ACTIONABLE THREE-WAY DECISIONS

A Thesis

Submitted to the Faculty of Graduate Studies and Research

for the Degree of Doctor of Science

In

Computer Science

University of Regina

by

Cong Gao

Regina, Saskatchewan

September, 2018

ABSTRACT

In this thesis, we analyze both the trisecting and acting aspects of three-way decisions. In an evaluation based model of three-way decisions, there are two steps: trisecting and acting. The trisecting step constructs three regions based on an evaluation function and a pair of thresholds. The acting step adopts proper strategies to deal with objects in these regions.

For the trisecting step, this thesis examines statistical interpretations for the construction of three regions. The interpretations rely on an understanding that the middle region consists of normal or typical instances in a population, while two side regions consist of, abnormal or atypical instances. By using statistical information such as median, mean, percentiles, and standard deviation, two interpretations are discussed. One is based on non-numeric values and the other is based on numeric values. For non-numeric values, median and percentiles are used to construct three pair-wise disjoint regions. For numeric values, mean and standard deviation are used. The interpretations provide a solid statistical basis of three-way decisions for applications.

This thesis analyzes a chi-square statistic as a measure for searching for the optimal pair of thresholds for trisecting. An optimization based method for determining the pair of thresholds is to minimize or maximize an objective function that quantifies the quality, cost, or benefit of a trisection. We use the chi-square statistic to interpret and establish an objective function in the context of classification. The maximization of the chi-square statistic searches for a strong correlation between the trisection and the classification.

For the acting step, this thesis introduces actionable strategies to three-way decision. We present a general framework of actionable three-way decisions with four change-based actionable models according to action benefit and action cost. Two of the four models provide the bounds of the cost and benefit and the other two models quantify the maximum benefit under limited cost and the minimum cost for a desired benefit, respectively. We design and analyze algorithms for these models.

To reduce action cost and increase benefit, we introduce the R4 reduction framework for actionable three-way decision. The framework consists of reductions of attributes, attribute-value pairs, classification rules, and actions for creating more benefit and reducing cost. The first three types of reductions are redefined for the context of three-way decisions and the action reduction is proposed. Attribute reduction removes some attributes from all classification rules to reduce the action cost. Attribute-value pair reduction shortens the left hand side of a rule to reduce the action cost without sacrificing any classification power or action benefit. Rule reduction and action reduction remove redundant classification rules and actions, respectively, to reduce computational cost. The Addition strategy for reduction is adapted and its correctness is proven. Based on this strategy, an algorithm for attribute and attribute-value pair reductions is designed.

Finally, we report experimental results to support the proposed four actionable three-way decision models and the R4 reduction framework.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor Dr. Yiyu Yao and my advisor Dr. Howard Hamilton. Dr. Yao has encouraged me to follow my research topics in three-way decisions, has provided me with helpful the needed means and has shared with me his expert knowledge. Dr. Hamilton has advised me on my work in Mitacs project and NSERC project. Also, I would like to thank Dr. Fan Min for his constructive suggestions and comments to my research.

Thanks for my wife, Luyan Xu's understanding to my research, she took care of our daughter, Hayley Gao, for months when I was working on this thesis. My mother, Weizhen Lu, and my mother-in-law, Yuanlan Lu, they came from China to Canada to look after my wife when she was pregnant, and their granddaughter in the first three months after Hayley was born.

Thanks for Steven Swab and his family. Steven is one of my best friends in Canada. He helped me correcting some language issues for the thesis. He also invited me to his son's first birthday party and we discussed some future research topics of this thesis.

The financial support from NSERC, through a Discovery Grant awarded to my supervisor Dr. Yiyu Yao, the Faculty of Graduate Studies and Research, the donors of scholarships offered in the University of Regina, the NSERC grant, ISM Canada, Saskatchewan government, and the Department of Computer Science helped to satisfy my daily basic needs and kept me focused on my research.

POST DEFENSE ACKNOWLEDGEMENT

At this point I wish to thank to my external committee member Dr. Davide Ciucci and internal committee members, Dr. Shaun Fallat, Dr. Howard Hamilton, Dr. Daryl Hepting, and Dr. Xue-dong Yang for their insightful suggestions and comments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1

INTRODUCTION

The purpose of this chapter is to provide the background and overall picture of the present study of three-way decisions. We give our motivations with a brief introduction of related works. We also highlight our contributions and provide the thesis structure.

1.1 State of the Art of Three-way Decisions

The concept of three-way decisions was first introduced by Yao [116] in 2009. The basic idea of three-way decisions is to divide a whole into three parts and process the three parts. This idea is commonly used in human problem solving [21, 109, 115, 117]. In recent years, the interest in both the theory and application of three-way decisions has rapidly increased. For example, there are multiple areas of study which can be categorized as: three-way decisions space [33, 34], three-way classifications [14, 53, 141], three-way clustering [131, 130, 132, 134], three-way concept analysis [35, 48, 72, 76, 77, 91, 110], three-way decisions with game theory [1, 3, 5, 106], three-way recommender systems [2, 136, 137], three-way decision support systems [13, 105], three-way approximations of fuzzy sets [120], three-way approximations of many-valued logic [86], three-way email spam filtering [40, 144], three-way government decisions [54], three-way financial decisions [6], sequential three-way decisions [31, 45, 46, 103, 109], dynamic three-way decisions [104, 139], and many others [29, 38, 39, 47, 49, 50, 55, 84, 114, 133, 143]. These results open up new avenues of research on three-way decisions.

The theory of three-way decisions was originally introduced to interpret three types of rules, known as acceptance, rejection, and non-commitment rules, in the rough set theory proposed by Pawlak [65]. Rough sets use a pair of sets, called lower and upper approximations of a set that represents a class or concept X. The lower approximation is the largest set whose objects belong to X, and the upper approximation is the smallest set that contains the objects of X. When the data is consistent, the two approximations are the same and equal to X. When the data is inconsistent, i.e., some objects have the same attribute values but belong to different classes, then the two approximations are different, and the difference between the two approximation sets is called the boundary of the class X. Based on the pair of approximations, three regions $\mathrm{POS}(X)$, $\mathrm{NEG}(X)$, and $\mathrm{BND}(X)$, called the positive, negative, and boundary regions, can be constructed to approximate X. The $\mathrm{POS}(X)$, $\mathrm{NEG}(X)$, and $\mathrm{BND}(X)$ consist of objects that belong to X, do not belong to X, and are difficult to classify, respectively. The classic Pawlak rough sets do not tolerate any impurity in positive and negative regions. This makes the boundary region big and we can only make decisions for objects in $\mathrm{POS}(X)$ and $\mathrm{NEG}(X)$. To relax this constraint, some initial ideas of the probabilistic rough sets [67] were introduced. They generalized the Pawlak rough sets by allowing impurities existing to exit in the positive and negative regions.

A full and comprehensive, as well as semantically sound, probabilistic rough set model was developed under the name of decision-theoretic rough sets [122] (DTRS), which controls the level of impurity in the three regions by a pair of thresholds (α, β). DTRS introduces the Bayesian Decision Theory to rough sets to compute the pair of thresholds in order to make the decisions with minimal risk or minimal cost. After the introduction of the DTRS model, many other instances of probabilistic rough sets were proposed for obtaining the optimal three regions based on the computation of the pair of thresholds. These approaches adopted different measures from machine learning and data mining to quantify the three regions. For example, the information-theoretic rough sets (ITRS) [15] use information gain as a measure, the Gini index rough sets (GIRS) [141] use Gini index. Other measures such as the chi-square statistic [24], and divergence [4] were also adapted for computing the three regions. The game-theoretic rough sets (GTRS) [106] was introduced to balance two measures such as accuracy and coverage. Other examples of rough sets are confirmation-theoretic rough sets (CTRS) [27], Naive Bayesian rough sets (NBRS) [128], variable precision rough set (VPRS) [145], and Bayesian rough set (BRS) [93]. These approaches generalized rough sets in view of three-way decisions. The idea of three-way decisions can also be applied to other models, such as the shadowed set [68] and the three-way approximations of fuzzy sets [120].

Most of the set-based three-way decision models focus on how to obtain three regions, and do not pay enough attention to dealing with objects in the three regions. DTRS can generate decision rules based on the three regions for processing the objects:

(P) If $P(X|[x]) \geq \alpha$, then decide POS(X);

(N) If $\beta < P(X|[x]) < \alpha$, then decide NEG($X$);

(B) If $P(X|[x]) \leq \beta$, then decide BND(X),

where the $P(X|[x])$ is the conditional probability that an object is in X given that the object is in $[x]$, i.e., equivalence class of x. However, these decision rules only process objects in the corresponding regions, DTRS does not provide a general framework for processing.

The trisecting-and-acting model [114, 117] for three-way decisions is a general framework for three-way decision. This model has two steps, i.e., trisecting and acting. The trisecting step divides a universal set of objects into three pair-wise disjoint regions and the acting step adopts strategies and actions to deal with objects in different regions. Such an idea of dividing and processing is widely used in many applications [6, 9, 26, 36, 45, 52, 63, 70, 72, 80, 82, 88, 91, 134, 138, 136]. For the trisecting step, some statistical interpretations and threshold determination methods were proposed. For example, Yao and Gao [119] interpreted the trisecting of numeric values and non-numeric values by mean, standard deviation, median, and percentile, Azam and Yao [4] used mean and variance to measure a divergence for three regions, and Gao and Yao [24] explained the chi-square statistic as a measure of correlation between three regions and a pair of thresholds. Under the framework of this trisecting-and-acting model, the aforementioned publications focused mainly on the trisecting step and there are a few studies for the acting step. Gao and Yao [22] introduced the concept of change-based acting that is able to construct actionable rules from three regions to promote object movement between regions for benefit. Based on the concept of change-based acting, a general actionable three-way decision (A3WD) framework [22] is formed.

We use Figure 1.1 to give a categorization of these different three-way decision models under the trisecting-and-acting model, where the major contributions of this thesis are highlighted in bold font. As we can see, most of the studies are under the branch of trisecting. The majority of the trisecting methods are set-based, including the rough sets, fuzzy sets, shadowed sets, interval sets, and orthopairs. Under the statistical interpretations for three-way decisions, two statistics based interpretations and two statistics based methods were proposed. The majority of the trisecting methods were focused on determining a pair of thresholds to create the optimal three regions. Under the acting branch, the concept of change-based acting was newly

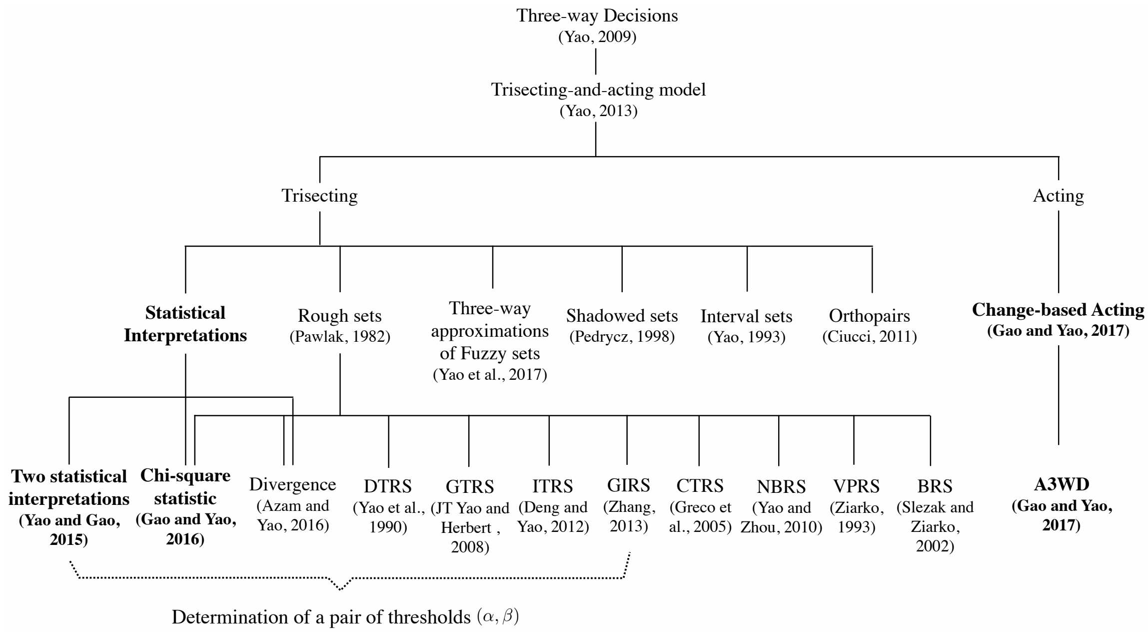


Figure 1.1: A categorization of three-way decision models.

introduced in 2017. Under this change-based acting, the A3WD framework provides the detailed actionable models and algorithms. In the following sections we give our motivations. The details of our contributions are given in following chapters: the two statistical interpretations and the chi-square statistic method for determining the pair of thresholds are introduced in Chapter 4, the change-based acting framework and A3WD models are introduced in Chapter 3, the A3WD algorithms are provided in Chapters 5 and 6.

1.2 Motivations and Related Works

In this section, we give the motivations and objectives of the thesis. Specifically, the motivations are the needs for and the potential values of the statistical interpretations of three-way decisions and actionable three-way decision making.

1.2.1 Trisecting in Three-way Decisions

The basic idea of three-way decisions may be explained in terms of the division and processing of a universal set of objects by using three regions [123]. A two-step trisecting-and-acting model has been proposed [114]. In the first step, one divides the universe into three pair-wise disjoint regions, that is, a trisection of the universe. In the second step, one designs the most effective strategies to process these three regions. The idea of dividing and processing the universe using three regions has been widely used in many fields, such as medicine [36, 63, 82], business [6], engineering and science [26, 144]. Examples of the three-way decision model include rough sets [65], interval sets [111], shadowed sets [69], three-way approximation of fuzzy sets [16], three-way classification [144], three-way clustering [132], orthopairs [11], and many more [14, 45, 54, 56, 70, 80, 134, 136, 138]. By integrating the results from these fields and exemplar models, a theory of three-way decisions has been proposed to study domain independent ways for fast and effective decision making and information processing [107].

Motivation and Objective 1: To Statistically Interpret Trisecting

Typically, a trisection of a universe is based on an evaluation function and a pair of thresholds. The evaluation function assigns each object in the universe an evaluation status value (ESV). The three regions are constructed by collecting, respectively, the set of objects whose ESVs are equal or greater than one threshold, the set of objects whose ESVs are equal or less than another threshold, and the set of objects whose ESVs are between the two thresholds. Therefore, the interpretation and determination of an evaluation function and a pair of thresholds is a fundamental issue for three-way decisions.

A variety of methods have been proposed to interpret and determine a pair of thresholds for trisecting. By generalizing Pawlak rough sets [65], Yao et al. [122]

proposed a decision-theoretic rough set (DTRS) model by using probability as an evaluation function and a pair of thresholds on probability to derive three regions known as the probabilistic positive, negative and boundary regions. The pair of thresholds is determined and interpreted by Bayesian decision theory. Yao and Herbert [106] proposed a game-theoretic rough set (GTRS) model, in which the pair of thresholds is determined by designing a game. Deng and Yao [15] and Zhang [140] investigated information entropy and the Gini index in rough sets, respectively, and the pair of thresholds is determined by maximizing the information gain or purity of the three regions. These probabilistic models of rough sets can be considered as specific examples of three-way decisions.

Some studies exploring statistical information and methods have bridged the gap between statistics and three-way decisions. For example, the sequential hypothesis testing framework introduced by Wald [96] influenced three-way decisions. If a test strongly supports a hypothesis, one accepts the hypothesis; if the test is strongly against the hypothesis, one rejects the hypothesis; otherwise, one performs further tests. Based on these further tests, the original hypothesis will be accepted or rejected. Due to the sequential nature of Wald's method, one can either accept or reject some hypotheses without the need for further testing. This sequential processing has the benefits of efficiency and effectiveness. Yao [109] adopted this idea to sequential three-way decisions and granular computing [69, 112]. Azam and Yao [4] used the variance and the mean to interpret the quality of a trisection, and therefore to find an optimal pair of thresholds that has best quality of three regions. These studies combining statistics and three-way decisions have provided a solid research foundation.

The first motivation of this thesis is that we further examine a general statistical interpretation of three-way decisions. We use statistical notions, such as mean, median, standard deviation, central tendency and dispersion, to interpret and determine the pair of thresholds for trisecting. Objects around the mean form one region, and regions on either side of mean form the other two regions. In particular, when $\mathbb{V}$ is a set of non-numeric values, some statistics such as median and mode, together with measures such as percentile and quartile can be used to trisect the OB. When $\mathbb{V}$ is a set of numeric values, some statistics such as mean, standard deviation, and moment can be used to trisect the OB. In other words, we search for statistical interpretations of three-way decisions that enable us to examine the structures of the data and to make inferences about that data.

Motivation and Objective 2: To Find the Optimal Pair of Thresholds for Trisecting

The problem of finding the optimal pair of thresholds that produces the optimal trisection is one of the central problem in three-way decisions. The evaluation-based three-way decision model [119] uses an evaluation function $e(\cdot)$ to map all objects into a totally ordered set $(\mathbb{V}, \succeq)$, and according to a pair of thresholds $(\alpha, \beta) \in \mathbb{V} \times \mathbb{V}$ with $\alpha \succ \beta$ (i.e., $\alpha \succeq \beta$ and $\neg(\beta \succeq \alpha)$), a universal set of objects can be divided into three regions: a region consists of objects whose values are at or above one threshold, a region of objects whose values are at or below the other threshold, and a region of objects whose values are between the two thresholds. To determine the optimal pair of thresholds, one method is to construct a meaningful objective function measuring the quality of trisections; the required pair of thresholds maximizes or minimizes the objective function. Examples of qualitative measures of a trisection are cost [121], Gini index [140], and information entropy [16].

The optimal trisection obtained by a pair of thresholds can be considered as the best approximation of a classification [24, 114]. Consider a classification problem in which all objects in OB are classified into one of two categories $\{X, X^C\}$, where X is a set of objects belonging to the given class and X^C is the set of objects not belonging to the given class. A fundamental task is to construct rules or a description function to achieve such a classification. Two-way classification models are typically used for such a task. However, these models may not produce desirable results with acceptable classification errors. In three-way classification [114], a trisection $\pi_{(\alpha,\beta)}(X) = (\mathrm{POS}_{(\alpha,\beta)}(X), \mathrm{BND}_{(\alpha,\beta)}(X), \mathrm{NEG}_{(\alpha,\beta)}(X))$ as an approximation of $\{X, X^C\}$ is obtained by a pair of thresholds (α, β) on an evaluation function. Different choices of thresholds lead to different three-way approximations. A good approximation shows a strong association or correlation between $\pi_{(\alpha,\beta)}(X)$ and $\{X, X^C\}$. In other words, $\pi_{(\alpha,\beta)}(X)$ and $\{X, X^C\}$ are correlated or dependent. The chi-square statistic is a measure of correlation and can be used as an objective function for measuring the goodness of a trisection [24]. The maximization of the chi-square statistic suggests the strongest correlation between a trisection and $\{X, X^C\}$. Therefore, the optimal pair of thresholds can be determined by maximizing the chi-square statistic.

1.2.2 Acting in Three-way Decisions

In a set-theoretical setting, three-way decisions can be formulated as a two-step process within a trisecting-and-acting model [117]. The trisecting step divides a universal set of objects into three pair-wise disjoint regions. The acting step adopts strategies to process objects in the three regions. We use elections as an example to illustrate the main ideas of the trisecting-and-acting model of three-way decisions. Based on an opinion poll, one typically divides a set of voters into three groups: voters who support the candidate, voters who oppose the candidate, and voters who are undecided or unwilling disclose their decisions. According to the poll results, the candidate may act to retain the group of supporters, to persuade the undecided voters, and to gain support from voters who oppose him / her.

Motivation and Objective 3: To Model an Actionable Three-way Decision Framework

Existing studies on three-way decisions focus mainly on the trisecting step [4, 15, 16, 24, 106, 141]. There is relatively little investigation into the acting step for devising actionable strategies. In this thesis, we combine ideas from actionable rule mining and three-way decisions to build a model of actionable strategies in three-way decisions. Drawing from the election example, we look at a framework in which actionable strategies facilitate the movement of voters from unfavorable regions to favorable regions. We represent and interpret actionable strategies in terms of the notation of actionable rules and action rules in machine learning and data mining [74, 75, 89, 90, 98, 102].

Silberschatz and Tuzhilin [89] introduced the concept of actionability that a user can react to realize his or her advantage. Ras and Wieczorkowska [75] adopted action rules to mine profitable patterns for banks. Yang et al. [102] introduced a postprocessing decision tree method to find beneficial actions. Su et al. [94] searched actionable behavioral rules with a high utility. Many studies on actionable rules cover topics in data mining and machine learning, such as association rule mining [51, 87], classfication [18, 74, 75, 101, 102], clustering [41, 62, 135], and outlier detection [10, 44]. To calculate action cost or measure the actionability of rules, attributes are categorized into two types [74, 75, 102]: attributes whose values are changeable or unchangeable, which we call flexible or stable attributes, respectively. Rules constructed from flexible attributes are actionable and those constructed from stable attributes are non-actionable. Issues of actionable rules such as comparative study are discussed in literature [43]. In the context of three-way decisions, we adopt actionable rules for moving objects from unfavorable regions to favorable regions to produce benefit or create value.

Based on the analysis of benefits and costs of actionable rules, an actionable three-way decision framework is required for different situations in practice. For example, a company may require an advertising solution for a product which obtains the maximum profit with a limited budget, or a candidate in an election demands an actionable solution (e.g., issue some polices) with the lowest cost to win an election. These problems are common constraint optimization problems which exist in many areas of our daily lives. With an actionable three-way decision framework, we can design four actionable models to deal with each type of problem. The first two models find the upper bounds of benefit and cost, the third model finds the maximum benefit under limited cost, and the last model finds the minimum cost for obtaining a desired benefit.

Motivation and Objective 4: To Decrease Cost and Increase Benefit for Actionable Three-way Decision Models

There are redundant attributes, attribute-value pairs, rules, and actions when building the actionable three-way decision framework. We can decrease the action cost, increase benefit, and speed up the computation of the four actionable models by removing these redundancies. In the actionable three-way decision framework, the cost of action is strongly related to the number of attributes whose values need to be changed in order to transfer objects. If some redundant attributes can be removed, the cost may decrease or the benefit may increase. Furthermore, the computation time of the four actionable models depends on the size of solution space, the fewer the number of classification rules and actions, the faster the computation.

In this thesis, the proposed R4 reduction framework removes the redundant attributes, attribute-value pairs, rules, and actions. On the one hand, removing redundant stable attributes may increase the number of possible actions which may increase the benefit because more objects can be transferred. On the other hand, removing redundant flexible attributes can decrease the action cost because it is not necessary to change the removed attributes' values. Removing redundant attribute-value pairs for each rule, which is also called rule simplification, shortens a rule and requires fewer attributes changes, such that the action cost can also be reduced. The removing of rules and actions will not change the cost or benefit, however, it will reduce the size of the solution space, and that can reduce the computation time.

