TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

NIINA SIMONEN

DISCRETIZATION IN SUBGROUP DISCOVERY

Master of Science thesis

Examiners: Prof. Tapio Elomaa,
M.Sc. (Tech) Juho Lauri
Examiner and topic approved by the
Faculty Council of the Faculty of
Computing and Electrical Engineering
on 9th March 2016

ABSTRACT

NIINA SIMONEN: Discretization in Subgroup Discovery
Tampere University of Technology
Master of Science thesis, 53 pages, 13 Appendix pages
May 2016
Master's Degree Programme in Information Technology
Major: Computer Science
Examiners: Prof. Tapio Elomaa, M.Sc. (Tech) Juho Lauri
Keywords: subgroup discovery, discretization

Subgroup discovery is a data mining technique to discoverer interesting subgroups from a selected population. It seeks to discover interesting relationships between different objects in a set with respect to a specific property. The discovered patterns are called subgroups and they are represented in the form of rules. Discretization is technique to replace numerical attributes with nominal ones, making it possible to use them with algorithms that do not support numerical attributes.

In this thesis two datasets are discretized for the application of subgroup discovery. For the discretizations four different methods were used and three different bin amounts were applied. The used datasets are the heart disease and the Australian credit approval from the UCI Machine Learning Repository. The subgroup discovery technique produced eleven subgroups sets as result, eight from heart disease dataset and three from Australian credit approval dataset. We observed that the bin amount affects greatly on the results. Also, with the binary discretization there are subgroup sets with a high share of subgroups with discretized attributes. In addition, the importance of expert guidance is emphasized.

TIIVISTELMÄ

NIINA SIMONEN: Diskretointi osajoukkojen haussa
Tampereen teknillinen yliopisto
Diplomityö, 53 sivua, 13 liitesivua
Toukokuu 2016
Tietotekniikan koulutusohjelma
Pääaine: Ohjelmitotiede
Tarkastajat: Prof. Tapio Elomaa, DI Juho Lauri
Avainsanat: osajoukon haku, diskretointi

Osajoukkojen haku on tiedonlouhintatekniikka, jolla pyritään löytämään mielenkiintoisia osajoukkoja väestöstä. Se löytää mielenkiintoisia suhteita eri objektien välillä joukosta, jonkin spesifioidun ominaisuuden perusteella. Löydetyt osajoukot kuvataan kaavojen avulla. Diskretisointi on tekniikka, jolla korvataan numeeriset atribuutit nominaaleilla. Tämä mahdollistaa sellaisien algoritmien käytön, jotka eivät suoraan tue numereelisien atribuuttien käsittely.

Tässä diplomityössä kaksi datasettiä on diskretisoitu ennen osajoukkojen hakua. Diskretisointiin on käytetty neljää erilaista tapaa ja kolmea eri siilomäärää. Käytetyt datasetit ovat sydäntautikanta ja austraalialainen luottopäätöksentekokanta. Osajoukkojen haut tuottivat yksitoista osajoukkoryhmää, joista yhdeksän on sydäntautikannasta ja loput kolme austraalialaisesta luottopäätöksentekokannasta. Kun tuloksia tarkastellaan, niin on huomattavissa, että siilojen lukumäärä vaikuttaa paljon lopputuloksiin. Lisäksi binääridiskretisoinnin kanssa saadaan osajoukkoryhmiä missä on korkea osuus osajoukkoja joilla on diskretisoituja attribuutteja. Myös asiantuntijuuden tarve kororstuu osajoukkojen mielenkiintoisuuden arvioinnissa.

PREFACE

This thesis is submitted as a part of my masters studies on Information Technology In Tampere University of the Technology. The challenge for me was to combine this and my studies, work, motherhood and spare time, but I'm happy to say that it seemed to sometimes painful, but still possible.

I like to thank my sister Suvi and friend Hanne for encouragement, my coworkers for patience and husband Toni for all of these and for the love and support. And for my dog Chico, who I lost while doing this, thank you for the eleven years, it was a blast.

Tampere, 23.5.2016

Niina Simonen

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

FP	Frequent pattern
GA	Genetic algorithm
KDD	Knowledge discovery in databases
MOEA	Multiobjective genetic algorithm
ROC	Receiver operating characteristics
SDRD	Supervised descriptive rule discovery

Class	Target property of interest
Cond	A conjunction of attribute values
FP	False positives
FPr	False positive rate
TP	True positives
TPr	True positive rate

1. INTRODUCTION

Knowledge discovery in databases (KDD) is the nontrivial process of identifying valid, original, and potentially useful patterns in data [18, 10]. A pattern that is interesting and certain enough, both according to the user's criteria, is called *knowledge* [18]. Discovered knowledge is the output of a program that monitors the set of facts in a database and produces patterns.

Within the KDD process the data mining stage is responsible for high-level automatic knowledge discovery from information obtained from real data [10]. *Predictive* and *descriptive induction* are two high-level goals in data mining [16]. Predictive induction produces classification and predictive rules with classical rule learning algorithms and descriptive induction involves mining of association rules, subgroup discovery and other approaches to non-classificatory induction [29]. The boundaries between prediction and description are not sharp and the tasks may overlap, the distinction is useful for understanding the overall discovery goal [16].

In this thesis the focus is on subgroup discovery. The concept of subgroup discovery was introduced by Klösgen [27] with the EXPLORA algorithm and Wrobel [42] with the MIDOS algorithm. The problem of subgroup discovery can be defined as follows [27, 42, 19]. Given a population of individuals and a property of those individuals, we are interested in finding population subgroups that are statistically "most interesting"; for example they are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest.

Subgroup discovery is a data mining technique aimed at discovering interesting relationships between different objects in a set with respect to a specific property [27, 42]. The discovered patterns are normally represented in the form of rules and called subgroups [40]. The discovered patterns are easy to interpret by the users and the domain experts [19]. The subgroups discovered in the data have an explanatory nature, and the interpretability for the final user of the extracted knowledge is

a crucial aspect in this field [39]. As on examples of subgroup descriptions, the following rules describe subgroups of heart disease patients who have asymptomatic chest pain:

- *fasting blood sugar* $\leq$ *120 mg/dl AND exercise induced angina = yes*, and
- *fasting blood sugar* $\leq$ *120 mg/dl AND sex = male AND maximum heart rate achieved = (-*∞*, 142.50].*

Decision support in targeting campaign and planning a population screening campaign aimed at detecting individuals with high disease risk are examples of applications of subgroup discovery [28].

Many subgroup discovery algorithms cannot handle numerical or continuous target attributes which must be *discretized* before using these algorithms [34]. When attributes are discretized, continuous attributes are replaced by nominal attributes [12]. In this thesis two different datasets are discretized for the application subgroup discovery. The used datasets are the heart disease and the Australian credit approval from the UCI Machine Learning Repository. The heart disease dataset contains 270 instances of heart disease patients and the Australian credit approval dataset contains 690 instances of credit applications of customers. There are four different discretization ways that are used and three different bin amounts. The subgroup discovery technique produced eleven subgroups sets as result, eight from heart disease dataset and three from Australian credit approval dataset. We observed that the bin amount affects greatly on the results and that with the binary discretization there are subgroup sets with a high share of subgroups with discretized attributes. Also, the importance of expert guidance is emphasized.

This thesis is organization is as follows. In Chapter 2 background of the topic is introduced. In Chapter 3 the problem of discretization of numeric variables is covered. In Chapter 4 are the descriptions of used and discretized datasets. In addition, used tools and options for the subgroup discovery tasks are presented. In Chapter 5 are the test results and in Chapter 6 are the conclusions.

2. BACKGROUND

Subgroup discovery has the following four main properties [5].

- *Type of target variable* may be binary, nominal, or numeric.

- *Description language* specifying the individuals from the reference population belonging to the subgroup. Mainly conjunctive languages are used. The subgroup description consists of a set of selection expressions or selectors.

- *Quality function* measures the interestingness of the subgroups. The type of the quality function is determined by the type of the target variable.

- *Search strategy* is a very important factor in subgroup discovery. The search space is exponential in the number of possible selectors of a subgroup description.

This chapter is organized as follows. After subgroup discovery is introduced in Section 2.1, the main focus of the next three sections is the quality of the subgroups. These three sections cover the rule quality, ROC analysis for subgroup and quality measurements including different quality functions for subgroup discovery. Search strategies and other techniques used for handling the large size of the search space of the subgroup discovery problem are presented in 2.5 and subgroup discovery algorithms are introduced in 2.6.

2.1 Subgroup discovery

The result of standard rule induction is a classification model which consists of a set of rules [28]. Subgroup discovery aims at finding patterns in data described with individual rules. An induced subgroup description has the form of implication $Class \leftarrow Cond$. In rule learning terms *Class* means the target class that appears in the rule consequent for the property of interest in subgroup discovery. *Cond* is a conjunction of attribute-value pairs selected from features describing the training

instances. With the heart disease dataset which is described in Section 4.2 the target class could be *fbs*, *sex*, *class*, or any other attribute. The *Cond* for target class *sex* would be 0 or 1, depending whether the target is to find subgroups of male or female population.

The goal of standard classification rule learning is to generate models, one for each class, consisting of rule sets describing class characteristics in terms of properties occurring in the descriptions of training examples [31]. Subgroup discovery on the other hand aims at discovering individual rules or patterns of interest without generating any models. Standard classification rule learning algorithms cannot address the task of subgroup discovery as they use covering algorithm for rule set construction and they use search heuristics aimed for rule set accuracy [28]. Subgroup discovery task can often tolerate more false positives than classification task.

Subgroup discovery and classification rule learning can be unified under the umbrella of *cost-sensitive classification* [31]. In cost-sensitive classification also the misclassification costs are taken into account [13]. With both subgroup discovery and classification rule learning, when deciding optimal classifiers, the thing that matters is the expected profit in a given context [28].

As mentioned before, predictive and descriptive inductions are high-level discovery goals of the KDD process for finding autonomously new patterns [16]. In predictive induction a system finds patterns for predicting the future behavior of some entities and in descriptive induction the system finds patterns for presentation to a user in a human-understandable form. There are several techniques that lie halfway between descriptive and predictive data mining [24]. *Supervised descriptive rule discovery (SDRD)* is a proposed paradigm which includes techniques combining the features of both type of inductions, and its main objective is to extract descriptive knowledge from data of a property of interest [36, 24]. Common to these techniques is that they use supervised learning to solve descriptive tasks. Within these techniques included are *contrast set mining*, *emerging pattern mining* and subgroup discovery [36]. Contrast set mining task is defined as a conjunction of attribute-value pairs defined on groups with no attribute occurring more than once [24]. Emerging pattern mining task is defined as patterns whose frequencies in two classes differ by large ratio. While all of these research areas aim at discovering patterns in the form of rules induced from labeled data, they solve different problems with the usage of different terminology, task definitions and techniques [36]. They all aim at optimiz-

ing a trade off between rule coverage and precision. The main difference between these techniques is that while subgroup discovery task attemps to describe unusual distributions in the search space with respect a value of the target variable, contrast set and emerging pattern tasks seek relationships of the data with respect to the possible values of the target variable [24].

2.2 Rule quality

Let us first consider classification problem with only two classes [14]. Each instance I is mapped to one element of a set $\{p, n\}$ of positive and negative class labels. A classifier maps instances to predicted classes. To separate actual class from predicted class we use labels $\{p', n'\}$ for the class predictions produced by the model. Given an classifier and an instance, there are four possible outcomes that are described in following list.

- *True positive (TP)*: an instance is positive and it is classified as positive.
- *False negative (FN)*: an instance is positive, but it is classified as negative.
- *True negative (TN)*: an instance is negative and it is classified as negative.
- *False positive (FP)*: an instance is negative, but it is classified as positive.

In Figure 2.1 is shown a *confusion matrix*, that is a two-by-two matrix that represents dispositions of the set of instances. On the Y axis there are actual class values and on X axis there are predicted classifications. Confusion matrix forms the basis for many common metrics. *Pos* is the total number of positive instances and it is the sum of true positives and false negatives. *Neg* is the total number of negative instances and it is a sum of false positives and true negatives.

Each rule describing a subgroup can be extended with the information about the *rule quality*. A standard rule describing a subgroup has the following form [28].

$$Class \leftarrow Cond[TPr, FPr]$$

Where Class is the target property of interest and Cond is a conjunction of attribute-values. *TPr* is the *true positive rate* or the *sensitivity* and it is computed as follows:

$$TPr = p(Cond \mid Class) = \frac{n(Class \cdot Cond)}{Pos}.$$

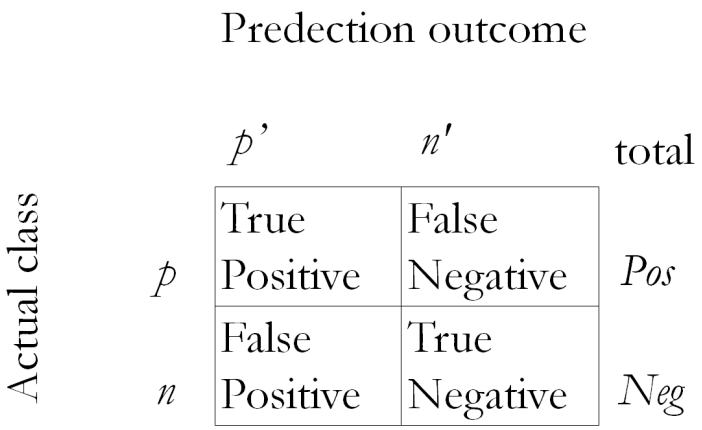

Figure 2.1 *Confusion matrix [14]*

In the formula $n(Class \cdot Cond)$ is the number of true positives and Pos is the number of positives instances in the target class. FPr is the *false positive rate* or the *false alarm* and it is computed as follows:

$$FPr = p(Cond \mid \overline{Class}) = \frac{n(\overline{Class} \cdot Cond)}{Neg}.$$

In the formula $n(\overline{Class} \cdot Cond)$ is the number of false positives and Neg is the number of negatives instances in the target class. $N = Pos + Neg$ is the size of the entire population.

2.3 ROC analysis for subgroup discovery

A receiver operating characteristics (ROC) graph is a technique for visualizing, organizing, and selecting classifiers based on their performance [14]. ROC graphs are two-dimensional graphs which have true positive rate on the Y axis and false positive rate on the X axis. A ROC graph depicts relative trade offs between benefits and costs.

ROC graphs have been used in signal detection theory and ROC analysis has been extended for use in visualizing and analysing the behavior of diagnostic systems. ROC graphs are used in machine learning because simple classification accuracy is often a poor metric for measuring performance [38]. ROC analysis has properties that make it especially useful for domain with skewed class distribution and unequal

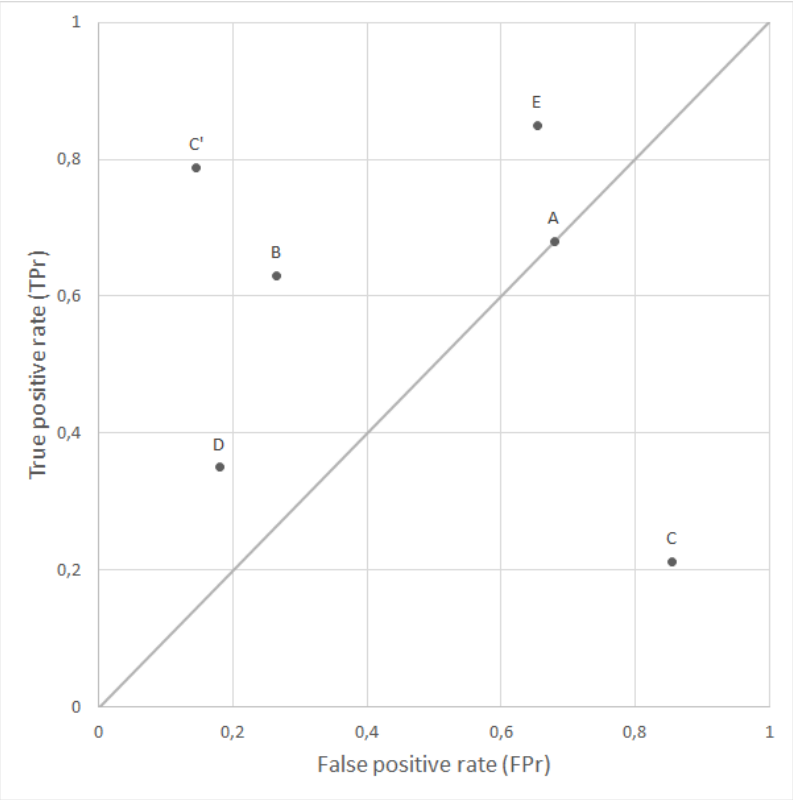

Figure 2.2 *A basic ROC graph with six discrete classifiers.*

classification error costs [14]. These characteristics are important in research areas of cost-sensitive learning and learning in the presence of unbalanced classes.

Classifiers like a decision trees and rule sets, are designed to produce only a class decision, a yes or no for each instance. When such *discrete* classifier is applied to a test set, it yields a single confusion matrix that is a (TPr, FPr) pair. This pair which corresponds to a single point in a ROC space. The rule sets that are outcome of subgroup discovery are discrete classifiers [26]. The classifiers in Figure 2.2 are all discrete classifiers [14].

There are several points in a ROC space that are important to understand. The point (0,0) represents the strategy of never issuing a positive classification. This kind of classifier commits no false positives but also gains no true positives. The point (1,1) represent the opposite strategy of unconditionally issuing positive classifications. The point (0,1) represent the perfect classification which have only true positives and no false positives instances. The point (1,0) represent the negation of the perfect

classification which have no true positives and only false positives.

When points in a ROC space are compared, one point is better than other if it is to the northwest of the first. This means that TPr is higher, FPr is lower or both. A classifier may be thought of as *conservative* if it is located on the left-hand side of a ROC graph near the X axis. Conservative rules make positive classifications only with strong evidence. This means that they make few false positives errors but they also have low true positives rates. A classifier may be thought of as *liberal* if it is located on the upper right-hand side of a ROC space. Liberal rules make positive classifications with weak evidence. This means that they classify nearly all positives correctly, but they often have high false positive rates. In Figure 2.2 the point D is more conservative than the point B and E is more liberal than B and D.

