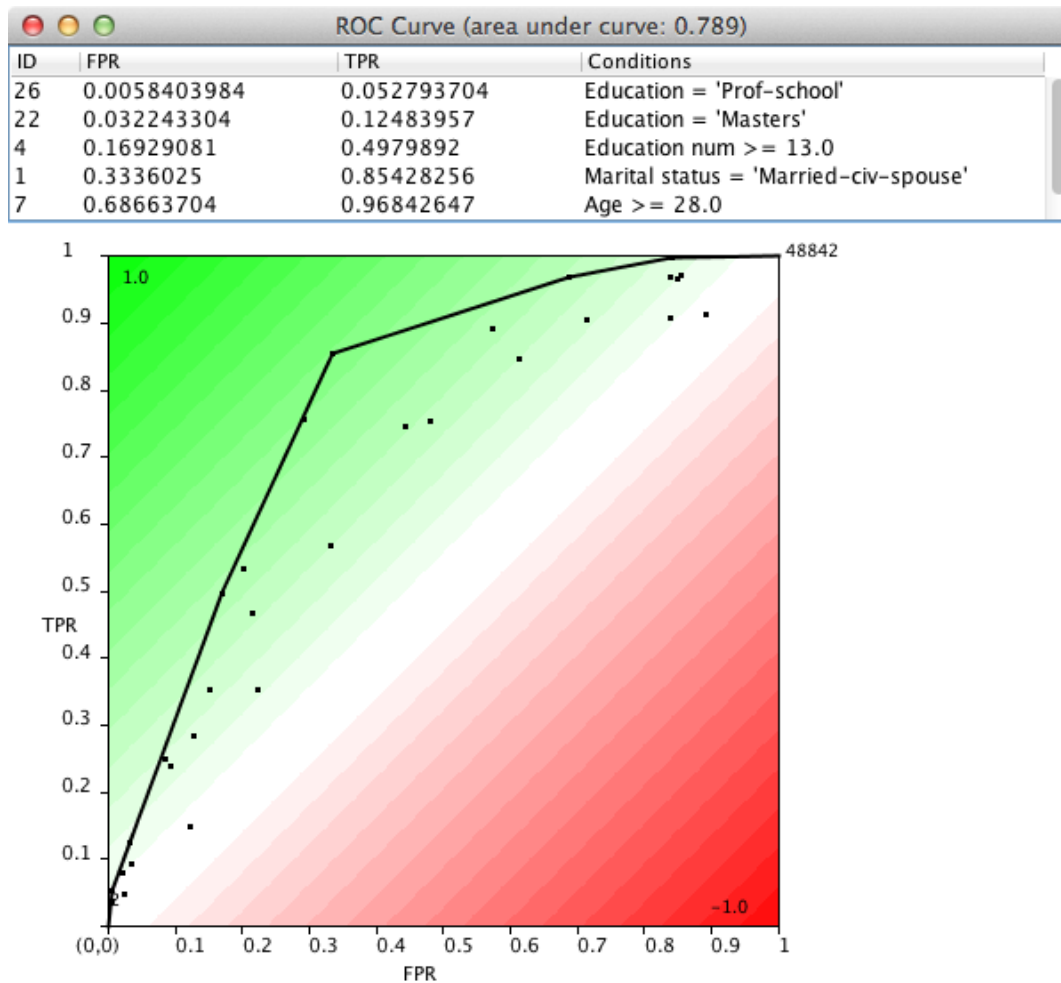


Figure 7.1: ROC Curve for the first run.

As we can see the best subgroup is represented by the conditions *Marital status* = 'Married-civ-spouse' that means that the beam search, because of the way that explores the hypothesis space, it will find that the best subgroups most probably will start with the first attribute set to that one. After this analysis we decide to **like** that subgroup and see how the phi coefficient spreads along the best subgroup:

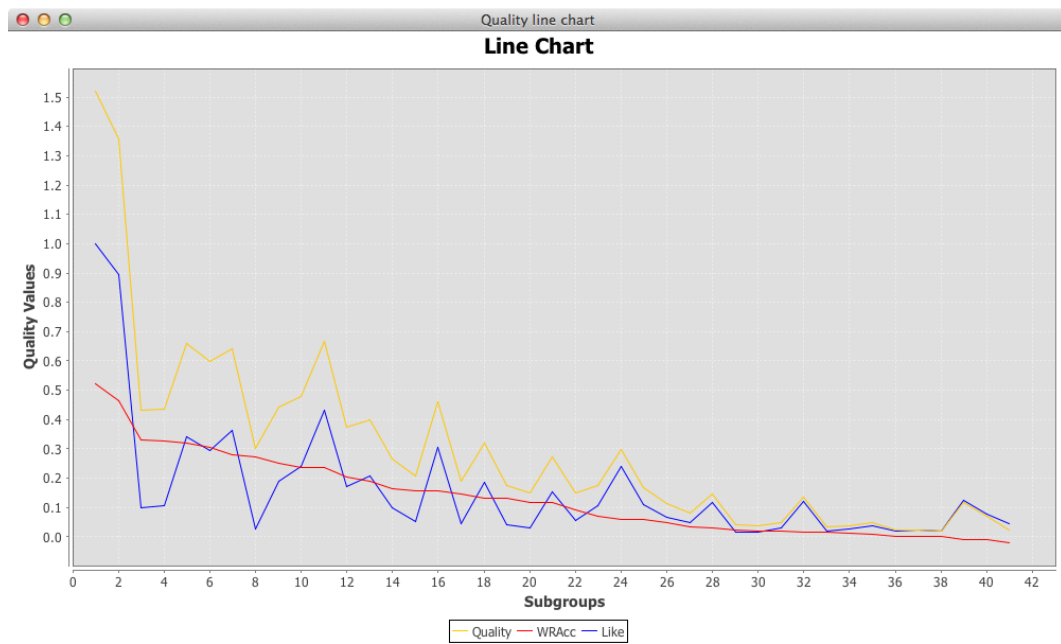


Figure 7.2: Line chart.

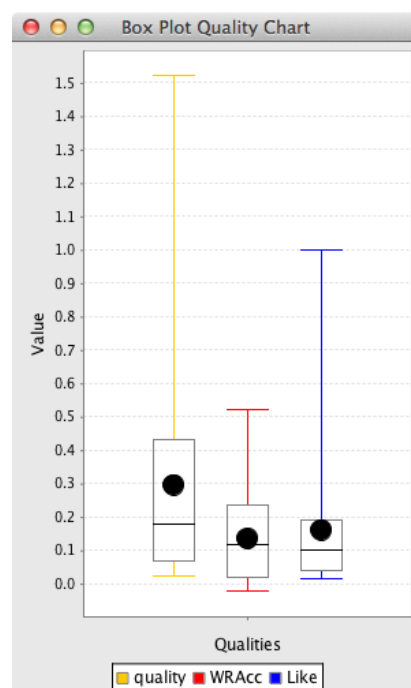


Figure 7.3: Box and whisker chart.

From the analysis of the first two charts we can get a general idea on how the coefficients are distributed. With the box and whisker chart we can see

that the phi coefficients for more than the 50% of the results is below 0.2 that means that the results do not show a general correlation with the subgroup. The line chart shows the same concept but with that we can see that there are 3 subgroups that reach good values: the first one is *Marital-status* with phi coefficient = 1, obviously the subgroup is equal to itself. The other two similar subgroups are:

Results position	Description	Coverage	WRAcc	Probability	Phi Coefficient	p-value
1	Marital status = 'Married civ-spouse'	22379	0.52068	0.446133	1	< 0.0001
2	Relationship = 'Husband'	19716	0.464351	0.448671	0.893676	< 0.0001
11	Sex = 'Male'	32650	0.23682	0.3037	0.431126	< 0.0001

Table 7.2: Interesting subgroups like (89 Results).

These subgroups show that the civilians people that are married essentially are made by husbands (and male of course), probably the subgroup with the wives is smaller so that's why there is no correlation.

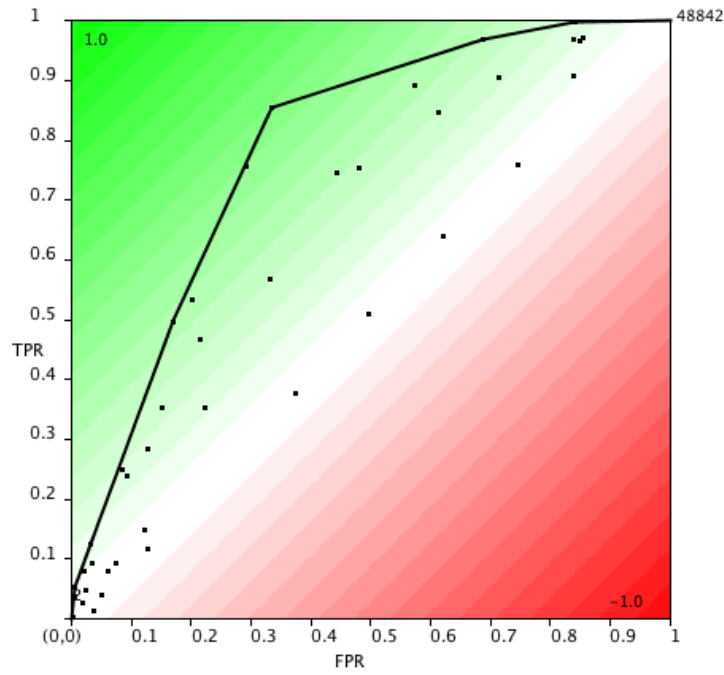


Figure 7.4: ROC Curve with liked subgroups.