1.3 Contributions

In this thesis, we analyze three-way decision theory on concerning trisecting and acting, and propose an actionable three-way decision (A3WD) framework. With regard to three-way decisions, the contributions of this thesis are as follows:

For the trisecting of A3WD, we present: (1) two statistical interpretations; and (2) one chi-square statistic based method for determining the pair of thresholds.

For acting of A3WD, we propose: (1) four actionable three-way decision models; (2) four actionable rule mining algorithms for these models; (3) the R4 reduction framework, a comprehensive, four-step approach for improving the performance of actionable three-way decision models; (4) an Addition algorithm schema for the first two steps of the R4 framework and proof of its correctness; (5) specific instances of this schema for both attribute reduction and attribute-value pair reduction (i.e., rule simplification).

1.4 Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 reviews the trisecting-and-acting three-way decision model. Based on two illustrative examples, Chapter 3 introduces the actionable three-way decision framework and its four models. In this framework, some assumptions are discussed and a cost-benefit analysis of actions is given. Chapter 4 presents two statistical interpretations for trisecting and a chi-square statistic based method for determining the values of the pair of thresholds. Chapter 5 proposes four algorithms for mining actions for the four models introduced in the Chapter 3. Chapter 6 introduces the R4 reduction framework that consists of four steps of reduction for decreasing action costs and increasing benefits. In the R4 framework, an Addition algorithm schema for attribute reduction and attribute-value pair reduction is proposed and its correctness is proven. Chapter 7 experimentally evaluates the effectiveness of the four proposed actionable models and the R4 framework. The last chapter concludes the thesis and presents some future works. All proofs of property, proposition, lemma, and theorem are given in the appendix.

Chapter 2

A TRISECTING-AND-ACTING MODEL OF THREE-WAY DECISIONS

This chapter overviews the trisecting-and-acting three-way decision models, introduces basic notions and formal setting used throughout the thesis, and provides general ideas of statistical interpretations of trisecting and change-based acting.

2.1 An Overview of Three-way Decisions

The basic ideas of three-way decisions can be explained in terms of the division and processing of a universal set of objects in three regions [114, 117, 119]. Figure 2.1 shows a two-step, trisecting-and-acting three-way decision model. The first step, called trisecting, divides a universal set OB into three pair-wise disjoint regions, called region L, region M, and region R, respectively, which may be viewed as the left, middle, and right regions in an evaluation based three-way decision model [114]. The names of three regions may vary from application to application. For example, in decision-theoretic rough sets [108], three regions are named POS, BND, and NEG, standing for the positive, boundary, and negative regions, respectively. The second step, called acting, adopts effective strategies to process three regions. Actions can be taken to enhance the effectiveness or quality of the trisection [89, 90].

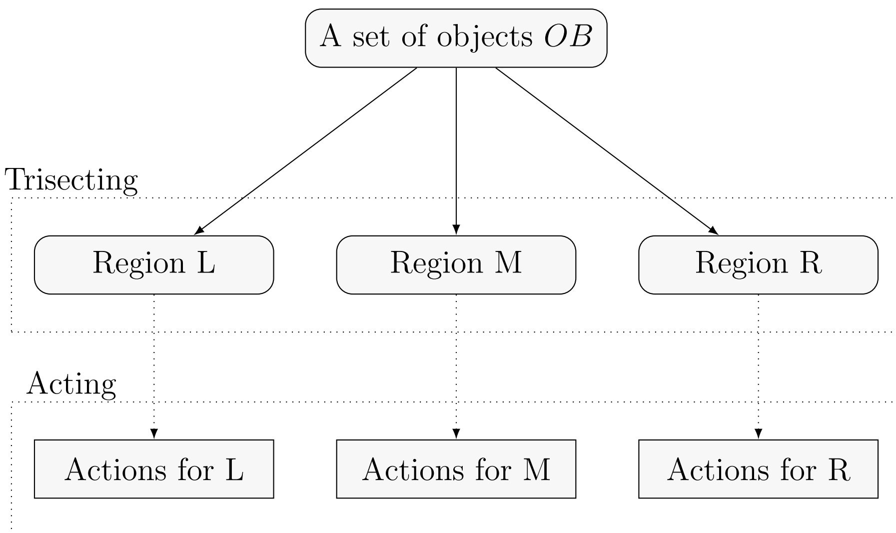


Figure 2.1: Trisecting-and-acting three-way decision model [114].

Ideas of dividing and processing the universe with three regions have been widely used in many fields, such as medicine [36, 63, 82], social networks [70], recommender systems [136], investment [6], and engineering and sciences [9, 26, 45, 52, 72, 80, 88, 91, 134, 138]. In some applications, the trisecting and acting are woven together as one and cannot be easily separated. A good trisection depends on the acting strategies for processing regions and conversely, effective acting relies on an appropriate trisection.

2.1.1 Trisecting

With respect to trisecting, we may use evaluation-based methods [119] to divide the universe OB into three pair-wise disjoint regions. An evaluation function $e(\cdot) : OB \longrightarrow \mathbb{V}$ that maps each object in the universe to an evaluation status value (ESV) in a totally ordered set $(\mathbb{V}, \succeq)$. A pair of thresholds $(\alpha, \beta) \in \mathbb{V} \times \mathbb{V}$ with $\alpha \succ \beta$ (i.e., $\alpha \succeq \beta \wedge \neg(\beta \succeq \alpha)$) trisects the ESVs into three sections. Objects are divided into three pair-wise disjoint regions according to their ESVs:

$$
\begin{aligned}
\mathrm{L}_{(\alpha,\beta)}(e) &= \{x \in OB \mid e(x) \succeq \alpha\}, \\
\mathrm{M}_{(\alpha,\beta)}(e) &= \{x \in OB \mid \beta \prec e(x) \prec \alpha\}, \\
\mathrm{R}_{(\alpha,\beta)}(e) &= \{x \in OB \mid e(x) \preceq \beta\}, \qquad (2.1)
\end{aligned}
$$

where the $e(x) \preceq \beta$ means $\neg(e(x) \succ \beta)$. The region L consists of objects with ESVs greater than or equal to one threshold, region R consists of objects with ESVs less than or equal to the other threshold, and region M consists of objects with ESVs between the two thresholds. In real applications, many evaluation functions can be designed, such as probability functions [65], fuzzy membership functions [16], and the Stanford-Binet test of IQ [81].

For example, because the human body is sensitive to changes in blood pressure, we may use blood pressure to predict health. In the classification of people's systolic blood pressure (sbp), OB is a set of people, $e(\cdot)$ is a measure of people's sbp in millimeters of mercury, $\mathbb{V}$ is a set of integers denoting sbp readings, and the order $\succeq$ is relational operator $\geq$. Generally, people with high blood pressure may have or be more likely to have certain diseases, people with low blood pressure may have or be more likely to have other health problems. Therefore, people can be trisected into three groups, called the hypotension, normal, and hypertension regions, via a pair of thresholds, e.g., $(\alpha = 140, \beta = 90)$ [63]. That is, people $x \in OB$ with $e(x) \geq 140$, $90 < e(x) < 140$, and $e(x) \leq 90$ is considered to have high blood pressure, normal blood pressure, and low blood pressure, respectively. With respect to different clinical cases, the pair of thresholds may be determined differently. Trisecting can be applied again for any of these regions if needed. For example, in ambulatory blood pressure analysis [95], the hypertension region is again trisected into the white coat hypertension region, the dippers with ambulatory hypertension region, and the nondippers with ambulatory hypertension region.

Generally, the goodness, quality, or cost of a trisection can be measured by an objective function [24]:

$$Q(\pi) = w_L Q(\mathrm{L}) + w_M Q(\mathrm{M}) + w_R Q(\mathrm{R}), \tag{2.2}$$

where $Q(\pi)$ is the goodness (quality, cost, or other measurement) of the trisection $\pi = (\mathrm{L}, \mathrm{M}, \mathrm{R})$, $Q(\mathrm{L})$, $Q(\mathrm{M})$, and $Q(\mathrm{R})$ are qualities or goodness of the regions L, M, and R, respectively, and w_L, w_M, and w_R are weights associated to different regions, representing their relative importances. Examples of objective functions are cost [108], information entropy [15], Gini index [141], chi-square statistic [24], and variance [4]. We have $w_L = w_M = w_R = 1$, if the three regions are treated equally.

The optimal trisection is the one that maximizes or minimizes the objective function in Equation (2.2), according to criteria used in particular applications. For example, the objective functions based on cost [108], information entropy [15], and Gini index [141] are to be minimized and those based on chi-square statistic [24] and variance [4] should be maximized.

2.1.2 Acting

In the acting step, strategies and actions for processing each region take a decision maker's advantage. These strategies and actions may be adopted to handle different applications, such as description of concept, prediction of objects, and transference of objects. To fully understand these three regions, descriptive rules can be constructed from objects in the three regions. Descriptive rules summarize the main features of each region and each rule characterizes a portion of a specific region. We can also construct predictive rules from the three regions to classify or cluster new instances. In some situations, the decision maker desires to transfer objects between regions. We can use actionable rules to achieve this purpose. Take classification as an example, the trisecting step obtains three regions with respect to an objective class X, i.e., POS, BND, and NEG, representing the regions of objects that are classified in X, not in X, and difficult to classify, respectively. The acting step may construct classification rules for POS and use these classification rules to classify unseen objects. In three-way concept analysis, the acting step constructs descriptive rules for a concept to describe and summarize the properties of the concept.

By acting, the quality or effectiveness of trisection can be improved or the cost can be reduced. We give some examples to show this idea. In military triage [36], victims are divided into three categories: those who are likely to survive regardless of what care they receive, those who are unlikely to survive regardless of what care they receive, and those for whom immediate care might make a positive difference in outcome. A decision maker may adopt strategy of giving treatment priority to the last category in order to cure the most victims when the medical supply is limited. In a loan application, a bank may categorize the applicants into three groups with respect to their credit rating, one group consists of applicants whose credit ratings are high, one group consists of applicants whose credit ratings are medium, and the last group consists of the clients whose credit ratings are low. In order to make profit, the bank may adopt different acting strategies for these groups of clients: approve the loan applications from the first group, carefully review and approve a portion of applications from the second group, and reject the applications from the last group. In making a governmental decision, a government may categorize citizens into three groups with respect to their health conditions, one group consists of healthy people, one group consists of relatively healthy but sometime having health problems, the last group consists of people who have bad health problems. In order to more effectively spend government budget, in the acting step, the governor may take different actions for different groups: do nothing for the first group, improve the living environment for the second group of people to avoid deterioration of their health conditions, offer periodical medical examination for the third group and treat them. In business, a company produces a product and people are categorized into three groups: people who will probably buy the product, people who probably will not, and people who are ignorant of or indifferent towards the product. In order to create profit from the product, the company may do nothing to the first group, but advertise on TV and Internet for the second and third group.

2.2 Statistical Analysis for Trisecting

From a statistical point of view, a good trisecting step aims to obtain a trisection that has good statistical properties such as central tendency, distribution, or other aspects implied in OB. The statistical analysis for trisecting is to analyze trisecting and generate three regions through statistical methods.

There is some research on the statistical analysis of trisecting. For example, Wald [96] used a sequential hypothesis testing in three-way decision problems. If a test strongly supports a hypothesis, one accepts the hypothesis; if the test is strongly against the hypothesis, one rejects the hypothesis; otherwise, one performs further tests. Based on further tests, the original hypothesis will either be accepted or rejected. Yao [109] adapted these ideas of sequential processing to three-way decision and granular computing [69, 112]. Yao and Gao [119] statistically analyzed the evaluation-based trisecting step when the set of ESVs, i.e., $\mathbb{V}$ is a set of non-numeric values and numeric values. When $\mathbb{V}$ is a set of non-numeric values, statistics such as median and mode, together with percentiles and quartiles can be used to trisect the OB. When $\mathbb{V}$ is a set of numeric values, statistics such as mean and standard deviation can be used to trisect the OB. In order to determining the values of a pair of thresholds, Gao and Yao [24] suggested to maximize the chi-square statistic, and

Azam and Yao [4] used a divergence that is constructed based on variance and mean.

2.3 Change-based Acting

A change-based strategy facilitates transference of objects between regions. Such a strategy analyzes similarity and dissimilarity between the objects in different regions, in order to design strategies of action so that objects in one region may be transferred to another region. In the election example, a candidate may want to persuade some voters who are undecided to be supporters by addressing specific relevant issues.

The change-based acting promotes movements of objects among regions by changing objects' attribute values. According to the decision maker's needs, we can categorize these directions of movements into two main types: (1) possible and (2) impossible, in which possible (or impossible) direction means the movement of this direction is possible (or impossible). Additionally, the possible movements consist of three sub-types: (1) desirable, (2) undesirable, and (3) indifferent, respectively, indicating that the decision maker wants the movement to happen, does not want the movement to happen, and does not care. Given two regions, we show all possible movements in Figure 2.2. The three types of possible movements are shown in different types of lines. The solid lines, dashed lines, and solid lines with slashes represent desirable, indifferent, and undesirable movements, respectively. Accordingly, actions can be classified into corresponding types, i.e., desirable actions, indifferent actions, and undesirable actions, respectively. Generally, there is one and only one possible movement from one region to another.

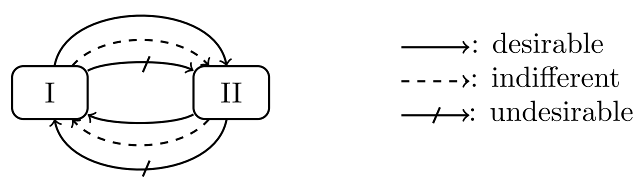


Figure 2.2: Six possible movements between two regions.

Chapter 3

AN ACTIONABLE THREE-WAY DECISION FRAMEWORK

This chapter introduces actionable rules to the trisecting-and-acting three-way decision model and proposes an actionable three-way decision framework, or A3WD for short. In this framework, four change-based A3WD models are proposed based on the analysis of action cost and benefit. Two of the four models provide the bounds of the cost and benefit and the other two models quantify the maximum benefit under cost constraint and the minimum cost for a desired benefit.

3.1 Two Illustrative Examples

This section illustrates the main ideas of A3WD by two examples, one for medicine and the other for election. The two examples are not based on real data, but the method can be used for dealing with real data.

3.1.1 A Medical Example

Doctors use symptoms to determine disease and cure disease by taking proper actions that change abnormal symptoms to be normal. Table 3.1 is an example decision table describing relationship between heart disease and some symptoms. The table consists of 9 patients (rows) and 4 symptoms or attributes (columns). *chol* and *bp* stand for cholesterol level and blood pressure, respectively. The first three attributes are symptoms and the last column is the diagnosis of the heart disease. Symbols - and + denote that a patient has heart disease and does not have heart disease, respectively.

Table 3.1: A decision table for medicine.

#	sex	chol	bp	result
o_1	female	medium	normal	+
o_2	female	medium	normal	-
o_3	female	low	normal	+
o_4	female	low	normal	-
o_5	female	low	normal	-
o_6	female	medium	low	+
o_7	female	high	high	-
o_8	male	high	low	-
o_9	male	low	normal	+

In the trisecting step, a doctor trisects the patients into three regions, in which region R_+ consists of people who are considered not to have heart disease, region R_- consists of people who are considered to have heart disease, and region $R_?$ consists of people who cannot be determined based on their symptoms. o_1 and o_2 have the same symptoms but have different diagnosis results, similar to o_3, o_4, and o_5. This means that we cannot correctly classify these patients based on their symptoms. Based on different criteria, a doctor may get different trisections of patients. These criteria could be to minimize risk (or cost) [122], minimize uncertainty [15], minimize impurity [140], balance accuracy and coverage [106], or maximize some statistical measures [4, 24]. In the next chapter, we analyze some statistical information from the data and propose a method to classify these patients into three regions with maximum correlation to the diagnosis results. For convenience, we suppose that a doctor trisects these patients into three regions $\mathrm{R}_+ = \{o_1, o_2, o_6, o_9\}$, $\mathrm{R}_- = \{o_7, o_8\}$, and $\mathrm{R}_? = \{o_3, o_4, o_5\}$. The construction of these three regions is illustrated in Example 3.1 in Section 3.4.

In the acting step, suppose we want to cure patients who have heart disease. Figure 3.1 shows the desirable, undesirable, and indifferent movement patterns. It should be noted that movement from R_- to $\mathrm{R}_?$ could be desirable.

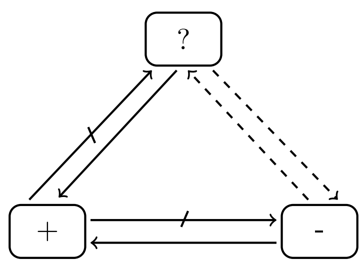


Figure 3.1: Movement patterns of Table 3.1 based on user's requirement.

By analyzing o_7 and o_8 in R_-, we construct two rules as follows:

$$r_7: \quad sex = female \wedge chol = high \wedge bp = high \Rightarrow result = -,$$
$$r_8: \quad sex = male \wedge chol = high \wedge bp = low \Rightarrow result = -.$$

A rule $X \Rightarrow Y$ indicates that if a patient has symptoms X then the patient has a diagnosis Y, where the left part X consists of conjunction of attribute-value pairs for symptoms and the right part Y is the diagnosis result. Similarly, we can construct rules for objects in R_+:

$$r_1: \quad sex = female \wedge chol = medium \wedge bp = normal \Rightarrow result = +,$$
$$r_6: \quad sex = female \wedge chol = medium \wedge bp = low \Rightarrow result = +,$$

$$r_9: \quad sex = male \wedge chol = low \wedge bp = normal \Rightarrow result = +.$$

We assume that patients will have the same diagnosis results if they have the same symptoms. Then patients in region R_- can be cured if we can take actions to change their attribute-value pairs to look like the ones in R_+. Generally, sex cannot be changed, while cholesterol level and blood pressure can be changed. Therefore, we may adopt actions to change o_7's cholesterol level and blood pressure to be the same as that of a patient from R_+. Patients o_1 and o_6 in R_+ have the same sex as o_7. As a result, o_1 or o_6 can be chosen as a reference to design actions. If o_1 is chosen, the action may be designed as follows:

a_1 : Reduce the cholesterol level from high to medium by taking 3 doses of medicine A and lower blood pressure from high to normal by taking 4 doses of medicine B.

If o_6 is chosen, the action may be:

a_2 : Reduce the cholesterol level from high to medium by taking 3 doses of medicine A and lower blood pressure from high to low by taking 8 doses of medicine B.

We notice that o_1 and o_2 have the same symptoms but different diagnosis results, which means a patient has about a 50% chance to be cured if a doctor chooses o_1 as a reference to design actions. As for o_6, there is no such ambiguity. Therefore we believe that action a_2 is more likely to cure o_7. The strategies to cure people in $R_?$ can be analyzed similarly.

Taking any action may produce benefit and incur cost. In this example, the benefit is that patients may be cured and the cost may be money, time, and / or other resources that the actions require. For treating o_7, actions a_1 and a_2 have the same benefit if o_7 is cured (or a_2 is more likely to cure o_7). Generally, the costs of two solutions may be difficult to compare. Because different decision makers may have different preferences and criteria. For simplicity, we may consider that the action cost is a combined cost of all types of resources demanded by actions.

3.1.2 An Election Example

Table 3.2 describes some voters in an opinion poll, where *hc* denotes voter's health care level and *fp* denotes the voter's attitude towards the nation's foreign policies. Values of *hc* can be high, medium, and low. Values of *fp* can be open, neutral, and closed. The attribute *result* means the voter's decision, values +, -, and ? denote a voter supports the candidate, oppose the candidate, and has not decided or is not willing to divulge his / her decision.

Table 3.2: A decision table of poll.

#	age	hc	fp	result
o_1	18-29	high	neutral	+
o_2	30-49	medium	closed	-
o_3	50-64	low	open	-
o_4	65+	low	open	-
o_5	18-29	medium	neutral	?
o_6	65+	high	closed	+
o_7	50-64	low	closed	?
o_8	30-49	high	open	+
o_9	18-29	medium	open	-

Three regions are constructed based on the voters' decisions: $\text{R}_+ = \{o_1, o_6, o_8\}$, $\text{R}_- = \{o_2, o_3, o_4, o_9\}$, and $\text{R}_? = \{o_5, o_7\}$. A candidate may win the election if he / she obtains at least 50% support from a poll, which means the candidate needs to transfer at least 2 voters from R_- or $\text{R}_?$ to R_+ to win the election. By analyzing the attributes in the table, we can determine that *age* is a stable attribute whose value cannot be easily changed and *hc* and *fp* are flexible attributes whose values can be changed by taking some actions.

The candidate wants to retain voters in region R_+, move voters from R_- and $\text{R}_?$ to R_+, and avoid moving voters from R_- to $\text{R}_?$. The movement patterns are shown in Figure 3.2.

We first construct rules from region R_+:

$$r_1: \quad age = 18\text{-}29 \wedge hc = high \wedge fp = neutral \Rightarrow result = +,$$

$$r_6: \quad age = 65 + \wedge hc = high \wedge fp = closed \Rightarrow result = +,$$

$$r_8: \quad age = 30\text{-}49 \wedge hc = high \wedge fp = open \Rightarrow result = +.$$

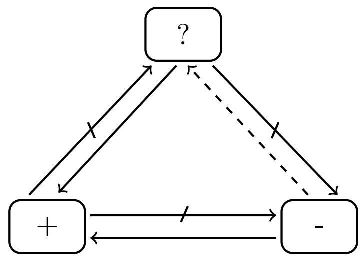


Figure 3.2: Movement patterns for the election poll.

There are four voters in R_- or $\mathrm{R}_?$ whose ages are the same as some objects in R_+. For example, o_2's age is the same as o_8. Therefore, a rule for action can be constructed:

$$r_2 \rightsquigarrow r_8 : hc : medium \rightsquigarrow high \wedge fp : closed \rightsquigarrow open \Rightarrow result : - \rightsquigarrow +,$$

where r_2 and r_8 are two rules constructed based on o_2 and o_8, expression $X \rightsquigarrow Y$ means that X is changed to Y, and $r_2 \rightsquigarrow r_8$ is a rule which indicates that the attribute-value pairs are changed from r_2 to r_8, accordingly. The rule $r_2 \rightsquigarrow r_8$ denotes that if we take some actions for the voters in the 30-49 age group to improve their health care level from medium to high and their attitude towards the current foreign policies from closed to open, then they may change their mind to support the candidate. Similarly, we can get the following rules for o_4, o_5, and o_9:

$$r_4 \rightsquigarrow r_6: \quad hc : low \rightsquigarrow high \wedge fp : open \rightsquigarrow closed \Rightarrow result : - \rightsquigarrow +,$$

$$r_5 \rightsquigarrow r_1: \quad hc: medium \rightsquigarrow high \wedge fp: neutral \rightsquigarrow neutral \Rightarrow result: ? \rightsquigarrow +,$$

$$r_9 \rightsquigarrow r_1: \quad hc: medium \rightsquigarrow high \wedge fp: open \rightsquigarrow neutral \Rightarrow result: - \rightsquigarrow +.$$

Now, we have four rules, each transfers one non-supporter to be a supporter. The candidate may choose two of them with minimum costs to achieve 50% of support.