The diagonal line of a ROC graph, y is equal to x, represents the strategy of randomly guessing the class. If a classifier randomly guesses the positive half of the time and half the negatives correct, it yields point in the ROC space that is located on the diagonal line. The location of the point on the diagonal line is based on the frequency which it guesses the positive class. In order to get away from diagonal line to the upper left triangle region, the classifier must exploit some information in the data. In Figure 2.2 A's performance is virtually random. At the point (0.68, 0.68), A may be said to be guessing the positive class 68% of the time.

A classifier that appears below the diagonal line performs worse than random guessing. Any classifier that produces a point in the lower right triangle can be negated to produce a point in the upper left triangle by reversing its classification decisions on every instance. In Figure 2.2 the point C is below diagonal line and it performs much worse than the random. The point C' that is above the diagonal line is negated C.

Any classifier on the diagonal may be said to have no information about the class. A classifier below the diagonal line may be said to have useful information, but it is applying the information incorrectly [17].

2.4 Quality measurements

A most significant factor in the quality of any subgroup discovery algorithm is the quality measure to be used both to select the rules and to evaluate the results

of the process [39]. Each rule describing a subgroup can be extended with the information about the rule quality [28]. The basic information about the rule quality is usually attached to the induced rule itself. In order to enable the comparison of the performance of different algorithms and other quality measures are computed separately as output of the learning algorithm.

Quality measures can be divided to categories, *objective quality measures* and *subjective measures of interestingness* [41]. Both the objective and subjective measures should be considered to solve subgroup discovery tasks [28]. The following list introduces subjective measures of interestingness.

- *Usefulness* is an aspect of rule interestingness which relates to finding the goals of the user [27].

- *Actionability* means that a rule is interesting if the user can do something with it to his or her advantage [37, 41]. Actionable is an important subjective measure of interestingness because users are most interested in the knowledge that they can benefit. Actionability is a special case of usefulness.

- *Operationality* is a special case of actionability, that enables performing an action which can operate on the target population [28]. It is the most valuable form of induced knowledge because if an operational rule is effectively executed, this operation can affect the target population and change the rule coverage.

- *Unexpectedness* means that a rule is interesting if it is "surprising" to the user [41]. Unexpected rules are interesting because they contradict expectations which depend on the system of beliefs.

- *Novelty* means a finding is interesting if it deviates from prior knowledge of the user [27].

- *Redundancy* amounts to the similarity of a finding with respect to other findings [27].

Predictive accuracy of a rule set is a typical predictive quality measure and it is defined as percentage of correctly predicted instances [28]. Descriptive quality measure evaluates each individual subgroup and is thus appropriate for evaluating the success of subgroup discovery.

The following *objective quality measurements* are also descriptive quality measures and they turn out to be most appropriate for measuring the quality of individual rules. The *coverage* is measure of *generality*, computed as the relative frequency of all the examples covered by the rule [28]. Coverage for rule R^i is defined as follows:

$$Cov(R^i) = Cov(Class_j \leftarrow Cond^i) = p(Cond^i) = \frac{n(Cond^i)}{n_s},$$

where $n(Cond^i)$ is the number of examples which verify the condition $Cond^i$ described in the antecedent, and n_s is the number of examples [39]. The *Support* is computed as the relative frequency of correctly classified covered examples [28]. Support is calculated with the following formula:

$$Sup(Class_j \leftarrow Cond^i) = p(Class_j.Cond^i) = \frac{n(Class_j.Cond^i)}{n_s},$$

where $p(Class_j.Cond^i)$ is the number of examples which satisfy the conditions for the antecedent $Cond^i$ and also belong to the value for the target variable *Class* indicated in the consequent part of the rule [39].

The *Size* of the set of rules is a complexity measure calculated as the number of introduced rules n_r [39]. Another way to measure complexity is to measure it as the mean number of rules obtained for each class, or the mean of variables per rule. The *Accuracy* is the fraction of predicted positives that are true positives [28]. Rule accuracy is called *precision* in information retrieval and *confidence* in association rule learning. Rule accuracy is computed as follows [39]:

$$Acc(Class_j \leftarrow Cond^i) = p(Class_j \mid Cond^i) = \frac{n(Class_j.Cond^i)}{n(Cond^i)}.$$

The *Significance* is measured in terms of likelihood ratio static of a rule [28] and it is defined as follows:

$$Sig(Class \leftarrow Cond^i) = 2 \cdot \sum_{j=1}^{n_c} n(Class_j.Cond^i) \cdot \log \frac{n(Class_j.Cond^i)}{n(Class_j) \cdot p(Cond^i)},$$

where $p(Cond^i) = n(Cond^i)/n_s$ is used as normalizing factor [39]. Although each rule is for a specific class value, the significance measures the novelty in the distribution impartially, for all class values.

The *Unusualness* of a rule is computed by the *weighted relative accuracy* of a rule and it is defined as follows [30]:

$$WRAcc(Class_j \leftarrow Cond^i) = p(Cond^i) \cdot [p(Class_j \mid Cond^i) - p(Class_j)].$$

The weighted relative accuracy of a rule can be described as the balance between the coverage of the rule $p(Cond^i)$ and accuracy gain $p(Class_j|Cond^i) - p(Class_j)$. WRAcc is appropriate for measuring the unusualness of separate subgroups, because it is proportional to the vertical distance of the subgroup to the ascending diagonal in a ROC space [28]. It also reflects rule significance, larger WRAcc means more significant rule. These are the most important quality measures for subgroup discovery. In addition to significance, WRAcc takes the rule coverage in to account. WRAcc heuristic can be used in the search of optimal subgroups and evaluating the quality of the introduced subgroup descriptions.

The measures for evaluating each individual rule can be complemented by their variants that compute the average over induced set of subgroup descriptions [28]. Average quality measures for the set of rules are as calculated as sum of measures divided by the number of introduced rules. For example the *average coverage for the set of rules (COV)* is calculated as

$$COV = \frac{1}{n_r} \sum_{i=1}^{n_r} Cov(R^i),$$

where n_r is the number of induced rules [39].

The goal of subgroup discovery is to find subgroups of object relation that are unusual distributional characteristics respect to the entire group or population [27, 42]. If we want to find k best subgroups, we need to measure quality of candidate groups. Since the interestingness of a group depends on its unusualness and size, evaluation function needs to combine both of these factors.

2.5 Techniques for subgroup discovery

In subgroup discovery the search space grows exponentially according all the possible selectors of a subgroup description [5]. This is why it is important to have techniques to reduce the search space. Another important goal is to improve the quality of discovered subgroups. This section introduces pruning, different search heuristics,

and set selection.

2.5.1 Top-k pruning

As the result of subgroup discovery, the applied subgroup discovery algorithm returns a result set containing subgroups [1]. That result set can contain subgroups that are above a certain minimal quality threshold, or are included on the top-k subgroups, that can be postprocessed further. The top-k approach is more flexible for applying different pruning options in the subgroup discovery process.

The set of the top-k subgroups is determined according to a given quality function in a top-k setting. After this different pruning strategies can be applied for restricting the search space of a subgroup discovery algorithm. A simple option is given by *minimal support pruning* based on antimonotone constraint of the subgroup size. The principle beyond *minimal support pruning* goes as follows, since we are not interested in solutions that cover too few members of the population, as soon as we reach a hypothesis that fails to cover that many elements, we can prune the entire subtree rooted at this hypothesis [42]. More powerful approaches are enabled by properties of certain quality functions [1].

Optimistic estimates can be applied for determining upper quality bounds for several quality functions. In the search of the k best subgroups, if it can be proven that no subset of currently investigated hypothesis is interesting enough to be included in the result set of top-k subgroups, then the evaluation of any subsets of this hypothesis can be skipped, but still the optimality of the result can be guaranteed. The basic principle of optimistic estimates is to safely prune parts to the search space and it was first proposed for binary target variables. The idea exploits the fact that only top-k subgroups are interesting [42]. If the k best hypotheses so far have already been obtained, and the optimistic estimate of the current subgroup is below the quality of the worst subgroup contained in the k best subgroups, then the current branch of the search space tree can safely be pruned [1].

Generalization-aware pruning is a pruning mechanism that estimates the quality of the subgroup against the qualities of its generations. A pattern can be compared to its generalizations in order fulfill minimal improvement constraint, such that subgroups with a lower target share than its generalizations are removed.

2.5.2 Search heuristics

Exhaustive evaluation of the candidate rules allows the best subgroup to be found, but when the search space becomes too large this is not affordable [24]. A heuristic search can be used to reduce the number of potential subgroups. In heuristic approaches a *beam search* strategy is commonly used because of its efficiency [1]. The search starts with a list of subgroup hypotheses of size w, corresponding to the *beam width*. The list can initially be empty. The w subgroup hypotheses contained in the beam are expanded iteratively, and only the best w expanded subgroups are kept implementing a hill-climbing greedy search. Beam search traverses the search space non-exhaustively and does not guarantee to discover the complete set of the k best subgroups, or all subgroups above a minimal quality threshold. It can be regarded as a variant of an anytime algorithm, since the search process can be stopped at any point such that the currently best subgroups are available.

Exhaustive approaches guarantee to discover the best solutions. The downside is that the runtime of a naive exhaustive algorithm usually prohibits its application for larger search spaces [1]. Depending on the applied algorithm, there are different pruning options that can be used for the subgroups discovery task. Many advanced algorithms apply extensions of frequent pattern trees (FP-trees) in a pattern-growth fashion. Also optimistic estimate pruning is applied, while generalization-aware pruning is better supported by layer-wise algorithms.

2.5.3 Set selection

Subgroup set selection is one of the critical issues for removing redundancy and improving the interestingness of the overall subgroup discovery result [1]. Constrains denoting redundancy filters can be used to prune large regions of search space. This is especially important for search strategies which do not constrain the search space. There are logical and heuristic redundancy filters [27]. According to their types, the filters include either logical or heuristic implications for the truth value of a constraint condition with respect to a predecessor/successor pair of subgroups [1]. Logical filters can be used as *strong filters*, since they can definitely exclude a region of the search space. Heuristic filter are *weak filters*, since they are applied as a first step in a brute force search, where the excluded regions of the search space can be determined later.

Condensed representations of frequent item sets have been developed for reducing the size of the association rules that are generated. These representations are used for redundancy management, since condensed patterns describe the specifically interesting patterns, and can significantly reduce the size of the result sets. For subgroup discovery target-closed representations can be formalized and this way perform an implicit redundancy management based on the subgroup descriptions.

Since often a set of very similar, overlapping subgroup patterns is retrieved, methods for extracting a set of relevant subgroups are required [33]. For redundancy management of subgroups for binary targets, the *(ir-)relevance* of subgroup with respect to a set of subgroups is quite simple method [1]. It is defined as follows. A subgroup hypothesis S_N is *irrelevant* if there exist subgroup hypothesis S_P such that the true positives of S_N are a subset of true positives of S_P and the false positives of S_N are a subset of false positives of S_P. This redundancy management technique can be embedded to the search process testing relevancy when a subgroup hypothesis is considered in to the set of k best subgroups.

A subgroup set can be selected according to its overall coverage of the dataset. The *weighted covering algorithm* is such an approach that works by example reweighting. On the subgroup selection method, it iteratively focuses on the space of the target records not covered so far, by reducing the weight of the already covered data records. Reweighting can also be used as search heuristic, with a combination of suitable quality function.

2.6 Subgroup discovery algorithms

In this section different algorithms for solving subgroup discovery task are introduced. First the pioneering algorithms are introduced and then there are different sections for algorithms based on classification and association rule learners and for evolutionary algorithms.

2.6.1 The pioneering algorithms

The first algorithms for subgroup discovery are extensions of classification algorithms and they use decision trees [24]. They can employ exhaustive and heuristic strategies for search and several quality functions to evaluate the quality of subgroups.

EXPLORA [27] was the first algorithm for subgroup discovery task and it was introduced by Klösgen in 1996. EXPLORA treats the learning as a single relation problem [26]. This means that all the data is assumed to be available in one relation. The algorithm uses decision trees for the extraction of rules [39]. The rules are specified by first defining a descriptive scheme and then by implementing a statical verification method. The interestingness of the rule is measured using criteria such as evidence, generality, redundancy, and simplicity. EXPLORA can apply exhaustive and heuristic subgroups discovery strategies without pruning [24].

MIDOS [42] was introduced by Wrobel in 1997 and it applied subgroup discovery task for multiple relational tables. MIDOS algorithm uses optimistic estimation and minimal support pruning, an optimal refinement operator and sampling to ensure efficiency and easy parallel use. The quality measure of MIDOS is a combination of unusualness and size.

2.6.2 Algorithms based on classification rule learners

Several subgroup discovery algorithms have been developed by adapting classification rule learners [24]. Some modifications must be implemented since the objective of classification rule learning differs from the objective of subgroup discovery as shown in Section 2.1. The following algorithms use a modified weighted covering algorithm and introduce example weights to modify the search heuristic.

SubgroupMiner [39] is an extension of EXPLORA and MIDOS. It enables usage of very large databases by efficient database integration, multirelational hypotheses, visualization-based interaction options, and discovery of causal subgroup structures [26]. It uses interactive beam search and it is the first algorithm which considers the usage of numeric target variables [24]. SubgroupMiner uses significance as the quality function to rank rules during the beam search and a special post-processing approach to eliminate redundant subgroups [26]. SubgroupMiner also uses the classical binomial test to verify whether statistical distribution of the target is significantly different in the extracted subgroup compared to the entire population.

SD algorithm [19] is a rule induction system guided by expert knowledge. SD does not define an optimal measure for automated subgroup search and selection. The goal of SD is to support the expert in performing flexible and effective search of broad range of optimal solutions. Thus, the decision of the subgroups in the final

solution is left to the expert. Targets of SD algorithm are to have sufficiently large coverage and a positive bias towards target class coverage, sufficiently diverse for detecting of the population and to fulfill experts subjective measures of acceptance which are understandability, simplicity and actionability.

CN2-SD [31] is a modified version of CN2 classification rule learning algorithm. CN2-SD uses weighted covering algorithm for ruleset construction [26]. CN2-SD induces the subgroups in the form of rules using the relation between true positives and false positives as quality measure [39].

RSD [32] is a relational subgroup discovery algorithm. RSD algorithm performs a simple form of predictive invention through first-order feature construction and use the constructed features for relational rule learning. The approach of RSD algorithm is to use a first-order feature construction that can be applied individual-centered domains. This means that there is a clear notion of individuals and learning occurs at the level of individuals only. RSD algorithm uses weighted covering algorithm.

CN2-SD and RSD uses the unusualness as heuristics, while SD uses the generalisation quotient [24]. SD and CN2-SD are both propositional, while RSD is a relational subgroup discovery algorithm.

2.6.3 Algorithms based on association rule learners

The objective of an association rule algorithm is to obtain relations between variables of the dataset [24]. In it several variables can appear both in the antecedent and consequent of the rule. In subgroup discovery the consequent of the rule consisting the property of interest is prefixed.

APRIORI-SD [26] was developed by modifying APRIORI-C algorithm. The modifications involved the implementation of example weighting scheme in rule post-processing, a modified rule quality function incorporating example weights into the weighted relative accuracy heuristic, a probabilistic classification scheme, and the use of the ROC space for improving the evaluation of discovered rules. APRIORI-SD produces smaller rulesets, where individual rules have higher coverage, significance, and unusualness.

SD4TS [35] is test selection based subgroup discovery algorithm and it is based on APRIORI-SD. The object of SD4TS algorithm is to find individuals which are

sharing the same optimal test. One application is used it for identifying subgroups of patients for which the optimal test for breast cancer diagnosis is the same. SD4TS uses cost-sensitive variant prediction quality, which corresponds to the benefits of the prediction rather than to its costs.

SD-Map [4] is an exhaustive subgroup discovery algorithm. SD-Map is based on FP-growth method and it computes subgroup quality directly without referring to other intermediate results by using modified FP-growth step [39]. SD-Map$\star$ algorithm [3] extends SD-Map by including optional strategies, utilizes quality functions with tight optimistic estimates and can handle continuous target variables directly without discretization.

DpSubgroup is a algorithm that uses optimistic estimates for pruning [22]. DpSubgroup [21] is generic pruning algorithm which is similar to the SD-Map. The main difference is that DpSubgroup algorithm incorporates pruning and provides a generic hook for optimistic estimates by a function. The algorithm makes double use of the optimistic estimates. First, the terms with an insufficient estimate are not considered for recursion and second, these terms are omitted in the construction of conditional FP-trees, which results in smaller memory requirements.

Merge-SD [20] is subgroup discovery algorithm which can handle numerical variables, but it can be also applied to ordinal attributes. Merge-SD prunes large parts of the search space by exploiting bounds between related numerical subgroup descriptions. It performs a depth-first-search in the space of subgroup descriptions and in each recursive step it checks all combinations endpoints.

BSD [33] is a subgroup discovery algorithm based on a vertical data structure, that also integrates efficient filtering for overlapping subgroups. It is tailored to the task of discovering relevant subgroup patterns. BSD algorithm combines a vertical *bitbased* representation of the information with advanced pruning strategies and efficient relevancy check. Bitsets are implemented time and memory efficiently using logical operators like OR and AND. BSD uses a branch-and-bound strategy, where a conditioned search space is mined recursively, similar to the SD-Map$\star$ and the DpSupgroup algorithms. There is an extension that enable parallelization of the search in multiple processes in order to distribute discovery effort and gain performance.

Some of described algorithms like APRIORI-SD and SD4TS are adapted from as-

sociation rule learning algorithm APRIORI, but others like SD-MAP, DpSubgroup, and Merge-SD are adaptations of FP-Growth [24]. FP-growth it similar to APRIORI, but it has a feature of avoiding multiple scans of database for testing each frequent pattern. Instead, it applies a recursive divide-and-conquer technique [4]. All of these algorithms use decision trees for representation [24].