The associated ROC Curve with the liked interaction demonstrates that we introduced subgroups with poor accuracy, most of them are subgroups that entered in the beam even if they had bad quality measure and bad coefficient value.

Looking at the position in the beam search and their scores these subgroups are good for getting a roc Curve with good *AUC*. We **liked** them and then we expanded all the subgroups two times setting the quality like threshold to 0.5 to obtain moderate correlated subgroups. The corresponding ROC Curve is plotted:

Even if the phi coefficient threshold is set to 0.5 the corresponding whisker and box chart shows that the subgroups in the results have good correlations.

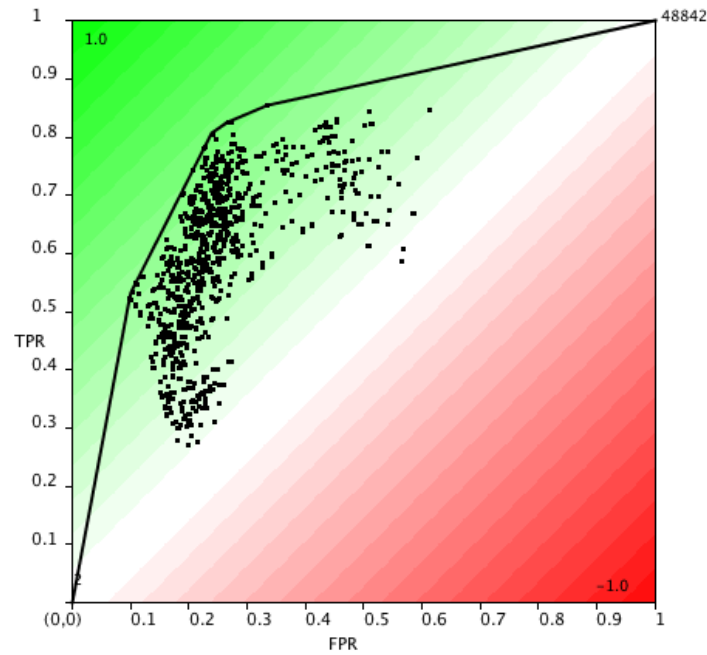


Figure 7.5: ROC Curve like Marital status = 'Married-civ-spouse', Relationship = husband, Sex = 'Male'.

For the local knowledge acquiring process we also **focused** the following subgroup that was the first in terms of quality measure and phi coefficient:

Results Position	Description	Coverage	WRAcc	Probability
1	Marital status = 'Married-civ-spouse' \wedge Education-num $\geq 9 \wedge$ Age ≥ 30	17467	0.5562	0.52241

Table 7.3: Best subgroup (89 results)

We obtained another one singular subgroup made by just one example:

Results Position	Description	Coverage	WRAcc	Probability
90	Marital status = 'Widowed' \wedge Education-num $\geq 16 \wedge$ Age ≥ 80	1	-0.000027	0

Table 7.4: Focused subgroup (90 Results).

This subgroup shows an old individual that does not earn a lot of money even if his education is high, contrary the other rest of the dataset that shows high

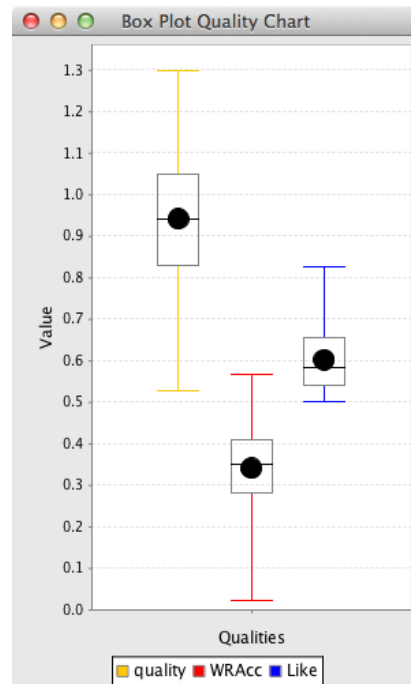


Figure 7.6: Box plot like Marital status = 'Married-civ-spouse', Relationship = husband, Sex = 'Male'.

probability of earning money with good education. This method shows the possibility to explore exhaustively the hypothesis space underlying singular characteristics. Moreover due to the *brute-forcing* nature of the focus we can be sure that no other interesting subgroups with those attributes exist.

To compare the results with the civilian population not married we have just to dislike the corresponding subgroups that we liked before. The ROC Curve that we obtain is:

This ROC Curve clearly shows that the population male and married earns more money than the population that is not married.

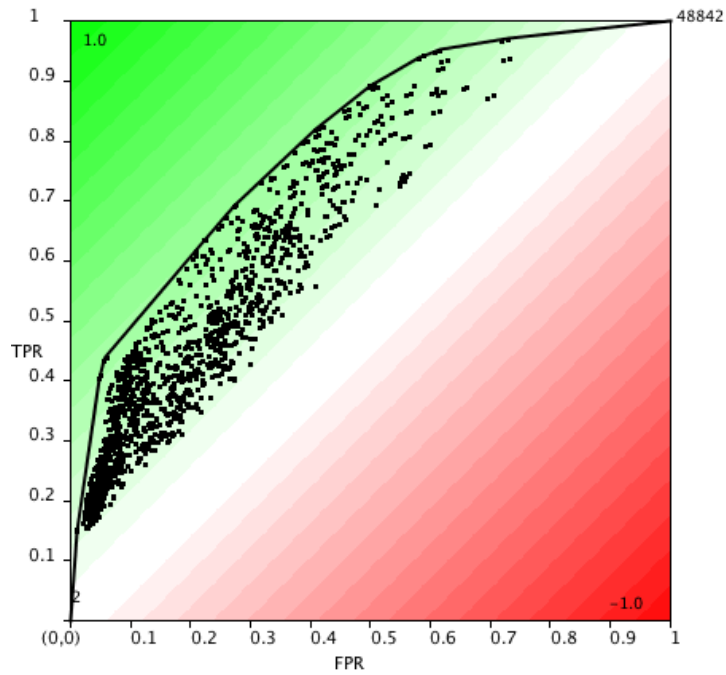


Figure 7.7: ROC Curve dislike Marital status = 'civ-spouse', Relation-husband, Sex = 'Male'.

Now, the user is interested to see how the **young population** is described in terms of money income and descriptions. In the first search results there are different subgroups described by the attribute *Age*.

Results Position	Description	Coverage	WRAcc	Probability
5	Age \geq 32	31724	0.3180	0.3284
6	Age \geq 37	25148	0.3061	0.3475
7	Age \geq 28	36830	0.2817	0.3073
10	Age \geq 42	18976	0.2382	0.3509
16	Age \geq 23	42945	0.15567	0.2715
18	Age \geq 48	12366	0.1328	0.3347
28	Age \geq 56	6248	0.029	0.2805

Table 7.5: Population described by Age (34 Results)

What we do now is disliking the intent of the third subgroup to search on subgroups with $Age \leq 28$. Unfortunately the new results are the same as before except without the subgroups described by the Age attribute. To show them we have to lower the quality measure to -1 to be sure that all the subgroups will appear. In fact we obtain this particular subgroup:

Results Position	Description	Coverage	WRAcc	Probability
121	Age \leq 18 Age(-inf, 28)	1457	-0.039	0

Table 7.6: Young subgroup (149 Results).

First of all looking at the subgroup's probability, clearly the target attribute is false for all the tuples in the subgroup. Therefore for sure a ROC Curve analysis here is unfeasible, probably all the subgroups that are similar to this one have most of the target attribute at false value.

The line chart below shows that the subgroups correlated with this other subgroup have bad quality measure thus are no interesting for the target.

Among the results however there are several interesting subgroups:

Results position	Description	Coverage	WRAcc	Probability	Phi Coefficient	p-value
4	Education = 'Some-college'	10878	-0.0607	0.081	0.142	< 0.0001
5	Workclass = 'Private'	33906	-0.8167	0.2178	0.1179	< 0.0001
14	Relationship = 'Own-child'	7581	-0.1915	0.0146	0.442	< 0.0001
68	Marital-status = 'Never-married'	16117	-0.3511	0.045	0.5442	<0.0001

Table 7.7: Interesting young subgroups (68 Results).

So we decide to investigate them to obtain better descriptions. What we get anyway is that only *Relationship = 'Own-child'* remains in the beam but the subgroups in the results contain the other attributes as descriptions except for *Marital-status = 'never-married'*, in fact we obtain:

Results position	Description	Coverage	WRAcc	Probability	Phi Coefficient	p-value
128	Relationship = 'Own-child' \wedge Education = 'some-college'	2573	0.2591	0.0112	0.32	< 0.0001
154	Relationship = 'Own-child' \wedge Workclass = 'Private'	5683	-0.1442	0.0135	0.3817	< 0.0001

Table 7.8: Interesting own child subgroup (184 Results).