3.2 Actionable Rules

We give a formal definition of decision table.

Definition 3.1 *A* ***decision table*** *is a tuple*

$$S = (OB,\ AT = A_s \cup A_f \cup \{d\},\ \{V_a \mid a \in AT\},\ \{I_a \mid a \in AT\})$$

where OB is a nonempty finite set of objects, AT is a finite nonempty set consisting of attributes composed by three subsets, in which A_s are ***stable attributes****, A_f are* ***flexible attributes*** *and d is a* ***decision attribute****, V_a is a nonempty set of values for every attribute $a \in AT$, and $I_a : OB \longrightarrow V_a$ is a mapping. For every $x \in OB$, attribute $a \in AT$, and value $v \in V_a$, $I_a(x) = v$ means that the object x has the value v for attribute a.*

In Definition 3.1, the entire set of attributes is categorized into three subclasses. Stable attributes are attributes, such as age and sex, whose values cannot be modified, flexible attributes are attributes, such as cholesterol level and blood pressure, whose values can be modified by actions. All attributes in $A_s \cup A_f$ are called condition attributes or features, while the attribute d is also called class label.

Given an object $x \in OB$, $[x]_{A_s \cup A_f}$ is the equivalence class of x based on the values of attributes $A_s \cup A_f$:

$$[x]_{A_s \cup A_f} = \{y \in OB \mid I_a(y) = I_a(x), \forall a \in A_s \cup A_f\}.$$

In the remainder of this thesis, we use $[x]$ instead of $[x]_{A_s \cup A_f}$ if there is no ambiguity. Given two objects with equivalence classes $[x]$ and $[y]$, we can get two classification rules:

$$
\begin{aligned}
r_{[x]}: \quad & \left[\bigwedge_{s \in A_s} s = I_s(x)\right] \wedge \left[\bigwedge_{f \in A_f} f = I_f(x)\right] \Rightarrow d = I_d(x), \\
r_{[y]}: \quad & \left[\bigwedge_{s \in A_s} s = I_s(y)\right] \wedge \left[\bigwedge_{f \in A_f} f = I_f(y)\right] \Rightarrow d = I_d(y).
\end{aligned}
$$

Classification rules have $X \Rightarrow Y$ form that indicates if X then Y. The left hand side of the rule, X, is a conjunction of all stable and flexible attribute-value pairs and the right hand side of the rule, Y, is the decision attribute-value pair. Let $ST(r_{[x]})$ be the stable attributes part in the left hand side of the rule $r_{[x]}$, $FL(r_{[x]})$ be the flexible attributes part in the left hand side of the rule $r_{[x]}$, i.e.,

$$
\begin{aligned}
ST(r_{[x]}) &= \left[\bigwedge_{s \in A_s} s = I_s(x)\right], \\
FL(r_{[x]}) &= \left[\bigwedge_{f \in A_f} f = I_f(x)\right].
\end{aligned}
\tag{3.1}
$$

Let $ST(r_{[y]}) = ST(r_{[x]})$ denote that $[x]$ and $[y]$ have the same values on each stable attribute. If $ST(r_{[y]}) = ST(r_{[x]})$, then $[x]$ and $[y]$ can be changed to each other by changing the flexible attributes values via actions. If a user wants to change $[x]$ into $[y]$, the action is to execute the following actionable rule [75]:

$$
r_{[x]} \rightsquigarrow r_{[y]}: \overset{\rightsquigarrow}{\bigwedge_{f \in A_f}} I_f(x) \rightsquigarrow I_f(y), \quad \textit{subject to} \bigwedge_{s \in A_s} I_s(x) = I_s(y), \tag{3.2}
$$

where $I_f(x) \rightsquigarrow I_f(y)$ means that the value of attribute f is changed from $I_f(x)$ to $I_f(y)$ and the symbol $\overset{\rightsquigarrow}{\bigwedge}$ means all the flexible attributes' values have to be changed.

Based on above concepts, the formal definition of an actionable rule is given in

Definition 3.2.

Definition 3.2 *An equivalence class* $[x] \subseteq OB$ *is called* ***actionable*** *if* $\exists [y] \subseteq OB, [y] \neq [x]$*, such that* $ST(r_{[x]}) = ST(r_{[y]})$*. A rule* $r_{[x]} \rightsquigarrow r_{[y]}$ *is called an* ***actionable rule*** *for changing* $[x]$ *into* $[y]$ *and each clause* $I_f(x) \rightsquigarrow I_f(y)$ *for* $f \in A_f$ *is called a* ***sub-actionable rule****.* $[x]$ *is called* ***non-actionable*** *if* $\nexists [y] \subseteq OB$ *satisfying* $ST(r_{[x]}) = ST(r_{[y]})$*.*

In Definition 3.2, equivalence classes $[x]$ and $[y]$ can be from different regions or the same region. If $[x]$ is non-actionable, then we cannot find any actionable rule for transferring $[x]$ to a different region.

An actionable rule can serve a guideline for action and strategy design. Given one actionable rule, many actions can be designed, because there may exist many options to change a flexible attribute value. For example, we may lower the blood pressure by taking pills, controlling diet, or doing exercises. In this thesis, we consider the simplest case, in which each actionable rule corresponds to one action. We analyze the benefits and costs from actionable rules instead of those of actions. Without ambiguity, we also refer to an actionable rule as an *action* and a sub-actionable rule as a *sub-action*. We assume that applying actions does not change the classification rules. We also assume that any action that is applied causes the stated transfer to occur.

3.3 Cost-benefit Analysis of Actions

Each action incurs a cost and brings a beneficial effect. Depending on a particular application, the motivation of taking actions is to minimize or maximize the objective function in Equation (2.2) with the lowest cost. Suppose $Q(\pi')$ is the quality of a new trisection π' by acting on π, then the benefit can be defined as the difference between two qualities:

$$B = Q(\pi') - Q(\pi), \tag{3.3}$$

or

$$B = Q(\pi) - Q(\pi'). \tag{3.4}$$

Specifically, applications whose objective functions are based on cost [108], information entropy [15], or Gini index [141] use Equation (3.4) and those based on chi-square statistic [24] or variance [4] use Equation (3.3). We use $B_{r_{[x]} \rightsquigarrow r_{[y]}}$ to denote the benefit of taking action $r_{[x]} \rightsquigarrow r_{[y]}$.

There are many types of cost involved with changing attribute values, such as money, time, and other resources. We suppose that all kinds of cost associated with a sub-action $I_f(x) \rightsquigarrow I_f(y)$ can be synthesized as one cost defined by a function C_f:

$$C_f : V_f \times V_f \longrightarrow \Re, \ \forall f \in A_f.$$

For each $f \in A_f$, $C_f(v_1, v_2)$ denotes the cost of changing the value of attribute f from v_1 to v_2. If $|V_f|$ is limited, then the C_f can be presented by a table shown in Table 3.3, where $V_f = \{v_1, v_2, \cdots, v_n\}$. Commonly, $C_f(v_i, v_i) = 0, i = 1, \cdots, n$, because we do not have to take a sub-action to change its value. Generally, $C_f(v_1, v_2)$ does not have to be equal to $C_f(v_2, v_1)$ and the cost functions $\{C_f \mid f \in A_f\}$ are given by domain experts.

Table 3.3: Cost function C_f.

	v_1	v_2	$\cdots$	v_n
v_1	$C_f(v_1, v_1)$	$C_f(v_1, v_2)$	$\cdots$	$C_f(v_1, v_n)$
v_2	$C_f(v_2, v_1)$	$C_f(v_2, v_2)$	$\cdots$	$C_f(v_2, v_n)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
v_n	$C_f(v_n, v_1)$	$C_f(v_n, v_2)$	$\cdots$	$C_f(v_n, v_n)$

We use two assumptions for computing action cost.

(A1) All attribute values are independent, which means that one value change will not affect any other values.

(A2) All actions are independent, which means that any action will only affect two equivalence classes, all other equivalence classes will not be affected.

Assumption (A1) allows us to calculate the cost of transferring one object by simply summing up the costs of all sub-actions. For example, the cost of transferring one object from $[x]$ to $[y]$ is:

$$\sum_{f \in A_f} C_f(I_f(x), I_f(y)). \tag{3.5}$$

Let $C_{r_{[x]} \rightsquigarrow r_{[y]}}$ denote the cost of action $r_{[x]} \rightsquigarrow r_{[y]}$ and it can be computed by:

$$C_{r_{[x]} \rightsquigarrow r_{[y]}} = |[x]| \sum_{f \in A_f} C_f(I_f(x), I_f(y)), \tag{3.6}$$

where $|\cdot|$ denote the cardinality of a set.

Assumption (A2) allows us to calculate and analyze the benefit and cost of any action independently. Based on the assumption (A2), given any two actions $r_{[x]} \rightsquigarrow r_{[y]}$ and $r_{[p]} \rightsquigarrow r_{[q]}$, the $B_{r_{[x]} \rightsquigarrow r_{[y]}}$ and $C_{r_{[x]} \rightsquigarrow r_{[y]}}$ will not be changed whether or not we take the action $r_{[p]} \rightsquigarrow r_{[q]}$.

3.4 Benefit in Classification Problems

In three-way approximation of a concept [114], three regions are named as POS, BND, and NEG, representing positive, boundary, and negative regions to approximate a subset $X \subseteq OB$ denoting a concept or class. Objects in POS, NEG, and BND are considered in class X, not in X, and difficult to classify, respectively. Let $Pr(X|[x])$ denote the conditional probability that an object is in X given that the object is in

Table 3.4: Misclassification cost matrix.

	POS	BND	NEG
X	λ_{PP}	λ_{BP}	λ_{NP}
X^C	λ_{PN}	λ_{BN}	λ_{NN}

$[x]$, which may be computed by a rough membership function [100]:

$$Pr(X|[x]) = \frac{|[x] \cap X|}{|[x]|}. \tag{3.7}$$

Given a pair of thresholds (α, β) with $0 \leq \beta < \alpha \leq 1$, the three regions are constructed by the DTRS model [121]:

$$\begin{aligned} \text{POS}_{(\alpha,\beta)}(X) &= \{x \in OB \mid Pr(X|[x]) \geq \alpha\}, \\ \text{BND}_{(\alpha,\beta)}(X) &= \{x \in OB \mid \beta < Pr(X|[x]) < \alpha\}, \\ \text{NEG}_{(\alpha,\beta)}(X) &= \{x \in OB \mid Pr(X|[x]) \leq \beta\}. \end{aligned} \tag{3.8}$$

We may use a misclassification cost matrix shown in Table 3.4 to measure the quality of three regions. In Table 3.4, λ_{PP}, λ_{BP}, λ_{NP} indicate the costs of classifying an object in X into the positive, boundary, and negative regions, respectively, and $X^C = OB - X$. Others are explained similarly. Therefore, the qualities of three regions can be computed by:

$$\begin{aligned} Q(\text{POS}_{(\alpha,\beta)}(X)) &= |X \cap \text{POS}_{(\alpha,\beta)}(X)|\lambda_{PP} + |X^C \cap \text{POS}_{(\alpha,\beta)}(X)|\lambda_{PN}, \\ Q(\text{BND}_{(\alpha,\beta)}(X)) &= |X \cap \text{BND}_{(\alpha,\beta)}(X)|\lambda_{BP} + |X^C \cap \text{BND}_{(\alpha,\beta)}(X)|\lambda_{BN}, \\ Q(\text{NEG}_{(\alpha,\beta)}(X)) &= |X \cap \text{NEG}_{(\alpha,\beta)}(X)|\lambda_{NP} + |X^C \cap \text{NEG}_{(\alpha,\beta)}(X)|\lambda_{NN}. \end{aligned}$$

In this setting, the Equation (3.4) is used for benefit. The $B_{r_{[x]} \rightsquigarrow r_{[y]}}$ can be analyzed as follows. Let a denote the number of objects in $[x]$ belonging to class X and b the number of objects from $[x]$ belonging to class X' after taking action:

$$\begin{aligned} a &= |X \cap [x]|, \\ b &= |X' \cap [x]|, \end{aligned} \tag{3.9}$$

where X' is a new set of objects obtained by changing $[x]$ according to $[y]$, representing the same concept as X. a is easy to compute and we use an assumption to compute b:

(A3) After taking an action $r_{[x]} \rightsquigarrow r_{[y]}$, the changed equivalence class $[x]$ will have the same probability with $[y]$'s, i.e., $Pr(X'|[x]) = Pr(X|[y])$, where $Pr(X|[y]) = |X \cap [y]|/|[y]|$.

The idea of this assumption can be explained by an example. Some people in Canada believe that changing their vehicle's all season tires to winter tires will make their vehicle safer to drive in winter. This assumption suggests that switching to winter tires will improve their vehicle's safety to the level of those vehicles using winter tires. Specifically, suppose there are three objects in $[x]$ moved to $[y]$, where $[y]$ has 2/3 objects labeled + and 1/3 labeled -, then these three objects will be transformed into two + and one -. Therefore, after taking action $r_{[x]} \rightsquigarrow r_{[y]}$, b can be computed by:

$$b = |X' \cap [x]| = |[x]|Pr(X'|[x]) = |[x]|Pr(X|[y]) = |[x]||X \cap [y]|/|[y]|. \tag{3.10}$$

Further, we have the following proposition:

Proposition 3.1 *Taking action $r_{[x]} \rightsquigarrow r_{[y]}$ to transfer objects from region V to W, the benefit is computed by:*

$$B_{r_{[x]} \rightsquigarrow r_{[y]}} = w_W \big[-b\lambda_{WP} - (|[x]| - b)\lambda_{WN} \big] + w_V \big[a\lambda_{VP} + (|[x]| - a)\lambda_{VN} \big], \tag{3.11}$$

where $V, W \in \{P, B, N\}$, in which P, B, and N represent positive, boundary, and *negative regions, respectively.*

Proof. See Appendix A.1.

We show how to compute the cost and benefit by an example below.

Example 3.1 *We continue to use the example in Section 3.1.1. We define* $X = \{x \in OB \mid I_d(x) = +\}$ *representing the group of people who do not have heart disease and we enlarge this group by transferring people in other groups to it. We use the cost matrix in Table 3.5 to compute the quality of three regions. The three regions to*

Table 3.5: Cost matrix.

	POS	BND	NEG
X	2	4	8
X^C	11	9	8

approximate X *are constructed as follows:*

$$\begin{aligned} \text{POS}_{(0.5,0.2)}(X) &= \{x \in OB \mid Pr(X|[x]) \geq 0.5\} = \{o_1, o_2, o_6, o_9\}, \\ \text{BND}_{(0.5,0.2)}(X) &= \{x \in OB \mid 0.2 < Pr(X|[x]) < 0.5\} = \{o_3, o_4, o_5\}, \\ \text{NEG}_{(0.5,0.2)}(X) &= \{x \in OB \mid Pr(X|[x]) \leq 0.2\} = \{o_7, o_8\}, \end{aligned}$$

where $\alpha = 0.5$ *and* $\beta = 0.2$ *are two thresholds minimizing the* $Q(\pi)$*.*

Table 3.6: Cost function C_{chol}.

	low	medium	high
low	0	1	3
medium	2	0	1
high	4	1	0

Table 3.7: Cost function C_{bp}.

	low	normal	high
low	0	1	2
normal	1	0	1
high	2	1	0

Using notation of actionable rule, we have $a_1 = r_{[o_7]} \rightsquigarrow r_{[o_1]}$ *and* $a_2 = r_{[o_7]} \rightsquigarrow r_{[o_6]}$*. Suppose the cost functions* C_{chol} *and* C_{bp} *are given in Table 3.6 and 3.7, respectively.*

According to Equation (3.6), the costs of action a_1 and a_2 can be computed as follows:

$$
\begin{aligned}
C_{r_{[o_7]} \rightsquigarrow r_{[o_1]}} &= |[o_7]|(C_{chol}(high, medium) + C_{bp}(high, normal)) = 2, \\
C_{r_{[o_7]} \rightsquigarrow r_{[o_6]}} &= |[o_7]|(C_{chol}(high, medium) + C_{bp}(high, low)) = 3.
\end{aligned}
$$

Now, we compute the benefit of the two actions. According to Equation (3.11) and using $w_P = w_B = w_N = 1$, we get:

$$
\begin{aligned}
B_{r_{[o_7]} \rightsquigarrow r_{[o_1]}} &= w_P\big[-b\lambda_{PP} - (|[o_7]| - b)\lambda_{PN}\big] + w_N\big[a\lambda_{NP} + (|[o_7]| - a)\lambda_{NN}\big] \\
&= -0.5 * 2 - (1 - 0.5) * 11 + 0 * 8 + (1 - 0) * 8 = 1.5, \\
B_{r_{[o_7]} \rightsquigarrow r_{[o_6]}} &= w_P\big[-b\lambda_{PP} - (|[o_7]| - b)\lambda_{PN}\big] + w_N\big[a\lambda_{NP} + (|[o_7]| - a)\lambda_{NN}\big] \\
&= -1 * 2 - (1 - 1) * 11 + 0 * 8 + (1 - 0) * 8 = 6.
\end{aligned}
$$

Obviously, action $r_{[o_7]} \rightsquigarrow r_{[o_1]}$ has a lower cost and a smaller benefit, while $r_{[o_7]} \rightsquigarrow r_{[o_6]}$ has a higher cost and a larger benefit.

3.5 Four Actionable Three-way Decision Models

Generally, to obtain greater benefits require higher costs and lower costs receive limited benefits. We propose the following four models for dealing with different situations.

(i) One requires the maximum benefit solution without cost limitation.

(ii) One requires the minimum cost solution to obtain the maximum benefit.

(iii) One requires the maximum benefit solution under a limited action cost.

(iv) One requires the minimum action cost solution to obtain a desired benefit.

The first two models provide the bounds of the cost and benefit and models (iii) and (iv) are constrained optimization problems representing a wide range of applications.

Consider an example in business, a company produces a product and wants to obtain the maximum benefit from it. With respect to the product, people are categorized into three groups, in which the first group consists of people who are likely to buy the product, the second group consists of people who are unlikely to buy the product, and the third group consists of people who are indifferent to the product or do not know the product. People in these three groups might be called buyers, potential buyers, and non-buyers, respectively. The company may have a limited budget to attract new buyers and retain existing clients. They may choose any combination of actions, such as extending the product's warranty period, advertising on TV, and / or reducing accessories' prices, etc. Each different action will consume different cost from the budget. Model (iii) is used to find a set of actions that will produce the maximum profit for the company when the budget is limited. Similarly in military triage, model (iii) can help to save most lives when medical supplies are limited.

Consider the election example in Section 3.1.2 again. The candidate acts to retain existing supporters and gain additional supporters to win the election. Suppose there are four actions a_1, a_2, a_3, and a_4 available and taking $\{a_1, a_3\}$ will help the candidate to win the election. Therefore, any super-set of $\{a_1, a_3\}$ will not change the result, but consumes more cost. Model (iv) is used to find a set of actions to obtain the desired benefit at the minimal cost.

These four models are independent from movement patterns and can be combined with any movement pattern, such as the one in Figure 3.1 or Figure 3.2. To formalize the four models, we define some concepts as follows:

Definition 3.3 *Given movement patterns, an action $r_{[x]} \rightsquigarrow r_{[y]}$ is called a **desirable action** if $[x]$ and $[y]$ are from unfavorable and favorable regions, respectively.*

*Similarly, we can define **undesirable action** and **indifferent action**.*

We construct three sets of actions DES, UND, and IDF, representing the set of desirable actions, undesirable actions, and indifferent actions, respectively:

$$\begin{aligned} \text{DES} &= \{r_{[x]} \rightsquigarrow r_{[y]} \mid r_{[x]} \rightsquigarrow r_{[y]} \text{ is desirable}\}, \\ \text{UND} &= \{r_{[x]} \rightsquigarrow r_{[y]} \mid r_{[x]} \rightsquigarrow r_{[y]} \text{ is undesirable}\}, \\ \text{IDF} &= \{r_{[x]} \rightsquigarrow r_{[y]} \mid r_{[x]} \rightsquigarrow r_{[y]} \text{ is indifferent}\}. \end{aligned}$$

Once a trisection and movement patterns are given, the DES, UND, and IDF can be easily found.

We are interested in the desirable actions because they can improve the quality of trisection and produce benefit. We further define:

$$\text{SOURCE} = \{[x] \mid \exists [y], r_{[x]} \rightsquigarrow r_{[y]} \in \text{DES}\}, \tag{3.12}$$

where each $[x] \in$ SOURCE is desirable to be transferred to another region. SOURCE is a source providing all these equivalence classes for a decision maker to chose. For $[x_i] \in \text{SOURCE}, i = 1, \ldots, |\text{SOURCE}|$, there may exist many equivalence classes $[y_1], \cdots, [y_{n_i}]$ that $r_{[x_i]} \rightsquigarrow r_{[y_j]} \in \text{DES}, j = 1, \ldots, n_i$, where n_i is the number of desirable actions that transfer $[x_i]$. We use c_{ij} and b_{ij} to denote the cost and benefit of $[x_i]$'s j^{th} action, i.e., $r_{[x_i]} \rightsquigarrow r_{[y_j]}, j = 1, \ldots, n_i$ and use $a_{ij} \in \{0, 1\}$ to indicate taking or not taking $[x_i]$'s j^{th} action. For example, c_{23} denotes the cost of $[x_2]$'s 3^{rd} action, b_{35} denotes the benefit of $[x_3]$'s 5^{th} action, $a_{24} = 1$ indicates that $[x_2]$'s 4^{th} action is taken, and $a_{21} = 0$ indicates that $[x_2]$'s 1^{st} action is not taken. For all actions transferring $[x_i]$, we may take none or one of them. In other words, given $[x_i]$, all a_{ij} satisfy $\sum_{j=1}^{n_i} a_{ij} \leq 1,\ a_{ij} \in \{0, 1\}$.

By using these notations, the model (i), (ii), (iii), and (iv) are formalized in

Definition 3.4, 3.5, 3.6, and 3.7, respectively.