2.6.4 Evolutionary algorithm for extracting subgroups

Subgroup discovery task can be approached and solved as optimization and search problem [24]. Evolutionary algorithms imitate the principles of natural evolution in order to form processes for searching. They utilize the collective learning process of a population of individuals and means of evaluating individuals in their environment, a measure of quality or fitness value can be assigned to individuals [6]. *Genetic algorithms (GAs)* are one of the most widely used evolutionary algorithms [24]. They are search algorithms based on natural genetics that provide robust search capabilities in complex spaces [11]. The heuristic used by them is defined by a fitness function. A fitness function determinates which individuals, or rules in the case of subgroup discovery task, will be selected to form part of the new population in competition process.

SDIGA [11] is an evolutionary fuzzy rule induction system. SDIGA uses linguistic rules as description language to specify the subgroups [39]. For rule learning SDIGA uses *iterative rule-learning (IRL)* approach, in which each chromosome represent a rule, but the GA solution and the global solution is formed by the best individuals obtained when [11] algorithm is run multiple times [11]. For rule quality measure SDIGA uses weighted sum of confidence and support. SDIGA uses DNF fuzzy rules.

MESDIF [7] is a *multiobjective genetic algorithm (MOEA)* which obtains fuzzy rules for subgroup discovery in disjunctive normal form. The objectives of MESDIF are to use a restriction in the rules in order to obtain a set of rules called as *the Pareto front* with high degree of coverage and take into account the support and the confidence of the rules. The MOEA of MESDIF is based on the SPEA2 approach and it uses DNF fuzzy rules.

NMEEF-SD [8] is a non-dominated MOEA for extracting fuzzy rules in subgroup discovery. It is a evolutionary fuzzy system based on NSGA-II model. NMEEF-SD is oriented toward the subgroup discovery task using special operators to promote the

extraction of interpretable and high quality subgroup rules. The quality measures considered as objectives in the evolutionary process can be support, fuzzy confidence, or unusualness. There is a post-processing tuning step proposed to improve the results of subgroup discovery algorithm NMEEF-SD by allowing the partitions to be adapted to the context of the variables [9].

Evolutionary algorithms for subgroup discovery are based on a evolutionary fuzzy systems, which are hybridisation between fuzzy logic and evolutionary algorithms [24]. A fuzzy is a approach in subgroup discovery which considers linguistic variables with linguistic terms in descriptive fuzzy rules that allows obtain knowledge in similar way to human reasoning [11]. Fuzzy rules enables representing the knowledge about patterns of interest in an explanatory and understandable form which can be used by the expert [7]. DNF fuzzy rules contribute a flexible structure to the rules, allowing each variable to take more than one value, and facilitating the extraction of more general rules. Evolutionary fuzzy systems provide novel and useful tools for pattern analysis and for extracting new kinds of useful information. They are especially useful in domains where the boundaries of a piece of information used may not be clearly defined [24]. The evolutionary algorithm allows the inclusion of quality measures in order to obtain rules with suitable values for both selected and other quality measures. The best approach to obtain solutions with good compromise between the quality measures for subgroup discovery is to use a multi-objective evolutionary algorithm. MOEAs combines the approximated reasoning method of fuzzy systems with the learning capabilities of genetic algorithms.

3. DISCRETIZATION OF CONTINUOUS TARGET ATTRIBUTES

The target attributes of subgroup discovery may be *nominal*, or they can be *continuous* [15]. The term continuous refers to attributes taking on numerical values; or in general an attribute with a linearly ordered range of values. Many subgroup discovery algorithms can handle only binary target attributes and continuous target attributes must be *discretized* before using these algorithms [34]. Continuous-valued attributes are discretized prior to selection, typically by partitioning the range of the attribute into subranges [15]. A discretization is *logical* condition, in terms of one or more attributes that serves to partition the data into subsets. There are the following three main goals of target attribute discretization [34].

- Clusters should be densely populated since then they are likely to represent similar cases.
- Clusters should be clearly distinct since two clusters located close may actually correspond to a similar target group.
- Isolated points that do not convincingly fall into a cluster should be efficiently skipped since they are unlikely part of a interesting target group.

Equal width interval binning is the simplest method to discretize data and has often been applied as means of producing nominal values from continuous ones [12]. It involves sorting the observed values of a continuous feature and dividing the range of observed values for the variable into k equal sized bins, where a k is a parameter given by user. If a variable x is observed to have values bounded by x_{min} and x_{max} then the following formula computes bin width:

$$\delta = \frac{x_{max} - x_{min}}{k}.$$

It also constructs bin boundaries, or *thresholds*, at $x_{min} + i\delta$ where $i = 1, ..., k - 1$.

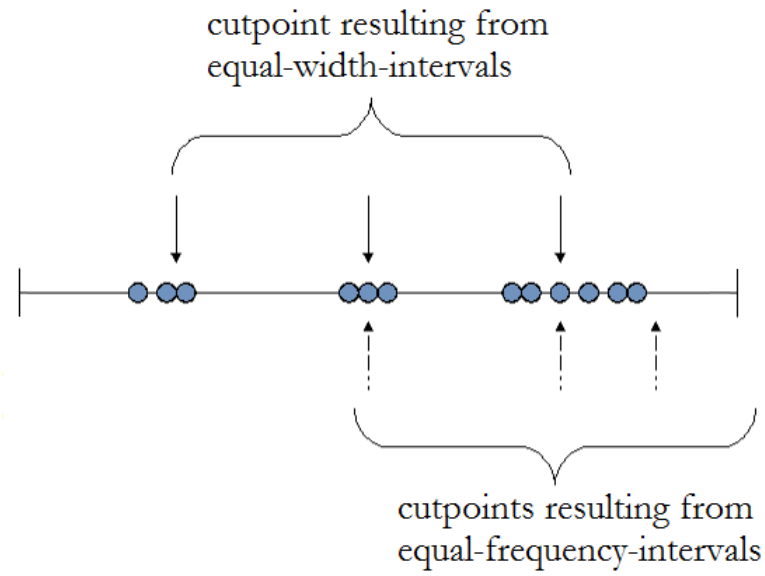

Figure 3.1 *Cutpoints from equal width and equal frequency intervals [34]*

The method is applied each continuous feature independently. It makes no use of the class information of the instance and is thus unsupervised method. This type of discretization is vulnerable to outliers that may drastically skew the range.

Equal frequency intervals divides a continuous variable into k bins where given m instances each bin contains m/k adjacent values. In Figure 3.1 there is an interval of some continuous attribute and points on the interval represent values of instances on that attribute. On top of the interval cutpoints resulting of equal width intervals are shown by downwards arrows pointing to the interval. On the bottom of the interval cutpoints resulting from equal frequency intervals are shown by upwards arrows pointing for the interval. Both of these methods identify clusters with lower density that are located very close to neighboring clusters [34]. These approaches do not satisfy the third goal of target discretization since they assign all points to clusters with exception of outliers. There are methods that solve to achieve all three goals of target discretization, but they use more complex techniques like for example dynamic programming approach.

In *binary discretization* the range of continuous-valued attribute is discretized by dividing it in two intervals [15]. It is used during decision tree generation. Threshold value T for continuous-valued A is determined, and the test $A \leq T$ is designed to left branch while $A > T$ is designed to the right branch. Threshold value T is called

a cutpoint.

Supervised learning methods utilize the class labels [12]. Equal width interval binning, equal frequency intervals, and binary discretization are unsupervised discretization methods. Holte's *1R Discretizer* [25] is an example of supervised discretization method. It is a simple classifier that induces one-level decision trees [12]. In order to properly deal with domain that contains continuous valued features, a simple supervised discretization method *1RD (One-Rule Discretizer)* is given. 1RD sorts the observed values of a continuous feature and attempts to greedily divide the domain of the feature into bins that each contain only instances of one particular class. Since such a scheme could lead to one bin for each observed real value, algorithm is constrained to forms bins of at least some minimum size. Each discretization interval is made as "pure" as possible by selecting cutpoints such that moving partition boundary to add an observed value to particular bin cannot make the count of the dominant class in that bin greater.

The standard approach to replace every numeric value with by a single nominal causes that subsequent subgroup discovery will typically find only suboptimal subgroup descriptions as only subset of all valid features are preserved [20]. Straight forward discretization does not take into account overlapping intervals which is one of the reasons that it finds only suboptimal subgroups. There is Fayyad-Irani algorithm for discretization with multiple interval [15]. There are also algorithms which support continuous variables for example SD-Map$\star$ [2]. Using continuous variables without discretization is slower. It demands more complex structure and the time consumption grows with every numerical variable. The advantages of discretization is that it is fast compered to more complex solutions and it can be used with any subgroup discovery algorithm.

4. INITIALIZATIONS FOR SUBGROUP DISCOVERY

In this chapter the solutions and setups to do the discretizations and the actual subgroups discovery searches are covered. First the tools and settings used for the subgroup discovery task are introduced. Then the used two datasets are presented in their own sections which hold the information about attribute types and distributions. The used discretizations of the presented datasets are shown in the last section of the chapter.

4.1 Used software and settings

Subgroup discovery searches where executed in R 3.2.3 environment with rsubgroup 0.6 extension package. RStudio version 0.99.491 was used for a graphical user interface to improve usability.

For subgroup discovery tasks the following settings where applied. BSD algorithm with WRAcc quality function was used for solving tasks. Description of BSD algorithm is in Section 2.6.3 and WRAcc in Section 2.4. The maximum number of patterns to discover was set to be ten. The maximum number of conjunctions was set to be five. Irrelevant patterns were filtered during pattern mining. For the post-processing minimum improvement filter was used for checking the patterns against all possible generations.

Two different datasets were used and both of them are from UCI Machine Learning Repository. For the heart disease dataset the target of subgroup discovery task was attribute *cp* with value 4. In other words rules where extracted for asymptomatic chest pain type. Attributes used for the tasks were all other attributes except *cp* and *class*. For the Australian credit card dataset the target of subgroup discovery task was attribute *A4* with value 2. Attributes used for the tasks were all other attributes except *A4* and *Class*.

4.2 Description of the heart disease dataset

The heart disease dataset has 270 instances and no missing values. Attributes of the dataset are described as follows. The amounts of the instances for the values of non-continuous attributes are marked inside brackets after the descriptions.

Binary attributes:

- *sex* of the patient. (0: female (87), 1: male (183))
- *fbs* is fasting blood sugar is greater than 120 mg/dl.
  (0: false (230), 1: true (40))
- *exang* is exercise induced angina. (0: no (181), 1: yes (89))
- *class* is a predicted class value (1: 150. 2: 120).

Nominal attributes:

- *cp* is a chest pain type. Possible values are as follows.
  1: typical angina (20)
  2: atypical angina (42)
  3: non-anginal pain (79)
  4: asymptomatic (129)
- *restecg* resting electrocardiographic results
  0: normal (131)
  1: having ST-T wave abnormality, T wave inversions and/or ST elevation or depression of > 0.05 mV (2)
  2: showing probable or definite left ventricular hypertrophy by Estes' criteria (137)
- *slope (ordered)* is the slope of the peak exercise ST segment.
  1: upsloping (130)
  2: flat (122)
  3: downsloping (18)
- *thal* Possible values are as follows.
  3: normal (152)
  6: fixed defect (14)
  7: reversable defect (104)

Numeric attributes:

- *age* is an age in years.
- *trestbps* resting blood pressure (in mm Hg on admission to the hospital).
- *chol* serum cholestoral in mg/dl.
- *thalach* is the maximum heart rate achieved.
- *oldpeak* ST depression induced by exercise relative to rest.
- *ca* number of major vessels (0 – 3) colored by flourosopy.

4.3 Description of the Australian credit approval dataset

The second dataset is the Australian credit approval involves credit card applications. It consists of 690 instances with no missing values. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. Attributes of the dataset are described as follows.

Binary attributes:

– A1: {0, 1} (0: 222, 1: 468)

– A8: {0, 1} (0: 329, 1: 361)

– A9: {0, 1} (0: 395, 1: 295)

– A11: {0, 1} (0: 374, 1: 316)

– Class: {0, 1} (0: 383, 1: 307)

Nominal attributes:

– A4: {1-3} (1: 163, 2: 525, 3: 2)

– A5: {1-14} (1: 53, 2: 30, 3: 59, 4: 51, 5: 10, 6: 54, 7: 38, 8: 146, 9: 64, 10: 25, 11: 78, 12: 3, 13: 41, 14: 38)

– A6: {1-9} (1: 57, 2: 6, 3: 8, 4: 408, 5: 59, 6: 0, 7: 6, 8: 138, 9: 8)

– A12: {1-3} (1: 57, 2: 625, 3: 8)

Numeric attributes: A2, A3, A7, A10, A13, A14

4.4 Discretizations of the datasets

In the heart disease dataset discretization is done for numerical attributes *age, trestbps, chol, thalach, oldpeak* and *ca*. The dataset is discretized the following ways.

- Equal interval width
- Equal interval width with removing unnecessary bins
- Equal frequency intervals
- Binary discretization

For the heart disease dataset each discretization is done with bin amounts three, five and ten. The Australian credit dataset is discretized with equal interval width with removing unnecessary bins, equal frequency intervals and binary discretization. For the Australian credit dataset the used bin amount for all its discretizations is five. There are twelve discretized datasets in for the heart disease dataset and three for the Australian credit dataset. All discretizations are done with data mining software Weka version 3.6.13 [23].

The following sections go through each discretization ordered under the way they are discretized. For each way there are a figure and a table containing the distributions of discretization the heart disease dataset with bin amount three. Tables of distributions with bin amount five and ten of the heart disease dataset can be found from Appendix A. For all other discretizations, except the discretization with equal interval width, there is a table containing distribution of the Australian credit approval discretization. The bin amount for discretizations of the Australian credit approval dataset is five.

4.4.1 Discretization with equal interval width

Figure 4.1 gives the distribution of attributes of the heart disease dataset that are discretized with three equal interval width (*3BinsDis_heart*). It can be seen on the figure that the distributions of the values differs much from each other.

In Table 4.1 are the distributions of the dicretized attributes of *3BinsDis_heart*. From the table the exact amounts that instances have value on the attribute can be seen. Attribute *chol* has value (-∞, 272.00] in 192 instances, but value (418.00, ∞) only in one. Because subgroup discovery aims to discover subgroups that are as

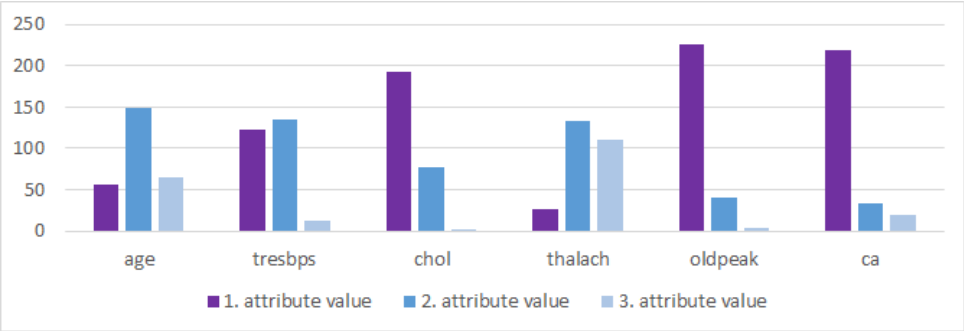

Figure 4.1 *The discretized attributes of the 3BinsDis_heart.*

large as possible, it is obvious that values like (418.00, ∞) are not going to be in any subgroup description.

Table 4.1 *The discretized attributes of the 3BinsDis_heart.*

age	tresbps	chol
(-∞, 45.00]: 56	(-∞, 129.33]: 123	(-∞, 272.00]: 192
(45.00, 61.00]: 149	(129.33, 164.67]: 135	(272.00, 418.00]: 77
(61.00, ∞): 65	(164.67, ∞): 12	(418.00, ∞): 1
thalach	**oldpeak**	**ca**
(-∞, 114.67]: 26	(-∞, 2.07]: 226	(-∞, 1.00]: 218
(114.67, 158.33]: 133	(2.07, 4.13]: 40	(1.00, 2.00]: 33
(158.33, ∞): 111	(4.13, ∞): 4	(2.00, ∞): 19

Disadvantages of equal interval division can be seen more clearly when the amount of bins increases. Table A.1 in Appendix A contains the distributions of the dicretized attributes with five equal intervals (*5BinsDis_heart*). Value (1.2, 1.8] of attribute *ca* does not appear in any instance. There are several values that are true only in few instances.

Table A.5 in Appendix A has the distributions of the dicretized attributes with ten equal intervals (*10BinsDis_heart*). In Table A.5 attribute *chol* has two values that do not appear for any instance in dataset. Attribute *oldpeak* has two values and attribute *ca* has six values that does not appear for any instance.

The value intervals which have zero instances do not affect the subgroup discovery

task since they do not appear in the dataset. With discretization with ten bins there is also more values that appear only in few instances. The distributions between values of the attributes are uneven because they have not been taken account while doing the discretization. The biggest disadvantage is that value group that could be part of some high quality subgroup is divided in smaller meaningless intervals.