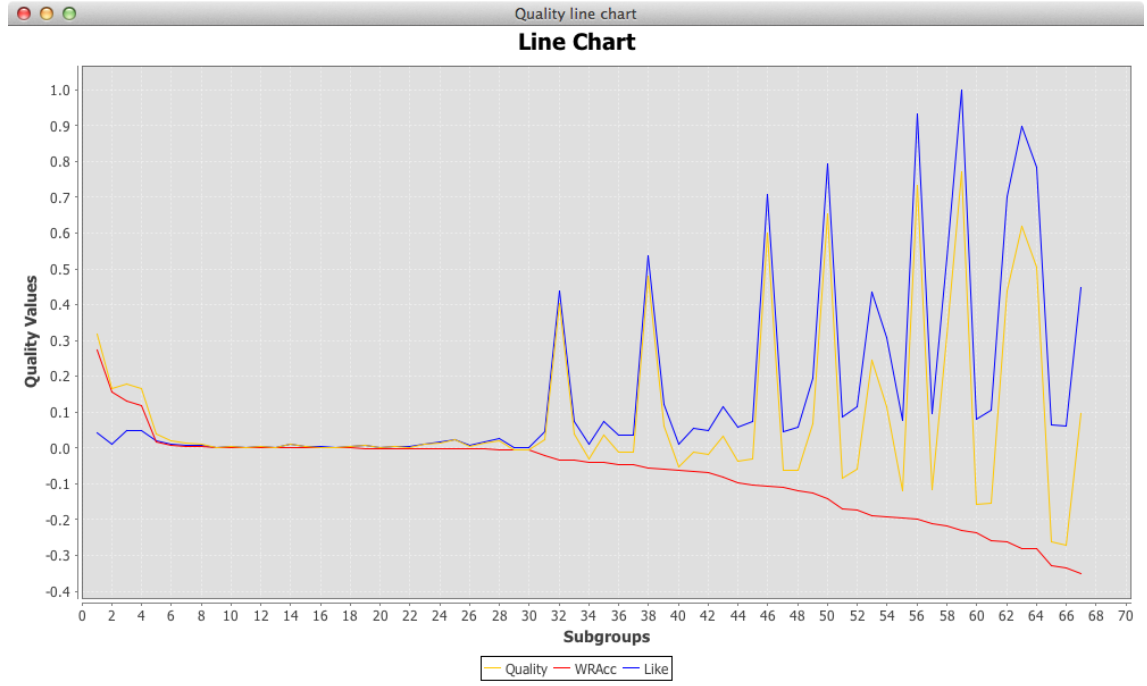


Figure 7.8: Line chart like young people.

These two subgroups are really similar for both quality measure, phi coefficient and probability so they can be interpreted as good descriptors for the young population.

In the third use case we want to impersonate a domain expert interested to analyse the dataset respect to the populations that are not native of the United States (*Native-country* \neq 'United-States'), in particular we are interested also in the population coming from Italy. For this use case we will also take into consideration how the patterns extracted are described. We want in fact exclude which ones that are not interesting because they describe the subgroups in terms of attributes that can be considered trivial by the user. To do that we will apply different intent dislikes.

After the first run with *depth* = 1 we get the following subgroup representing the United States people, the corresponding ROC Curve is the same of the first use case:

Results Position	Description	Coverage	WRAcc	Probability
30	Native country = 'United States'	43832	0.023148	0.243977

Table 7.9: United States population.

After this, we click the correspondent **opposite** checkbox and we re-run the search maintaining the same search parameters and we get a different results set.

Below the corresponding charts:

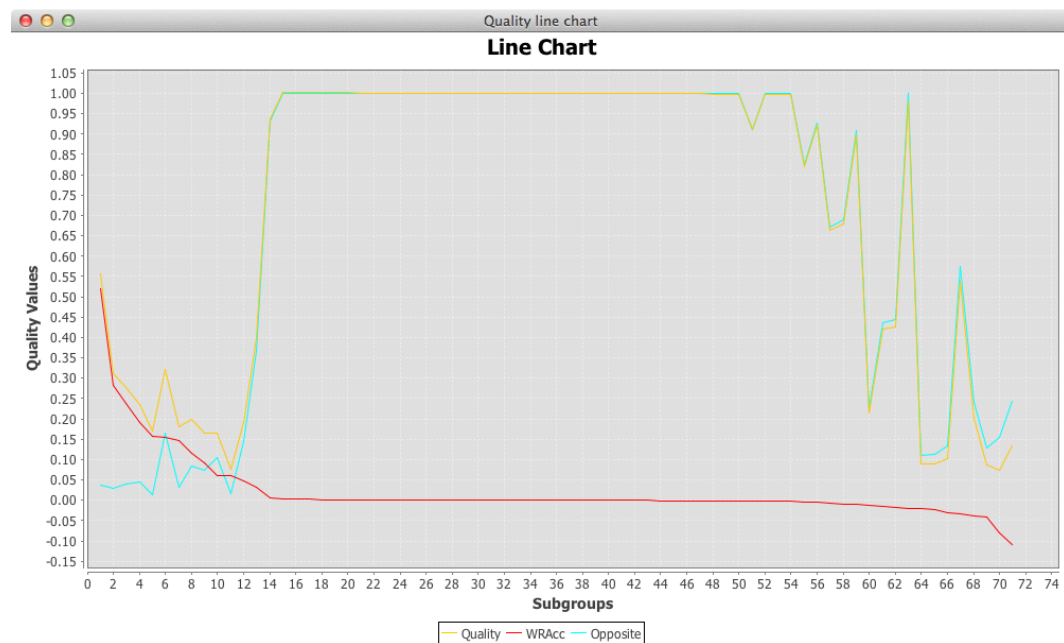


Figure 7.9: Line Chart opposite subgroups.

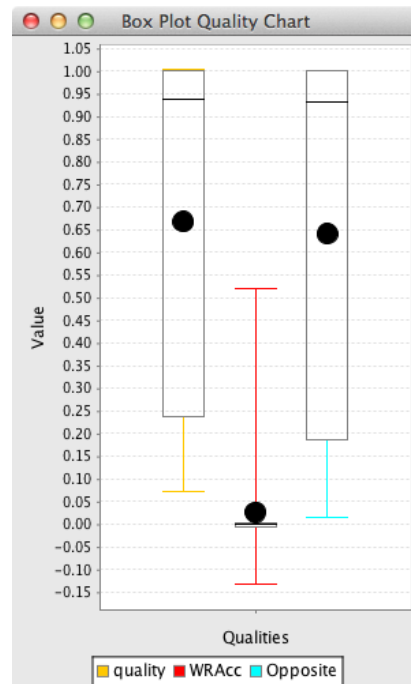


Figure 7.10: box and whisker chart opposite subgroups.

The box and whisker chart shows that the results subgroup have good opposite scores and bad quality measure, the line chart in fact show that most of the opposite subgroups have quality measure equal to zero.

We decide to expand all the subgroups to get a full view of the opposite subgroups and we obtain the following Roc Curve and box and whisker chart: The Roc Curve shows that the population that is not from the United States have really bad target attribute.

Looking at the first line chart we produced (Figure 7.9) we see a lot of subgroups with Yule's Q coefficient at the maximum value. These subgroups are described by attributes with *Native-country* \neq *United-States*. Someone could argue that these attributes are trivial and he wants to get more different patterns, so to do that he can disable the attribute *Native-country* and proceed with the search without this attribute.

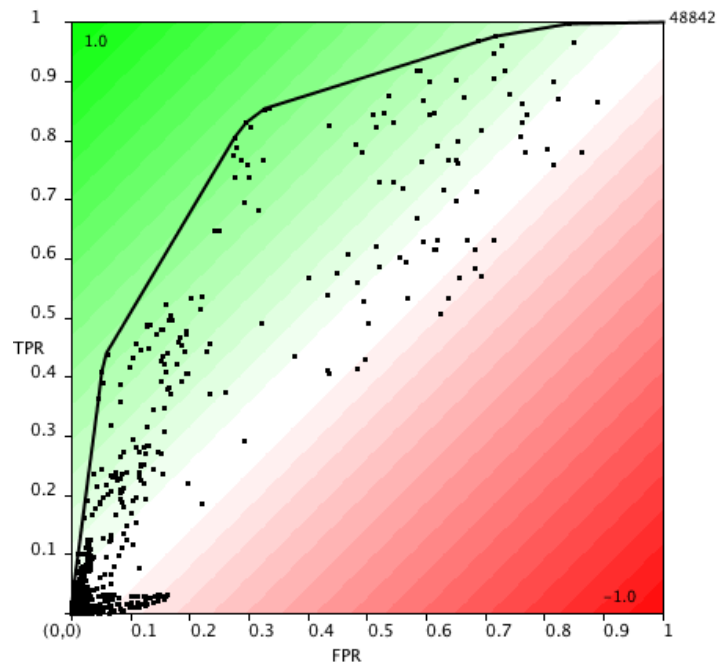


Figure 7.11: Roc Curve total opposite subgroups.