Definition 3.4 *Given a trisection of a universe* $\pi = (\mathrm{L}, \mathrm{M}, \mathrm{R})$*, sets* DES *and* SOURCE*, where* $|\mathrm{SOURCE}| = n$*,* $[x_i] \subseteq \mathrm{SOURCE}$ *has* n_i *actions, and the cost and benefit of* $[x_i]$*'s* j^{th} *action are denoted as* c_{ij} *and* b_{ij}*, respectively,* $j = 1, \ldots, n_i$*. The solution that produces the maximum benefit is a set of* a_{ij} *that*

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij} b_{ij},$$

where $\sum_{j=1}^{n_i} a_{ij} \leq 1, \; a_{ij} \in \{0, 1\}, \; i = 1, \ldots, n.$

Definition 3.5 *Given a trisection of a universe* $\pi = (\mathrm{L}, \mathrm{M}, \mathrm{R})$*, sets* DES *and* SOURCE*, where* $|\mathrm{SOURCE}| = n$*,* $[x_i] \subseteq \mathrm{SOURCE}$ *has* n_i *actions, and the cost and benefit of* $[x_i]$*'s* j^{th} *action are denoted as* c_{ij} *and* b_{ij}*, respectively,* $j = 1, \ldots, n_i$*. The solution that requires the minimum action cost when benefit is maximized is a set of* a_{ij} *that*

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij} c_{ij}, \quad \text{subject to} \quad \sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij} b_{ij} = \bar{B},$$

where $\sum_{j=1}^{n_i} a_{ij} \leq 1, \; a_{ij} \in \{0, 1\}, \; i = 1, \ldots, n,$ *and* $\bar{B}$ *is the maximum benefit satisfying Definition 3.4.*

Definition 3.6 *Given a trisection of a universe* $\pi = (\mathrm{L}, \mathrm{M}, \mathrm{R})$*, sets* DES *and* SOURCE*, where* $|\mathrm{SOURCE}| = n$*,* $[x_i] \subseteq \mathrm{SOURCE}$ *has* n_i *actions, and the cost and benefit of* $[x_i]$*'s* j^{th} *action are denoted as* c_{ij} *and* b_{ij}*, respectively,* $j = 1, \ldots, n_i$*. The solution that produces maximum benefit with limited action cost* c_a *is a set of* a_{ij} *that*

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij} b_{ij}, \quad \text{subject to} \quad \sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij} c_{ij} \leq c_a,$$

where $\sum_{j=1}^{n_i} a_{ij} \leq 1, \; a_{ij} \in \{0, 1\}, \; i = 1, \ldots, n.$

Definition 3.7 *Given a trisection of a universe* $\pi = (\mathrm{L}, \mathrm{M}, \mathrm{R})$*, sets* DES *and* SOURCE*, where* $|\mathrm{SOURCE}| = n$*,* $[x_i] \subseteq \mathrm{SOURCE}$ *has* n_i *actions, and the cost and benefit of* $[x_i]$*'s* j^{th} *action are denoted as* c_{ij} *and* b_{ij}*, respectively,* $j = 1, \ldots, n_i$*. The solution that requires the minimum action cost to reach the desired benefit,* b_l*, is a set of* a_{ij} *that*

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij} c_{ij}, \quad \text{subject to } \sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij} b_{ij} \geq b_l,$$

where $\sum_{j=1}^{n_i} a_{ij} \leq 1,\ a_{ij} \in \{0, 1\},\ i = 1, \ldots, n.$

In the context of this thesis, the cost and benefit of an action are independent, which means that they are two values associated with each action. The computational relations of these models are shown in Figure 3.3. In this figure, the model (ii) is a special case of model (i) by giving a constraint on cost, and it is also a special case of model (iii) when $b_l = \bar{B}$. Model (i) is a special case of model (iii) when $c_a = +\infty$. Model (iii) and model (iv) are dual problems.

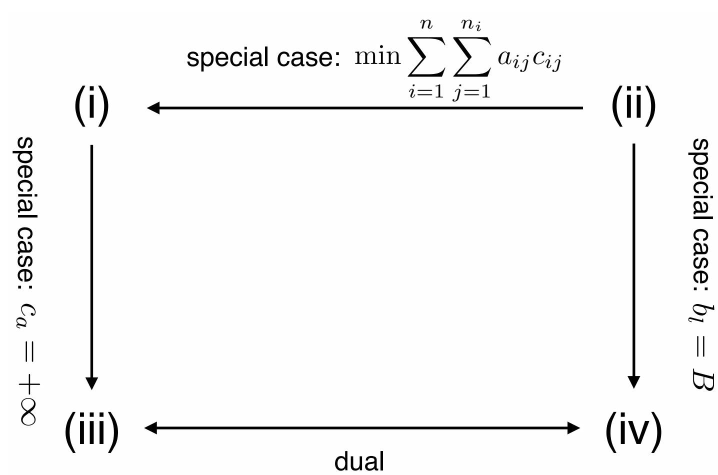


Figure 3.3: Computational relations of models.

Figure 3.4 illustrates the solutions of four models in solution space. In this picture, each curved line depicts the procedure of taking a possible set of actions. The circles show the corresponding solutions, in which the top right circle indicates the solution for models (i) and (ii). The solution for model (iii) is the circle indicating the highest intersection of vertical line c_a and curved lines. The solution for model (iv) is the circle of left most intersection of horizontal line b_l and curved lines. Both the solutions for models (iii) and (iv) are positioned below and to the left of the solution for models (i) and (ii).

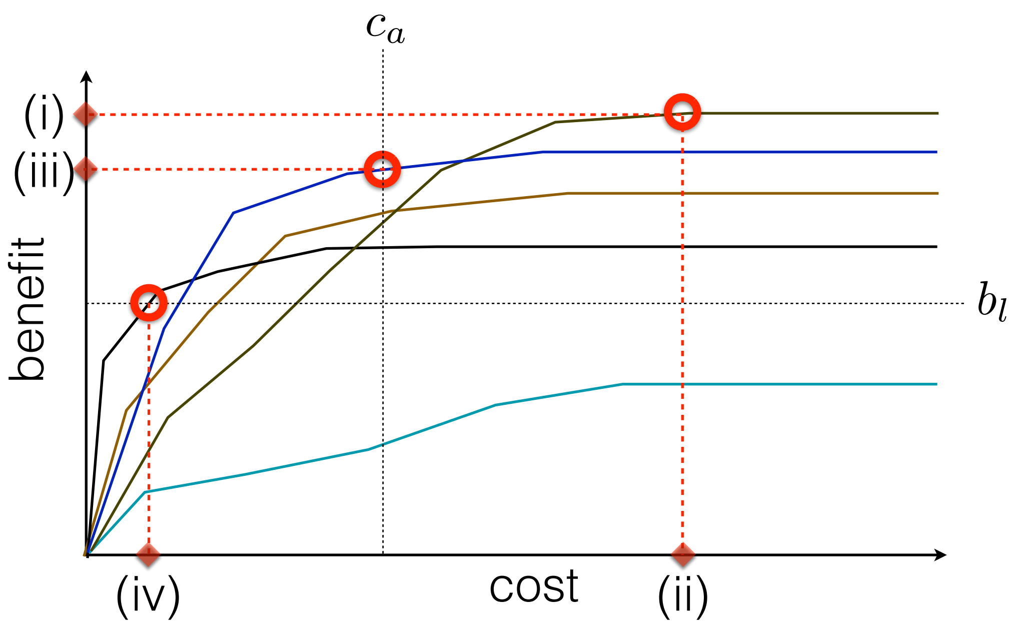


Figure 3.4: Illustration of models.

Chapter 4

STATISTICAL INTERPRETATIONS OF TRISECTING

In this chapter, we consider the task of dividing OB into three regions. For such a task, we consider the following fundamental issues:

(1) construction and interpretation of a totally ordered set $\mathbb{V}$,

(2) construction and interpretation of an evaluation function $e : OB \longrightarrow \mathbb{V}$,

(3) determination and interpretation of a pair of thresholds (α, β), and

(4) measurement of the quality of a tripartition {L,M,R}.

We examine two specific statistical interpretations of three-way decisions depending on the structures of $\mathbb{V}$, and a chi-square statistic based method for computing the pair of thresholds for three regions.

The studies on combining statistics and three-way decisions provide a promising research direction. The main objective of this chapter is to further examine the statistical interpretations of three-way decisions. In particular, we use statistical notions, such as the mean, median, standard deviation, central tendency and dispersion, to construct and interpret an evaluation function and a pair of thresholds for a three-way decision. Objects around the mean value form one region, and two tails form the other two regions. In other words, we search for statistical interpretations of three-way decisions that enable us to examine the structures of the data and to make inferences about the data.

4.1 General Considerations

In many applications, we typically have statistical information about objects in OB. For example, we may have frequencies of measurement values with respect to a particular feature of objects. Such information may be used to construct both an evaluation function and a pair of thresholds. In the cases when an evaluation function is given, we may use a distribution of the evaluation status values (ESVs) to find a pair of thresholds.

For convenience, we use an ordering $\preceq$ on the set of evaluation status values (ESVs) $\mathbb{V}$ instead of the ordering $\succeq$ used in Chapter 2, and the three regions are constructed as:

$$\begin{aligned}
\mathrm{L}_{(\alpha,\beta)}(e) &= \{x \in OB \mid e(x) \preceq \beta\}, \\
\mathrm{M}_{((\alpha,\beta)}(e) &= \{x \in OB \mid \beta \prec e(x) \prec \alpha\}, \\
\mathrm{R}_{((\alpha,\beta)}(e) &= \{x \in OB \mid e(x) \succeq \alpha\}. \qquad (4.1)
\end{aligned}$$

According to Equation (4.1), we divide OB into three regions. As shown in Figure 4.1, the region L consists of objects with low ESVs, the region M with medium ESVs, and the region R with high ESVs. The example of blood pressure classification in the last chapter also implies that the middle region M consists of normal or typical instances from a population, while regions L and R consist of abnormal or atypical instances. In other words, the blood pressure of a healthy person is expected to fall within a certain range, e.g., between 90 and 140 for systolic blood pressure. An interesting question is how to interpret the intuitive notions of low, medium, and high values used in three-way decisions based on concepts from statistics.

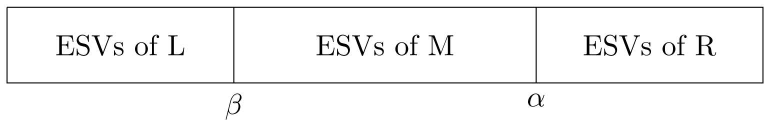


Figure 4.1: Division of $(\mathbb{V}, \preceq)$.

In statistics, distributional characteristics such as median, mean, percentile, and standard deviation can be used to describe a population. To establish a connection to three-way decisions, we may collect objects with ESVs around the median or mean value to form region M. The percentile or standard deviation may be used to calculate the distances of objects with ESVs from the median or mean, which in turn determines a pair of threshold values. Two special cases of $\mathbb{V}$ may be considered. One is a set of non-numeric values and the other is a set of numeric values.

When $\mathbb{V}$ is a set of non-numeric values, we can perform comparisons based on the total order $\preceq$ and we cannot carry out arithmetic operations such as addition and multiplication. In other words, we can only consider the ranking of values in $\mathbb{V}$ and the distribution of ESVs. The ordering enables us to locate the median, that is, an object at the middle point of a ranked list. In addition, we can also use the frequency information to compute percentiles. Consequently, we can use the median as the middle point of region M and use two percentiles to determine the size of region M. One percentile is used to calculate the left boundary of M, and the other percentile is used to calculate the right boundary of M. Region L is the set of objects whose ESVs are below the left boundary and region R is the set of objects whose ESVs are above the right boundary.

When $\mathbb{V}$ is a set of numeric ESVs, we can apply the median based interpretation by using the object order induced by $\preceq$. Moreover, since we can perform arithmetic operations on $\mathbb{V}$, we can consider another interpretation which uses the mean and standard deviation. That is, we can use the mean to set up the middle position, and use standard deviations to calculate the distances of the thresholds from the mean. Based on standard deviations, region M is the set of objects with ESVs around the mean, region L is the set of objects whose ESVs are much less than the mean, and region R is the set of objects whose ESVs are much greater than the mean.

These two interpretations make use of different types of statistical information. We discuss their detailed formulations in the next two sections.

4.2 Interpretations through Median and Percentile

When $\mathbb{V}$ is a set of non-numeric values, the ordering $\preceq$ only allows us to arrange objects in OB into a ranked list according to their ESVs, as shown in Figure 4.2. The median is the value at the middle position of this list and the positions of two thresholds around the median are determined by two percentiles. A user can determine the three regions L, M, and R by the pair of percentiles. We use a simple example to demonstrate the main ideas.

Example 4.1 *Suppose* $OB = \{x_1, x_2, x_3, x_4, x_5\}$ *is a set of five eggs, and* $\mathbb{V}$ *is a set of words describing the size of eggs:* {*smallest, smaller, small, medium, large, larger, largest*} *with the ordering smallest* $\preceq$ *smaller* $\preceq$ *small* $\preceq$ *medium* $\preceq$ *large* $\preceq$ *larger* $\preceq$ *largest. We want to divide* OB *into three subsets according to their sizes. Given an evaluation function, suppose objects in* OB *have the following ESVs:* $e(x_1) = small, e(x_2) = smaller, e(x_3) = largest, e(x_4) = large, e(x_5) = larger$. *We can arrange all objects in* OB *into a ranked list based on their ESVs,* x_2, x_1, x_4, x_5, x_3, *according to the ordering* $\preceq$. *The object at the middle position of this list is* x_4 *and*

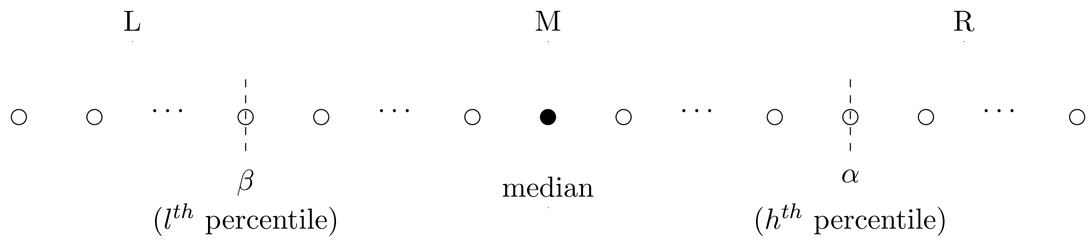


Figure 4.2: Illustration of division on rank ordered list through median and percentile.

its value is large, i.e., the median is large. Suppose we want the region M to include 60% of the objects and regions L and R each consists of 20% of the objects. We use 20% for computing the position of the left threshold and 20% for computing the position of the right threshold. The position of the left threshold is 1 with object x_2 and ESV smaller, and the position of the right threshold is 4 with object x_3 and ESV largest. That is, β = smaller and α = largest. Therefore, the tripartition is given L=$\{x_2\}$, M=$\{x_1, x_4, x_5\}$ and R=$\{x_3\}$.

From the construction process of the example, we have an interpretation of three-way decisions using the median and percentile, as depicted in Figure 4.2. In Example 4.1, the two percentiles are $l = 20^{th}$ percentile and $h = 80^{th}$ percentile. The ESV of the object denoted by the solid circle in the middle position is the median and (α, β) is a pair of thresholds based on a pair of percentiles.

The trisection of three-way decisions can be constructed as follows. Suppose that the size of OB is n. Step 1: arrange the set of objects into a ranked list according to their ESVs in ascending order, in which objects with the same ESV can be ranked in any order. In this way, we have a list of ESVs, $v_1, v_2, \ldots, v_n$, where v_1 is the smallest value and v_n is the largest value. Step 2: we search for ESVs at l^{th} and h^{th} percentiles with $l < 50$ and $h > 50$, we can calculate the pair of thresholds by:

$$\begin{aligned} \beta &= v_{\lfloor ln/100 \rfloor}, \\ \alpha &= v_{\lceil hn/100 \rceil}, \end{aligned} \tag{4.2}$$

where the floor operator $\lfloor a \rfloor$ gives the largest integer that is not greater than a and the ceiling operator $\lceil a \rceil$ gives the smallest integer that is not less than a. The floor and ceiling operators used in β and α, respectively, are needed because $ln/100$ and $hn/100$ may not be integers. As a result, three regions are constructed by:

$$\begin{aligned}
\mathrm{L}_{(\alpha,\beta)}(e) &= \{x \in OB \mid e(x) \preceq \beta\} \\
&= \{x \in OB \mid e(x) \preceq v_{\lfloor ln/100 \rfloor}\}, \\
\mathrm{M}_{(\alpha,\beta)}(e) &= \{x \in OB \mid \beta \prec e(x) \prec \alpha\} \\
&= \{x \in OB \mid v_{\lfloor ln/100 \rfloor} \prec e(x) \prec v_{\lceil hn/100 \rceil}, \\
\mathrm{R}_{(\alpha,\beta)}(e) &= \{x \in OB \mid e(x) \succeq \alpha\} \\
&= \{x \in OB \mid e(x) \succeq v_{\lceil hn/100 \rceil}\}. \qquad (4.3)
\end{aligned}$$

In order to have three pair-wise disjoint regions, we require that $\beta \prec \alpha$, i.e., β and α cannot be the same value in $\mathbb{V}$. This requires that the two percentiles must be chosen to satisfy the criterion.

Equation (4.3) provides an interpretation of three-way decisions using the median and percentile. Such an interpretation has been widely used in many applications. For example, in boxplots [78], the values of β and α are obtained by first and third quartiles, i.e., 25^{th} and 75^{th} percentiles, and the middle region M by interquartile range (IQR).

4.3 Interpretations through Mean and Standard Deviation

When $\mathbb{V}$ consists of numeric values, statistical measures based on arithmetic operations such as the mean and standard deviation can be applied. For simplicity, we assume that $\mathbb{V}$ is the set of real numbers. Suppose $e(x_1)$, $e(x_2)$, ..., $e(x_n)$ are the

ESVs of objects in OB, where n is the cardinality of OB. The mean and standard deviation are calculated by:

$$\begin{aligned} \mu &= \frac{1}{n}\sum_{i=1}^{n} v(x_i), \\ \sigma &= \left(\frac{1}{n}\sum_{i=1}^{n}(e(x_i)-\mu)^2\right)^{\frac{1}{2}}. \end{aligned}$$

As shown by Figure 4.3 and Figure 4.4, we may interpret μ as the ESV for representing objects in M and σ as a unit to measure the positions of the two thresholds β and α. In some cases, we only have a data that is a sample of the whole population. Thus, we have to use sample standard deviation s instead of σ:

$$s = \left(\frac{1}{n-1}\sum_{i=1}^{n}(e(x_i)-\mu)^2\right)^{\frac{1}{2}}.$$

Suppose two non-negative numbers k_1 and k_2 represent the distances of two thresholds from the mean in terms of the number of the standard deviations. The pair of thresholds can be constructed as follows:

$$\begin{aligned} \beta &= \mu - k_1\sigma, \quad k_1 \geq 0, \\ \alpha &= \mu + k_2\sigma, \quad k_2 \geq 0. \end{aligned} \tag{4.4}$$

Generally, k_1 and k_2 need not be equal. According to β and α, three regions can be constructed by:

$$\begin{aligned} \mathrm{L}_{(k_1,k_2)}(e) &= \{x \in OB \mid e(x) \leq \beta\} \\ &= \{x \in OB \mid e(x) \leq \mu - k_1\sigma\}, \\ \mathrm{M}_{(k_1,k_2)}(e) &= \{x \in OB \mid \beta < e(x) < \alpha\} \\ &= \{x \in OB \mid \mu - k_1\sigma < e(x) < \mu + k_2\sigma\}, \end{aligned}$$

$$\begin{aligned}\mathrm{R}_{(k_1,k_2)}(e) &= \{x \in OB \mid e(x) \geq \alpha\} \\ &= \{x \in OB \mid e(x) \geq \mu + k_2\sigma\}, \end{aligned} \tag{4.5}$$

where $\leq$, $<$, and $\geq$ are standard relations on a set $\mathbb{V}$ of numeric values. It is worth noting that k_1 and k_2 are related to z-score. Thus, we can interpret three-way decisions in terms of z-scores.

Equation (4.5) makes no assumption of distribution of ESVs of objects. In many real applications, it is common that the objects' ESVs satisfy a certain distribution. For example, Figure 4.3 and Figure 4.4 illustrate the trisection based on two kinds of distributions. In Figure 4.3, the normal distribution shows a unimodal and symmetric curve, in which the mean μ is the normal or typical point of the distribution. While Figure 4.4 shows a monotonic curve, the region around the mean represents the average area of the distribution, i.e., not too high and not too low.

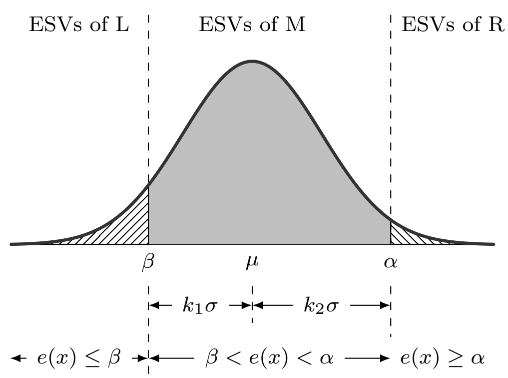


Figure 4.3: Illustration of three-way decisions on a normal distribution.

There are many applications based on this model. For example, Pater [63] suggested using $k_1 = k_2 = 2$ for blood pressure classifications. In other words, $\mathrm{M}_{(2,2)}(e) = \{x \in OB \mid \mu - 2\sigma < e(x) < \mu + 2\sigma\}$ is the region of normal blood pressure, while $\mathrm{L}_{(2,2)}(e) = \{x \in OB \mid e(x) \leq \mu - 2\sigma\}$ and $\mathrm{R}_{(2,2)}(e) = \{x \in OB \mid e(x) \geq \mu + 2\sigma\}$ regions are abnormal, that is, the hypotension and hypertension regions, respectively. The Stanford-Binet test [81] is the most often used approach to measure people's

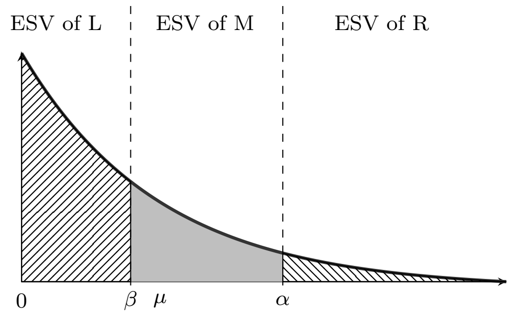


Figure 4.4: Illustration of three-way decisions on an exponential distribution.

Intelligence Quotient (IQ), and $k_1 = k_2 = 2$ is usually used to classify people into three categories [81, 83]. People with an IQ between $\mu - 2\sigma$ and $\mu + 2\sigma$ are considered average, people with an IQ greater than $\mu + 2\sigma$ are considered above average, and people with an IQ less than $\mu - 2\sigma$ are considered below average.

The two statistical interpretations of three-way decisions are strongly related. Both people's IQ and blood pressure satisfy the normal distribution [63, 81], and based on distributional properties, we know that the mean is equal to the median and the middle region from $\mu - 2\sigma$ to $\mu + 2\sigma$ includes about 95% ESVs [12], while both left and right regions include 2.5% ESVs. This conversion from mean-deviation interpretation to median-percentile interpretation also can be done inversely. In real practices, an imperial rule, namely 68-95-99.7 rule [12] is widely used, which means the middle region is about 68%, 95%, and 99.7% of all ESVs in a normal distribution when $k = 1$, $k = 2$, and $k = 3$, respectively. For non-symmetric distribution such as Figure 4.4, we also can use the mean and deviation to calculate three regions immediately, then the percentage of each region of all ESVs can be calculated by distribution parameters. Another way is that we can do standardization, the new standardized ESVs satisfy normal distribution with $\mu = 0$ and $\sigma^2 = 1$.

The interpretations through mean and standard deviation can also be described by z-score or standard score [79]. An object x's z-score is calculated by $z = (e(x) - \mu)/\sigma$. The z-score integrates mean and deviation together and outcomes only one value.

This value is the ratio of the distance between the object and mean and standard deviation. According to Equation (4.4), k_1 and k_2 are actually the two z-scores of the pair of thresholds. The z-score provides another measure to determine three regions.