4.4.2 Discretization with equal interval width with removing unnecessary bins

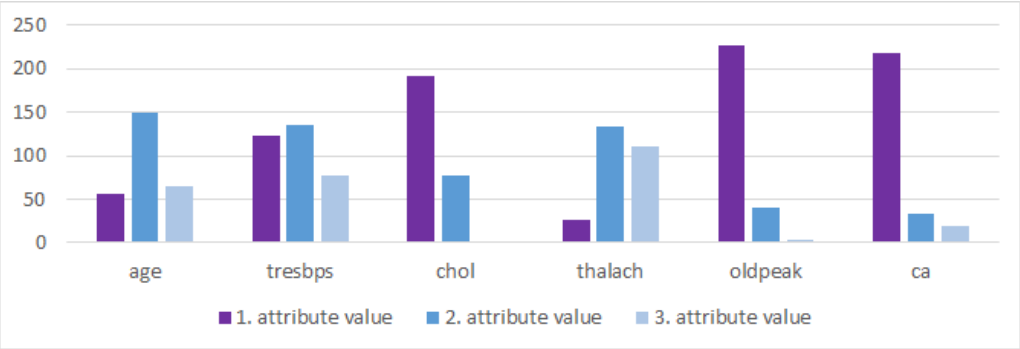

Figure 4.2 *The discretized attributes of the 3findBinsDis_heart.*

Equal interval width with removing unnecessary bins differs from equal interval width by removing those intervals which seem to be unnecessary. Removing unnecessary bins is made by using *findNumBins* option in discretization with Weka. In Figure 4.2 has the discretization distribution with three equal intervals with removal of unnecessary bins (*3findBinsDis_heart*). This discretization differs *3BinsDis_heart* only with *chol* attribute that is divided in to two values instead of three.

Table 4.2 *The discretized attributes of the 3findBinsDis_heart*

age	tresbps	chol
(-∞, 45.00]: 56	(-∞, 129.33]: 123	(-∞, 272.00]: 192
(45.00, 61.00]: 149	(129.33, 164.67]: 135	(272.00, ∞): 78
(61.00, ∞): 65	(164.67, ∞): 12	
thalach	oldpeak	ca
(-∞, 114.67]: 26	(-∞, 2.07]: 226	(-∞, 1.00]: 218
(114.67, 158.33]: 133	(2.07, 4.13]: 40	(1.00, 2.00]: 33
(158.33, ∞): 111	(4.13, ∞): 4	(2.00, ∞): 19

Table 4.2 shows exact distributions of attributes in *3findBinsDis_heart* with value ranges of the nominal attributes. Compered with *3binsDis_heart* in Table 4.1 it can be seen that from attribute *chol* the nominal range (418.00, ∞) that only contains one instant is removed and combined previous nominal range.

The usage of *findNumBins* option can be seen more clearly when the amount of dividable bins is greater. With bin amount five attribute *chol* includes three less and attribute *ca* includes two less values than in discretization without removal of unnecessary bins. With these attributes also numeric value limits differs from discretization with equal width. Table A.2 in Appendix A has discretization distribution with five equal intervals with removal of unnecessary bins (*5findBinsDis_heart*).

Table A.6 in Appendix A has discretization distribution with ten equal intervals with removal of unnecessary bins (*10findBinsDis_heart*). With *findNumBins* option there are no intervals that are empty. There still are values that only few instances belong to and the distribution is uneven.

Table 4.3 *The discretized attributes of the 5findBinsDis_aus.*

A2	A3	A7
(-∞, 1617.80]: 119	(-∞, 5267.00]: 643	(-∞, 2883.00]: 664
(1617.80, 3219.60]: 340	(5267.00, 10534.00]: 25	(2883.00, 5766.00]: 17
(3219.60, 4821.40]: 170	(10534.00, 15801.00]: 17	(5766.00, 8649.00]: 5
(4821.40, 6423.20]: 51	(15801.00, 21068.00]: 2	(8649.00, ∞): 4
(6423.20, ∞): 10	(21068.00, ∞): 3	
A10	**A13**	**A14**
(-∞, 13.40]: 667	(-∞, 400.00]: 634	(-∞, 20001.00]: 685
(13.40, ∞): 23	(400.00, ∞): 56	(20001.00, ∞): 5

Table 4.3 contains distributions of discretization of the Australian credit approval dataset with equal interval witdth and *findNumBins* option (*5findBinsDis_aus*) and the bin amount five. For attribute *A7* there are four different values and for attributes *A10*, *A13* and *A14* there are only two different values. Even for those that have only two values, the distributions are far away from even.

4.4.3 Discretization with equal frequency intervals

In Figure 4.3 is distribution of attributes of the heart disease dataset that are discretized with three equal frequency intervals (*3equalFreqDis_heart*). With equal frequency distribution is more even between values. Only with attribute *ca* there is an obvious difference between first and other values.

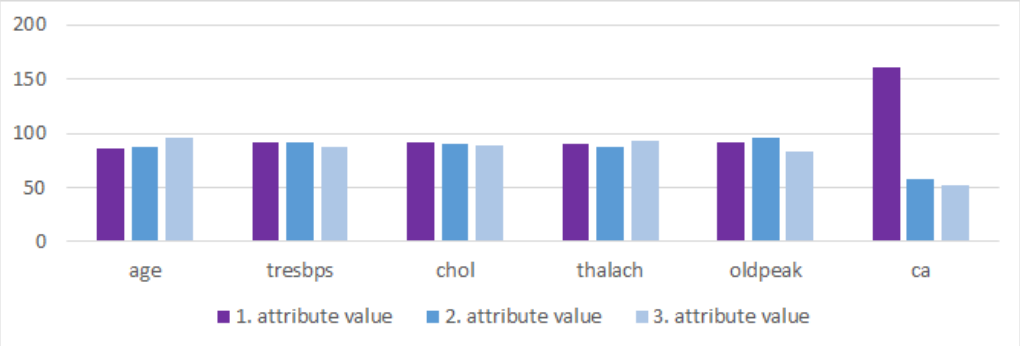

Figure 4.3 *The discretized attributes of the 3equalFreqDis_heart.*

In Table 4.4 are distributions of attributes of the *3equalFreqDis_heart*. As seen in Table 4.4, value (-∞, 0.50] of attribute *ca* has much more occurrences than other values of the attribute combined.

Table 4.4 *The discretized attributes of the 3equalFreqDis_heart*

age	tresbps	chol
(-∞, 50.50]: 86	(-∞, 121.00]: 91	(-∞, 226.50]: 91
(50.50, 58.50]: 88	(121.00, 139.00]: 91	(226.50, 267.50]: 90
(58.50, ∞): 96	(139.00, ∞): 88	(267.50, ∞): 89
thalach	**oldpeak**	**ca**
(-∞, 142.50]: 90	(-∞, 0.15]: 91	(-∞, 0.50]: 160
(142.50, 161.50]: 87	(0.15, 1.45]: 96	(0.50, 1.50]: 58
(161.50, ∞): 93	(1.45, ∞): 83	(1.50, ∞): 52

In Table A.3 are distributions of the dicretized attributes of the heart disease dataset with five bins and equal frequency intervals (*5equalFreqDis_heart*). Attribute *ca* has quite uneven distribution and attribute *oldpeak* has much more occurrences with value (-∞, 0.05] than with other values.

In Table A.7 are distributions of the dicretized attributes of heart disease dataset with ten bins that have equal frequency intervals (*10equalFreqDis_heart*). Attribute *ca* has the same distribution as with five bins that have equal frequency. Attribute *oldpeak* has the same issue as in *5equalFreqDis_heart*. With ten bins the occurrences of values becomes quite small and their probability to be part of subgroup descriptions decreases.

Table 4.5 *The discretized attributes of the 5equalFreqDis_aus.*

A2	A3	A7
(-∞, 1787.50]: 137	(-∞, 10.50]: 138	(-∞, 3.50]: 128
(1787.50, 2321.00]: 137	(10.50, 69.00]: 138	(3.50, 23.00]: 142
(2321.00, 2979.00]: 138	(69.00, 204.50]: 141	(23.00, 90.50]: 150
(2979.00, 3862.50]: 140	(204.50, 1023.00]: 136	(90.50, 385.50]: 139
(3862.50, ∞): 138	(1023.00, ∞): 137	(385.50, ∞): 131
A10	**A13**	**A14**
(-∞, 0.50]: 395	(-∞, 26.00]: 138	(-∞, 1.50]: 295
(0.50, 1.50]: 71	(26.00, 120.50]: 149	(1.50, 22.50]: 99
(1.50, 3.50]: 73	(120.50, 197.50]: 129	(22.50, 294.00]: 99
(3.50, 7.50]: 72	(197.50, 296.00]: 132	(294.00, 1082.00]: 99
(7.50, ∞): 79	(296.00, ∞): 142	(1082.00, ∞): 98

Table 4.5 contains the results of equal frequency discretization of the Australian credit approval dataset with five bins (*5equalFreqDis_aus*). The distributions of the values are quite even, except for the attributes *A10* and *A14*, which both have the first value with much more instances than the others.

4.4.4 Binary discretization

In Figure 4.4 gives the distribution of attributes of the heart disease dataset that are discretized with binary discretization with three as a bin amount (*3binaryDis_heart*). As seen on the figure there are two (bin amount -1) different attributes for each attribute in the original dataset. For the attribute *age* there are *age_1* and *age_2*. Values of these new attributes are divided into two intervals. Intervals are $(-\infty, x]$ and (x, ∞) where x is a cutpoint. The attributes that describe the same

original attribute, for example *age_1* and *age_2*, both involve the hole range of the original attribute, but differ in the location of the cutpoint.

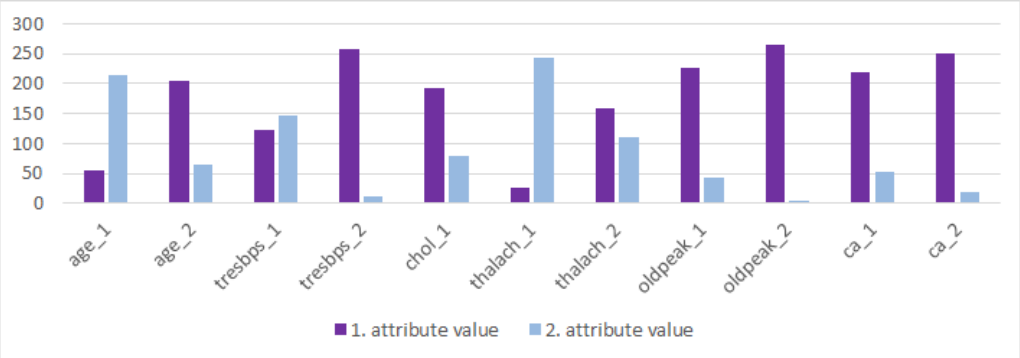

Figure 4.4 *The discretized attributes of the 3binary_heart.*

In Table 4.6 contains the distributions of the dicretized attributes of *3binaryDis_heart*. Two attributes suffices to divide continuous value for three bins. As an example the bins for *age* are $(-\infty, 45.00]$, $(45.00, 61.00]$ and $(61.00, -\infty]$. The value is on range $(45.00, 61.00]$ when $age\_1 = (45.00, \infty]$ and $age\_2 = (-\infty, 61.00]$. It is the intersection of those two values. In general $n - 1$ attributes suffices dividing n bins. The amount of the attributes can be smaller since *findNumBins* option was used in discretization. In Table 4.6 there is just one attribute for *chol*, so the numeric area is just divided for two intervals.

Table 4.6 *The discretized attributes of the 3binaryDis_heart.*

age_1	age_2	tresbps_1	tresbps_2
(-∞, 45.00]: 56	(-∞, 61.00]: 205	(-∞, 129.33]: 123	(-∞, 164.67]: 258
(45.00, ∞): 214	(61.00, ∞): 65	(129.33, ∞): 147	(164.67, ∞): 12
chol_1	thalach_1	thalach_2	oldpeak_1
(-∞, 272.00]: 192	(-∞, 114.67]: 26	(-∞, 158.33]: 159	(-∞, 2.07]: 226
(272.00, ∞): 78	(114.67, ∞): 244	(158.33, ∞): 111	(2.07, ∞): 44
oldpeak_2	ca_1	ca_2	
(-∞, 4.13]: 266	(-∞, 1.00]: 218	(-∞, 2.00]: 251	
(4.13, ∞): 4	(1.00, ∞): 52	(2.00, ∞): 19	

In Table A.4 there are the distributions of the binary dicretized attributes of the heart disease dataset with five bins (*5binaryDis_heart*). There is one attribute for

chol and two for *ca* just like in the discretization with three bins, but the cut points differ. In *3binaryDis_heart* cut point for *chol* is 272.00, but in *5binaryDis_heart* it is 213.60. The distribution shifts from (192, 78) to (69, 201). For attribute *ca* the distributions seem to go more even in *5binaryDis_heart*.

In Table A.8 gives the distributions of the dicretized attributes with binary then bins discretization (*10binaryDis_heart*). Non of the attributes in the dataset have nine $(n-1)$ attributes in this discretization. That is because *findNumBins* option was used in discretization. Greatest amount of attributes for a single dataset attribute is eight. For *age* and *tresbps* there are eight attributes, but for others there are less. For *chol* there is one attribute which has a different cutpoint than in *3binaryDis_heart* and *5binaryDis_heart*. There are also two attributes to describe attribute *ca*. The first cutpoint of *ca* is same as the *5binaryDis_heart*, but the second is not. The attributes *ca_1* and *ca_2* have same distribution, so there are no instances in the dataset that are between values $(0.30, \infty)$ and $(-\infty, 0.60)$ for original attribute *ca*.

Table 4.7 *The discretized attributes of the 5binaryDis_aus.*

A2_1	A2_2	A2_3
(-∞, 1617.80]: 119	(-∞, 3219.60]: 459	(-∞, 4821.40]: 629
(1617.80, ∞): 571	(3219.60, ∞): 231	(4821.40, ∞): 61
A2_4	A3_1	A3_2
(-∞, 6423.20]: 680	(-∞, 5267.00]: 643	(-∞, 10534.00]: 668
(6423.20, ∞): 10	(5267.00, ∞): 47	(10534.00, ∞): 22
A3_3	A3_4	A7_1
(-∞, 15801.00]: 685	(-∞, 21068.00]: 687	(-∞, 2883.00]: 664
(15801.00, ∞): 5	(21068.00, ∞): 3	(2883.00, ∞): 26
A7_2	A7_3	A10_1
(-∞, 5766.00]: 681	(-∞, 8649.00]: 686	(-∞, 13.40]: 667
(5766.00, ∞): 9	(8649.00, ∞): 4	(13.40, ∞): 23
A13_1	A14_1	
(-∞, 400.00]: 634	(-∞, 20001.00]: 685	
(400.00, ∞): 56	(20001.00, ∞): 5	

Table 4.7 contains the distribution of continuous attributes of the Australian credit data approval dataset with binary discretization with five bins (*5binaryDis_aus*).

For attribute *A2* and *A3* there are four attributes and for *A7* there are three. For the attributes *A10*, *A13* and *A14* there is only one attribute for each attribute in original dataset and the distributions for them are far from even. There seems to be many attributes that have the other value much larger and almost describes the whole group.

Binary discretization is interesting since every attribute divides all instances. By increasing the bin amount, the probability that subgroup could use a rule resulting from binary discretization does not decrease, since the size of the intervals does not decrease. On the contrary, probability can increase since the subgroups could be described with better quality. Another interesting use case is that if a cut point is not wanted to the subgroup discovery for some reason, for example it is too obvious for expert or does not offer new information, the discretized attribute could be left out from the search without affecting other attributes.

5. EXTRACTED SUBGROUPS

The results of the subgroups discovery task are presented in following three sections. The two first sections introduce separately subgroups extracted from the datasets. These firts sections focus more on the individual subgroups that are discovered and quality measures of these individual subgroups. These subgroups are presented with tables and ROC graphs which contains all different subgroups extracted from each of the datasets. Tables containing subgroup descriptions and quality measures for each of the subgroup sets and figures in a ROC space describing quality of these subgroup sets are found from Appendix B. The third section covers average measures for all found subgroup sets and focuses on comparison between those sets that are discovered by performing subgroup discovery tasks on each discretized dataset.

5.1 Subgroups extracted from the heart disease dataset

Subgroup discoveries for discretizations of the heart disease dataset produced eight different subgroup sets. This is because with some of the discretizations, subgroup discovery task produced same subgroup sets as result. The following list of the names of the extracted subgroup sets includes the used discretizations in the brackets.

- 3bins_heart (*3BinsDis_heart*, *3findBinsDis_heart*)
- 3equalFreq_heart (*3equalFreqDis_heart*)
- 3binary_heart (*3binaryDis_heart*)
- 5and10bins_heart (*5findBinsDis_heart*, *10findBinsDis_heart*, *5equalFreqDis_heart*, *10equalFreqDis_heart*)
- 5findBins_heart (*5findBinsDis_heart*)
- 5binary_heart (*5binaryDis_heart*)
- 10findBins_heart (*10findBinsDis_heart*)
- 10binary_heart (*10binaryDis_heart*)

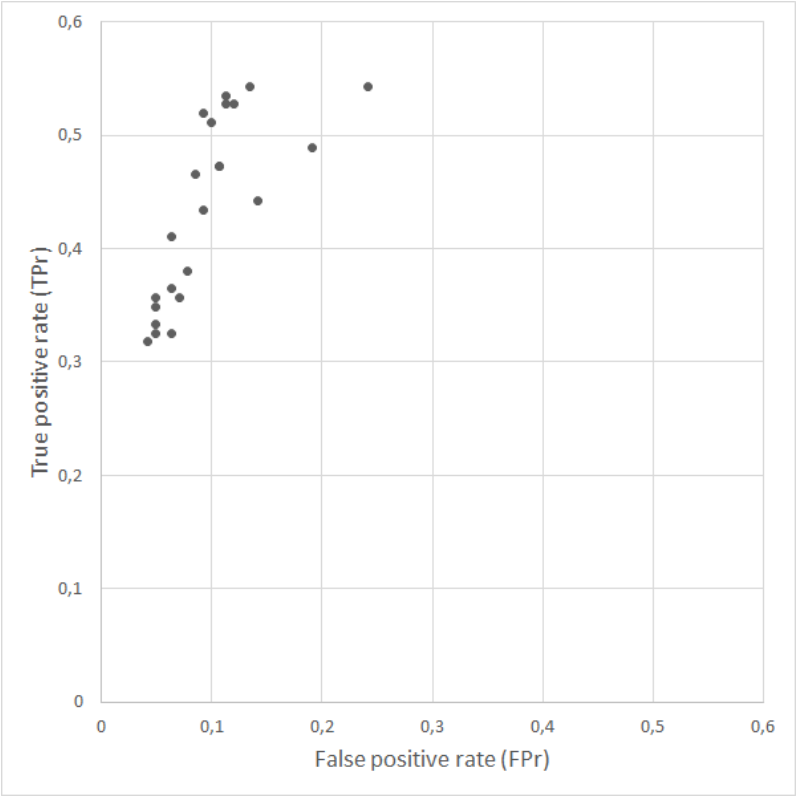

Figure 5.1 *All subgroups extracted from the heart disease dataset*

Rules can be depicted in the ROC space with (TPr, FPr) pair as point. Figure 5.1 consist of all subgroups that subgroup discovery task discovered for the heart disease dataset. Note that the TPr and FPr ranges are from 0 to 0.6 rather than 0 to 1. Almost all subgroups have FPr lower than 0.2 and all have lower FPr than 0.25. TPr values are above 0.3. Since all subgroups are in northwest corner of the ROC space, they can be called conservative.