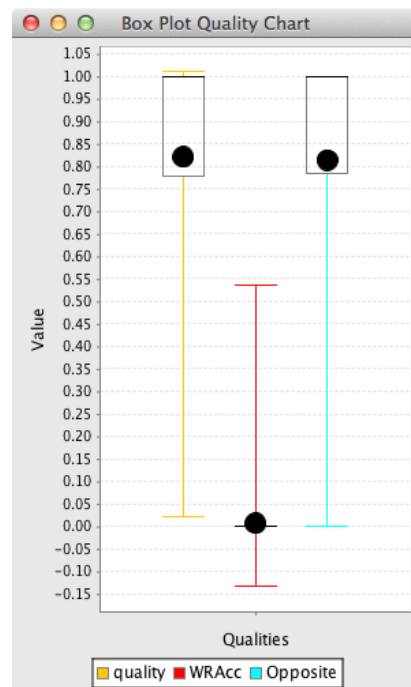


Figure 7.12: Box chart total opposite subgroups.

Some other opposite subgroups come in the result:

Results position	Description	Coverage	WRAcc	Probability	Yule's Q	p-value
16	Race = 'Asian-Pac-Islander'	1519	0.005	0.2698	0.9313	< 0.0001
17	Education = 'Preschool'	83	0.9099	0.0123	0.9113	< 0.0001
18	Race = 'Other'	406	-0.0053	0.1215	0.8253	< 0.0001
19	Education = '1st-4th'	247	-0.0057	0.032	0.9272	< 0.0001
20	Occupation = 'Priv-house-serv'	242	-0.0061	0.12	0.6705	< 0.0001

Table 7.10: Opposite subgroups (34 Results).

The first and third subgroup are trivial because it is obvious that people that do not are natives of the United States could have different races. The other three instead are interesting because show that these populations have bad education and most of the people that work as servants are not american people but foreigners. So we dislike the two subgroups with the Race attribute. Unfortunately no other new subgroups come in the results. As we said before anyway we were interested in people from Italy, so we use the Cortana's history to go back with the results and we like *Native-country = 'Italy'* but we maintain the disabled attribute native country because if we would maintain it all the subgroup with *Native-country = 'Italy'* in the description obviously would get a high score but we are not interested in them because we want to see if Italian people are correlated with other aspects of the dataset. However as we can see from the following line chart any subgroups is related with italian people probably because the subgroup is really small (Coverage = 105) but we can conclude anyway that any subgroups contain a large piece of italian population.

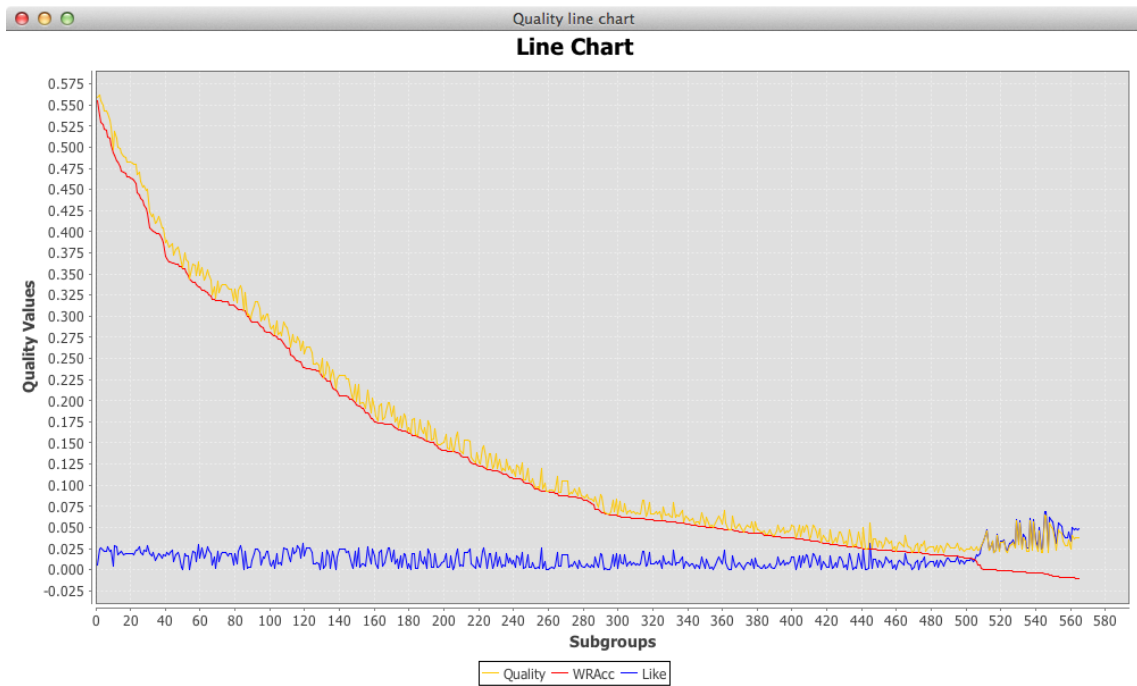


Figure 7.13: Italians correlated coefficients.

7.4 Mammals Dataset



The second dataset that we will analyse is the mammals dataset. This dataset divides the geography of Europe into clusters based on their fauna, which is a core activity of biology. The dataset was created by combining two datasets: one documenting presence or absence of 101 mammals for a set of 2221 grid cells covering Europe [25], and one documenting climate and elevation of the corresponding land areas [16]. The cell resolution of the grid is approximately 50×50 km, and the grid system is based on the Universal Transverse Mercator (UTM) projection and the Military Grid Reference System (MGRS). We use a version of this dataset that has been pre-processed [13]. The climate and geo data is described by several attributes:

- latitude
- longitude
- mean month temperature
- max month temperature
- min month temperature
- mean precipitation
- 10 bioclimatic variables [4]

This is the summary of the data examples and table attributes:

nr of examples	2221
columns	171
nominals	1
numerics	69
binaries	101

Table 7.11: Mammals dataset

As we can see there are a lot of binaries attributes. They represent in fact for each mammal if it is included in the clusters or not. The numerics attributes are the geo data and the nominals are the name of the species.

This dataset does not have a single binary attribute, therefore to use it with our interactive implementation we created a *dummy variable* set to 0 to all the tuples in the dataset so the quality measure will be equal to zero for every subgroup. What we want to do is just analyse them with the use of the coefficients. Obviously with this approach a ROC curve analysis will be impossible because there is not a target concept.

Anyway what can we do is correlate the clusters with the animals and vice versa. The results from the dataset will be ranked by the coverage of the subgroups rather than the quality measure. Looking at the results the first thing

to note is the difference in the ranking between the subgroups, the subgroups where there is not a particular species are larger than the ones with the same specie, this mean that the clusters are not so populated most probably. After this analysis we can for example try to find patterns that describe a group of mammals.

The ones that we want analyse are: the *capra ibex*, the *apodemus alpicola*, the *Microtus multiplex* and the *chionomys nivalis*. These mammals live in alpine environments so most probably we can be lucky to find patterns correlated with them. We like the subgroups and the most relevant patterns correlated are (here we do not take into consideration the results position because it is irrelevant):

Description	Coverage	Phi Coefficient	p-value
Chionomys_nivalis = '1'	210	0.5924	< 0.001
Rupicapra_rupicapra = '1'	200	0.5125	< 0.001
Sorex_alpinus = '1'	170	0.4481	< 0.001
Marmota_marmota = '1'	114	0.5572	< 0.001
Capra_ibex = '1'	58	0.6266	< 0.001

Table 7.12: Correlated alpine mammals.

Basically what we found are animals that live with them, however there were no correlated patterns described by the geo data. What we have to do is try to expand the results and to disable the attributes related to the species. After multiple subgroups expansions we got a singular subgroup described by multiple patterns, the phi coefficient reaches good value:

Description		Coverage	Phi Coefficient	p-value
$\text{prec_may_utm} \geq 85.38 \wedge \text{min_temp_nov_utm} \leq -0.13 \wedge \text{latitude} \leq 47.63 \wedge \text{latitude} \geq 42.67 \wedge \text{bioclim19_utm} \geq 101.0$		77	0.5605	< 0.001
$\text{prec_may_utm} \geq 85.38 \text{ AND } \text{min_temp_nov_utm} \leq -0.13 \text{ AND } \text{latitude} \leq 47.63 \text{ AND } \text{latitude} \geq 42.67 \text{ AND } \text{bioclim17_utm} \geq 99.5$		77	0.5605	< 0.001
$\text{prec_may_utm} \geq 85.38 \text{ AND } \text{min_temp_nov_utm} \leq -0.13 \wedge \text{latitude} \leq 47.63 \wedge \text{latitude} \geq 42.67 \wedge \text{bioclim16_utm} \geq 287.5$		77	0.5605	< 0.001
$\text{prec_may_utm} \geq 85.38 \wedge \text{min_temp_nov_utm} \leq -0.13 \wedge \text{latitude} \leq 47.63 \wedge \text{latitude} \geq 42.67 \wedge \text{bioclim14_utm} \geq 32.0$		77	0.5605	< 0.001
$\text{prec_may_utm} \geq 85.38 \wedge \text{min_temp_nov_utm} \leq -0.13 \wedge \text{latitude} \leq 47.63 \wedge \text{latitude} \geq 42.67 \wedge \text{bioclim12_utm} \geq 690.42$		77	0.5605	< 0.001
$\text{prec_may_utm} \geq 85.38 \wedge \text{min_temp_nov_utm} \leq -0.13 \wedge \text{latitude} \leq 47.63 \wedge \text{latitude} \geq 42.67 \wedge \text{prec_dec_utm} \geq 35.08$		77	0.5605	< 0.001

Table 7.13: Alpine mammals environment.