4.4 Determining Thresholds in Chi-square Statistic

In this section, we introduce a chi-square statistic based method for determining thresholds in three-way approximation of a concept.

4.4.1 Three-way Approximation of Concepts

Consider a classification problem in which all objects in OB are classified into one of the two categories $\{X, X^C\}$, where X is a set of objects belonging to the given class, $X^C = OB - X$ is the set of objects not belonging to the given class. A fundamental task is to construct rules or a description function to achieve such a classification. Binary classification models are typically used for such a task. However, these models may not produce desirable results such as acceptable classification accuracy. In three-way classification [114], a trisection $\pi_{(\alpha,\beta)}(X)$=($\text{POS}_{(\alpha,\beta)}(X), \text{BND}_{(\alpha,\beta)}(X)$, $\text{NEG}_{(\alpha,\beta)}(X)$) as an approximation of $\{X, X^C\}$ shown in Equation (4.6) is obtained by a pair of thresholds (α, β) on an evaluation function $e(\cdot)$:

$$\begin{aligned}
\text{POS}_{(\alpha,\beta)}(X) &= \{x \in OB \mid e(x) \succeq \alpha\}, \\
\text{BND}_{(\alpha,\beta)}(X) &= \{x \in OB \mid \beta \prec e(x) \prec \alpha\}, \\
\text{NEG}_{(\alpha,\beta)}(X) &= \{x \in OB \mid e(x) \preceq \beta\},
\end{aligned} \tag{4.6}$$

where $\text{POS}_{(\alpha,\beta)}(X)$, $\text{NEG}_{(\alpha,\beta)}(X)$, and $\text{BND}_{(\alpha,\beta)}(X)$ consist of objects that are considered belonging to the class X, not belonging to X, and difficult to classify, respectively; they are called the positive region, negative region, and boundary region of the class X, respectively.

In probabilistic rough set approximation of a concept [113], the probability of X given an equivalence class of object $[x] \subseteq OB$ is used as an evaluation function, i.e., $e(x) = Pr(X|[x])$, where X is a subset of OB. The conditional probability $Pr(X|[x])$ is the ESV of object x and all ESVs are real numbers between 0 and 1. The relation $\succeq$ is the "greater than or equal" relation $\geq$. Under the assumption $0 \leq \beta < 0.5 \leq \alpha \leq 1$, one easily obtains three probabilistic regions by Equation (4.6).

Different choices of thresholds lead to different three-way approximations. A good approximation shows a strong association or correlation of $\pi_{(\alpha,\beta)}(X)$ and $\{X, X^C\}$. In other words, $\pi_{(\alpha,\beta)}(X)$ and $\{X, X^C\}$ are correlated or dependent. The chi-square statistic is a measure of correlation and can be used as an objective function for measuring the goodness of a trisection $\pi_{(\alpha,\beta)}(X)$.

4.4.2 Contingency Table of Three-way Decisions

The connection of the actual classification $\{X, X^C\}$ and a three-way approximation $\pi_{(\alpha,\beta)}(X) = (\mathrm{POS}_{(\alpha,\beta)}(X), \mathrm{BND}_{(\alpha,\beta)}(X), \mathrm{NEG}_{(\alpha,\beta)}(X))$ of $\{X, X^C\}$ can be represented by a contingency table [19] as shown in Table 4.1. The two factors, i.e., the class X and the pair of thresholds (α, β), form the rows and columns, respectively, are two variables of the contingency table. A contingency table has two directions, i.e., row and column; it is also called a cross-classification table.

Table 4.1: A contingency table of three-way decision.

	$\mathrm{POS}_{(\alpha,\beta)}(X)$	$\mathrm{BND}_{(\alpha,\beta)}(X)$	$\mathrm{NEG}_{(\alpha,\beta)}(X)$	Total
X	n_{XP}	n_{XB}	n_{XN}	$n_{X\cdot}$
X^C	n_{X^CP}	n_{X^CB}	n_{X^CN}	$n_{X^C\cdot}$
Total	$n_{\cdot P}$	$n_{\cdot B}$	$n_{\cdot N}$	n

The numbers in the table such as n_{XP} and n_{X^CN} represent the numbers of objects in the corresponding category of a class and a region. Numbers with subscripts having a dot such as $n_{X^C\cdot}$ and $n_{\cdot N}$ are called marginal totals, denoting the numbers of objects in the corresponding row or column. The number n is the grand total. It is the number of all objects in the table, i.e., $n = |OB|$, where $|\cdot|$ is the cardinality of a set. In probabilistic rough sets, numbers in the first column of Table 4.1 are $n_{XP} = |X \cap \mathrm{POS}_{(\alpha,\beta)}(X)|$, $n_{X^CP} = |X^C \cap \mathrm{POS}_{(\alpha,\beta)}(X)|$, and $n_{\cdot P} = |\mathrm{POS}_{(\alpha,\beta)}(X)|$, respectively. Additionally, we can estimate probabilities, such as $Pr(\mathrm{POS}_{(\alpha,\beta)}(X)) = n_{\cdot P}/n$, $Pr(X|\mathrm{POS}_{(\alpha,\beta)}(X)) = n_{XP}/n_{\cdot P}$, and $Pr(X^C|\mathrm{POS}_{(\alpha,\beta)}(X)) = n_{X^CP}/n_{\cdot P}$.

4.4.3 Chi-square Statistic as An Objective Function

The chi-square statistic, also referred to as χ^2 statistic, plays an important role in testing the independence of two variables. Given a contingency table, the χ^2 statistic is computed by:

$$\chi^2 = \sum \frac{(observed - expected)^2}{expected}, \tag{4.7}$$

where the "observed" is the actual observed number in a contingency table cell and the "expected" is the corresponding expected number under the independence assumption. For example, consider the cell $(X, \mathrm{POS}_{(\alpha,\beta)}(X))$, the observed number of objects is n_{XP} and the expected number of objects is computed by assuming independence of $\{X, X^C\}$ and $\pi_{(\alpha,\beta)}(X)$. With the marginal numbers $n_{\cdot P}$ and $n_{X\cdot}$, the expected number is computed by:

$$Pr(X) * Pr(\mathrm{POS}_{(\alpha,\beta)}(X)) * |OB| = (\frac{n_{\cdot P}}{n}\frac{n_{X\cdot}}{n})n = \frac{n_{X\cdot}n_{\cdot P}}{n}. \tag{4.8}$$

The value $(n_{XP} - n_{X\cdot}n_{\cdot P}/n)^2$ measures the divergence of the observed number n_{XP} from the expected number $n_{X\cdot}n_{\cdot P}/n$ under the independence assumption. If the observed value is close or equal to the expected number, then $(n_{XP} - n_{X\cdot}n_{\cdot P}/n)^2$ is close or equal to 0 and $(n_{XP} - n_{X\cdot}n_{\cdot P}/n)^2/(n_{X\cdot}n_{\cdot P}/n)$ is close or equal to 0 as well.

This suggests that the actual number is highly probable due to chance and there is a lack of dependence of X and $\mathrm{POS}_{(\alpha,\beta)}(X)$. By summing up all cells, the chi-square statistics can be used to measure the independence / dependence of $\{X, X^C\}$ and $\pi_{(\alpha,\beta)}(X)$. A higher value of chi-statistic suggests a stronger dependency. Therefore, the chi-square statistic may be used as a measure of the goodness of fit of a three-way approximation $\pi_{(\alpha,\beta)}(X)$.

We can demonstrate the appropriateness of chi-square statistics as an objective function by relating it to the general formulation of objective function as given by Equation (2.2). Each region occupies a column with two cells in the contingency table. We may quantify the quality of each region as a sum of two cells' divergences of observed numbers from their expected numbers as follows:

$$\begin{aligned} Q(\mathrm{POS}_{(\alpha,\beta)}(X)) &= \frac{(n_{XP} - n_{X\cdot}n_{\cdot P}/n)^2}{n_{X\cdot}n_{\cdot P}/n} + \frac{(n_{X^C P} - n_{X^C\cdot}n_{\cdot P}/n)^2}{n_{X^C\cdot}n_{\cdot P}/n}, \\ Q(\mathrm{BND}_{(\alpha,\beta)}(X)) &= \frac{(n_{XB} - n_{X\cdot}n_{\cdot B}/n)^2}{n_{X\cdot}n_{\cdot B}/n} + \frac{(n_{X^C B} - n_{X^C\cdot}n_{\cdot B}/n)^2}{n_{X^C\cdot}n_{\cdot B}/n}, \\ Q(\mathrm{NEG}_{(\alpha,\beta)}(X)) &= \frac{(n_{XN} - n_{X\cdot}n_{\cdot N}/n)^2}{n_{X\cdot}n_{\cdot N}/n} + \frac{(n_{X^C N} - n_{X^C\cdot}n_{\cdot N}/n)^2}{n_{X^C\cdot}n_{\cdot N}/n}. \end{aligned} \quad (4.9)$$

By summing up the three quantities with $w_P = w_B = w_N = 1$, we have:

$$\begin{aligned} Q(\pi_{(\alpha,\beta)}(X)) &= Q(\mathrm{POS}_{(\alpha,\beta)}(X)) + Q(\mathrm{BND}_{(\alpha,\beta)}(X)) + Q(\mathrm{NEG}_{(\alpha,\beta)}(X)) \\ &= \chi^2_{(\alpha,\beta)}. \end{aligned} \quad (4.10)$$

That is, the χ^2 statistic of contingency table of three-way decisions may be viewed as a special case of a measure of the quality of a three-way approximation $\pi_{(\alpha,\beta)}(X)$ as defined by Equation (2.2).

If the χ^2 statistic is statistically significant, that means $\{X, X^C\}$ and $\pi_{(\alpha,\beta)}(X)$ are correlated or dependent; otherwise, they are independent. A larger χ^2 statistic indicates a stronger correlation. Each pair of thresholds (α, β) induces a trisection of

OB. We want to find a pair of thresholds that provides the strongest correlation. In other words, we search for a pair of thresholds by maximizing the χ^2 statistic:

$$(\alpha^*, \beta^*) = \arg \max_{(\alpha,\beta)} \chi^2_{(\alpha,\beta)} \tag{4.11}$$

where (α^*, β^*) is the optimal pair of thresholds. As pointed out by Miller and Siegmund [60], "if the chi-square value is statistically significant, then it can be judged that a predictor variable has been found." In the context of three-way decisions, a good pair of (α, β) is obtained.

4.4.4 Maximizing Chi-square Statistic to Find Thresholds

Based on the framework shown in Equations (4.9) and (4.10), we take a look at every component of the objective function. When $0 \leq \beta < 0.5 \leq \alpha \leq 1$, $Q(\mathrm{POS}_{(\alpha,\beta)}(X))$ is only related to the threshold α and $Q(\mathrm{NEG}_{(\alpha,\beta)}(X))$ is only related to the threshold β. When α changes from 0.5 to 1, $n_{\cdot P}$, n_{XP}, and n_{X^cP} become smaller. However, $Q(\mathrm{POS}_{(\alpha,\beta)}(X))$ may either increase or decrease, that is, $Q(\mathrm{POS}_{(\alpha,\beta)}(X))$ is non-monotonic with respect to α. Similarly, $Q(\mathrm{NEG}_{(\alpha,\beta)}(X))$ and $\chi^2_{(\alpha,\beta)}$ are non-monotonic as well. Thus, a pair of thresholds (α, β) that maximizes the statistic cannot be obtained in a simple analytical expression like in a decision-theoretic rough sets model [121] (i.e., in decision-theoretic rough sets, once the cost matrix is given, the pair of thresholds can be computed directly by equations). Fortunately, given a finite universal set OB, the number of possible values for α and β are limited. The exhaustive search method may work well in these cases.

Many studies [7, 8, 32, 60, 129] discussed the computation of maximally selected chi-square statistic. Boulesteix [7] analyzed maximally selected chi-square statistics in the case of one binary response and nominal predictor. Miller and Siegmund [60], and Boulesteix and Strobl [8] discussed the situation that a predictor variable is generated by two cut-points that have some relationships between each other and combined the two columns of contingency table together. Hothorn and Zeileis [32] explained the general maximally selected statistics and proposed an efficient algorithm that can be applied to compute the maximally selected χ^2 statistic for a 2 by 2 contingency table.

4.4.5 An Illustrative Example

We use an example from [15] to demonstrate the main idea of the proposed method. Suppose that we have a partition of a universal set with 15 equivalence classes $X_1, X_2, \ldots, X_{15}$. Table 4.2 gives the conditional probability of a class X given an equivalence class X_i, that is $Pr(X|X_i)$. To derive three-way decisions to approximate X, we use a pair of thresholds (α, β) with $0 \leq \beta < 0.5 \leq \alpha \leq 1$. The three regions are given by:

$$
\begin{aligned}
\mathrm{POS}_{(\alpha,\beta)}(X) &= \bigcup \{X_i \mid Pr(X|X_i) \geq \alpha\}, \\
\mathrm{BND}_{(\alpha,\beta)}(X) &= \bigcup \{X_i \mid \beta < Pr(X|X_i) < \alpha\}, \\
\mathrm{NEG}_{(\alpha,\beta)}(X) &= \bigcup \{X_i \mid Pr(X|X_i) \leq \beta\}. \qquad (4.12)
\end{aligned}
$$

According to Table 4.2, the sets of possible values of α and β for consideration are $D_\alpha = \{0.5, 0.6, 0.8, 0.9, 1.0\}$ and $D_\beta = \{0.0, 0.1, 0.2, 0.4\}$, respectively.

Table 4.2: Probabilistic information of a class X [15].

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
$Pr(X_i)$	0.0177	0.1285	0.0137	0.1352	0.0580	0.0069	0.0498	0.1070
$Pr(X\|X_i)$	1.0	1.0	1.0	1.0	0.9	0.8	0.8	0.6

	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
$Pr(X_i)$	0.1155	0.0792	0.0998	0.1299	0.0080	0.0441	0.0067	
$Pr(X\|X_i)$	0.5	0.4	0.4	0.2	0.1	0.0	0.0	

Given a sample size n, a pair of thresholds (α, β) produces a contingency table and the corresponding chi-square statistic. Although, the sample size n is not given in [15], the optimal pair of thresholds will not be affected by providing different values of n, i.e., all magnitudes of χ^2 corresponding to different pairs of thresholds will change under different sample sizes, but the ranking of these magnitudes will not change. Based on this setting, n_{XP} can be computed as the closest integer of the following expression:

$$\left(\sum_{X_i \in \text{POS}_{(\alpha,\beta)}(X)} Pr(X|X_i)Pr(X_i) \right) n.$$

The numbers in other cells can be similarly computed. Table 4.3 shows the contingency table for $(\alpha = 0.6, \beta = 0.4)$ and $n = 1000$. All computed numbers in Table 4.3 are modified to their nearest integers, since some computed numbers are not integers when n is set to some numbers. The χ^2 statistic of Table 4.3 is 351.18. By computing contingency tables and the corresponding χ^2 statistics for all possible combinations of α and β, we obtain Table 4.4. Accordingly, $(\alpha = 0.8, \beta = 0.2)$ is selected as the optimal pair of thresholds due to its maximal χ^2 statistic. This means $(\alpha = 0.8, \beta = 0.2)$ provides the strongest correlation between the class X and approximation of $\pi_{(\alpha,\beta)}(X)$.

Table 4.3: The contingency table for $(\alpha = 0.6, \beta = 0.4)$ and $n = 1000$.

	$\text{POS}_{(0.6,0.4)}(X)$	$\text{BND}_{(0.6,0.4)}(X)$	$\text{NEG}_{(0.6,0.4)}(X)$	Total
X	457	58	98	613
X^C	60	58	269	387
Total	517	116	367	1000

Table 4.4: χ^2 statistics for all combinations of (α, β).

	$\beta = 0.0$	$\beta = 0.1$	$\beta =$**0.2**	$\beta = 0.4$
$\alpha = 1.0$	311.24	316.04	368.05	373.31
$\alpha = 0.9$	355.18	358.97	397.12	389.58
$\alpha =$ **0.8**	381.39	384.36	**411.35**	394.72
$\alpha = 0.6$	356.15	358.20	374.02	351.18
$\alpha = 0.5$	310.29	311.53	318.29	292.50

For comparison, the Gini index method [140] also chooses $(\alpha = 0.8, \beta = 0.2)$, the game theory method [3] chooses $(\alpha = 0.5, \beta = 0)$ (using initial search point $(\alpha = 1, \beta = 0.5)$), while the information entropy method [15] chooses $(\alpha = 0.9, \beta = 0.2)$ that provides the second largest χ^2 statistic in Table 4.4. The binary classification in this case chooses $(\alpha = 0.5, \beta = 0.4)$, which provides the empty boundary region and also produces the smallest χ^2 statistic. The Pawlak rough sets method [65] chooses $(\alpha = 1, \beta = 0)$, which provides the largest boundary region and produces the third smallest χ^2 statistic.

Chapter 5

ACTIONABLE RULE MINING

In this chapter, we propose four algorithms for mining actionable rules for the four actionable three-way decision models introduced in Chapter 3.

Because of assumptions (A1) and (A2) in Section 3.3, we can analyze the costs and benefits for each equivalence class individually. For consistent analysis of this chapter, we continue using the medical example in Section 3.1.1. Based on the movement patterns and trisection obtained in Example 3.1, we list all the benefits and costs for each desirable action in Table 5.1.

Table 5.1: All desirable actions in DES with costs and benefits for the medical example.

$[o_3]$	b_{ij}	c_{ij}
$r_{[o_3]} \rightsquigarrow r_{[o_1]}$	2.5	3
$r_{[o_3]} \rightsquigarrow r_{[o_6]}$	16	6

$[o_7]$	b_{ij}	c_{ij}
$r_{[o_7]} \rightsquigarrow r_{[o_1]}$	1.5	2
$r_{[o_7]} \rightsquigarrow r_{[o_6]}$	6	3

$[o_8]$	b_{ij}	c_{ij}
$r_{[o_8]} \rightsquigarrow r_{[o_9]}$	6	5

5.1 Determining the Bounds of Benefit and Cost

Obviously, the lower bounds of both the benefit and the cost of a three-way decision are zero when no action is taken. The upper bound of the benefit $\bar{B}$ can be easily computed by following equation without considering cost:

$$\bar{B} = \sum_{[x_i] \in \text{SOURCE}} \max_{j=1,\ldots,n_i} \{b_{ij}\}, \tag{5.1}$$

However, the upper bound of the benefit may not be unique, because there may exist many actions that have the same benefit to transfer $[x_i]$. Therefore, there may be many solutions of a_{ij} that satisfy model (i). Among these solutions, the one with the minimum cost is the solution for the model (ii). This can be achieved by choosing the action with the maximum benefit to transfer each $[x_i] \in$ SOURCE. If there are two or more actions that have the same maximum benefit, then the action with minimum cost among them is chosen.

We design the Algorithm 1 for the model (i) and the model (ii). The set of a_{ij} found by Algorithm 1 is the actionable three-way solution to obtain the upper bound of benefit $\bar{B}$ and the upper bound of cost $\bar{C}$. Based on Table 5.1, we have $\bar{B} = 28$ and $\bar{C} = 14$.

Algorithm 1: An algorithm to find the upper bounds of benefit and cost.

Input: DES with costs and benefits
Output: $\bar{B}$, $\bar{C}$, and action set a_{ij}.

let $\bar{B} = 0$ and $\bar{C} = 0$;
compute SOURCE;
foreach $[x_i] \in$ SOURCE **do**
    find all $[y_1], \cdots, [y_{n_i}]$, where $r_{[x_i]} \rightsquigarrow r_{[y_j]} \in \text{DES}, j = 1, \cdots, n_i$;
    let $c_{ij} = C_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$, $b_{ij} = B_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$;
    let all $a_{ij} = 0,\ j = 1, \cdots, n_i$;
    let $p = 0$, $q = +\infty$, and $k = 1$;
    for $j = 1$ to n_i **do**
        if *($b_{ij} > p$)* or *($b_{ij} = p$ and $c_{ij} < q$)* **then**
            let $p = b_{ij}$, $q = c_{ij}$;
            let $k = j$;
        endif
    end
    let $a_{ik} = 1$;
    let $\bar{B} = \bar{B} + b_{ik}$;
    let $\bar{C} = \bar{C} + c_{ik}$;
end
return $\bar{B}$, $\bar{C}$, and all a_{ij}.

In Algorithm 1, given DES with actions' costs and benefits, for each equivalence class in SOURCE, all desirable actions in DES will be checked. Therefore, the computational complexity of Algorithm 1 is $|\text{DES}||\text{SOURCE}||A_s \cup A_f|$. The computation for SOURCE is $|AT||OB|^2$.

5.2 Maximum Benefit with Cost Constraints

The problem defined in Definition 3.6 is similar to the multiple-choice knapsack problem (MCKP) [71], where the constraint of a_{ij} in our problem is looser (MCKP requires $\sum_{j=1}^{n_i} a_{ij} = 1,\ a_{ij} \in \{0, 1\},\ i = 1, \ldots, n$). Suppose there are n actionable equivalence classes and each has m actions, then the exhaustive search for the solution has to check m^n combinations. Due to the similarity to the MCKP, it is also NP-hard to find the optimal solution of model (iii).

To efficiently search for an approximate optimal solution of Definition 3.6, a dynamic programming based strategy can be adopted. Suppose we have n actionable equivalence classes given in an order, denoted as $[x_1]$, $\ldots$, $[x_n]$. Any order can be used and will not affect the result of the algorithm. Let $f(i, k)$ denote the maximum benefit for the first i actionable equivalence classes (i.e., $[x_1], \ldots, [x_i]$, $i \leq n$) and k is the limited action cost ($k \leq c_a$). Therefore, $f(n, c_a)$ is the maximum benefit under limited cost c_a. Suppose we know all the values of $f(i-1, k'), k' = 0, \ldots, k$ (i.e., the maximum benefit when we have the first $i-1$ equivalence classes under different limited action costs from 0 to k). To calculate the maximum benefit when we take the i^{th} equivalence class $[x_i]$ into account, we have to consider all $[x_i]$'s actions and the $f(i, k)$ will be computed as the maximum one from the following $n_i + 1$ cases:

(0) $f(i, k) = f(i-1, k)$, if none of $[x_i]$'s actions is taken;

(1) $f(i, k) = f(i-1, k - c_{i1}) + b_{i1}$, if $[x_i]$'s first action is taken;

(2) $f(i,k) = f(i-1, k-c_{i2}) + b_{i2}$, if $[x_i]$'s second action is taken;

$\dots$

(n_i) $f(i,k) = f(i-1, k-c_{in_i}) + b_{in_i}$, if $[x_i]$'s last action is taken.