All different subgroups are presented as combined to reduce the the repetation, since there are same subgroup descriptions in different subgroup sets. Table 5.1 consist of all subgroups extracted from the heart disease dataset. The table is in descending order in respect to the value of the WRAcc. The lines with subgroups with discretized attributes in their subgroup descriptions are highlighted for separating them of subgroups without them. Notice that the order of rules describing a subgroup is not meaningful, for example *fbs=0, exang=1* describes the same subgroup as would *exang=1, fbs=0*. In descriptions of the subgroups comma between the rules can be replaced with AND.

Table 5.1 *WRAcc, size, TP and FP for all subgroups from the heart disease dataset*

WRAcc	size	description	TP	FP
0.1066	80	age_4=(-∞, 67.40], thalach_4=(-∞, 175.80], exang=1	67	13
0.1051	85	thalach_4=(-∞, 175.80], exang=1	69	16
0.1032	84	age_4=(-∞, 67.40], exang=1	68	16
0.1029	80	exang=1, age_8=(-∞, 67.40], trestbps_1=(104.60, ∞)	66	14
0.1018	89	exang=1	70	19
0.1014	85	exang=1, trestbps_1=(104.60, ∞)	68	17
0.0948	72	exang=1, thalach_2=(-∞, 158.33]	60	12
0.0914	76	fbs=0, exang=1	61	15
0.0914	76	exang=1, age_1=(45.00, ∞)	61	15
0.0866	62	fbs=0, exang=1, thalach_2=(-∞, 158.33]	53	9
0.0853	69	exang=1, age_2=(-∞, 61.00]	56	13
0.0766	53	thal=7, exang=1	46	7
0.0753	60	fbs=0, sex=1, thalach=(-∞, 142.50]	49	11
0.0752	104	thal=7	70	34
0.0750	56	exang=1, age=(45.00, 61.00]	47	9
0.0749	77	fbs=0, thalach=(-∞, 142.50]	57	20
0.0747	52	exang=1, thalach=(-∞, 142.50]	45	7
0.0741	90	thalach=(-∞, 142.50]	63	27
0.0713	56	slope=2, exang=1	46	10
0.0708	50	fbs=0, slope=2, exang=1	43	7
0.0688	49	exang=1, ca=(0.60, ∞)	42	7
0.0687	47	fbs=0, exang=1, age=(45.00, 61.00]	41	6
0.0653	51	restecg=2, exang=1	42	9

Table 5.2 consist of more quality measures for all subgroups extracted from the heart disease dataset. The table uses same ordering, descending order in respect to the value of the WRAcc, as the previous table.

Table 5.2 *Coverage, support, accuracy and significance quality measures for all subgroups from the heart disease dataset*

description	Cov	Sup	Acc	Sig
age_4=(-∞, 67.40], thalach_4=(-∞, 175.80], exang=1	0.2963	0.2481	0.8375	19.61
thalach_4=(-∞, 175.80], exang=1	0.3148	0.2556	0.8118	17.71
age_4=(-∞, 67.40], exang=1	0.3111	0.2519	0.8095	17.92
exang=1, age_8=(-∞, 67.40], trestbps_1=(104.60, ∞)	0.2963	0.2444	0.8250	18.90
exang=1	0.3296	0.2593	0.7865	16.02
exang=1, trestbps_1=(104.60, ∞)	0.3148	0.2519	0.8000	16.77
exang=1, thalach_2=(-∞, 158.33]	0.2667	0.2222	0.8333	17.57
fbs=0, exang=1	0.2815	0.2259	0.8026	15.62
exang=1, age_1=(45.00, ∞)	0.2815	0.2259	0.8026	14.85
fbs=0, exang=1, thalach_2=(-∞, 158.33]	0.2296	0.1963	0.8548	17.00
exang=1, age_2=(-∞, 61.00]	0.2556	0.2074	0.8116	15.03
thal=7, exang=1	0.1963	0.1704	0.8679	17.76
fbs=0, sex=1, thalach=(-∞, 142.50]	0.2222	0.1815	0.8167	13.03
thal=7	0.3852	0.2593	0.6731	7.68
exang=1, age=(45.00, 61.00]	0.2074	0.1741	0.8393	13.97
fbs=0, thalach=(-∞, 142.50]	0.2852	0.2111	0.7403	9.55
exang=1, thalach=(-∞, 142.50]	0.1926	0.1667	0.8654	14.98
thalach=(-∞, 142.50]	0.3333	0.2333	0.7000	7.99
slope=2, exang=1	0.2074	0.1704	0.8214	13.33
fbs=0, slope=2, exang=1	0.1852	0.1593	0.8600	14.60
exang=1, ca=(0.60, ∞)	0.1815	0.1556	0.8571	13.70
fbs=0, exang=1, age=(45.00, 61.00]	0.1741	0.1519	0.8723	14.34
restecg=2, exang=1	0.1889	0.1556	0.8235	11.61

A closer look reveals that there are a high share of the discovered subgroups that have discretized attributes in their descriptions. This means that there are subgroups that have high quality with descriptions that have discretized attributes. With closer viewing it shows that there are overlapping subgroups. These subgroups

may overlap by discretized value ranges, but they describe different subgroups and are listed separately. For an example of this compare subgroups *thalach_4=(-∞, 175.80], exang=1* and *exang=1, thalach=(-∞, 142.50]*. These subgroups have both the attribute *exang* with same value and they both have attribute *thalach* with ranges that overlap because they are results from subgroup discovery tasks with different discretizations. The first rule is from subgroup set *3equalFreq_heart* and the second is from *5binary_heart*.

The subgroup with description *exang=1* is a great comparison point. It is a subgroup with a single binary attribute and it has a high WRAcc value. It appears in every subgroup set and in most of those sets it is the best subgroup that is found according its WRAcc value. There are four different subgroups with higher WRAcc value than the subgroup *exang=1*. These subgroups are from the subgroup sets *5binary_heart* and *10binary_heart*. When these four best subgroups are examined, it can be seen that they all have *exang* attribute with same value as part of their subgroup description. In addition of that attribute they have discretized numerical attributes to limit more the subgroups and increase the quality.

The sizes of the subgroup vary from 47 to 104. The smallest subgroup *fbs=0, exang=1, age=(45.00, 61.00]* has also the highest accuracy. The biggest subgroups *thal=7* has the highest coverage and highest support value together with subgroup *exang=1*. Subgroup *thal=7* has low significance. The subgroup that has highest significance, also haves highest WRAcc value.

5.2 Subgroups extracted from the Australian credit approval

With the Australian credit approval dataset each discretization produced different set of subgroups. The names of the subgroup sets are 5findBins_aus, 5equalFreq_aus and 5binary_aus.

Figure 5.2 consist of all subgroups extracted from the Australian credit approval dataset. Note that the TPr and FPr ranges are from 0 to 0.6 instead of from 0 to 1. Compared to Figure 5.1 that contains all the subgroups extracted from the heart disease dataset the quality of subgroups is more spread apart. Those subgroups which have higher TPr are more on the east side of the ROC space, meaning that they have higher FPr. On east side there are subgroups with lower TPr. If there would be a trendline drawn in a graph, it would be almost parallel to the diagonal

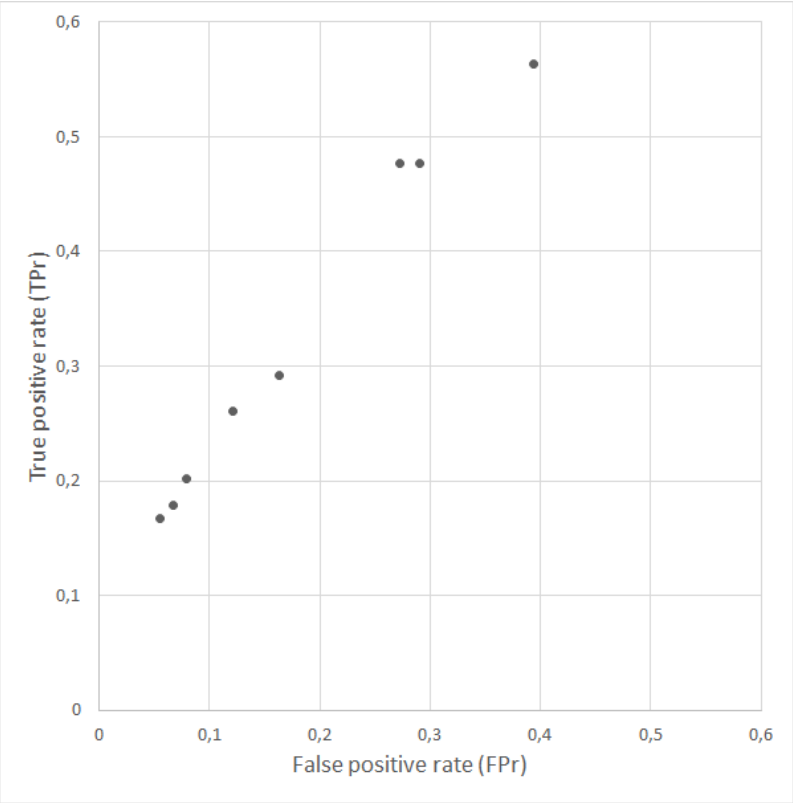

Figure 5.2 *All subgroups extracted from the Australian credit approval dataset*

line. This means that there is almost a straight consequence that when TPr gets higher then FPr gets higher in same with respect to it.

Table 5.3 *WRAcc, size, TP and FP for all subgroups from the Australian credit approval dataset*

WRAcc	size	description	TP	FP
0.0370	295	A9=1	250	45
0.0337	298	A2_1=(1617.80, ∞), A8=1	250	48
0.0309	361	A8=1	296	65
0.0254	157	A11=0, A9=1	137	20
0.0233	180	A11=0, A8=1	153	27
0.0224	119	A11=0, A8=1, A9=1	106	13
0.0206	97	A9=1, A1=0, A10_1=(-∞, 13.40]	88	9
0.0204	105	A9=1, A1=0	94	11

Table 5.3 consist of all the subgroups that were extracted from the Australian credit approval dataset. Table is in descending order in respect to the value of the WRAcc. The lines with subgroups with discretized rules are highlighted for separation.

Table 5.4 *Coverage, support, accuracy and significance quality measures for all subgroups from the Australian credit approval dataset*

description	Cov	Sup	Acc	Sig
A9=1	0.4275	0.3623	0.8475	6.31
A2_1=(1617.80, ∞), A8=1	0.4319	0.3623	0.8389	5.24
A8=1	0.5232	0.4290	0.8199	3.89
A11=0, A9=1	0.2275	0.1986	0.8726	5.58
A11=0, A8=1	0.2609	0.2217	0.8500	4.07
A11=0, A8=1, A9=1	0.1725	0.1536	0.8908	5.80
A9=1, A1=0, A10_1=(-∞, 13.40]	0.1406	0.1275	0.9072	6.14
A9=1, A1=0	0.1522	0.1362	0.8952	5.51

Table 5.4 consist of futher quality measures for all subgroups extracted from the Australian credit approval dataset. The table uses same ordering, descending order respect to the value of the WRAcc, as the previous table.

As seen on the results there are only two subgroups with discretized attributes. The subgroup *A2_1=(1617.80, ∞), A8=1* is in subgroup set *5binary_aus*. The subgroup *A9=1, A1=0, A10_1=(-∞, 13.40]* is in subgroup sets *5findBins_aus* and *5binary_aus*. For both of these subgroups there is a subgroup in the result which has the same nominal attributes with the same values, but do not have the discretized attributes. With the discretized attributes in their descriptions the quality of those subgroups is lifted.

The sizes of the subgroups vary from 97 to 361. The smallest subgroup *A9=1, A1=0, A10_1=(-∞, 13.40]* has low coverage and support, but it also hast the highest accuracy. The subgroup *A8=1* is the biggest subgroup and it has the highest coverage and support value. The same subgroup *A9=1* haves the highest WRAcc and significance values.

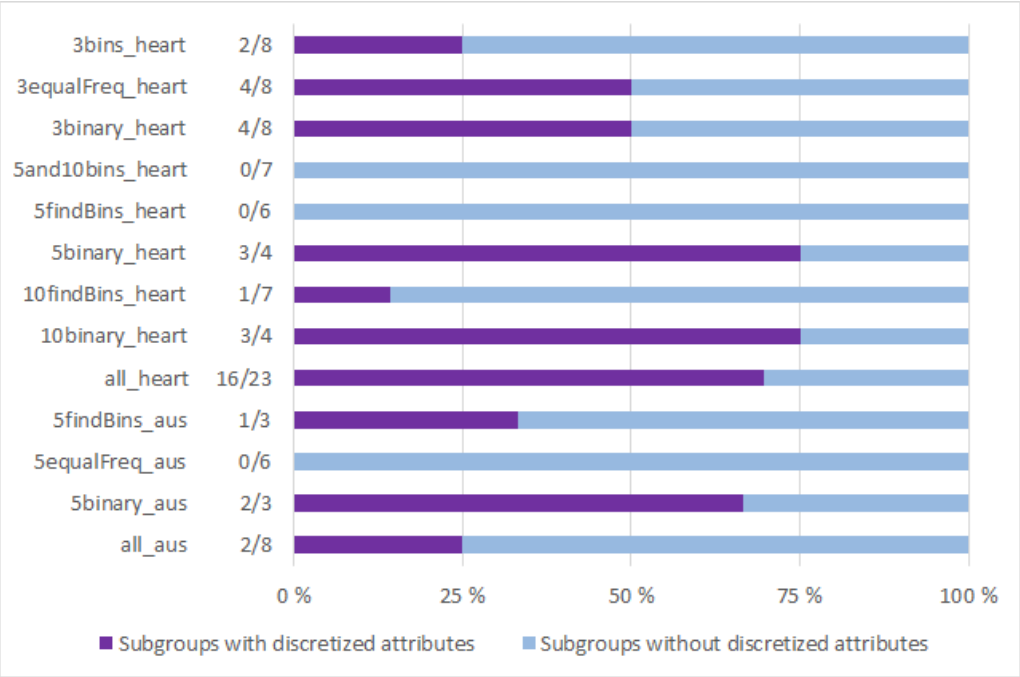

Figure 5.3 *The division between subgroups with and without discretized rules*

5.3 Average measures for rule sets

To understand better the average measures let us examine how different discretizations affect the results. Figure 5.3 shows how subgroups are divided between subgroups that have discretized attributes in rules and those without them. In addition for subgroup set extracted with different discretizations, there are subgroup sets *all_heart* and *all_aus* which contains all subgroups extracted from the both of the datasets. In the figure, next to the subgroup set names, are the amounts of subgroups with discretized attributes along with the size of subgroups in the set. The diagram shows the relative shares as the darker blue bar marks subgroups with discretized attributes and lighter blue marks it without them. There are subgroup sets *5and10bins_heart*, *5findBins_heart* and *5equalFreq_aus* with none with discretized attributes on their subgroup descriptions. When these subgroup sets are looked more closely it can be seen that subgroup set *5and10bins_heart* is result of subgroup discovery task of four different discretizations *5binsDis_heart, 5equalFreqDis_heart, 10bins* and *10equalFreqDis_heart*. Different discretizations produce the same results because discretized attributes do not appear among the best discovered subgroups. Out of a total of fifteen discretizations six of them did not produce any subgroups with discretized attributes among the best extracted subgroups.

Equal interval bin discretization was only used for the heart disease dataset. Results for it can be seen in subgroup sets *3bins_heart* and *5and10bins_heart*. Subgroup set *3bins_heart* contains two subgroups with discretized attributes and *5and10bins_heart* contains none of them.

The results of usage of equal interval bin discretization with removal of unnecessary bins, it can been seen on subgroup sets *3bins_heart*, *5findBins_heart*, *10findBins_heart* and *5findBins_aus*. The set *5findBins_heart* does not have any subgroups discretized attributes, but the others have small shares of them.

The usage of equal frequency interval discretization produced subgroup sets *3equalFreq_heart*, *5and10bins_heart* and *5equalFreq_aus*. Of these subgroup sets only *3equalFreq_heart* has subgroups discretized attributes and the relative share of those subgroups is 50%.

The subgroup sets *3binary_heart*, *5binary_heart*, *10binary_heart*, and *5binary_heart* are all results from subgroup discovery task with binary discretization and they have the highest share of subgroups discretized attributes in their rules. The shares vary from 50% to 75%.