With this example we showed that with this interactive approach we can compare mammals subgroups to each other and also to compare geo data with the subgroups.

7.5 About running time of the algorithm

Time consumption of the algorithms is generally a relevant aspect that has to be considered for evaluation of machine learning algorithms. In our research however this aspect becomes difficult to measure due to the heavy interaction with the dataset, every interaction has different timing and it would be useless to report every running time. Anyway because we did not touch the search algorithm, we changed only the evaluation of the candidate subgroups and the contingency table analysis time is irrelevant, we can be sure that we developed a good solution also in terms of time consuming.

Chapter 8

Conclusions and future work

The first aspect to underline in our research is that we really made the beam search *interactive*, the user can easily narrow the search to select the hypothesis space on which is interested in and select among the results which regions to expand exhaustively. The interaction with the patterns has simple semantic, the descriptions are easy to understand and the coefficients values too, the knowledge extracted is still *actionable* as with the standard subgroup discovery. The ROC Space analysis with the support of the box and whisker chart, can easily make the user able to draw the conclusions from the research. An enrichment of the research is done with the local knowledge extraction and the line chart. Examining the coefficients spreading over the results and the use of statistics tests on the significance of the comparisons can give significant aspects of the dataset, even if the ROC space analysis is unfeasible as seen with the mammals dataset. Moreover the timing consumption has to be considered: with the contingency table analysis we developed a fast method to compare the subgroups. More complex approaches such as *Learning to rank* with the use of machine learning models in fact would have been an impassable approach because of the relevant time that it would have taken. The results are "balanced", the search does not change so much considering only subgroups with false target attribute. This can be easily seen looking at the ROC Curve produced in the results section. All the subgroups remain above the diagonal or only some of them go under it. However some aspects could be improved, searching

for local knowledge in fact is a difficult task for the user, with the increasing depth of the beam search in fact more and more redundancy is included in the results and the manual filtering method could be seen as boring for the human user even if with that he can see significant aspects of the datasets. A possible way to avoid this could be to suggest the user some filters or to build a predictive model that can *learn from the interaction* which subgroups have to remain in the beam and which have to be discarded, maybe creating different experiments and record them. After that we could give them as input to a machine learning algorithm. Another possible way could be to learn how to generate the refinements of the subgroups, focusing on refinements closer to the interaction subgroups perhaps considering the binary variables and the attributes descriptions. The most frustrating aspect of the search in fact is when to stop it or not, since the search is an approximation, he does not know if with the next description he will be able to see more interesting subgroups or not. He has to be smart to investigate and focus exhaustively on only the interesting subgroups. Something that could be easily implemented is the *subgroup discovery inside the subgroup discovery*: when the user finds some interesting local knowledge he could run another subgroup discovery process between the interaction subgroups and the interesting subgroups, setting the target attribute as the same of the original search with the χ^2 test as quality measure. In this way he can see which attributes most describe the subgroups as similar or opposite depending on the interaction. Finally a relevant aspect to take into consideration is the interaction with models. However our approach works only on symmetric quality measure as WRAcc and the quality measure for linear regressions has bounds between 0 and 1 so it is not symmetric and the quality measure for bayesian networks has no bounds. Perhaps other quality measures can be developed or maybe with some mathematic manipulations the same result can be achieved. In conclusion it could be interesting moreover to investigate the interaction with the z-score as quality measure and the interaction with subgroups described by intervals and nominal value sets. The last interaction has already been developed in Cortana but it has not been taken into consideration because we wanted to focus first on basic interactions and

then maybe proceed further with the research in this field. Unfortunately we have tested the knowledge discovery only on datasets that did not have any economic or really interesting aspects. It would have been very challenging to see if this approach could have raised helpful knowledge in a real scenario.

Glossary

Attribute-Relation File Format ARFF file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files were developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with the Weka machine learning software. ARFF files have two distinct sections. The first section is the Header information, which is followed the Data information. The Header of the ARFF file contains the name of the relation a list of the attributes *thecolumnsinthedata* and their types. 7

DAG directed graph with no directed cycles. 12

degree of freedom Statisticians use the terms degrees of freedom to describe the number of values in the final calculation of a statistic that are free to vary. 30

Pearson's coefficient is a coefficient that expresses the linearity between their covariance and the product of their respective standard deviations. The mean Pearson correlation analysis using Pearson correlation analysis. Given two statistical variables X and Y, the Pearson correlation coefficient is defined as their covariance divided by the product of the standard deviations of the two variables: $\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$. 32

table the input of a data mining algorithm is usually represented by a table which is described by its rows and its columns. The rows are called examples and the columns attributes. 8

List of Figures

2.1	A linear regression Model	13
2.2	A naive bayes classifier with three variables, X influences both Y and Z, the simplest bayesian network.	14
3.1	Z-score graph representation.	23
3.2	Isometrics lines for Accuracy and WRAcc.	27
3.3	Roc Curve with WRAcc isometrics lines in blue.	29
4.1	Coefficients ranges of Phi and Yule's Q coefficients.	37
5.1	Interactive data mining systems.	43
5.2	Use case for interacting with the beam search.	44
5.3	User activity diagram.	52
6.1	Cortana logo.	54
6.2	Cortana's main window.	56
6.3	Dataset details.	57
6.4	Target concept section.	57
6.5	Search conditions.	58
6.6	Search strategy.	59
6.7	Cortana results.	60
6.8	Cortana ROC Curve.	62
6.9	UML classes diagram.	64
6.10	UML GUI classes diagram.	65
6.11	UML Interfaces diagram.	67
6.12	Dislike option window.	68

6.13	Bounds for numeric attributes on the subgroup's description. .	69
6.14	Additional Information Window.	69
6.15	Filtering.	70
6.16	File Menu.	71
6.17	Disable attributes window.	71
6.18	Interactive buttons.	72
6.19	Interaction with data and new subgroups showing	73
6.20	Line chart for qualities.	74
6.21	Box and whisker plot for qualities values.	75
7.1	ROC Curve for the first run.	82
7.2	Line chart.	83
7.3	Box and whisker chart.	83
7.4	ROC Curve with liked subgroups.	85
7.5	ROC Curve like Marital status = 'Married-civ-spouse', Relationship = husband, Sex = 'Male'.	86
7.6	Box plot like Marital status = 'Married-civ-spouse', Relationship = husband, Sex = 'Male'.	87
7.7	ROC Curve dislike Marital status = 'civ-spouse', Relation-husband, Sex = 'Male'.	88
7.8	Line chart like young people.	90
7.9	Line Chart opposite subgroups.	91
7.10	box and whisker chart opposite subgroups.	92
7.11	Roc Curve total opposite subgroups.	93
7.12	Box chart total opposite subgroups.	93
7.13	Italians correlated coefficients.	95
A.1	Coefficients ranges.	118
A.2	WRAcc range.	119

List of Tables

4.1	Contingency table 2x2.	33
4.2	Critical chi-squared values.	38
4.3	Coefficients values.	41
7.1	Adults dataset	81
7.2	Interesting subgroups like (89 Results).	84
7.3	Best subgroup (89 results)	86
7.4	Focused subgroup (90 Results).	86
7.5	Population described by Age (34 Results)	88
7.6	Young subgroup (149 Results)..	89
7.7	Interesting young subgroups (68 Results).	89
7.8	Interesting own child subgroup (184 Results).	89
7.9	United States population.	91
7.10	Opposite subgroups (34 Results).	94
7.11	Mammals dataset	96
7.12	Correlated alpine mammals.	97
7.13	Alpine mammals environment.	98
1	Adults dataset attributes descriptions.	112

Bibliography

- [1] Eric K. Y. Ho Arno J. Knobbe. Pattern teams. *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2006.
- [2] Alan H. Cheetham and Joseph E. Hazel. Binary (presence-absence) similarity coefficients. *Journal of Paleontology*, 43(5):1130–1136, September 1969.
- [3] P. Clark and T. Niblett. The cn2 induction algorithm. *Machine Learning*, 1989.
- [4] WorldClim Global Climate Data Free climate data for ecological modeling and GIS. <http://www.worldclim.org/bioclim>.
- [5] Ad Feelders Dennis Leman and Arno Knobbe. Exceptional model mining. *ECML/PKDD*, 2008.
- [6] M.C. Pike D.Hill. Algorithm 299 chi squared integral. *Commun. ACM* 01/1967, 1965.
- [7] Population Division. <http://www.census.gov/sipp/weights.html>.
- [8] Ralph Johnson Erich Gamma, Richard Helm and John Vlissides. *Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.
- [9] R. A Fisher. On the interpretation of chi-squared from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 1922.