We define $c_{i0} = 0$ and $b_{i0} = 0$, $i = 0, \dots, n$, i.e., there is no benefit or cost if we do not take any of $[x_i]$'s actions. Thus, the first case (0) can be rewritten in the same form as others, i.e., $f(i,k) = f(i-1, k-c_{i0}) + b_{i0}$. By combining all cases, $f(i,k)$ is computed by:

$$f(i,k) = \max\{f(i-1, k-c_{ij}) + b_{ij} \mid c_{ij} \le k\}, j = 0, \dots, n_i.$$

The number j that maximizes $f(i,k)$ is chosen, which means $[x_i]$'s j^{th} action is taken:

$$a_{ij} = \begin{cases} 1, & j = \arg\max_{l=0,\dots,n_i} \{f(i-1, k-c_{il}) + b_{il} \mid c_{il} \le k\}; \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

This is an iterative strategy gradually reducing the size of problem (i.e., the number of equivalence classes). That is, to compute $f(i,k)$, we have to know $f(i-1,0), \dots,$ and $f(i-1,k)$, and to compute $f(i-1,k)$, we have to know $f(i-2,0), \dots,$ and $f(i-2,k)$. Finally, the base conditions $f(0,0), \dots, f(0,k)$ will be reached. We define $f(0,k) = 0, k = 0, \dots, c_a$, because there is no benefit when no equivalence class can be transferred. Thus, a complete iterative formula for computing $f(i,k)$ can be formulated as follows:

$$f(i,k) = \begin{cases} 0 & \text{if } i = 0; \\ \max\{f(i-1, k-c_{ij}) + b_{ij} \mid c_{ij} \le k\}, j = 0, \dots, n_i & \text{otherwise.} \end{cases} \tag{5.3}$$

We use following example to show how this strategy works.

Example 5.1 *We consider the medical example again. The set of desirable actions* DES *is given in Table 5.1.*

We compute all values of $f(i,k)$ in a table by considering the equivalence classes one by one. Without losing generality, we use the order: $[o_3], [o_7], [o_8]$ and notations $[x_1] = [o_3]$, $[x_2] = [o_7]$, and $[x_3] = [o_8]$. In the beginning, $i = 0$. According to Equation (5.3), we have Table 5.2, in which the column $[x_i]$ lists equivalence classes,

Table 5.2: A maximum benefit computing table when $i = 0$.

$[x_i]$	c_{ij}	b_{ij}	$k=1$	2	3	4	5	6	7	8	9	10
$[x_0]$	0	0	0	0	0	0	0	0	0	0	0	0

column c_{ij} and b_{ij} show action costs and benefits, respectively, and columns from $k = 1$ to 10 stand for different action cost k from 1 to c_a. The $[x_0]$ does not exist, we use it as a symbol to compute $f(i,k)$. The first row is the base condition computed by $f(0,k) = 0, k = 1,\ldots,10$ according to Equation (5.3).

Next, we take $[x_1]$ into account and get Table 5.3 according to Equation (5.3). We

Table 5.3: A maximum benefit computing table when $i = 1$.

$[x_i]$	c_{ij}	b_{ij}	$k=1$	2	3	4	5	6	7	8	9	10
$[x_0]$	0	0	0	0	0	0	0	0	0	0	0	0
$[x_1]$	3, 6	2.5, 16	0(0)	0(0)	2.5(1)	2.5(1)	2.5(1)	16(2)	16(2)	16(2)	16(2)	16(2)

analyze the computations of cell $f(1,1)$ and $f(1,10)$ here, other cells are similar. To compute $f(1,1)$, the current limited action cost is $k = 1$, there is none $[x_1]$'s actions requiring a cost less than or equal to 1. Therefore, 0(0) is written into the cell $(1,1)$, where the row of $[x_0]$ is not counted here, which means the first row is the row of $[x_1]$. The first number 0 before the parenthesis denotes the obtained benefit, the second number 0 in the parenthesis denotes the sequence number of action (i.e., j) that is taken to get the benefit, 0 means that no action is taken. Similarly, when $k = 10$, we have three options for $f(i-1, k-c_{ij}) + b_{ij}$, where $i = 1$ and $j = 0,\ldots,2$. The values are 0(0), 2.5(1), and 16(2), respectively. Therefore, $f(1,10) = \max\{0, 2.5, 16\} = 16$ and 16(2) is written into the cell $(1,10)$.

By repeating the procedure for $[x_2]$ and $[x_3]$, we get Table 5.4. The maximum

Table 5.4: The complete maximum benefit computing table $f(i,k)$.

$[x_i]$	c_{ij}	b_{ij}	$k=1$	2	3	4	5	6	7	8	9	10
$[x_0]$	0	0	0	0	0	0	0	0	0	0	0	0
$[x_1]$	3, 6	2.5, 16	0(0)	0(0)	2.5(1)	2.5(1)	2.5(1)	16(2)	**16(2)**	16(2)	16(2)	16(2)
$[x_2]$	2, 3	1.5, 6	0(0)	1.5(1)	6(2)	6(2)	6(2)	16(0)	16(0)	17.5(1)	22(2)	**22(2)**
$[x_3]$	5	6	0(0)	1.5(0)	6(0)	6(0)	6(0)	16(0)	16(0)	17.5(0)	22(0)	**22(0)**

benefit, 22, is in the bottom right cell of the table, i.e., $f(3,10)$. It is worth mentioning that the optimal solution may be not unique.

Once the maximum benefit is found, the associated set of actions to obtain this benefit, i.e., the set of a_{ij}, can be inferred. According to Table 5.4, the maximum benefit of 22 is reached by taking none of $[x_3]$'s actions. Thus, we consider $f(3-1,10)=f(2,10)$. We get 22(2) in cell (2,10) and it indicates $[x_2]$'s 2^{nd} action is taken. Then we have a remaining cost of 7 by subtracting cost of 3 (the taken action's cost $c_{22}=3$) from 10. Next, we check the cell of $f(2-1,10-3)=f(1,7)$ and we get 16(2), which shows that $[x_1]$'s 2^{nd} action is taken. Finally, by checking $f(1-1,7-6)=f(0,1)=0$, we reach the base condition, inference procedure completes. We get $a_{22}=1$, $a_{12}=1$, and all other a_{ij} are 0. In other words, the optimal solution for obtaining the maximum benefit of 22 with a limited cost of 10 is realized by taking the following actions:

$$r_{[o_7]} \rightsquigarrow r_{[o_6]} \text{ and } r_{[o_3]} \rightsquigarrow r_{[o_6]}.$$

The inference procedure is indicated by arrows in Table 5.4.

According to the strategies analyzed above, an algorithm is designed and shown in Algorithm 2. The algorithm consists of three parts. Part one is from line 1 to line 5, it computes all action costs and benefits for each actionable equivalence class. The second part is from line 6 to line 23, it is the main part of the algorithm computing the complete maximum benefit table (i.e., $f(i,k)$). The last part is from line 24 to line 34, it infers actions (i.e., a_{ij}) which are taken to obtain the maximum benefit. $h(i,k)$ is an action table associated with $f(i,k)$ by simply collecting all numbers in parentheses in Table 5.4. For example, $h(2,4) = 3$ means that $[x_2]$'s 3^{rd} action maximizes the benefit when action cost is limited at 4. Thus, the last part of the algorithm is accomplished by $h(i,k)$ table.

Algorithm 2: Compute maximum benefit with limited action cost.

Input: DES with costs and benefits, c_a.
Output: B, a_{ij}. $//B$ is the approximate maximum benefit

compute SOURCE;
foreach $[x_i] \in$ SOURCE **do**
  find all $[y_1], \ldots, [y_{n_i}]$, where $r_{[x_i]} \rightsquigarrow r_{[y_j]} \in \text{DES}, j = 1, \ldots, n_i$;
  let $c_{ij} = C_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$, $b_{ij} = B_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$;
end
let $f(0,k) = 0$, $h(0,k) = 0$, where $k = 0, \ldots, \lceil c_a \rceil$; $//f$ and h are benefit table and action table, respectively
for $i = 1$ to n **do**
  for $k = 1$ to c_a **do**
    let $b = 0$, $t = f(i-1,k)$, $p = 0$; //temporary variables
    for $j = 1$ to n_i **do**
      if $c_{ij} \leq k$ **then**
        let $b = f(i-1, \lfloor k - c_{ij} \rfloor) + b_{ij}$;
      else
        let $b = 0$;
      endif
      if $b > t$ **then**
        let $t = b$;
        let $p = j$;
      endif
    end
    let $f(i,k) = t$, $h(i,k) = p$;
  end
end
let $B = f(n, c_a)$, $k = c_a$, all $a_{ij} = 0$;
for $i = n$ to 0 **do**
  if $k \leq 0$ **then**
    break;
  endif
  let $t = h(i,k)$;
  if $t > 0$ **then**
    let $a_{it} = 1$;
    let $k = \lfloor k - c_{it} \rfloor$; $//c_{it}$ is the action cost of i^{th} equivalence class' t^{th} action
  endif
end
return B and a_{ij}.

In Algorithm 2, the $\lceil a \rceil$ is a ceil operator that offers the smallest integer larger than or equal to a and the $\lfloor a \rfloor$ is a floor operator that offers the largest integer less than or equal to a. $\lfloor k - c_{ij} \rfloor$ is used to ensure that the column index is always an integer, because the action cost c_{ij} may be a real number in real applications. Thus, $k - c_{ij}$ as a column index might be a non-integer, this makes an incorrect reference to a cell $(i-1, k-c_{ij})$. By using $\lfloor \cdot \rfloor$, each reference to a cell gets an equal or less benefit than the maximum benefit that can be obtained. Therefore, the computed maximum benefit from the algorithm is an approximate value that is equal to or less than the actual maximum benefit. Suppose B' is the actual maximum benefit, B is the benefit obtained by Algorithm 2, they satisfy $(B' - c_a) < B \leq B'$. Specifically, we have $B = B'$ when all c_{ij} are integers.

The computational complexity analysis of Algorithm 2 is straightforward when DES with costs and benefits is given. In the first part of this algorithm, each equivalence class in SOURCE has to check every action in DES by comparing every condition attribute's value. Therefore, the maximum computation of this part is $|\text{DES}||\text{SOURCE}||A_s \cup A_f|$, or is simply denoted as $|OB|^2|AT|$. The second part has three nested loops, the computation is nc_am, where m is the average of all n_i, i.e., $m = 1/n \sum_{i=1}^{n} n_i$. The last part has one loop and its computation is n. Overall, the algorithm reduces the time complexity from exponential to polynomial.

5.3 Minimum Action Cost for A Desired Benefit

In this section, we provide two algorithms for the model (iv).

On the one hand, model (iv) can be solved by the same strategy used in Section 5.2 with a similar complexity. The Table 5.4 can also be used to search for solutions of model (iv). For example, if $b_l = 15$, we only have to consider the first six columns (until $k = 6$), because in the bottom row, the left most column offering benefit greater than or equal to b_l is the sixth column, i.e., $6 = \arg\min_k f(3,k) \geq 15$. However, the table does not offer any benefit greater than 22. Therefore, we need more columns to find such a benefit, but we do not know how many columns are needed for the table. Fortunately, we know the maximum number of columns, that is the upper bound of the cost $\lceil \bar{C} \rceil$. Accordingly, we can make a slight modification to Algorithm 2 by setting $c_a = \lceil \bar{C} \rceil$, then the left most cell in the bottom row of the table with benefit greater than or equal to b_l offers the obtained benefit. Part 3 of the algorithm deriving the set of actions starts from this cell. The summed cost of these derived taken actions is the minimum cost required by model (iv). Such a modified algorithm is designed and shown in Algorithm 3.

Because the obtained benefit is underestimated by Algorithm 2, the action cost needed to obtained a desired benefit by Algorithm 3 is overestimated. Suppose C' is the minimum action cost needed for the desired benefit b_l, C is the approximate action cost computed by Algorithm 3, they satisfy $C' \leq C < (C' + \lceil \bar{C} \rceil)$. We may use $\lceil k - c_{ij} \rceil$ instead of $\lfloor k - c_{ij} \rfloor$ in Algorithm 3, then we have $(C' - \lceil \bar{C} \rceil) < C \leq C'$, it underestimates the cost. The time complexity of the second part is $O(n\lceil \bar{C} \rceil m)$, where m is the average of all n_i. The amount of computation may be huge when some c_{ij} are large.

On the other hand, we can design a different algorithm for model (iv). We use the same notations as the last section and introduce a new function $g(i,k)$, which denotes the minimum action cost with respect to the first i equivalence classes and a desired benefit of k. Therefore, the objective is to compute $g(n, b_l)$ and to find the associated set of a_{ij}. Suppose we know all the values of $g(i-1,k)$, $k = 0, \ldots, b_l$ when we take the i^{th} equivalence class, $[x_i]$, into account to calculate $g(i,k)$. We have to consider all $[x_i]$'s actions and $g(i,k)$ will be computed from one of the following $n_i + 1$ cases:

(0) $g(i,k) = g(i-1,k)$, if none of $[x_i]$'s actions is taken;

(1) $g(i,k) = g(i-1,k-b_{i1}) + c_{i1}$, if $[x_i]$'s first action is taken;

Algorithm 3: Compute minimum action cost for a desired benefit.

Input: DES with costs and benefits, b_l.
Output: C, B, a_{ij}. //C and B are the approximate minimum action cost and approximate obtained benefit, respectively

```
compute SOURCE;
foreach $[x_i] \in$ SOURCE do
    find all $[y_1], \ldots, [y_{n_i}]$, where $r_{[x_i]} \rightsquigarrow r_{[y_j]} \in$ DES, $j = 1, \ldots, n_i$;
    let $c_{ij} = C_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$, $b_{ij} = B_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$;
end
compute $\bar{C}$ by Algorithm 1;
let $f(0,k) = 0$, $h(0,k) = 0$, where $k = 0, \ldots, \lceil \bar{C} \rceil$;  //$f$ and $h$ are benefit table and action
  table, respectively
for $k = 1$ to $\lceil \bar{C} \rceil$ do
    for $i = 1$ to $n$ do
        let $b = 0$, $t = f(i-1,k)$, $p = 0$;  //temporary variables
        for $j = 1$ to $n_i$ do
            if $c_{ij} \leq k$ then
                let $b = f(i-1, \lfloor k - c_{ij} \rfloor) + b_{ij}$;
            else
                let $b = 0$;
            endif
            if $b > t$ then
                let $t = b$;
                let $p = j$;
            endif
        end
        let $f(i,k) = t$, $h(i,k) = p$;
    end
    if $f(i,k) \geq b_l$ then
        break;
    endif
end
let $C = k$, $B = f(n,k)$, all $a_{ij} = 0$;
for $i = n$ to $0$ do
    if $k \leq 0$ then
        break;
    endif
    let $t = h(i,k)$;
    if $t > 0$ then
        let $a_{it} = 1$;
        let $k = \lfloor k - c_{it} \rfloor$;  //$c_{it}$ is the action cost of $i^{th}$ equivalence class' $t^{th}$ action
    endif
end
return $C$, $B$, and $a_{ij}$.
```

(2) $g(i,k) = g(i-1, k - b_{i2}) + c_{i2}$, if $[x_i]$'s second action is taken;

$\ldots$

(n_i) $g(i,k) = g(i-1, k - b_{in_i}) + c_{in_i}$, if $[x_i]$'s last action is taken.

We define $c_{i0} = 0$ and $b_{i0} = 0$ for the first case, then the final iterative formula for $g(i,k)$ is given by:

$$g(i,k) = \begin{cases} +\infty & \text{if } i = 0; \\ \min\{g(i-1, k-b_{ij}) + c_{ij} \mid b_{ij} \leq k\}, j = 0, \ldots, n_i & \text{otherwise,} \end{cases} \tag{5.4}$$

where all $g(0,k)$ are initialized with $+\infty$, $k = 0, \ldots, b_l$, because the iterative part of Equation (5.4) always chooses the minimum value. If $g(0,k)$ are set to 0, the solution will be 0 as well. Additionally, all $g(i,0)$ are defined to 0, $i = 0, \ldots, n$. We use the following example to demonstrate this idea.

Example 5.2 *Consider the same setting as last example. We want to find the solution with minimum action cost for a desired benefit of* 10 *(i.e.,* $b_l = 10$*). We can compute a table shown in Table 5.5.*

Table 5.5: The complete minimum action cost computing table $g(i,k)$.

$[x_i]$	c_{ij}	b_{ij}	k=1	2	3	4	5	6	7	8	9	10
$[x_0]$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
$[x_1]$	3, 6	2.5, 16	2.5(1)3	2.5(1)3	16(2)6	16(2)6	16(2)6	16(2)6	16(2)6	16(2)6	16(2)6	**16(2)6**
$[x_2]$	2, 3	1.5, 6	1.5(1)2	6(2)3	6(2)3	6(2)3	6(2)3	6(2)3	16(0)6	16(0)6	16(0)6	**16(0)6**
$[x_3]$	5	6	1.5(0)2	6(0)3	6(0)3	6(0)3	6(0)3	6(0)3	16(0)6	16(0)6	16(0)6	**16(0)6**

*The content in each cell of the main part of the Table 5.5 has three components. For example, '*6(2)3*' denotes that this cell gets an accumulated benefit of* 6 *(the number on the left hand side of the parentheses), chooses the* 2^{nd} *(the number in parentheses) action of current equivalence class, and requires an accumulated action cost of* 3 *(the number on the right hand side of parentheses). The bottom right cell, i.e.,* $g(3,10) =$ 16(0)6 *provides the minimum action cost of* 6 *to obtain a benefit of* 16*, which satisfies the condition* $16 \geq b_l = 10$*. This cell has the same meaning as the cell* $f(3,6)$ *in Table 5.4.*

The inference procedure for a_{ij} *is similar as previous. Step 1, according to '*$g(3,10) =$ 16(0)6*', none of* $[x_3]$*'s actions is chosen, then we consider* $g(3-1,10)$*. Step 2, by* *checking cell* $(2, 10)$*, its content '*$16(0)6$*' means none of* $[x_2]$*'s actions is chosen as well and we continue to consider* $g(2-1, 10)$*. Step 3, the content in cell* $(1, 10)$ *is '*$16(2)6$*', which means* $[x_1]$*'s* 2^{nd} *action is chosen. Because we have reached the first equivalence class, the inference procedure stops. Therefore, to obtain a desired benefit of* 10*, the minimum action cost is* 6 *and it is achieved by following action:*

$$r_{[o_3]} \rightsquigarrow r_{[o_6]}.$$

The corresponding values for a_{ij} *are* $a_{12} = 1$ *and all others are* 0*.*

According to the strategy analyzed above, we provide Algorithm 4. This algorithm has three parts as well, where the first and third parts are the same as Algorithm 2. The second part computes three tables, i.e., $g(i, k)$, $h(i, k)$, and $l(i, k)$, where $g(i, k)$ is explained above, $h(i, k)$ and $l(i, k)$ are used to save chosen actions and accumulated benefit associated with $g(i, k)$, respectively. By using the ceil operator $\lceil \cdot \rceil$, the computed minimum cost C satisfies $C' \leq C < (C' + b_l)$. Because using $l(i, k)$, Algorithm 4 is able to provide the actually obtained benefit that satisfies the condition of model (iv).

The computational complexity of the second part (i.e., line 6 to line 26) of Algorithm 4 is $O(nb_l m)$, where m is the average number of actions of all equivalence classes. It is similar to the complexity of Algorithm 3, whose computational complexity is $O(n\lceil \bar{C} \rceil m)$. It is obvious that Algorithm 4 will be faster than Algorithm 3 if $b_l < \lceil \bar{C} \rceil$. Therefore, we may choose one of these two algorithms in real practice according to the values of $\lceil \bar{C} \rceil$ and b_l.

Algorithm 4: Compute minimum action cost for a desired benefit (2nd version).

Input: DES with costs and benefits, b_l.

Output: C, B, a_{ij}. //C and B are the approximate minimum action cost and actually obtained benefit, respectively

compute SOURCE;
foreach $[x_i] \in$ SOURCE **do**
    find all $[y_1], \ldots, [y_{n_i}]$, where $r_{[x_i]} \rightsquigarrow r_{[y_j]} \in$ DES, $j = 1, \ldots, n_i$;
    let $c_{ij} = C_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$, $b_{ij} = B_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$;
end
let $g(i, 0) = 0$, $g(0, k) = +\infty$, $h(0, k) = 0$, $l(0, k) = 0$, where $i = 0, \ldots, n$, $k = 1, \ldots, b_l$; //g, h, and l are cost table, action table, and benefit table, respectively
for $i = 1$ to n **do**
    for $k = 1$ to b_l **do**
        let $c = 0$, $b = 0$, $p = 0$, $t = g(i-1, k)$, $q = l(i-1, k)$, $p = 0$;
        for $j = 1$ to n_i **do**
            if $b_{ij} \leq k$ **then**
                let $c = g(i-1, \lceil k - b_{ij} \rceil) + c_{ij}$;
                let $b = l(i-1, \lceil k - b_{ij} \rceil) + b_{ij}$;
            else
                let $c = c_{ij}$;
                let $b = b_{ij}$;
            endif
            if $(c < t)$ or $(c = t$ and $b > q)$ **then**
                let $t = c$;
                let $p = j$;
                let $q = b$;
            endif
        end
        let $g(i, k) = t$, $h(i, k) = p$, $l(i, k) = q$;
    end
end
let $C = g(n, b_l)$, $B = l(n, b_l)$, $k = b_l$, all $a_{ij} = 0$;
for $i = n$ to 0 **do**
    if $k \leq 0$ **then**
        break;
    endif
    let $t = h(i, k)$;
    if $t > 0$ **then**
        let $a_{it} = 1$;
        let $k = \lceil k - b_{it} \rceil$; //$b_{it}$ is the benefit of i^{th} equivalence class' t^{th} action
    endif
end
return C, B, and a_{ij}.

Chapter 6

THE R4 REDUCTION FRAMEWORK FOR ACTIONABLE THREE-WAY DECISIONS

This chapter introduces the R4 reduction framework for actionable three-way decisions. The framework specifies reductions of attributes, attribute-value pairs, classification rules, and actions. The first three types of reductions are based on existing methods that are redefined for the context of actionable three-way decisions and the fourth is novel. Attribute reduction removes attributes from all classification rules to reduce the action cost. Attribute-value pair reduction shortens the left hand side of particular rules to reduce their action cost without sacrificing any of their classification power or action benefit. Rule reduction and action reduction remove redundant classification rules and actions to reduce the computational cost. For the first two types of reductions in R4, the Addition strategy for reduction is adapted and its correctness is proven. Based on this strategy, an algorithm for reductions of attributes and attribute-value pairs is designed.

The major contributions of this chapter are as follows: (1) we propose the R4 framework, a comprehensive, four-step approach to reduce the costs and increase the benefits of finding and applying actions; (2) we modify the existing definitions of attribute reduction, attribute-value pair reduction, and rule reduction to suit the context of actionable three-way decisions; (3) we provide the Addition algorithm schema for the first two steps of the framework and prove its correctness; (4) we design specific instances of this schema for attribute reduction and attribute-value pair reduction.

6.1 Motivation of Reduction and Related Works

In the acting step of three-way decision making, actions are taken with the goal of transferring objects from unfavorable regions to favorable regions to gain benefit [22]. Given a trisection, classification rules are first induced to classify every object in one of the three regions based on the obtained trisection. Then actionable rules for transferring objects are constructed based on the classification rules, and actions for specific objects are chosen according to the actionable rules. Each action may incur a cost (the *action cost*) and bring a benefit (the *action benefit*).