Table 5.5 *WRACC, SIZE, COV, ACC, and SIG of the subgroup sets*

Subgroup set	WRACC	SIZE	COV	ACC	SIG
3bins_heart	0.0788	66.38	0.2458	0.8154	14.17
3equalFreq_heart	0.0805	75.12	0.2782	0.7816	12.83
3binary_hearty	0.0919	74.00	0.2741	0.8152	16.02
5and10bins_heart	0.0789	68.43	0.2534	0.8050	13.80
5findBins_heart	0.0812	71.33	0.2642	0.8019	14.17
5binary_heart	0.1042	84.50	0.3130	0.8113	17.82
10findBins_heart	0.0794	68.14	0.2524	0.8098	14.10
10binary_heart	0.1023	84.50	0.3130	0.8053	17.40
all_heart	0.0844	69.70	0.2581	0.8136	14.76
5findBins_aus	0.0295	251.00	0.3638	0.8582	5.45
5equalFreq_aus	0.0266	202.83	0.2940	0.8627	5.19
5binary_aus	0.0304	230.00	0.3333	0.8645	5.90
all_aus	0.0267	201.50	0.2902	0.8652	5.41

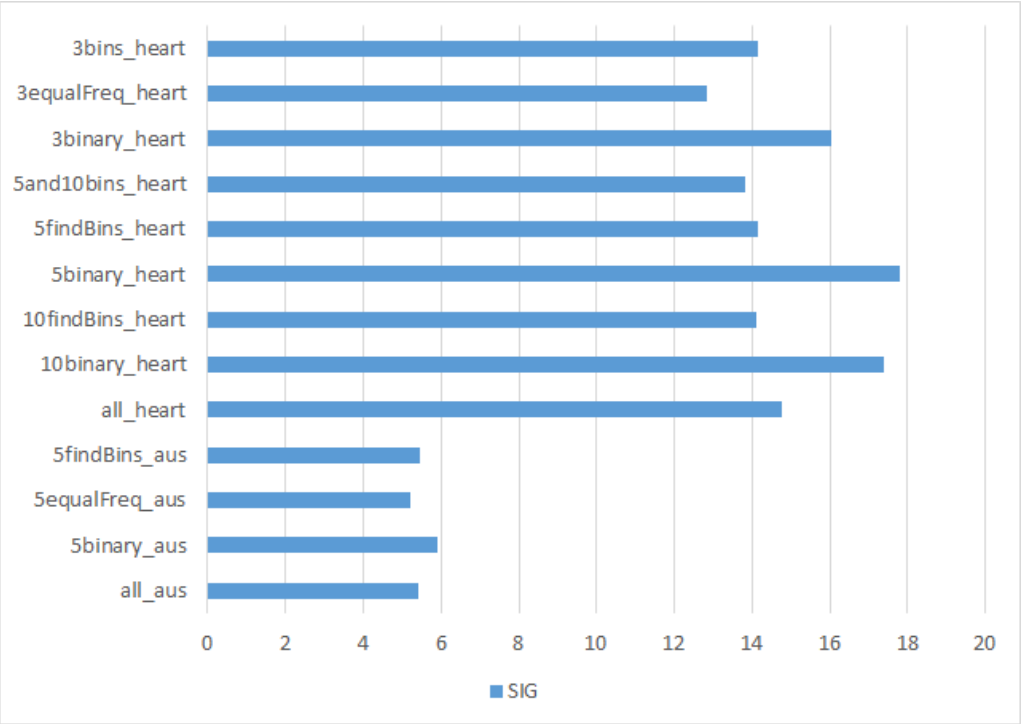

Figure 5.4 *SIG of subgroup sets*

Table 5.5 consist of average measures of subgroup sets.

Before considering the average size (SIZE) of the subgroups let us consider the subgroup size that is the target attribute with value used for subgroup discovery task. The heart disease dataset has 270 instances and for target attribute *cp* there are 129 instances where *cp* is 4. For the Australian credit approval dataset there exist 690 instances and 525 has value 2 for attribute *A4*. So for the heart disease dataset the the target consists 129 instances and for the Australian credit approval dataset 525 instances. The SIZE of subgroup sets extracted from the heart disease dataset varies from 66.38 to 84,50. The SIZE of subgroup sets extracted from Austaralian credit approval dataset varies from 202.53 to 251.00. The SIZE of for *all_heart*, which have all different subgroups extracted from the heart disease dataset, is 69.70. The average size for *all_aus* which have all the subgroups extracted from the Australian credit approval dataset is 201.50.

Figure 5.4 visualizes the average significance (SIG) from Table 5.5. The subgroup sets from the Australian credit approval dataset have not as high significance as the rules from the heart disease dataset. By examining the subgroup sets from

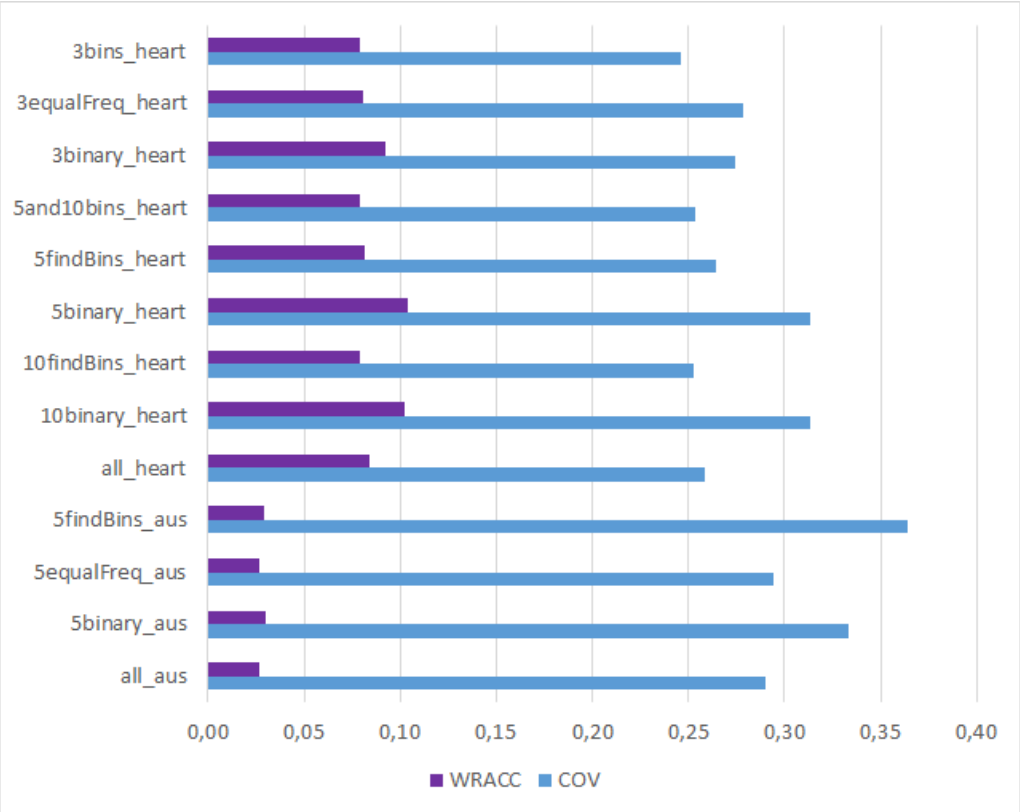

Figure 5.5 *WRACC, and COV of subgroup sets*

the heart disease dataset three subgroup sets reaches the SIG value 16 or above. These subgroup sets are *3binary_heart*, *5binary_heart* and *10binary_heart*. All of these subgroup sets are results of using binary discretization with subgroup discovery task and they have the highest relative share of discretized attributes used in their subgroups. Subgroup set *5binary_aus* has the highest SIG among the subgroup sets from the Australian credit approval dataset and it also has the highest share of subgroups with discretized attributes appearing in the descriptions. The subgroup set *3equalFreq_heart* has the lowest SIG from the heart disease dataset and *5equalFreq_aus* from the Australian credit approval dataset.

Figure 5.5 visualizes the average measures of WRAcc (WRACC), and coverage (COV) from Table 5.5. The measures are drawn in same figure since the values are located between 0.0 and 0.4.

WRAcc is the measure that is used as quality function for subgroup discovery. It is the measure that has the highest impact of the results of the subgroups discov-

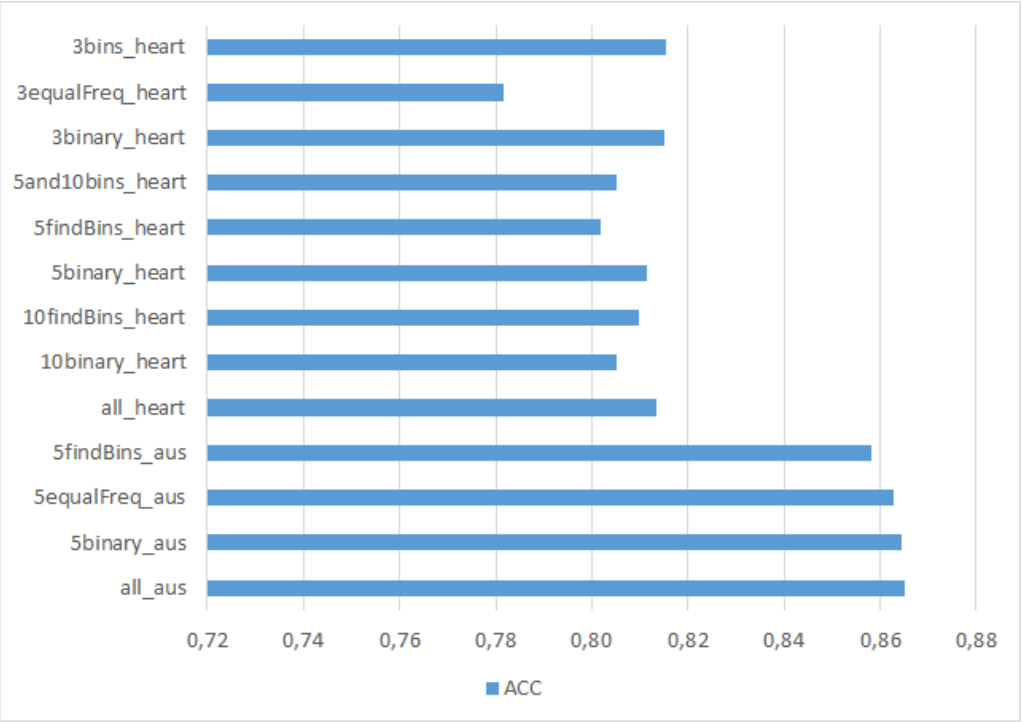

Figure 5.6 *ACC of subgroup sets*

ery tasks and it can be said to be the most meaningful. Discretizations from the heart disease dataset seem to have a higher average WRAcc than discretizations from the Australian credit approval dataset. The subgroup sets *5binary_heart* and *10binary_heart* has the highest WRACC values with values higher than on 0.10. The subgroup set *5binary_aus* has the highest WRACC among the subgroup sets resulting from subgroup discovery tasks on the Australian credit approval dataset. All of these subgroup sets has higher relative share of discretized attributes. The subgroup set *5equalFreq_aus* has the lowest WRACC value and *3bins_heart* has the lowest WRACC value among the the subgroup set from heat disease dataset.

According to the average coverage one cannot clear distinction between discretizations from different datasets. The subgroup set *5findBinsAus_aus* has the highest COV of all subgroup sets. The subgroup sets *5binary_heart* and *10binary_heart* have the highest COV among the subgroup sets resulting from subgroup discovery tasks on the heart disease dataset. These binary subgroup sets also have the highest relative share of subgroups with discretized attributes. The subgroup set *3bins_heart* has the lowest COV and *all_aus* has the lowest COV among the subgroup sets resulting from subgroup discovery tasks on the Australian credit approval

dataset. Coverage is counted as the relative frequency of all examples covered by the rules, so it is proportional to the size of the subgroup.

Figure 5.6 illustrates the average accuracy measures (ACC) from Table 5.5. Notice that the value range in the figure is from 0.72 to 0.88. Subgroup sets extracted from the Australian credit approval dataset have a higher ACC than those extracted from the heart disease dataset. Subgroup set *5binary_aus*, which have the highest relative share of discretized attributes, has also has the highest ACC. The subgroup sets *3bins_heart* and *3binary_heart* has the highest ACC among the subgroup sets from the heart disease dataset. The subgroup set *3equalFreq_heart* has the lowest ACC value and *5equalFreq_aus* has the lowest ACC value among the subgroup sets resulting from the Australian credit approval dataset.

When these average measures are looked more closely, some of the trade offs can be seen. For an example subgroup sets *5equalFreq_aus* and *all_aus* have low COV values among the subgroups from the Australian credit approval dataset, but the values of ACC are almost the highest ones. The subgroup sets from the Australian credit approval dataset have a lower WRACC and SIG, but they also have a higher ACC. Subgroup sets from heart disease dataset have a higher WRACC and SIG, but a lower ACC.

6. CONCLUSIONS

Discretization is useful for its simplicity and time consumption. It also makes it possible to use any subgroup discovery algorithm. The methods used in this thesis were straightforward, offering a soft approach towards handling continuous values in data mining and were able to produce subgroups with high unusualness values.

With all other discretizations except the binary one, the amount of bins in discretization step affect greatly whether there will be subgroups with discretized attributes in the result set. For them, the probability for subgroups with discretized attributes as result decreases when the bin amount increases. With binary discretization there is not this effect, since all attributes divide the whole population. So with equal interval width and equal frequency discretization, the bin amount should be proportional to the target class and value. With a small bin amount there is a danger of not describing anything or even if it does, it does not produce usable information.

Subgroup discovery with binary discretization produced sets with high share of discretized attributes. The other case when there were a high share of values was discretization with equal frequency and with three as a bin amount. With the binary discretization the quality of these subgroups was better.

The usage of discretized attributes can increase the quality of subgroups. With both datasets, there were subgroups with discretized attributes that where also subgroups of already extracted subgroups. Of course it is possible to have subgroups descriptions with only discretized or numerical attributes, the descriptions of best subgroups depends on the dataset.

As the subgroup discovery tasks were performed for some datasets without any knowledge guidance, the only usable thing to evaluate the subgroups are the subjective quality measures. The information is the subgroup really usable, actionable or operational needs background knowledge or expert's guidance. This is obvious, but it is emphasized when the results are examined.

BIBLIOGRAPHY

[1] M. Atzmueller. Subgroup Discovery - Advanced Review. *WIREs: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

[2] M. Atzmueller and F. Lemmerich. Fast and Effective Subgroup Mining for Continuous Target Variables. In *Proceedings of the LWA 2009 (Knowledge Discovery and Machine Learning Track)*. University of Darmstadt, Germany, 2009.

[3] M. Atzmueller and F. Lemmerich. Fast Subgroup Discovery for Continuous Target Concepts. In *Proceedings 18th International Symposium on Methodologies for Intelligent Systems*, volume 5722 of *Lecture Notes in Computer Science*, pages 35–44. Springer, 2009.

[4] M. Atzmueller and F. Puppe. SD-Map - A Fast Algorithm for Exhaustive Subgroup Discovery. In J. Fuernkranz, T. Scheffer, and M. Spiliopoulou, editors, *PKDD*, volume 4213 of *Lecture Notes in Computer Science*, pages 6–17. Springer, 2006.

[5] M. Atzmueller, F. Puppe, and H-P Buscher. Towards Knowledge-Intensive Subgroup Discovery. In *Proceedings of the LWA 2004 Workshop*, pages 117–123, Germany, 2004.

[6] T. Bäck, D. B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, 1st edition, 1997.

[7] F. Berlanga, M. del Jesus, P. González, F. Herrera, and M. Mesonero. Multiobjective Evolutionary Induction of Subgroup Discovery Fuzzy Rules: A Case Study in Marketing. In *Proceedings of the 6th Industrial Conference on Data Mining*, volume 4065 of *Lecture Notes in Computer Science*, pages 337–349. Springer, Leipzig, Germany, 2006.

[8] C. Carmona, P. González, M. del Jesus, and F. Herrera. NMEEF-SD: Non-Dominated Multiobjective Evolutionary Algorithm for Extracting Fuzzy Rules in Subgroup Discovery. In *IEEE Trans. on Fuzzy Systems*, pages 958–970. IEEE, 2010.

[9] C. Carmona, P. González, M. Gacto, and M. del Jesus. Genetic Lateral Tuning for Subgroup Discovery with Fuzzy Rules using the Algorithm NMEEF-SD. *International Journal of Computational Intelligence Systems*, 5(2):355–367, 2012.

[10] M. del Jesus, P. González, and F. Herrera. Multiobjective Genetic Algorithm for Extracting Subgroup Discovery Fuzzy Rules. In *Proceedings of the 2007 Intelligence in Multicriteria Decision Making (MCDM)*, pages 50–57, 2007.

[11] M. del Jesus, P. González, F. Herrera, and M. Mesonero. Evolutionary Fuzzy Rule Induction Process for Subgroup Discovery: A Case Study in Marketing. *IEEE Transactions on Fuzzy Systems*, 15(4):578–592, 2007.

[12] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and Unsupervised Discretization of Continuous Features. In *Machine Learning: Proceedings Of The Twelfth International Conference*, pages 194–202. Morgan Kaufmann, 1995.

[13] C. Elkan. The Foundations of Cost-Sensitive Learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978. Morgan Kaufmann, 2001.

[14] T. Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[15] U. Fayyad and K. Irani. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1029, 1993.

[16] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17:37–54, 1996.

[17] P. Flach and S. Wu. Repairing Concavities in ROC Curves. In *Proceedings of the 2003 UK Workshop on Computational Intelligence*, pages 38–44. University of Bristol, UK, 2003.

[18] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus. Knowledge Discovery in Databases: An Overview. In K. Ford, editor, *Knowledge Discovery in Databases*, pages 1–30. AAAI/MIT Press, 1991.

[19] D. Gamberger and N. Lavrač. Expert-Guided Subgroup Discovery: Methodology and Application. *Journal of Artificial Intelligence Research*, 17:501–527, 2002.

[20] H. Grosskreutz and S. Rüping. On Subgroup Discovery in Numerical Domains. In *Data Mining and Knowledge Discovery*, volume 19, pages 210–226, 2009.

[21] H. Grosskreutz, S. Rüping, N. Shabaani, and S. Wrobel. Optimistic Estimate Pruning Strategies for Fast Exhaustive Subgroup Discovery. *Technical report*, 2008.

[22] H. Grosskreutz, S. Rüping, and S. Wrobel. Tight Optimistic Estimates for Fast Subgroup Discovery. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *Lecture Notes in Artificial Intelligence*, pages 440–456. Springer-Verlag, Heidelberg, Germany, 2008.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and H. Ian. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11, 2009.