- [10] Tarek Abudawood Peter Flach. Evaluation measures for multi-class subgroup discovery. In Springer, editor, *European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 35–50, 2009.
- [11] LIACS Data Mining group. <http://www.datamining.liacs.nl/background.html>.
- [12] LIACS Data Mining group. <http://datamining.liacs.nl/>.
- [13] J. Eronen H. Heikinheimo, M. Fortelius and H. Mannila. Biogeography of european land mammals shows environmentally distinct and spatially coherent clusters. *Journal of Biogeography*, 2007.
- [14] Liquiang Geng Howard J. Hamilton. Interestingness measures for datamining: A survey. *ACM Comput. Surv.*, 2006.
- [15] Stefan Ruping Henrik Grosskreutz and Stefan Wrobel. Tight optimistic estimates for fast subgroup discovery. *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2008.
- [16] Hijmans. <http://www.worldclim.org>.
- [17] Mark A. Hall Ian H. Witten, Eibe Frank. *Data Mining Practical Machine Learning Tools and Techniques*. Mantesh, 3rd edition edition, 2011.
- [18] Constantin F. Aliferis Ioannis Tsamardinos, Laura E. Brown. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 2006.
- [19] Pearl J. Bayesian networks: A model of self-activated memory for evidential reasoning. *Proceedings of the 7th Conference of the Cognitive Science Society*, 1985.
- [20] Fürnkranz Johannes and Peter A. Flach. Roc 'n' rule learning - towards a better understanding of covering algorithms. *Machine Learning*, 2005.

- [21] Rob M. Konijn Wouter Duivesteijn Wojtek Kowalczyk Arno Knobbe. Discovering local subgroups, with an discovering local subgroups, with an application to fraud detection. *Advances in Knowledge Discovery and Data Mining*, 2013.
- [22] Richard J. Larsen and Morris L. Marx. *An Introduction to Mathematical Statistics and Its Applications*. Pearson, 2000.
- [23] Nada Lavrač. Subgroup discovery techniques and applications. *Advances in Knowledge Discovery and Data Mining*, 2005.
- [24] JFree Chart Library. <http://www.jfree.org/jfreechart/>.
- [25] Societas Europea Mammalogica. <http://www.european-mammals.org/>.
- [26] Ad Feeldersy Michael Mampaey, Siegfried Nijssenz and Arno Knobbe. Efficient algorithms for finding richer subgroup descriptions efficient algorithms for finding richer subgroup descriptions in numeric and nominal data. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, 2012.
- [27] Karl Pearson. On the theory of contingency and its relation to association and normal correlation. *Research Memoirs Biometric Series I*, 1904.
- [28] Barbara F.I. Pieters. Subgroup discovery on numeric and ordinal targets, with an application to biological data aggregation. In *ECML/PKDD-10 Tutorial and Workshop: Barcelona, 2010.*, 2010.
- [29] Nada Lavrač Jožef Stefan Nova Gorica Polytechnic. Subgroup discovery techniques and applications. *PAKDD'05 Proceedings of the 9th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, 2005.
- [30] UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml>.
- [31] Neil J. Salkind. *Encyclopedia of Measurement and Statistics*. SAGE Publications, Inc., 2007.

- [32] James Theiler. Combining statistical tests by multiplying p-values. *Astrophysics and Radiation Measurements Group, NIS-2*, 2004.
- [33] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
- [34] Matthijs van Leeuwen · Arno Knobbe. Diverse subgroup set discovery. *Data Min Knowl Disc*, 2012.
- [35] Matthijs J. Warrens. Similarity coefficients for binary data. *Properties of Coefficients, Coefficient Matrices, Multi-way Metrics and Multivariate Coefficients*, 2008.
- [36] Arno Knobbe Wouter Duivesteijn. Exploiting false discoveries – statistical validation of patterns and quality measures in subgroup discovery. *Proceeding ICDM '11 Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, 2011.
- [37] Yan Zhao and Yiyu Yao. On interactive data mining. *Proceedings of the Second Indian International Conference on Artificial Intelligence*, 2444-2454, 2005.
- [38] Peter V. Zysno. The modification of the phi coefficient reducing its dependence on the marginal distributions. *Methods of Psychological Research Online*, 1997.

Attribute name	Type	Values
Class	Binary	$[\leq 50K, \geq 50K]$
Age	Numeric	[17,90]
Workclass	Nominal	Private, self-emp-not-inc, self-emp-inc, federal-gov, local-gov, state-gov, without-pay, never-worked
Fnlwgt	Numeric	[12285,1490400]
Education	Nominal	Preschool, 1st-4th, 5th-6th, 7th-8th, 9th, 10th, 11th, 12th, HS-grad, some-college, assoc-voc, assoc-acdm, bachelors, masters, prof-school, doctorate
Education-num	Numeric	[1,16] (index in ordered list of Education)
Marital-status	Nominal	Married-civ-spouse, divorced, never- married, separated, widowed, married-spouse-absent, married-AF-spouse
Occupation	Nominal	Tech-support, craft-repair, other-service, sales, exec-managerial, prof-specialty, handlers-cleaners, machine-op-inspct, adm-clerical, farming-fishing, transport-moving, priv-house-serv, protective-serv, armed- forces
Relationship	Nominal	Wife, own-child, husband, not-in-family, other-relative, unmarried
Race	Nominal	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Sex	Nominal	Female, Male
Capital-gain	Numeric	[0,99999]
Capital-loss	Numeric	[0,4356]
Hours-per-week	Numeric	[1,99]
Native-country	Nominal	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US, India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France,Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad and Tobago, Peru, Hong, Holland-Netherlands

Table 1: Adults dataset attributes descriptions.

Appendix A

Coefficients Generator

Below we report a python script that we wrote to compare some coefficients found in the statistics literature. We compared the Yule's Q coefficient, the phi coefficient, a phi coefficient modification [38], the okai coefficient and the fager coefficients [2]. This coefficients have been chosen among several for their symmetrical nature. As we see the okai and the fager coefficient are in some way linear but they do not reach the zero value when the binary variables are completely different. Otherwise the linear coefficients are really similar and only little differences exist between them and that's why we chose the phi coefficient because it was the most linear between all (it is modification does not change in a significant way the coefficient values).

```
1 #Yule 's Q (ad-bc)/(ad+bc)
  #Phi (ad-bc)/sqrt(a+b * c+d * a+c * b+d)
3 #Phi2 (ad-bc)/|ad+bc| + N * e, e = min(b,c) if ad >= bc, min(a,d)
  ↪ ) otherwise
  #Fager a/sqrt((a+b)*(c+d)) - (1/(2*sqrt(c+d)))
5 #WRAcc (|G|/|S|*(1^G - 1^S)) 1<=G<=N S=|N| 1<=1^G<=N 1^S=K
  #okai = a/sqrt((a+b)*a+c)
7 import random
  import math
9 from decimal import Decimal

11 s1 = []
```

```

s2 = []
13
#WRacc
15 WRAccList = []

17 output_yule = open("outputYule.txt", "w")
   output_phi = open("outputPhi.txt", "w")
19 output_phi2 = open("outputPhi2.txt", "w")
   output_fager = open("outputFager.txt", "w")
21 output_okai = open("outputOkai.txt", "w")
   output_yule2 = open("outputYule2.txt", "w")
23 output_WRacc = open("outputWracc.txt", "w")

25
def dumpWRacc():
27
    for i in reversed(range(0,100)):
29        for j in reversed(range(0,100)):
            print i,j
31            WRacc = 0.0
            WRacc = (float(i)/100) * (float(j) - float(100-j))
33            WRaccMax = float(100/100 * 100)
            WRacc = (WRacc/WRaccMax)
35            WRAccList.append(WRacc)

37 k = 0
   WRAccList.sort(reverse=True)
39 for i in range(0, len(WRAccList)):
    output_WRacc.write(str(k) + "\t" + str(WRAccList[i]) + "\n")
41 k = k + 1

43
def Okai(a,b,c,d):
45     return str(Decimal(a/(math.sqrt((a+b)*(a+c))))))

47 def Fager(a,b,c,d):
    return str(Decimal(a/math.sqrt((a+b)*(c+d)) - (1/(2*math.sqrt(2
        ↪ c+d))))))
49

```

```

def Phi2(a,b,c,d):
51     e = 0.0
53     min1 = 0.0
        min2 = 0.0
55
        if(b < c):
57             min1 = b
        else:
59             min1 = c

        if(c < d):
61             min2 = c
        else:
63             min2 = d

65     if a*d >= b*c:
67         e = min1
        else:
69         e = min2

71     return str(Decimal((a*d - b*c)/(math.fabs(a*d + b*c)+ 100*e))
        ↪ )

73
def Phi(a,b,c,d):
75     return str(Decimal((a*d - b*c)/(math.sqrt((a+b)*(c+d)*(a+c)*(b
        ↪ +d))))))

77 def Yule(a,b,c,d):
        return str(Decimal((a*d - b*c)/(a*d + b*c)))
79

def Yule2(a,b,c,d):
81     return str(Decimal((math.sqrt(a*d) - math.sqrt(b*c)/(math.sqrt
        ↪ (a*d) + math.sqrt(b*c))))))

83
def count(s1,s2,j):
85