Some studies focused on a *local view*, which is to search for one action for transferring one object or a group of similar objects [75], and other studies focused on a *global view*, which is to search for a set of actions which optimizes an objective function [22, 102]. As an example of a local view, Ras and Wieczorkowska [75] first introduced actionable rules (referred to as action rules in their paper) to move specific customers of a bank from a low profit class to a high profit class to increase profits. Yang et al. [102] applied a similar step to decision trees; their method attempts to change customers from an undesired status to a desired status by taking actions to move them from one node of the decision tree to another. Since they attempted to obtain an optimal tree, they were taking a global view.

Gao and Yao [22] analyzed action costs and benefits, and proposed four actionable models (i.e., Chapter 3) to deal with costs and benefits for different situations. In these four models, the action costs are assumed to be determined by only the attributes and their values. However, some attributes, attribute values, classification rules, and actions may be redundant. In order to reduce costs and increase benefits, we propose here that the notions of a reduct and rule induction be adopted from rough sets [65] and machine learning [73] to remove those redundancies. Reducing the number of attributes and attribute-value pairs may increase benefits and reduce action costs, while reducing the number of rules and actions may reduce computational costs.

There is some existing work related to the reductions of attributes, attribute-value pairs, and rules. Most relevant publications discuss attribute reduction and relatively few discuss attribute-value pair and rule reduction. An *attribute reduct* is a minimal set of attributes that is necessary and sufficient to classify objects [65]. For attribute reduction in cost-sensitive situations, Min et al. [61] introduced test-cost-sensitive attribute reduction that aims to find a reduct with minimum test cost rather than a reduct with maximum accuracy. Jia et al. [37] and Ju et al. [42] discussed attribute reducts with minimum test costs for decision-theoretic rough sets. Miao et al. [59] studied three types of relative attribute reducts. Yao and Zhang [124] analyzed class-specific attribute reducts and their relations. Yao and Zhao [125] analyzed measures for attribute reduction, such as confidence, coverage, cost, and generality, which may be considered when designing attribute reduction algorithms. Ma et al. [57, 58] proposed monotonic uncertainty measures for attribute reduction in a probabilistic rough set model. The *attribute-value pair reduction* is also called *rule simplification*, which simplifies the left hand side of a rule by removing some redundant attribute-value pairs without making the rule unsatisfied. Shan and Ziarko [85] introduced the decision matrix to find all maximally general rules. Ziarko and Shan [146] designed an algorithm for attribute-value pair reduction based on a decision matrix. Yao and Fu [118] gave a formal definition of attribute-value pair reduction for a consistent decision table but did not provide an algorithm. *Rule reduction* removes redundant rules from a set of rules based on certain criteria. One widely applicable criterion for rule reduction is to find a minimum set of rules that has maximum generality, but other criteria can be used in specific applications. Hamilton et al. [30] introduced the RIAC framework that combines attribute reduction and rule simplification for approximate classification. Grzymala-Busse [28] introduced the LEM2 algorithm, which directly induces a set of simplified rules for classification. We adopt the idea of separating the attribute reduction, attribute-value pair reduction, and rule reduction into three sequential steps [118]. We also propose a method for action reduction, which may significantly reduce computational cost.

6.2 Four-step Analysis of Reductions

As previously stated, the goal of this chapter is to describe the R4 algorithmic framework for reducing costs and increasing benefits for four actionable models. Given a decision table, an objective concept, movement patterns, a misclassification cost matrix, and attribute-value changing cost functions, five processing steps are sufficient to produce a solution for any of four actionable models described in Chapter 3. Figure 6.1 illustrates this procedure, in which each processing step is denoted as a rectangle. The R4 framework specifies the first four steps, which are enclosed in a dashed box, i.e., attribute reduction, attribute-value pair reduction, rule reduction, and action reduction. Action reduction is a new process introduced in this thesis. With the R4 framework, the first two steps can reduce the action cost, increase the upper bound of benefit, and increase the maximum benefit under a limited cost, and the last two steps can reduce the computational cost.

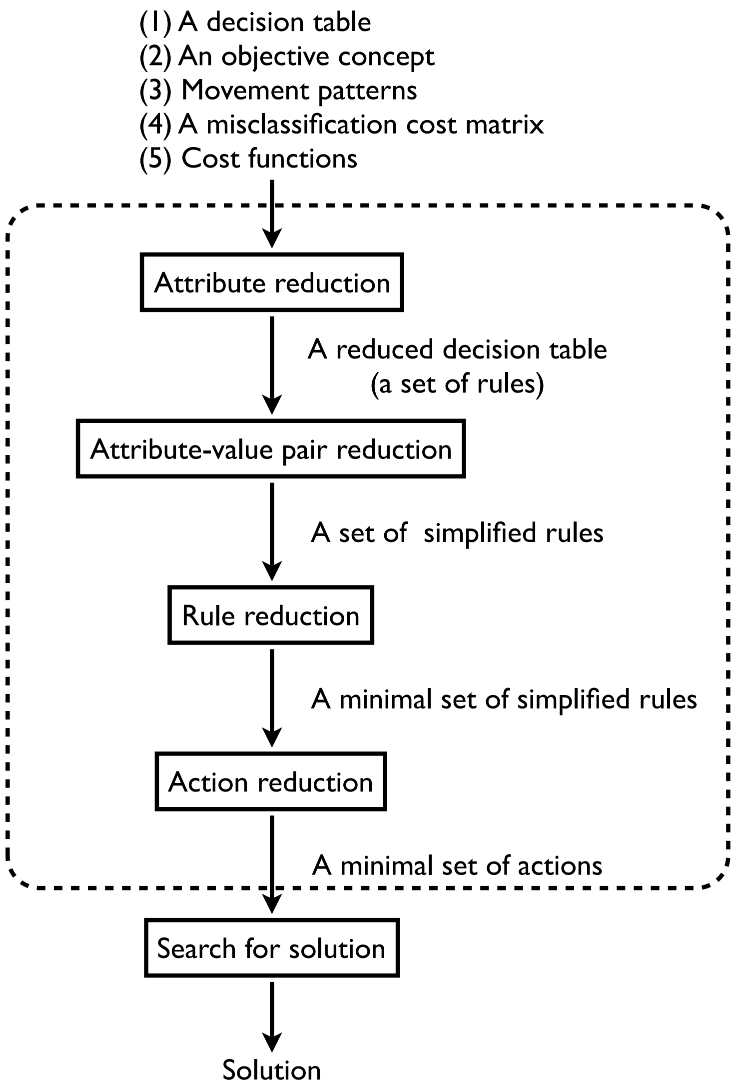


Figure 6.1: An overview of the procedure for acting in actionable three-way decision making with R4.

6.3 Attribute Reduction

An *attribute reduct* of a decision table is a minimal subset of $A_s \cup A_f$ that keeps the same classification power as the whole condition attribute set. The Addition strategy for attribute reduction constructs a reduct from an empty set and adds the attributes to the set one at a time without ever deleting any attributes [23]. Other common attribute reduction strategies, such as the Deletion strategy or the Addition-deletion strategy requires a deletion phase. Since the Addition strategy allows the order of the selection of attributes to be controlled by a heuristic selection function, it may construct a reduct with preferable attributes [142] (e.g., attributes lead to low action cost) compared to other strategies.

Generally, a decision table may have more than one reduct and it is an NP-hard problem to find all reducts [92]. Our motive is to reduce action cost and increase benefit. Therefore in this section, we introduce an algorithm based on the Addition strategy to search for a reduct with an approximately minimum cost.

6.3.1 Attribute Reducts in Three-way Decisions

Let us consider how we can increase benefits and decrease costs for the four actionable models. If some condition attributes are not required to distinguish objects, we can remove them and obtain a subset of $A_s \cup A_f$ without losing any classification power. It is possible to find a subset of the condition attributes that generates the same trisection of OB as $A_s \cup A_f$. According to Equation (3.2), if some stable attributes are removed without losing any classification power, then the resulting subset of $A_s \cup A_f$ may provide more actionable rules to transfer objects and therefore may increase the solution benefit and decrease the action cost. According to Equation (3.5), the cost of each action depends on the costs of its sub-actions, where the sub-actions are based on flexible attributes and their values. If some flexible attributes are removed, then the subset avoids executing some sub-actions for each action and therefore may reduce the action costs. Therefore, to reduce action cost, we must remove some flexible or stable attributes, and to increase solution benefit, we must remove some stable attributes.

To describe the search for a minimal subset of condition attributes that maps the same objects to the favorable and unfavorable regions as are mapped by the whole condition attribute set, we introduce some notation. Let (S_F, S_U, S_I) denote a trisection of OB where S_F, S_U, and S_I are the sets of objects in all favorable regions, all unfavorable regions, and all other regions, respectively, and any two of the sets are pair-wise disjoint; they can be obtained based on the initial trisection π and movement patterns. We use a set of three indicators $\mathbf{T} = \{U, F, I\}$ and a mapping τ to indicate the type of region that an object belongs to:

$$\tau : OB \longrightarrow \mathbf{T},$$

where U, F, and I indicate the unfavorable regions, favorable regions, and other regions, respectively. For example, if $x \in OB$ is in an unfavorable region, i.e, $x \in S_U$, then $\tau(x) = U$. We also define a *indiscernibility relation* [64] $\text{IND}(A \mid \tau)$ for a subset of attributes $A \subseteq (A_s \cup A_f)$ with respect to the mapping τ:

$$\text{IND}(A \mid \tau) = \{(x, y) \in OB \times OB \mid \tau(x) \neq \tau(y)\}. \tag{6.1}$$

Based on this definition, a relative attribute reduct [66] can be redefined as follows.

Definition 6.1 *An attribute set $R \subseteq (A_s \cup A_f)$ from a decision table S is called a* ***relative attribute reduct*** *of S with respect to the mapping τ if R satisfies the following two conditions:*

$$\text{(s1)} \quad \text{IND}(R \mid \tau) = \text{IND}(A_s \cup A_f \mid \tau);$$

$$\text{(n1)} \quad \forall a \in R,\ \text{IND}(R - \{a\} \mid \tau) \neq \text{IND}(A_s \cup A_f \mid \tau).$$

Condition (s1), which is called the *jointly sufficient* condition, ensures that R has the same trisecting power as the whole condition attribute set. Condition (n1), which is called the *individually necessary* condition, ensures that every attribute in A is necessary.

By adapting the idea of a discernibility matrix from rough sets research [92], we construct a version of a discernibility matrix for the context of three-way decision making:

Definition 6.2 *Given a decision table S and a mapping τ, the* ***discernibility matrix*** *$M = (m(x,y))$ is an $|OB| \times |OB|$ matrix, in which the element $m(x,y)$ for an object pair $(x,y) \in OB \times OB$ is defined by:*

$$m(x,y) = \begin{cases} \emptyset, & \tau(x) = \tau(y); \\ \{a \in (A_s \cup A_f) \mid I_a(x) \neq I_a(y)\}, & \tau(x) \neq \tau(y). \end{cases} \tag{6.2}$$

The discernibility matrix M is a symmetric and square matrix where all elements on the principal diagonal are the empty set, i.e., $m(x,y) = m(y,x), \forall x, y \in OB$ and $m(x,x) = \emptyset, \forall x \in OB$. Any element $m(x,y)$ of M is a set of attributes such that each attribute of $m(x,y)$ can distinguish x and y by its values. It is sufficient to consider only the lower triangle or the upper triangle of M. As another formulation, M also can be expressed as a set consisting of all distinct, nonempty elements, that is, $M = \{m(x,y) \mid \forall x, y \in OB \wedge m(x,y) \neq \emptyset\}$. The concept of a relative attribute reduct can also be characterized in terms of the discernibility matrix, as shown by the following theorem [92].

Theorem 6.1 *Given the discernibility matrix M of a decision table S, an attribute set R is a relative attribute reduct of S if and only if*

(s2) $\forall (x,y) \in OB \times OB,\ m(x,y) \neq \emptyset \Rightarrow R \cap m(x,y) \neq \emptyset$;

(n2) $\forall a \in R, \exists (x,y) \in OB \times OB,\ m(x,y) \neq \emptyset \wedge ((R - \{a\}) \cap m(x,y) = \emptyset)$.

The (s2) and (n2) conditions are the jointly sufficient condition and the individually necessary condition, respectively, and the explanations are the same as for (s1) and (n1). Theorem 6.1 provides a criterion to test whether a subset of $A_s \cup A_f$ is a reduct, but it does not directly offer a method to find a reduct. In the rest of this chapter, we use m for $m(x, y)$ and we use the set form $M = \{m \mid m(x, y) \neq \emptyset\}$ to represent the discernibility matrix if there is no confusion. We use RED(M) to denote the set of all reducts of a decision table with a discernibility matrix M.

We use Example 6.1 to demonstrate the above notations.

Example 6.1 *Suppose we have the decision table shown in Table 6.1 for an artificial medical data set, in which age is a stable attribute, chol and bp are flexible attributes, representing for cholesterol level and blood pressure, respectively, and diagnosis is a decision attribute.*

Table 6.1: A decision table for an artificial medical data set.

#	age	chol	bp	diagnosis
o_1	0-29	low	normal	+
o_2	30-59	low	normal	+
o_3	30-59	medium	low	+
o_4	0-29	low	low	+
o_5	30-59	high	high	-
o_6	60+	low	high	-
o_7	0-29	high	high	-
o_8	60+	medium	normal	-

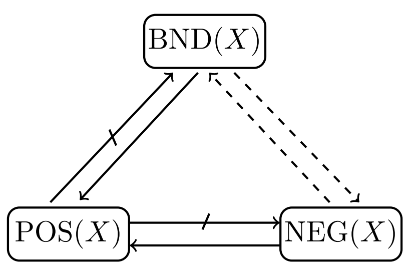


Figure 6.2: Movement patterns for Table 6.1.

Given an objective class $X = \{x \in OB \mid I_{diagnosis}(x) = +\} = \{o_1, o_2, o_3, o_4\}$ *and the misclassification cost matrix shown in Table 6.2, three regions are constructed:*

$$\mathrm{POS}_{(0.5,0.2)}(X) \quad = \{o_1, o_2, o_3, o_4\},$$

Table 6.2: Misclassification cost matrix.

	POS	BND	NEG
X	2	4	8
X^C	11	9	8

$$\begin{aligned} \mathrm{NEG}_{(0.5,0.2)}(X) &= \{o_5, o_6, o_7, o_8\}, \\ \mathrm{BND}_{(0.5,0.2)}(X) &= \emptyset, \end{aligned}$$

where the pair of thresholds $(\alpha = 0.5, \beta = 0.2)$ *is computed based on the given misclassification cost matrix, the computation method can be checked in [122]. The movement patterns shown in Figure 6.2 indicate that* $S_F = \mathrm{POS}(X)$*,* $S_U = (\mathrm{NEG}(X) \cup \mathrm{BND}(X))$*, and* $S_I = \emptyset$*. Since* $\mathrm{BND}(X) = \emptyset$*, we wish to transfer objects from only* $\mathrm{NEG}(X)$ *to* $\mathrm{POS}(X)$*.*

By applying Definition 6.2 to Table 6.1, the discernibility matrix shown in Table 6.3 is computed, only the lower triangle is given. The set representation of the discernibility matrix is:

$$M = \{\{age, chol\}, \{age, bp\}, \{chol, bp\}, \{age, chol, bp\}\}.$$

Table 6.3: Discernibility matrix of Table 6.1.

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8
o_1	$\emptyset$							
o_2	$\emptyset$	$\emptyset$						
o_3	$\emptyset$	$\emptyset$	$\emptyset$					
o_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$				
o_5	$\{age, chol, bp\}$	$\{chol, bp\}$	$\{chol, bp\}$	$\{age, chol, bp\}$	$\emptyset$			
o_6	$\{age, bp\}$	$\{age, bp\}$	$\{age, chol, bp\}$	$\{age, bp\}$	$\emptyset$	$\emptyset$		
o_7	$\{chol, bp\}$	$\{age, chol, bp\}$	$\{age, chol, bp\}$	$\{chol, bp\}$	$\emptyset$	$\emptyset$	$\emptyset$	
o_8	$\{age, chol\}$	$\{age, chol\}$	$\{age, bp\}$	$\{age, chol, bp\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

There are 3 condition attributes and an exhaustive search has to check $2^3 - 2 = 6$

combinations of them, i.e., all possible subsets except $\emptyset$ *and* $A_s \cup A_f$*. Therefore, we can use Theorem 6.1 to check each combination and get the set of reducts:*

$$\mathrm{RED}(M) = \{\{age, chol\}, \{age, bp\}, \{chol, bp\}\}.$$

6.3.2 Categorization of Attributes

In order to construct a reduct, attributes can be categorized in a variety of explicit or implicit way [61, 97, 99, 127, 142]. A widely used categorization is the division of attributes into core attributes and non-core attributes. To build a theoretical foundation for our Addition strategy, which was previously proposed in a somewhat different form [23], we categorize the condition attributes into three categories as follows:

Definition 6.3 *Given a decision table* S*, attributes in* $A_s \cup A_f$ *can be divided into three pair-wise disjoint classes:*

$$\begin{aligned}
&\mathrm{CORE} = \bigcap \mathrm{RED}(M),\\
&\text{USEFUL-NC} = \bigcup \mathrm{RED}(M) - \mathrm{CORE},\\
&\mathrm{USELESS} = A_s \cup A_f - \bigcup \mathrm{RED}(M).
\end{aligned}$$

Attributes in CORE*,* USEFUL-NC*, and* USELESS *are called* ***core****,* ***useful non-core****, and* ***useless*** *attributes, respectively. The set of attributes* $\mathrm{USEFUL} = \bigcup \mathrm{RED}(M) = \mathrm{CORE} \cup \text{USEFUL-NC}$ *is the set of all* ***useful*** *attributes.*

A core attribute is one that appears in a singleton set in a discernibility matrix [92]. Given a discernibility matrix M, the CORE can be easily constructed by visiting each element of M and computing $\mathrm{CORE} = \bigcup \{m \in M \mid |m| = 1\}$.

According to the above classification of attributes, the following properties hold:

(P1) CORE $\cup$ USEFUL-NC $\cup$ USELESS $= A_s \cup A_f$;

(P2) Every attribute in USEFUL-NC appears in at least one reduct, but not in all reducts;

(P3) Any attribute in USELESS does not appear in any reduct;

(P4) Any attribute in CORE appears in every reduct;

(P5) $\text{RED}(M') = \text{RED}(M)$, where $M' = \{m - \text{USELESS} \mid m \in M\}$.

Proof. Properties (P1) to (P4) are easy to prove and the proof of (P5) is given in Appendix A.2.

We further adopt two operations, element absorption [92] and element deletion [126] for simplifying discernibility matrix. They are defined in Definition 6.4.

Definition 6.4 *An* ***element absorption*** *on m' means replacing all m by m' in a discernibility matrix M if $m', m \in M$ and $m' \subset m$. The result of applying all possible element absorptions in some order to M is a new discernibility matrix, M^*, called an* ***absorbed discernibility matrix*** *of M. Given any $A \subseteq A_s \cup A_f$, an* ***element deletion*** *of $m \in M$ with regard to A means replacing m by $m - A$ in M.*

In the set representation of a discernibility matrix, some elements may be removed from M by absorption, i.e., $M^* \subseteq M$. Absorption will not affect any reduct [92], i.e., $\text{RED}(M^*) = \text{RED}(M)$. Although, element deletion may change $\text{RED}(M)$, the existence of at least one reduct is still guaranteed, as stated in Lemma 6.1 [126].

Lemma 6.1 *Given a discernibility matrix M, $W = \bigcup M$, and $A \subseteq W$, if the set of attributes $W - A$ is jointly sufficient (i.e., $\forall m \in M, (W - A) \cap m \neq \emptyset$), then $\text{RED}(M') \neq \emptyset$ and $\text{RED}(M') \subseteq \text{RED}(M)$, where $M' = \{m - A \mid m \in M\}$.*

By making use of the notation in Definition 6.5 [126], we provide Lemma 6.2.

Definition 6.5 *Given an attribute a from a discernibility matrix M, i.e., $a \in \bigcup M$, the group of sets of attributes defined by:*

$$Group_M(a) = \{m \in M \mid a \in m\}$$

is called the ***a-induced group of sets of attributes on*** *M.*

Lemma 6.2 *Given a discernibility matrix M, then $\bigcup M^* =$ USEFUL, i.e., every attribute in the absorbed discernibility matrix is useful.*

Proof. See Appendix A.3.

Lemma 6.2 provides an effective way of computing the USEFUL set. This lemma also leads to an effective way of computing the USEFUL-NC set, since the CORE set is easy to calculate from M. That is, USEFUL-NC = USEFUL − CORE = $\bigcup M^* - \bigcup\{m \in M \mid |m| = 1\} = \bigcup\{m \in M^* \mid |m| > 1\}$. Based on Lemma 6.2, Lemma 6.3 is readily proved.

Lemma 6.3 *Given a discernibility matrix M, an attribute $a \in$ USEFUL if and only if $\exists g \in Group_M(a)$, such that g cannot be absorbed by any other elements in M, i.e., $\forall m \in (M - Group_M(a)), m \not\subseteq g$.*

Proof. See Appendix A.4.

If an attribute a is useful, there must exist some $g \in Group_M(a)$ satisfying Lemma 6.3 that makes a useful. Thus, Lemma 6.3 provides an approach to test whether an attribute is useful without applying any actual element absorption to obtain an absorbed matrix. Thus, we can find some useful attributes and construct a reduct without needing to calculate any element absorptions.

Example 6.2 *Based on the discernibility matrix $M = \{\{age, chol\}, \{age, bp\}, \{chol, bp\}, \{age, chol, bp\}\}$ and $\text{RED}(M) = \{\{age, chol\}, \{age, bp\}, \{chol, bp\}\}$ from Example 6.1,* *the absorbed discernibility matrix and classification of attributes are as follows:*

$$M^* = \{\{age, chol\}, \{age, bp\}, \{chol, bp\}\}.$$

$$\text{CORE} = \emptyset, \quad \text{USEFUL-NC} = \{age, chol, bp\}, \quad \text{USELESS} = \emptyset.$$

6.3.3 The Addition Strategy for Reduct Construction

The Addition strategy for attribute reduct construction starts from either the empty set or the CORE set, adds attributes one at a time [142], and never removes any attributes after they have been added. If it starts from the empty set, the reduct construction procedure can be considered as a search on an attribute lattice $< 2^{A_s \cup A_f}, \subseteq>$ from the top to the bottom. Figure 6.3 illustrates the corresponding attribute lattice for the decision table in Table 6.1, where a downward link connecting a pair of nodes implies a proper subset relationship.

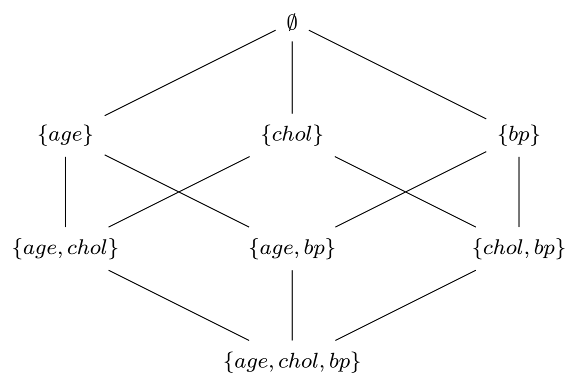


Figure 6.3: The attribute lattice of the decision table in Table 6.1.