[24] F. Herrera, C. Carmona, P. González, and M. del Jesus. An Overview on Subgroup Discovery: Foundations and Applications. In *Knowledge and Information Systems*, pages 1–31. Springer London, 2010.

[25] R. Holte. Very simple classification rules perform well on most commonly used datasets. In *Machine Learning*, volume 11, pages 63–91. Kluwer Academic Publishers, 1993.

[26] B. Kavsek and N. Lavrač. APRIORI-SD: Adapting Association Rule Learning to Subgroup Discovery. *Applied Artificial Intelligence*, 20(7):543–583, 2006.

[27] W. Klösgen. Explora: A Multipattern and Multistrategy Discovery Assistant. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI/MIT Press, California, USA, 1996.

[28] N. Lavrač, P. Flach, H. Motoda, and T. Fawcett. Decision Support through Subgroup Discovery: Three Case Studies and the Lessons Learned. In *Machine Learning*, volume 57, pages 115–143. Springer, 2004.

[29] N. Lavrač, P. Flach, and L. Todorovski. Rule Induction for Subgroup Discovery with CN2-SD. In M. Bohanec, B. Kasek, N. Lavrač, and D. Mladenic, editors, *2nd Int. Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and MetaLearning*, pages 77–87. University of Bristol, 2002.

[30] N. Lavrač, P. Flach, and B. Zupan. Rule Evaluation Measures: A Unifying View. In S. Dzeroski and P. Flach, editors, *Ninth International Workshop on Inductive Logic Programming (ILP'99)*, volume 1634 of *Lecture Notes in Computer Science*, pages 174–185. Springer-Verlag, 1999.

[31] N. Lavrač, B. Kavsek, P. Flach, L. Todorovski, and S. Wrobel. Subgroup Discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.

[32] N. Lavrač, F. Železný, and P. Flach. RSD: Relational Subgroup Discovery through First-order Feature Construction. In *Proceedings of the 12th International Conference on Inductive Logic Programming*, volume 2583 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 2002.

[33] F. Lemmerich, M. Rohlfs, and M. Atzmueller. Fast Discovery of Relevant Subgroup Patterns. In *Proceedings of the 23rd FLAIRS Conference*, 2010.

[34] K. Moreland and K. Truemper. Discretization of Target Attributes for Subgroup Discovery. In P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 5632 of *Lecture Notes in Computer Science*, pages 44–52. Springer, 2009.

[35] M. Mueller, R. Rosales, H. Steck, S. Krishnan, and S. Kramer. Subgroup Discovery for Test Selection: A Novel Approach and Its Application to Breast Cancer Diagnosis. In *Advances in Intelligent Data Analysis VIII*, volume 5772 of *Lecture Notes in Computer Science*, pages 119–130. Springer, 2009.

[36] P. Novak, N. Lavrač, and G. Webb. Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining. *Journal of Machine Learning Research*, 10:377–403, 2009.

[37] G. Piatetsky-Shapiro and subgroup set. Matheus. The Interestingness of Deviations. In *AAAI'94 Workshop on Knowledge Discovery in Databases*, pages 25–36. AAAI Press, 1994.

[38] F. Provost and T. Fawcett. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 43–48. AAAI Press, 1997.

[39] C. Romero, P. González, S. Ventura, M. del Jesus, and F. Herrera. Evolutionary Algorithms for Subgroup Discovery in e-Learning: A Practical Application Using Moodle Data. *Expert Systems with Applications Journal*, 36(2):1632–1644, 2009.

[40] A. Siebes. Data Surveying: Foundations of an Inductive Query Language. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, pages 269–274. AAAI Press, Montreal, Canada, 1995.

[41] A. Silberscatz and A. Tuzhilin. On Subjective Measures of Interestingness in Knowledge Discovery. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 275–281. AAAI Press, 1995.

[42] S. Wrobel. An Algorithm for Multi-relational Discovery of Subgroups. In *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery*, PKDD '97, pages 78–87. Springer-Verlag, London, UK, 1997.

APPENDIX A. REST OF DISCRETIZATIONS

Table A.1 *The discretized attributes of the 5BinsDis_heart.*

age	tresbps	chol
(-∞, 38.60]: 9	(-∞, 115.20]: 49	(-∞, 213.60]: 69
(38.60, 48.20]: 65	(115.20, 136.40]: 124	(213.60, 301.20]: 160
(48.20, 57.80]: 85	(136.40, 157.60]: 73	(301.20, 388.80]: 36
(57.80, 67.40]: 95	(157.60, 178.80]: 19	(388.80, 476.40]: 4
(67.40, ∞): 16	(178.80, ∞): 5	(476.40, ∞): 1
thalach	oldpeak	ca
(-∞, 97.20]: 6	(-∞, 1.24]: 173	(-∞, 0.60]: 160
(97.20, 123.40]: 34	(1.24, 2.48]: 63	(0.60, 1.20]: 58
(123.40, 149.60]: 76	(2.48, 3.72]: 27	(1.20, 1.80]: 0
(149.60, 175.80]: 125	(3.72, 4.96]: 5	(1.80, 2.40]: 33
(175.80, ∞): 29	(4.96, ∞): 2	(2.40, ∞): 19

Table A.2 *The discretized attributes of the 5findBinsDis_heart.*

age	tresbps	chol
(-∞, 38.60]: 9	(-∞, 115.20]: 49	(-∞, 213.60]: 69
(38.60, 48.20]: 65	(115.20, 136.40]: 124	(213.60, ∞): 201
(48.20, 57.80]: 85	(136.40, 157.60]: 73	
(57.80, 67.40]: 95	(157.60, 178.80]: 19	
(67.40, ∞): 16	(178.80, ∞): 5	
thalach	oldpeak	ca
(-∞, 97.20]: 6	(-∞, 1.24]: 173	(-∞, 0.60]: 160
(97.20, 123.40]: 34	(1.24, 2.48]: 63	(0.60, 1.20]: 58
(123.40, 149.60]: 76	(2.48, 3.72]: 27	(1.20, ∞): 52
(149.60, 175.80]: 125	(3.72, 4.96]: 5	
(175.80, ∞): 29	(4.96, ∞): 2	

Table A.3 *The discretized attributes of the 5equalFreqDis_heart.*

age	tresbps	chol
(-∞, 45.50]: 56	(-∞, 119.00]: 57	(-∞, 207.50]: 54
(45.50, 52.50]: 53	(119.00, 125.50]: 53	(207.50, 233.50]: 54
(52.50, 57.50]: 50	(125.50, 134.50]: 54	(233.50, 257.50]: 54
(57.50, 62.50]: 57	(134.50, 144.50]: 52	(257.50, 288.50]: 54
(62.50, ∞): 54	(144.50, ∞): 54	(288.50, ∞): 54
thalach	oldpeak	ca
(-∞, 128.50]: 54	(-∞, 0.05]: 85	(-∞, 0.50]: 160
(128.50, 146.50]: 52	(0.05, 0.75]: 46	(0.50, 1.50]: 58
(146.50, 158.50]: 53	(0.75, 1.35]: 43	(1.50, 2.50]: 33
(158.50, 169.50]: 54	(1.35, 2.05]: 52	(2.50, ∞): 19
(169.50, ∞): 57	(2.05, ∞): 44	

Table A.4 *The discretized attributes of the 5binaryDis_heart.*

age_1	age_2	age_3	age_4
(-∞, 38.60]: 9	(-∞, 48.20]: 74	(-∞, 57.80]: 159	(-∞, 67.40]: 254
(38.60, ∞): 261	(48.20, ∞): 196	(57.80, ∞): 111	(67.40, ∞): 16
tresbps_1	tresbps_2	tresbps_3	tessbps_4
(-∞, 115.20]: 49	(-∞, 136.40]: 173	(-∞, 157.60]: 246	(-∞, 178.80]: 265
(115.20, ∞): 221	(136.40, ∞): 97	(157.60, ∞): 24	(178.80, ∞): 5
chol_1	thalach_1	thalach_2	thalach_3
(-∞, 213.60]: 69	(-∞, 97.20]: 6	(-∞, 123.40]: 40	(-∞, 149.60]: 116
(213.60, ∞): 201	(97.20, ∞): 264	(123.40, ∞): 230	(149.60, ∞): 154
thalach_4	oldpeak_1	oldpeak_2	oldpeak_3
(-∞, 175.80]: 241	(-∞, 1.24]: 173	(-∞, 2.48]: 236	(-∞, 3.72]: 263
(175.80, ∞): 29	(1.24, ∞): 97	(2.48, ∞): 34	(3.72, ∞): 7
oldpeak_4	ca_1	ca_2	
(-∞, 4.96]: 268	(-∞, 0.60]: 160	(-∞, 1.20]: 218	
(4.96, ∞): 2	(0.60, ∞): 110	(1.20, ∞): 52	

Table A.5 *The discretized attributes of the 10BinsDis_heart.*

age	tresbps	chol
(-∞, 33.80]: 1	(-∞, 104.60]: 10	(-∞, 169.80]: 9
(33.80, 38.60]: 8	(104.60, 115.20]: 39	(169.80, 213.60]: 60
(38.60, 43.40]: 30	(115.20, 125.80]: 61	(213.60, 257.40]: 93
(43.40, 48.20]: 35	(125.80, 136.40]: 63	(257.40, 301.20]: 67
(48.20, 53.00]: 42	(136.40, 147.00]: 49	(301.20, 345.00]: 33
(53.00, 57.80]: 43	(147.00, 157.60]: 24	(345.00, 388.80]: 3
(57.80, 62.60]: 57	(157.60, 168.20]: 13	(388.80, 432.60]: 4
(62.60, 67.40]: 38	(168.20, 178.80]: 6	(432.60, 476.40]: 0
(67.40, 72.20]: 13	(178.80, 189.40]: 3	(476.40, 520.20]: 0
(72.20, ∞): 3	(189.40, ∞): 2	(520.20, ∞): 1
thalach	**oldpeak**	**ca**
(-∞, 84.10]: 1	(-∞, 0.62]: 130	(-∞, 0.30]: 160
(84.10, 97.20]: 5	(0.62, 1.24]: 43	(0.30, 0.60]: 0
(97.20, 110.30]: 11	(1.24, 1.86]: 40	(0.60, 0.90]: 0
(110.30, 123.40]: 23	(1.86, 2.48]: 23	(0.90, 1.20]: 58
(123.40, 136.50]: 31	(2.48, 3.10]: 18	(1.20, 1.50]: 0
(136.50, 149.60]: 45	(3.10, 3.72]: 9	(1.50, 1.80]: 0
(149.60, 162.70]: 71	(3.72, 4.34]: 5	(1.80, 2.10]: 33
(162.70, 175.80]: 54	(4.34, 4.96]: 0	(2.10, 2.40]: 0
(175.80, 188.90]: 24	(4.96, 5.58]: 0	(2.40, 2.70]: 0
(188.90, ∞): 5	(5.58, ∞): 2	(2.70, ∞): 19

Table A.6 *The discretized attributes of the 10findBinsDis_heart.*

age	tresbps	chol
(-∞, 33.80]: 1	(-∞, 104.60]: 10	(-∞, 169.80]: 9
(33.80, 38.60]: 8	(104.60, 115.20]: 39	(169.80, ∞): 261
(38.60, 43.40]: 30	(115.20, 125.80]: 61	
(43.40, 48.20]: 35	(125.80, 136.40]: 63	
(48.20, 53.00]: 42	(136.40, 147.00]: 49	
(53.00, 57.80]: 43	(147.00, 157.60]: 24	
(57.80, 62.60]: 57	(157.60, 168.20]: 13	
(62.60, 67.40]: 38	(168.20, 178.80]: 6	
(67.40, ∞): 16	(178.80, ∞): 5	
thalach	oldpeak	ca
(-∞, 84.10]: 1	(-∞, 0.62]: 130	(-∞, 0.30]: 160
(84.10, 97.20]: 5	(0.62, 1.24]: 43	(0.60, ∞): 110
(97.20, 110.30]: 11	(1.24, 1.86]: 40	
(110.30, 123.40]: 23	(1.86, 2.48]: 23	
(123.40, 136.50]: 31	(2.48, 3.10]: 18	
(136.50, ∞): 199	(3.10, ∞): 16	

Table A.7 *The discretized attributes of the 10equalFreqDis_heart.*

age	tresbps	chol
(-∞, 41.50]: 24	(-∞, 109.00]: 20	(-∞, 194.00]: 27
(41.50, 44.50]: 25	(109.00, 116.00]: 29	(194.00, 207.50]: 27
(44.50, 49.50]: 30	(116.00, 121.00]: 42	(207.50, 221.50]: 27
(49.50, 52.50]: 30	(121.00, 127.00]: 22	(221.50, 233.50]: 27
(52.50, 55.50]: 29	(127.00, 131.00]: 41	(233.50, 244.50]: 26
(55.50, 57.50]: 21	(131.00, 137.00]: 19	(244.50, 256.50]: 27
(57.50, 59.50]: 27	(137.00, 141.00]: 39	(256.50, 269.50]: 27
(59.50, 62.50]: 30	(141.00, 149.00]: 11	(269.50, 288.50]: 28
(62.50, 66.50]: 30	(149.00, 159.00]: 24	(288.50, 308.50]: 26
(66.50, ∞): 24	(159.00, ∞): 23	(308.50, ∞): 28
thalach	oldpeak	ca
(-∞, 115.50]: 27	(-∞, 0.05]: 85	(-∞, 0.50]: 160
(115.50, 128.50]: 27	(0.05, 0.35]: 20	(0.50, 1.50]: 58
(128.50, 140.50]: 28	(0.35, 0.65]: 25	(1.50, 2.50]: 33
(140.50, 147.50]: 29	(0.65, 0.95]: 15	(2.50, ∞): 19
(147.50, 154.50]: 29	(0.95, 1.15]: 14	
(154.50, 159.50]: 23	(1.15, 1.45]: 28	
(159.50, 162.50]: 24	(1.45, 1.70]: 16	
(162.50, 169.50]: 26	(1.70, 2.05]: 23	
(169.50, 177.50]: 29	(2.05, 2.85]: 22	
(177.50, ∞): 28	(2.85, ∞): 22	

Table A.8 *The discretized attributes of the 10binaryDis_heart.*

age_1	age_2	age_3	age_4
(-∞, 33.80]: 1	(-∞, 38.60]: 9	(-∞, 43.40]: 39	(-∞, 48.20]: 74
(33.80, ∞): 269	(38.60, ∞): 261	(43.40, ∞): 231	(48.20, ∞): 196
age_5	age_6	age_7	age_8
(-∞, 53.00]: 116	(-∞, 57.80]: 159	(-∞, 62.60]: 216	(-∞, 67.40]: 254
(53.00, ∞): 154	(57.80, ∞): 111	(62.60, ∞): 54	(67.40, ∞): 16
tresbps_1	tresbps_2	tresbps_3	tessbps_4
(-∞, 104.60]: 10	(-∞, 115.20]: 49	(-∞, 125.80]: 110	(-∞, 136.40]: 173
(104.60, ∞): 260	(115.20, ∞): 221	(125.80, ∞): 160	(136.40, ∞): 97
tresbps_5	tresbps_6	tresbps_7	tessbps_8
(-∞, 147.00]: 222	(-∞, 157.60]: 246	(-∞, 168.20]: 259	(-∞, 178.80]: 265
(147.00, ∞): 48	(157.60, ∞): 24	(168.20, ∞): 11	(178.80, ∞): 5
chol_1	thalach_1	thalach_2	thalach_3
(-∞, 169.80]: 9	(-∞, 84.10]: 1	(-∞, 97.20]: 6	(-∞, 110.30]: 17
(169.80, ∞): 261	(84.10, ∞): 269	(97.20, ∞): 264	(110.30, ∞): 253
thalach_4	thalach_5	oldpeak_1	oldpeak_2
(-∞, 123.40]: 40	(-∞, 136.50]: 71	(-∞, 0.62]: 130	(-∞, 1.24]: 173
(123.40, ∞): 230	(136.50, ∞): 199	(0.62, ∞): 140	(1.24, ∞): 97
oldpeak_3	oldpeak_4	oldpeak_5	
(-∞, 1.86]: 213	(-∞, 2.48]: 236	(-∞, 3.10]: 254	
(1.86, ∞): 57	(2.48, ∞): 34	(3.10, ∞): 16	
ca_1	ca_2		
(-∞, 0.30]: 160	(-∞, 0.60]: 160		
(0.30, ∞): 110	(0.60, ∞): 110		

APPENDIX B. SUBGROUP SETS

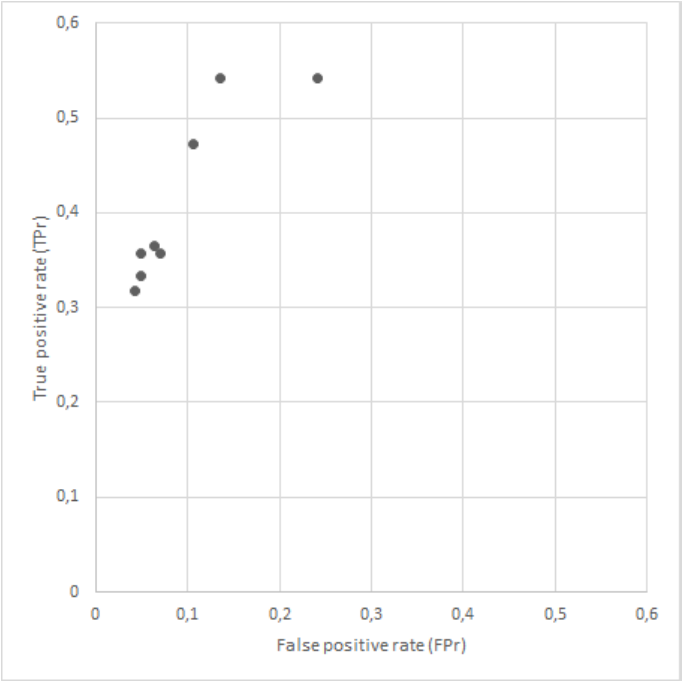

Figure B.1 *Subgroup set 3bins_heart in the ROC space*

Table B.1 *Subgroup set 3bins_heart*

WRAcc	Acc	size	description	TP	FP
0.1018	0.7865	89	exang=1	70	19
0.0914	0.8026	76	fbs=0, exang=1	61	15
0.0766	0.8679	53	thal=7, exang=1	46	7
0.0752	0.6731	104	thal=7	70	34
0.0750	0.8393	56	exang=1, age=(45.00, 61.00]	47	9
0.0713	0.8214	56	slope=2, exang=1	46	10
0.0708	0.8600	50	fbs=0, slope=2, exang=1	43	7
0.0687	0.8723	47	fbs=0, exang=1, age=(45.00, 61.00]	41	6