```



```

a = 0.0
87 b = 0.0
c = 0.0
89 d = 0.0

91 for i in range(0,100):
    if s1[i] == 0 and s2[i] == 0:
93         a = a + 1
    if s1[i] == 0 and s2[i] == 1:
95         b = b + 1
    if s1[i] == 1 and s2[i] == 0:
97         c = c + 1
    if s1[i] == 1 and s2[i] == 1:
99         d = d + 1
    print a,b,c,d
101
output_yule.write(str(j) + "\t" + Yule(a,b,c,d) + "\n")
103 output_phi.write(str(j) + "\t" + Phi(a,b,c,d) + "\n")
output_phi2.write(str(j) + "\t" + Phi2(a,b,c,d) + "\n")
105 output_fager.write(str(j) + "\t" + Fager(a,b,c,d) + "\n")
output_okai.write(str(j) + "\t" + Okai(a,b,c,d) + "\n")
107 output_yule2.write(str(j) + "\t" + Yule2(a,b,c,d) + "\n")

109
#create the other_vector with x differences
111 def other_vector(x):

113     k = 0
    for i in range(0,100):
115         if k < x:
            if s1[i] == 0:
117                 s2[i] = 1
                k = k + 1
119         else:
            s2[i] = 0
121             k = k + 1

123 for i in xrange(0,100):
    x = random.randrange(0, 2)

```

```
125     s1.append(x)
127
129 for i in range(0,100):
131     s2.append(s1[i])
133
135 for i in range(1,100):
137     count(s1,s2,i)
139     other_vector(i)
141
143 dumpWRacc()
```

coefficients.py

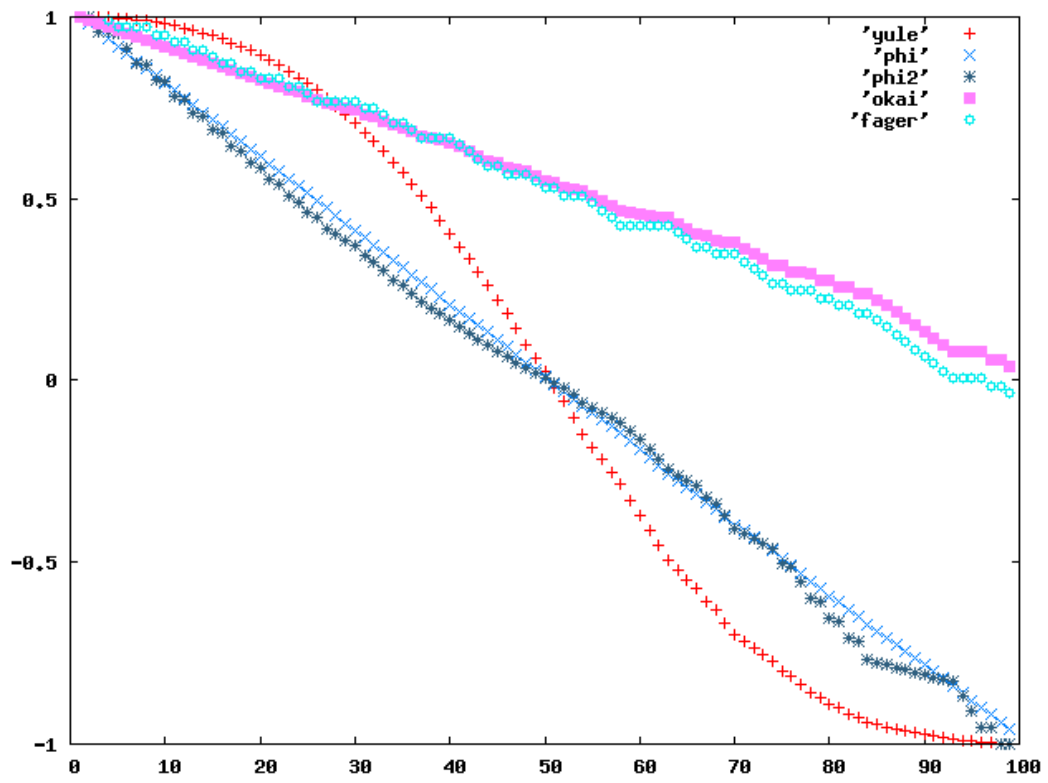


Figure A.1: Coefficients ranges.

Just to be sure that the quality measure could take range between 0 and 1 we plotted it and with the script we generated all its possible values. We generated them for a vector of 100 variables. We run two cycles, one inside the other. The first to cycle over all the possible size of the subgroups and the other for every possible number of true values that could result in the target's subgroup (100) and that's why there are 10000 total comparisons.

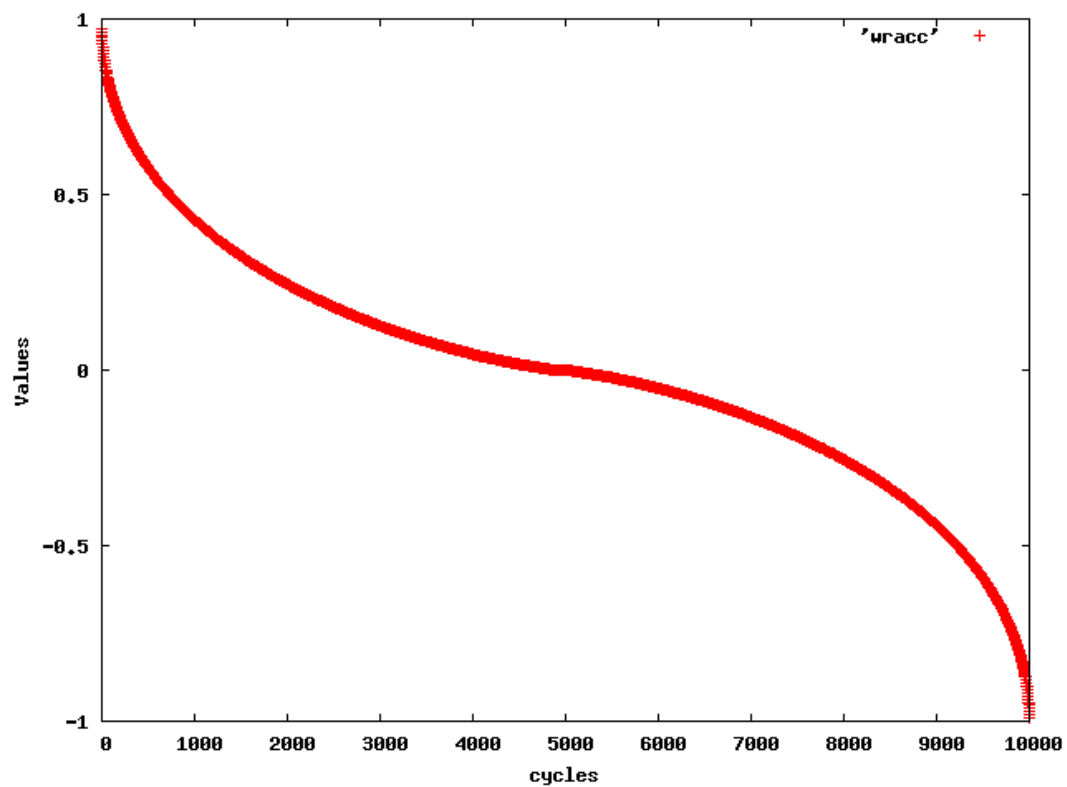


Figure A.2: WRacc range.

Appendix B

Relevant code

```
public void focus(SubgroupInteractive aSubgroup)
2 {
4     /* if we have only a description length equal to 1 we skip ↵
    ↵ */
    if (aSubgroup.getDepth() == 1)
6         return;

8     SubgroupInteractive aSubgroupCopy = aSubgroup.copy();

10    for (int i = 0; i < aSubgroup.getItsConditionsInteractive().↵
    ↵ size() - 1; i++)
    {
12        for (int j = 1; j < aSubgroup.getItsConditionsInteractive↵
    ↵ ().size(); j++)
        {
14            if (aSubgroup.getItsConditionsInteractive().get(i).↵
    ↵ getItsColumnInteractive().isNumericType())
16                {

18                    float [] aSplitPoints = aSubgroup
                        .getItsConditionsInteractive().get(i)
20                        .getItsColumnInteractive()
                        .getUniqueNumericDomain(aSubgroup.getMembers());
```

```

22         for (float aSplit : aSplitPoints)
23         {
24
25             ConditionInteractive aCondition = aSubgroup.↵
26             ↵ getItsConditionsInteractive().get(i).copy();
27             aCondition.setValue(Float.toString(aSplit));
28
29             if(aCondition.getColumn().evaluate(aCondition).↵
30             ↵ cardinality() > 0) {
31
32                 aSubgroupCopy.getItsConditionsInteractive().set(i, ↵
33                 ↵ aCondition);
34                 aSubgroupCopy.setItsMembers(aCondition.getColumn().↵
35                 ↵ evaluate(aCondition));
36
37                 if (aSubgroupCopy.getItsConditionsInteractive().get(↵
38                 ↵ j).getItsColumnInteractive().isNumericType())
39                     focusNumeric(aSubgroupCopy, j);
40                 if (aSubgroupCopy.getItsConditionsInteractive().get(↵
41                 ↵ j).getItsColumnInteractive().isNominalType())
42                     focusNominal(aSubgroupCopy, j);
43             }
44         }
45
46         if (aSubgroup.getItsConditionsInteractive().get(i).↵
47         ↵ getItsColumnInteractive().isNominalType())
48         {
49             TreeSet<String> aDomain = aSubgroup.↵
50             ↵ getItsConditionsInteractive().get(i).↵
51             ↵ getItsColumnInteractive().getDomain();
52
53             for (String aValue : aDomain)
54             {
55                 if (!checkValue(aValue))