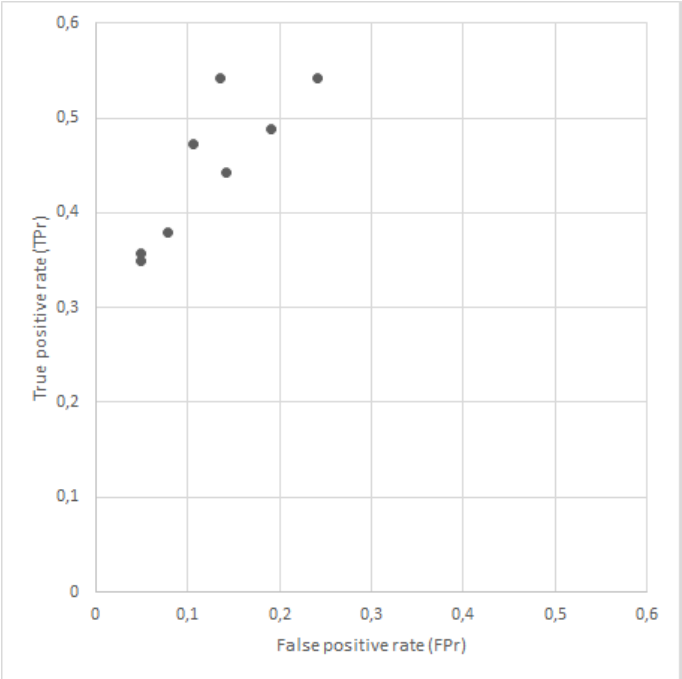

Figure B.2 *Subgroup set 3equalFreq_heart in the ROC space*

Table B.2 *Subgroup set 3equalFreq_heart*

WRAcc	Acc	size	description	TP	FP
0.1018	0.7865	89	exang=1	70	19
0.0914	0.8026	76	fbs=0, exang=1	61	15
0.0766	0.8679	53	thal=7, exang=1	46	7
0.0753	0.8167	60	fbs=0, sex=1, thalach=(-∞, 142.50]	49	11
0.0752	0.6731	104	thal=7	70	34
0.0749	0.7403	77	fbs=0, thalach=(-∞, 142.50]	57	20
0.0747	0.8654	52	exang=1, thalach=(-∞, 142.50]	45	7
0.0741	0.7000	90	thalach=(-∞, 142.50]	63	27

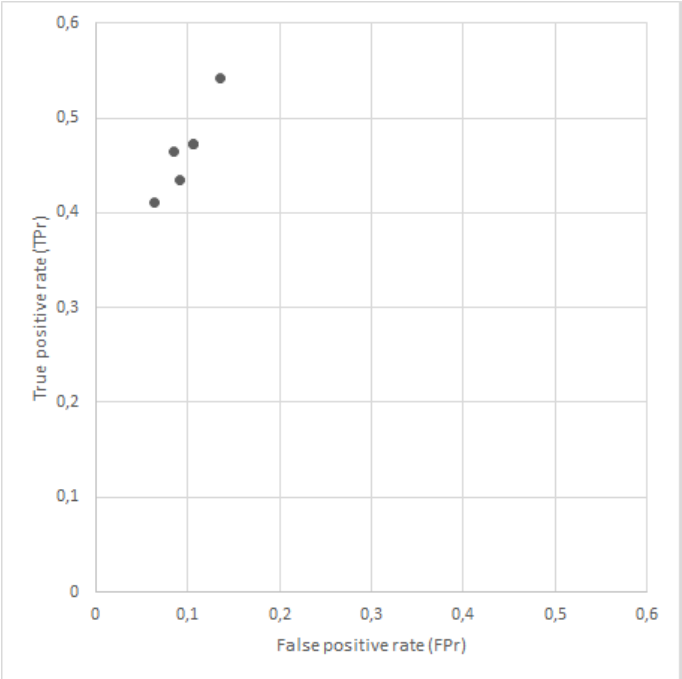

Figure B.3 *Subgroup set 3binary_heart in the ROC space*

Table B.3 *Subgroup set 3binary_heart*

WRAcc	Acc	size	description	TP	FP
0.1018	0.7865	89	exang=1	70	19
0.0948	0.8333	72	exang=1, thalach_2=(-∞, 158.33]	60	12
0.0914	0.8026	76	fbs=0, exang=1	61	15
0.0914	0.8026	76	exang=1, age_1=(45.00, ∞)	61	15
0.0866	0.8548	62	fbs=0, exang=1, thalach_2=(-∞, 158.33]	53	9
0.0853	0.8116	69	exang=1, age_2=(-∞, 61.00]	56	13

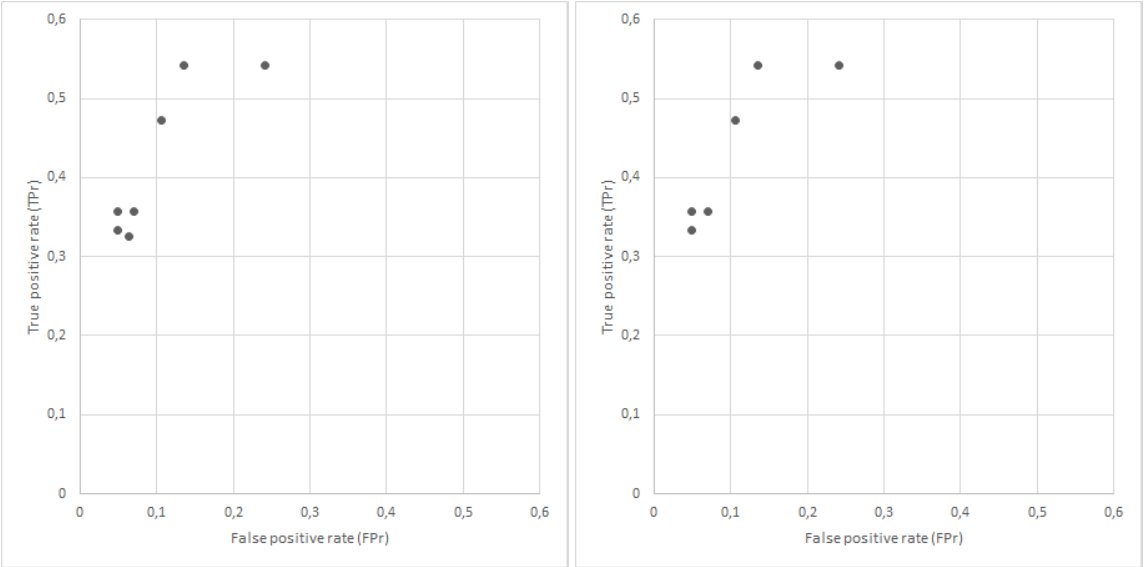

Figure B.4 *Subgroup set 5and10bins_heart (on the left) and the5findBins_heart (on the right) in the ROC space.*

Table B.4 *Subgroup set 5and10bins_heart*

WRAcc	Acc	size	description	TP	FP
0.1018	0.7865	89	exang=1	70	19
0.0914	0.8026	76	fbs=0, exang=1	61	15
0.0766	0.8679	53	thal=7, exang=1	46	7
0.0752	0.6731	104	thal=7	70	34
0.0713	0.8214	56	slope=2, exang=1	46	10
0.0708	0.8600	50	fbs=0, slope=2, exang=1	43	7
0.0653	0.8235	51	restecg=2, exang=1	42	9

Table B.5 *Subgroup set 5findBins_heart*

WRAcc	Acc	size	description	TP	FP
0.1018	0.7865	89	exang=1	70	19
0.0914	0.8026	76	fbs=0, exang=1	61	15
0.0766	0.8679	53	thal=7, exang=1	46	7
0.0752	0.6731	104	thal=7	70	34
0.0713	0.8214	56	slope=2, exang=1	46	10
0.0708	0.8600	50	fbs=0, slope=2, exang=1	43	7

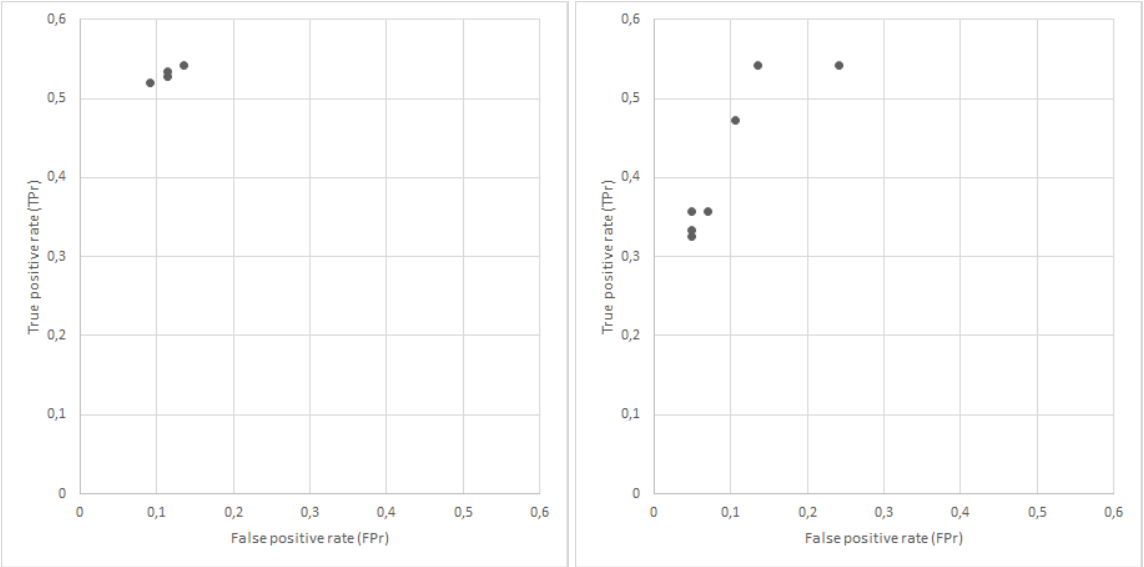

Figure B.5 *Subgroup set 5binary_heart (on the left) and 10findBins_heart (on the right) in the ROC space.*

Table B.6 *Subgroup set 5binary_heart*

WRAcc	Acc	size	description	TP	FP
0.1066	0.8375	80	age_4=(-∞, 67.40], thalach_4=(-∞, 175.80], exang=1	67	13
0.1051	0.8118	85	thalach_4=(-∞, 175.80], exang=1	69	16
0.1032	0.8095	84	age_4=(-∞, 67.40], exang=1	68	16
0.1018	0.7865	89	exang=1	70	19

Table B.7 *Subgroup set 10findBins_heart*

WRAcc	Acc	size	description	TP	FP
0.1018	0.7865	89	exang=1	70	19
0.0914	0.8026	76	fbs=0, exang=1	61	15
0.0766	0.8679	53	thal=7, exang=1	46	7
0.0752	0.6731	104	thal=7	70	34
0.0713	0.8214	56	slope=2, exang=1	46	10
0.0708	0.8600	50	fbs=0, slope=2, exang=1	43	7
0.0688	0.8571	49	exang=1, ca=(0.60, ∞)	42	7

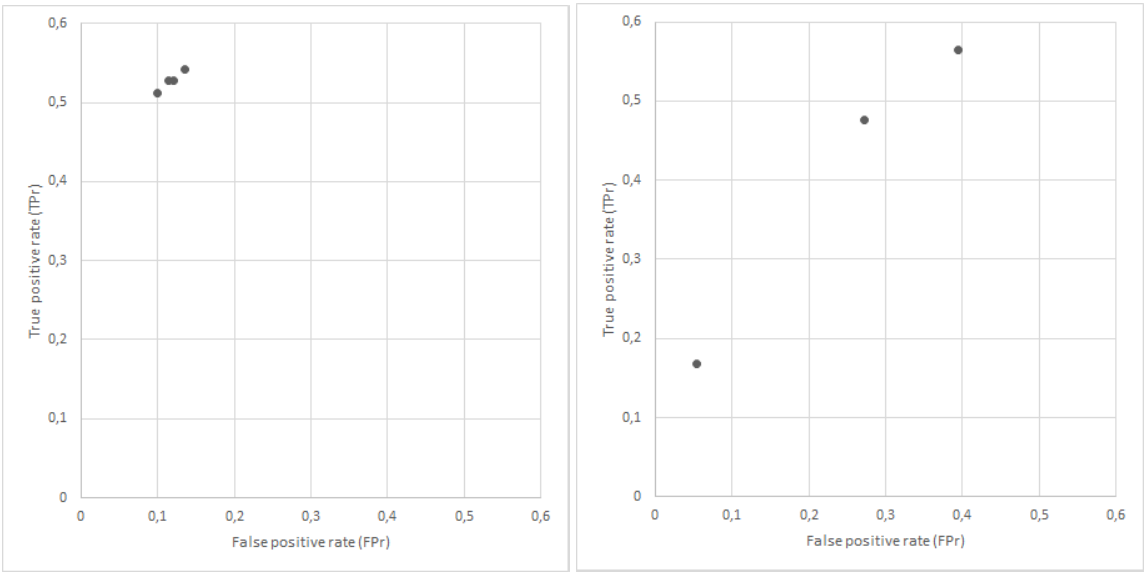

Figure B.6 *Subgroup set 10binary_heart (on the left) and 5findBins_aus (on the right) in the ROC space.*

Table B.8 *Subgroup set 10binary_heart*

WRAcc	Acc	size	description	TP	FP
0.1032	0.8095	84	exang=1, age_8=(-∞, 67.40]	68	16
0.1029	0.8250	80	exang=1, age_8=(-∞, 67.40], trestbps_1=(104.60, ∞)	66	14
0.1018	0.7865	89	exang=1	70	19
0.1014	0.8000	85	exang=1, trestbps_1=(104.60, ∞)	68	17

Table B.9 *Subgroup set 5findBins_aus*

WRAcc	Acc	size	description	TP	FP
0.0370	0.8475	295	A9=1	250	45
0.0309	0.8199	361	A8=1	296	65
0.0206	0.9072	97	A9=1, A10=(-∞, 13.40], A1=0	88	9

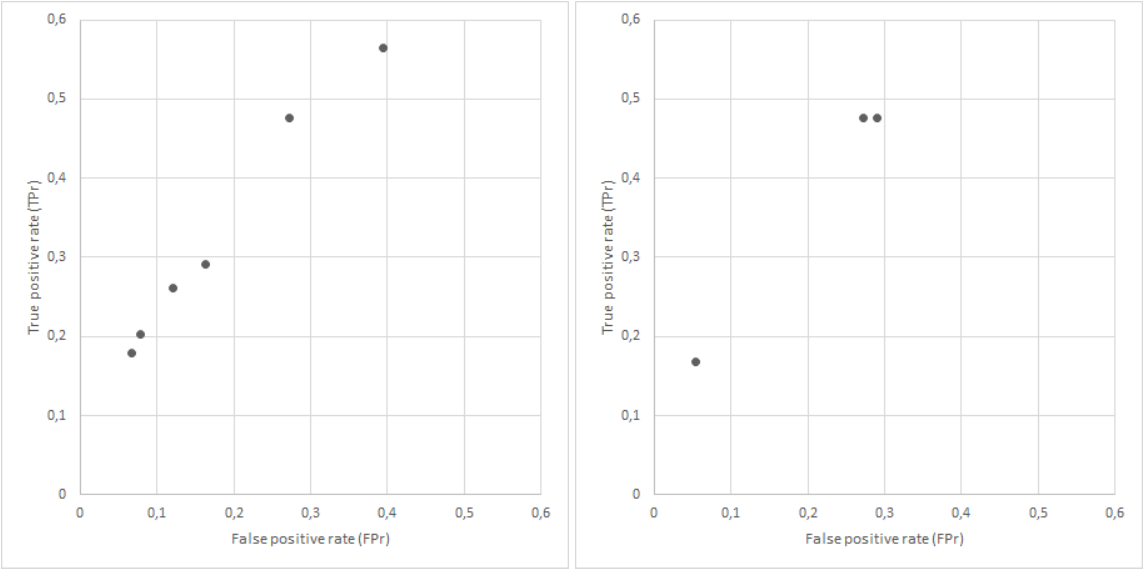

Figure B.7 *Subgroup set 5equalFreq_aus (on the left) and 5binary_aus (on the right) in the ROC space.*

Table B.10 *Subgroup set 5equalFreq_aus*

WRAcc	Acc	size	description	TP	FP
0.0370	0.8475	295	A9=1	250	45
0.0309	0.8199	361	A8=1	296	65
0.0254	0.8726	157	A11=0, A9=1	137	20
0.0233	0.8500	180	A11=0, A8=1	153	27
0.0224	0.8908	119	A11=0, A8=1, A9=1	106	13
0.0204	0.8952	105	A9=1, A1=0	94	11

Table B.11 *Subgroup set 5binary_aus*

WRAcc	Acc	size	description	TP	FP
0.0370	0.8475	295	A9=1	250	45
0.0337	0.8389	298	A2_1=(1617.80, ∞), A8=1	250	48
0.0206	0.9072	97	A9=1, A1=0, A10_1=(-∞, 13.40]	88	9