```

```

52         {
           ConditionInteractive aCondition = aSubgroup.↵
↵   getItsConditionsInteractive().get(i).copy();
54           aCondition.setValue(aValue);

           if (aCondition.getColumn().evaluate(aCondition).↵
↵   cardinality() > 0) {

58               aSubgroupCopy.getItsConditionsInteractive().set(i,↵
↵   aCondition);
               aSubgroupCopy.setItsMembers(aCondition.getColumn().↵
↵   .evaluate(aCondition));

60               if (aSubgroupCopy.getItsConditionsInteractive().↵
↵   get(j).getItsColumnInteractive().isNumericType())
62                   focusNumeric(aSubgroupCopy, j);
               if (aSubgroupCopy.getItsConditionsInteractive().↵
↵   get(j).getItsColumnInteractive().isNominalType())
64                   focusNominal(aSubgroupCopy, j);
               }
66           }
           }

68       }

70   }

72   }

74   itsResultInteractive.setIDs();
76 }

```

focus.java

```

package nl.liacs.subdisc.interactive;
2
import java.util.Comparator;
4
/* better to use a treseet comparator than compareTo methods */
6 public class SubgroupComparator implements Comparator<
    ↳ SubgroupInteractive>
    {
8     @Override
    public int compare(SubgroupInteractive s1, SubgroupInteractive
        ↳ s2)
10    {
12        /* if subgroup is equals not include it in the results */
        if (s1.equals(s2))
            return 0;
14
        if (s1.getOldQuality() > s2.getOldQuality())
16            return -1;
        else if (s1.getOldQuality() < s2.getOldQuality())
18            return 1;
        else if (s1.getCoverage() > s2.getCoverage())
20            return -1;
        else if (s1.getCoverage() < s2.getCoverage())
22            return 1;
24        return -1;
26    }
    }
}

```

SubgroupComparator.java


```

1  /* new check and log function */
   protected void checkAndLog(SubgroupInteractive theSubgroup, int ↵
       ↳ theOldCoverage)
3  {
   int aNewCoverage = theSubgroup.getCoverage();
5
   if (aNewCoverage < theOldCoverage && aNewCoverage >= super.↵
       ↳ getItsMinimumCoverage())
7  {
9
   float aQuality = evaluateCandidate(theSubgroup);
   theSubgroup.setMeasureValue(aQuality);
11  boolean okLike = true;
   boolean okDislike = true;
13  boolean okOpposite = true;
15
   if (likes.size() > 0)
   {
17       if (!ResultWindowInteractive.lowFilter
           && (theSubgroup.getItsQualityLike() <= 0 || ↵
↳ theSubgroup.getItsQualityLike() < itsSearchParameters.↵
↳ getMinimumLike()))
19           okLike = false;
21
           if (ResultWindowInteractive.lowFilter
               && (theSubgroup.getItsQualityLike() <= 0 || ↵
↳ theSubgroup.getItsQualityLike() > itsSearchParameters.↵
↳ getMinimumLike()))
23               okLike = false;
           }
25
   if (dislikes.size() > 0)
27  {
       if (!ResultWindowInteractive.lowFilter
           && (theSubgroup.getItsQualityDisLike() >= 0 || ↵
↳ theSubgroup.getItsQualityDisLike() < itsSearchParameters.↵
↳ getMinimumDislike()))
29           okDislike = false;

```

```

31         if (ResultWindowInteractive.lowFilter
33             && (theSubgroup.getItsQualityDisLike() >= 0 || ↵
↵ theSubgroup.getItsQualityDisLike() > itsSearchParameters.↵
↵ getMinimumDislike()))
        okDislike = false;
35     }

37     if (opposite.size() > 0)
    {
39         if (!ResultWindowInteractive.lowFilter
            && (theSubgroup.getItsQualityOpposite() <= 0 || ↵
↵ theSubgroup.getItsQualityOpposite() < itsSearchParameters.↵
↵ getMinimumOpposite()))
41             okOpposite = false;

43         if (ResultWindowInteractive.lowFilter
            && (theSubgroup.getItsQualityOpposite() <= 0 || ↵
↵ theSubgroup.getItsQualityOpposite() > itsSearchParameters.↵
↵ getMinimumOpposite()))
45             okOpposite = false;
        }

47         if (theSubgroup.getMeasureValue() > super.↵
↵ getItsQualityMeasureMinimum()
49             && aNewCoverage <= super.getItsMaximumCoverage()
            && okLike
51             && okDislike
            && okOpposite
53             && theSubgroup.getPValue() <= itsSearchParameters.↵
↵ getMinimumPValue()
            && !itsResultInteractive.contains(theSubgroup))
55         {
            itsResultInteractive.add(theSubgroup);
57         }

59     itsCandidateQueue.add(new CandidateInteractive(theSubgroup)) ↵
↵ ;

```

```

61     logCandidateAddition(theSubgroup);
        super.getItsCandidateCount().getAndIncrement();
63     }

65 }

67 /* new interactive evaluation */
    public float evaluateCandidate(SubgroupInteractive ↵
        ↵ theNewSubgroup)
69 {
    float aQuality = 0.0f;
71
    switch (itsSearchParameters.getTargetType())
73 {
        case SINGLE_NOMINAL:
75     {
            int aCountHeadBody = 0;
77         final BitSet aMembers = theNewSubgroup.getMembers();

79         for (int i = aMembers.nextSetBit(0); i >= 0; i = aMembers.↵
            ↵ nextSetBit(i + 1))
                if (getItsBinaryTarget().get(i))
81                 ++aCountHeadBody;

83         aQuality = getItsQualityMeasure().calculate(aCountHeadBody ↵
            ↵ , theNewSubgroup.getCoverage());

85
            if (getItsQualityMeasure().getROCHaven() != 0)
87                 aQuality = aQuality / getItsQualityMeasure().↵
            ↵ getROCHaven();

89         theNewSubgroup.setOldQuality(aQuality);
            theNewSubgroup.setSecondaryStatistic(aCountHeadBody/ (↵
            ↵ double) theNewSubgroup.getCoverage()); // relative ↵
            ↵ occurrence of positives in subgroup
91         theNewSubgroup.setTertiaryStatistic(aCountHeadBody); // ↵
            ↵ count of positives in the subgroup

```

```

93     float qualityLike = addQualityLike(theNewSubgroup);
    float qualityDislike = removeQualityDislike(theNewSubgroup↵
↵ );
95     float qualityOpposite = removeQualityOpposite(↵
↵ theNewSubgroup);

97     aQuality = (aQuality + qualityLike + qualityDislike + ↵
↵ qualityOpposite);

99     break;
    }

101
102     case SINGLE_NUMERIC:
103     {
        /* TODO: to be implemented */

105
        break;
107     }

109     default:
        break;
111 }

113 return aQuality;
}

```

SubgroupDiscoveryInteractive.java

```
2 public ContingencyTable(BitSet b1, BitSet b2)
3 {
4
5     for (int i = 0; i < b1.length(); i++)
6     {
7
8         if (!b1.get(i) && !b2.get(i))
9         {
10             a++;
11         }
12
13         if (!b1.get(i) && b2.get(i))
14         {
15             b++;
16         }
17
18         if (b1.get(i) && !b2.get(i))
19         {
20             c++;
21         }
22
23         if (b1.get(i) && b2.get(i))
24         {
25             d++;
26         }
27
28     }
29
30     this.N = b1.length() + b2.length();
31     this.n1 = a + b;
32     this.n2 = c + d;
33     this.n3 = a + c;
34     this.n4 = b + d;
35
36 }
```

ContingencyTable.java

Acknowledgements

I would like to express my gratitude to Dr. Arno Knobbe to have been always available to help me during my thesis and also I would like to thank all his data mining group and the InfraWatch group: Wouter Duivesteijn, Rob Konijn, Jan Van Rijn, Ricardo Cachucho. I had really good time with them. Special thanks also to my italian colleagues Ugo Vespier and Alberto Baggio to have supported me. I extend my gratitude also to my italian supervisor Proff.ssa Francesca Rossi that have helped me when I came back. Finally I would like to thank my family to have let me go to Netherlands to meet all these people and also to have supported me always during my university career. Last but not least my girlfriend and my roommates in Padova.