During construction, the Addition strategy ensures that the set of attributes is always a subset of at least one reduct in RED. Yao and Zhao [126] provided the formal definition of a partial reduct given in Definition 6.6.

Definition 6.6 *A set of attributes R' is called a* ***partial reduct****, if $\exists R \in$ RED, such that $R' \subseteq R$.*

Once an attribute $a \in \bigcup M$ is chosen to be added into a partial reduct, then M can be simplified by removing all elements in $Group_M(a)$ from M, because all object pairs associated with these elements can be discerned by a. We provide Theorem 6.2, which can be applied to ensure that each added attribute a will be in a partial reduct if two sequential steps are used to shrink M: (1) remove $Group_M(a)$ from M yielding M_1, i.e., $M_1 = M - Group_M(a)$, and (2) delete g from M_1 giving a new version of M, i.e., $M = \{m - g \mid m \in M_1\}$, where $g \in Group(a)$ makes a useful. By repeating these two steps, M will be shrunk to the empty set, and one reduct will be constructed.

Theorem 6.2 *Given a discernibility matrix M, if an attribute $a \in A_s \cup A_f$ is useful, then $\exists g \in Group_M(a)$ that satisfies the following property:*

$$\mathrm{RED}(\{\{a\} \cup M'\}) \subseteq \mathrm{RED}(M),$$

where $M' = \{m - g \mid m \in (M - Group_M(a))\}$.

Proof. See Appendix A.5.

Theorem 6.2 applies to any discernibility matrix, and it suggests an Addition strategy for reduct construction algorithms. In Algorithm 5, we provide an algorithm schema for this Addition strategy. To clearly illustrate the idea, the schema is given by a recursive procedure. As can be seen, the schema contains a tail recursion that can be replaced by a loop. Based on the schema, many algorithms for reduct construction can be designed. We later provide an algorithm instance of this schema by using a loop. We use Example 6.3 to demonstrate the idea of the Addition strategy in detail.

Example 6.3 *We continue using the data given in previous examples. We start to construct a relative attribute reduct from $\emptyset$ (i.e., search from the top of Figure 6.3). We use the order in which the attributes appear in the table for processing, and we use M, R, and CA for the discernibility matrix, partial reduct, and candidate attributes for the partial reduct, respectively.*

Algorithm 5: An algorithm schema of Addition strategy for reduct construction.

Procedure: Addition(M)
Input: A discernibility matrix M.
Output: A relative attribute reduct R.

if $M = \emptyset$ **then**
    return $\emptyset$;
else
    let $R = \emptyset$, $CA = \bigcup M$; // or $R =$ CORE, $CA =$ USEFUL-NC;
    select a useful attribute $a \in CA$;
    let $R = R \cup \{a\}$;
    let $M_1 = M - Group_M(a)$; // shrinking step 1
    select a $g \in Group_M(a)$ such that $\forall m \in M_1, m - g \neq \emptyset$;
    let $M = \{m - g \mid m \in M_1\}$; // shrinking step 2
    let $R = R \cup$ Addition(M);
    return R;
endif

[**Recursion Level 1**]. *Check attribute age.*

The initial values of the variables are as follows: $M = \{\{age, chol\}, \{age, bp\}, \{chol, bp\}, \{age, chol, bp\}\}$, $R = \emptyset$, $CA = \{age, chol, bp\}$, *and* $Group_M(age) = \{\{age, chol\}, \{age, bp\}, \{age, chol, bp\}\}$.

Because there exists $g = \{age, chol\} \in Group_M(age)$, *which makes attribute age useful, we add it to partial reduct:* $R = \emptyset \cup \{age\} = \{age\}$. *After performing the shrinking steps with* $g = \{age, chol\}$, *we have* $M = \{m - g \mid m \in (M - Group_M(age))\} = \{\{bp\}\}$.

The process of this level of recursion is illustrated in Figure 6.4. After adding age into the partial reduct, we will need to search for the reduct in a sublattice that contains all nodes containing the attribute age. Because the $g = \{age, chol\}$ *was deleted from* M, *we do not have to check chol in next level of recursion. This shrinking step further reduces the sublattice to the smaller one shown in Figure 6.5.*

[**Recursion Level 2**]. *Check attribute bp.*

The values of variables are as follows: $M = \{\{bp\}\}$, $R = \{age\}$, *and* $CA = \{bp\}$.

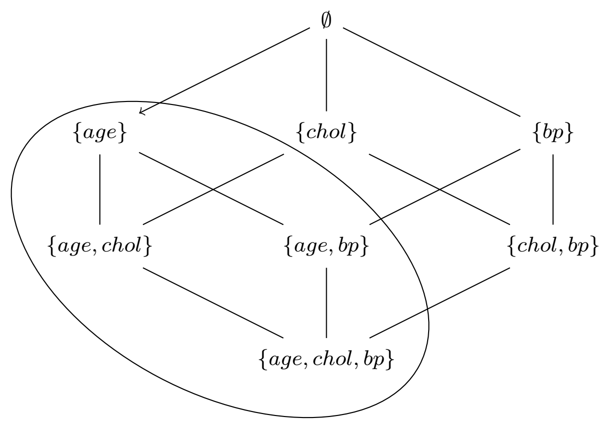


Figure 6.4: The attribute lattice after adding attribute *age*.

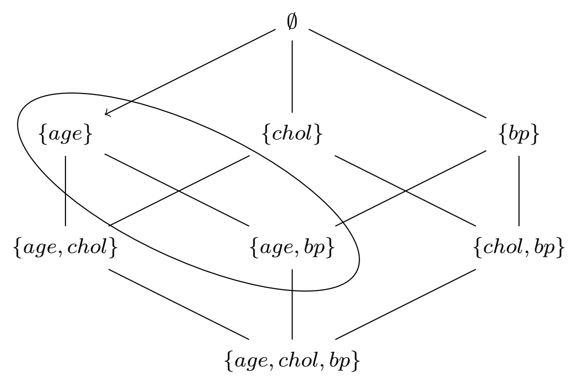


Figure 6.5: The attribute lattice after removing attribute *chol*.

Because $\{bp\}$ is a singleton element in M, bp is a core attribute and therefore it is an useful attribute. We add it to the partial reduct: $R = \{age\} \cup \{bp\} = \{age, bp\}$. We choose $g = \{bp\}$ because it makes attribute bp useful. After performing the shrinking steps with $g = \{bp\}$, we have $M = \{m - g \mid m \in (M - Group_M(bp))\} = \emptyset$.

After adding attribute bp into R, M is empty, which means all objects in OB can be discerned by the attributes in R. The process of this level of resursion is illustrated in Figure 6.6, in which the reduct node $\{age, bp\}$ is circled.

6.3.4 Algorithm and Fitness Functions

Ideally, we would like to construct a reduct that produces the greatest benefit and requires the lowest action cost for any actionable model. However, finding the reduct

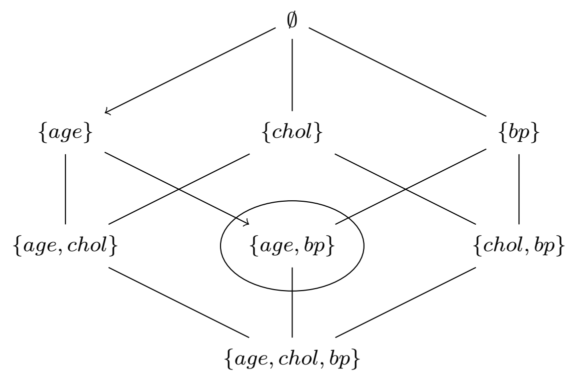


Figure 6.6: The attribute lattice at recursion level 2.

with either lowest cost or the greatest benefit is at least NP-hard, because finding all reducts has been proven to be NP-hard [92]. Therefore, based on the algorithm schema given in Section 6.3.3, we present Algorithm 6 for computing an attribute reduct with an approximately optimal solution. Algorithm 6 uses a while loop to implement the recursive function call in the schema, a foreach loop from line 9 to 13 to determine whether attribute a is useful, and the two subtractions in lines 6 and 17 to implement the two shrinking steps.

The overall time complexity of Algorithm 6 is $O(|M|^3|AT|)$, where $|M|$ is the cardinality of the set representation of discernibility matrix. The foreach loop from line 9 to 13 has to check $|Group_M(a)||M-Group_M(a)||A_s \cup A_f|$ times in the worst case and its time complexity is $O(|M|^2|AT|)$. In practice, M may be reduced significantly in size during the interations of the while loop because of the two shrinking steps in lines 6 and 17.

Algorithm 6 uses two fitness functions, σ and δ, where function σ evaluates each attribute a and function δ evaluates each element of G, i.e., a subset of $Group_M(a)$, for the chosen attribute a. Because we prefer to construct a reduct that requires the lowest action cost and produces the greatest benefit, the fitness functions should be designed to reflect these preferences. The reduct $\{age, bp\}$ found in Example 6.3 without using any fitness functions contains the stable attribute age, which may re-

Algorithm 6: An Addition strategy based algorithm for attribute reduction.

Input: A discernibility matrix M.

Output: A relative attribute reduct R.

```
R = ∅, CA = ⋃M; // or R = CORE, CA = USEFUL-NC;
sort CA by using a fitness function σ;
while M ≠ ∅ do
    let a be the first attribute of sorted CA;
    compute Group_M(a) for a;
    let M = M − Group_M(a);
    let CA = CA − {a};
    let G = ∅;
    foreach g_i ∈ Group_M(a) do
        if ∀m ∈ M, g_i ⊈ m then // g_i makes a useful
            let G = G ∪ {g_i};
        endif
    end
    if G ≠ ∅ then // a is useful
        let R = R ∪ {a};
        select the g ∈ G with the highest value of δ;
        let M = {m − g | m ∈ M};
        let CA = CA − g;
    endif
end
return R.
```

duce the number of possible actions, because an *age* object in an unfavorable region cannot have its value changed to match similar objects in a favorable region. Therefore, this reduct may not lead to a solution with the lowest cost or the highest benefit. In contrast, $\{chol, bp\}$ has two flexible attributes, which provide more potential for possible actions.

In order to generate a solution that requires lower action cost and produces higher benefit, the following heuristic information may be relevant when designing both σ and δ:

(i) Fewer attributes lead to fewer sub-actions, each of which has a cost, and therefore may lead to less overall action cost.

(ii) Fewer stable attributes lead to a greater number of desirable actions and therefore may lead to less overall action cost.

(iii) Attributes with smaller average costs of value changing may lead to less overall action cost.

Recall that function σ is used for choosing attributes. Heuristic information (i) suggests choosing fewer attributes and also choosing attributes with greater classification power. Heuristic information (ii) and (iii) suggest choosing the attribute with the lowest cost for changing attribute values of the objects in unfavorable regions. Thus, we propose fitness function $\sigma(a)$ as follows:

$$\sigma(a) = f(a) \cdot c(a)^{\lambda_\sigma}, \tag{6.3}$$

where $f(a)$ is a function evaluating the classification power of attribute a, $c(a)$ is a function evaluating the cost of changing the value of a for the objects in unfavorable regions, and $\lambda_\sigma \leq 0$ is a parameter controlling the magnitude of the cost penalty. Setting λ_σ to -1 gives the classification power and the cost the same importance. An attribute with the largest value for σ will be firstly considered for adding into a partial reduct.

The classification power, $f(a)$, can be evaluated by frequency or entropy, as shown as Equation (6.4) and Equation (6.5), respectively:

$$\begin{aligned} Frequency: f(a) &= |\{m \in M \mid a \in m\}|, && (6.4)\\ Entropy: f(a) &= H(\{a\}|\tau)^{-1} \\ &= \big(-\sum_{i=1}^{n} P(X_i) \sum_{j \in \{U,F,I\}} P(S_j|X_i) \log(P(S_j|X_i))\big)^{-1}, && (6.5) \end{aligned}$$

where $H(\{a\}|\tau)$ denotes the conditional information entropy of attribute a with regard to τ, and $\{X_1, \ldots, X_n\}$ is a partition of OB introduced by a. Intuitively, if an attribute has the highest frequency of appearance in the discernibility matrix, it may be able to distinguish the largest number of objects. Therefore, the frequency-based strategy selects the available attribute with the highest frequency to be added to the partial reduct. Intuitively, if an attribute has the lowest entropy, it may have the lowest uncertainty for distinguishing objects, i.e., it may have the greatest classification power. Therefore, the entropy-based strategy selects the attribute with the lowest entropy for adding into a partial reduct. In Equation (6.5), we use $H(\{a\}|\tau)^{-1}$ so that the Addition algorithm can consistently select the attribute with the largest value for $f(a)$.

By considering heuristic information (ii) and (iii), function $c(a)$ is defined as shown in Equation (6.6):

$$c(a) = \begin{cases} \frac{1}{|S_U|} \sum_{v \in V'_a} \sum_{y \in S_U} C_a(I_a(y), v), & a \text{ is flexible;} \\ \lambda_s, & a \text{ is stable,} \end{cases} \tag{6.6}$$

where V'_a is the set of all values of attribute a from all objects in S_F and λ_s is a large, fixed value, e.g., $\lambda_s = +\infty$. For any flexible attribute a, the function $c(a)$ needs to visit all condition attributes for each object in S_U. Therefore, the time complexity of computing $c(a)$ is $O(|S_U||AT|)$. An alternative, simpler cost penalty function is given in Equation (6.7):

$$c(a) = \begin{cases} \lambda_f, & a \text{ is flexible;} \\ \lambda_s, & a \text{ is stable,} \end{cases} \tag{6.7}$$

where λ_f and λ_s are fixed values and $\lambda_f \leq \lambda_s$.

The second fitness function δ is applied to evaluate every element in G for a useful attribute a. When shrinking M, we prefer to remove attributes with lowest classification power and highest cost. Therefore, frequency and entropy can be again used to design δ. Suppose we use frequency to evaluate classification power. In a practical method, we want to remove a subset g with a small $\sum_{m \in M} |m \cap g|$ and a large $1/|g|\sum_{b\in g} c(b)$. Thus, we propose the following definition for the fitness function δ:

$$\delta(g) = |\{m \in M \mid m \cap g \neq \emptyset\}|\big(\frac{1}{|g|}\sum_{b\in g} c(b)\big)^{\lambda_\delta}, \tag{6.8}$$

where $\lambda_\delta \leq 0$ is a parameter that controls the magnitude of cost and $c(\cdot)$ is given by Equation (6.6) or Equation (6.7). In Algorithm 6, a subset g with the smallest value for function δ will be chosen to shrink M.

Example 6.4 *We continue from previous examples. Suppose the cost functions for attributes chol and bp are given in Table 6.4 and Table 6.5, respectively. By applying Equations (6.5) and (6.6) to Equation (6.3), and setting $\lambda_\sigma = -1$, $\lambda_s = +\infty$, and $\lambda_\delta = -1$, the relative attribute reduct found by Algorithm 6 is $\{chol, bp\}$.*

Table 6.4: Cost function C_{chol}.

	low	medium	high
low	0	1	3
medium	2	0	1
high	4	1	0

Table 6.5: Cost function C_{bp}.

	low	normal	high
low	0	1	2
normal	1	0	1
high	3	1	0

Given a reduct $R \subseteq A_s \cup A_f$, the notation $[x]_R$ denotes the equivalence class of x with respect to R: $[x]_R = \{y \in OB \mid I_a(y) = I_a(x), \forall a \in R\}$. For simplicity, we will use $[x]$ to represent $[x]_R$ in the remainder of this chapter.

6.3.5 Applying Attribute Reduct to Only Favorable Regions

To produce greater benefit, we only apply the attribute reduct to favorable regions rather than to all regions. In the context of this chapter, objects are grouped into equivalence classes that can be transferred. If we apply the reduct for all regions, some equivalence classes in unfavorable regions based on $A_s \cup A_f$ will be grouped into a larger equivalence class. Such a larger equivalence class has a higher demand on cost, and may exceed the cost limit. By applying attribute reduct for only favorable regions, we will have more possible actions, which may produce a greater benefit solution. We use Example 6.5 with three cases to explain this idea.

Example 6.5 *We continue using the previously introduced decision table, objective class, movement patterns, misclassification cost matrix, and cost functions to transfer objects from* NEG(X) *to* POS(X). *In the remainder of this example, we consider the improvements of benefits, that is to compute the maximum benefit (model (i)), we also compute the maximum benefit under a limited action cost of 8 (model (iii)) in three specific cases.*

Case (1) *Not applying attribute reduction, i.e., all attributes in $A_s \cup A_f$ are used to induce classification rules. We get 8 equivalence classes, each containing one object. There are only 2 transferable objects: o_5 and o_7. Objects o_6 and o_8 are untransferable because there is no object in the favorable regions whose age is 60+. All desirable actions are computed (see Table 6.6). For model (i), the upper bound of the benefit is 12 by taking actions $r_{[o_5]} \rightsquigarrow r_{[o_3]}$ and $r_{[o_7]} \rightsquigarrow r_{[o_1]}$. For model (iii), the maximum benefit under a limited cost of 8 is 6 by taking action $r_{[o_5]} \rightsquigarrow r_{[o_3]}$.*

Table 6.6: All desirable actions with costs and benefits for case (1).

$[o_5]$	b_{ij}	c_{ij}
$r_{[o_5]} \rightsquigarrow r_{[o_2]}$	6	5
$r_{[o_5]} \rightsquigarrow r_{[o_3]}$	6	4

$[o_7]$	b_{ij}	c_{ij}
$r_{[o_7]} \rightsquigarrow r_{[o_1]}$	6	5
$r_{[o_7]} \rightsquigarrow r_{[o_4]}$	6	7

Case (2) *Applying attribute reduction for all regions with reduct $R = \{chol, bp\}$. The equivalence classes based on R are $[o_1] = \{o_1, o_2\}$, $[o_3] = \{o_3\}$, $[o_4] = \{o_4\}$, $[o_5] = \{o_5, o_7\}$, $[o_6] = \{o_6\}$, and $[o_8] = \{o_8\}$. All equivalence classes in unfavorable regions, i.e., $[o_5]$, $[o_6]$, and $[o_8]$ are transferable, because the only stable attribute age is removed after applying attribute reduction. Then, all desirable actions with their benefits and costs are computed (see Table 6.7). The upper bound of benefit is 24 by taking actions $r_{[o_5]} \rightsquigarrow r_{[o_3]}$, $r_{[o_6]} \rightsquigarrow r_{[o_1]}$, and $r_{[o_8]} \rightsquigarrow r_{[o_3]}$. The maximum benefit under a limited cost of 8 is 12 by taking actions $r_{[o_6]} \rightsquigarrow r_{[o_1]}$ and $r_{[o_8]} \rightsquigarrow r_{[o_3]}$.*

Table 6.7: All desirable actions with costs and benefits for case (2).

$[o_5]$	b_{ij}	c_{ij}
$r_{[o_5]} \rightsquigarrow r_{[o_1]}$	12	10
$r_{[o_5]} \rightsquigarrow r_{[o_3]}$	12	8
$r_{[o_5]} \rightsquigarrow r_{[o_4]}$	12	14

$[o_6]$	b_{ij}	c_{ij}
$r_{[o_6]} \rightsquigarrow r_{[o_1]}$	6	1
$r_{[o_6]} \rightsquigarrow r_{[o_3]}$	6	4
$r_{[o_6]} \rightsquigarrow r_{[o_4]}$	6	3

$[o_8]$	b_{ij}	c_{ij}
$r_{[o_8]} \rightsquigarrow r_{[o_1]}$	6	2
$r_{[o_8]} \rightsquigarrow r_{[o_3]}$	6	1
$r_{[o_8]} \rightsquigarrow r_{[o_4]}$	6	3

Case (3) *Restricting attribute reduction to only favorable regions with reduct* $R = \{chol, bp\}$. *In this case,* o_5 *and* o_7 *will not be grouped as one equivalence class. The transferable equivalence classes are* $[o_5] = \{o_5\}$, $[o_6] = \{o_6\}$, $[o_7] = \{o_7\}$, *and* $[o_8] = \{o_8\}$. *All desirable actions are computed (see Table 6.8). Thus, the upper bound of benefit is 24 by taking actions* $r_{[o_5]} \rightsquigarrow r_{[o_3]}$, $r_{[o_6]} \rightsquigarrow r_{[o_1]}$, $r_{[o_7]} \rightsquigarrow r_{[o_3]}$, *and* $r_{[o_8]} \rightsquigarrow r_{[o_3]}$. *We can take actions* $r_{[o_5]} \rightsquigarrow r_{[o_3]}$, $r_{[o_6]} \rightsquigarrow r_{[o_1]}$, *and* $r_{[o_8]} \rightsquigarrow r_{[o_3]}$ *to obtain the maximum benefit of 18 under a limited cost of 8.*

Table 6.8: All desirable actions with costs and benefits for case (3).

$[o_5]$	b_{ij}	c_{ij}
$r_{[o_5]} \rightsquigarrow r_{[o_1]}$	6	5
$r_{[o_5]} \rightsquigarrow r_{[o_3]}$	6	4
$r_{[o_5]} \rightsquigarrow r_{[o_4]}$	6	7

$[o_6]$	b_{ij}	c_{ij}
$r_{[o_6]} \rightsquigarrow r_{[o_1]}$	6	1
$r_{[o_6]} \rightsquigarrow r_{[o_3]}$	6	4
$r_{[o_6]} \rightsquigarrow r_{[o_4]}$	6	3

$[o_7]$	b_{ij}	c_{ij}
$r_{[o_7]} \rightsquigarrow r_{[o_1]}$	6	5
$r_{[o_7]} \rightsquigarrow r_{[o_3]}$	6	4
$r_{[o_7]} \rightsquigarrow r_{[o_4]}$	6	7

$[o_8]$	b_{ij}	c_{ij}
$r_{[o_8]} \rightsquigarrow r_{[o_1]}$	6	2
$r_{[o_8]} \rightsquigarrow r_{[o_3]}$	6	1
$r_{[o_8]} \rightsquigarrow r_{[o_4]}$	6	3

Also, we can reduce cost (models (ii) and (iv)) analogously. In Example 6.5, under a limited cost of 8, case (3) can obtain a benefit of 18, which is greater than the benefits obtained in case (2) and case (1). Furthermore, since the constraint $\bar{B}$, i.e., the upper bound of benefit may be changed after reduction, model (ii) can be analyzed through model (iv). For example, the upper bound of the benefit in case (1) is 12 and its associated minimum cost is 9. After applying attribute reduction for favorable regions in case (3), we are able to transfer all objects from S_U and the cost to obtain the same benefit (i.e., 12) is reduced to 2 by taking actions $r_{[o_6]} \rightsquigarrow r_{[o_1]}$ and $r_{[o_8]} \rightsquigarrow r_{[o_3]}$. This is because case (3) groups the objects of S_U at the finest granularity, i.e., with the smallest equivalence classes, constructed based on $A_s \cup A_f$. Intuitively, this process splits some actions in case (2) into smaller actions in case (3) and allows us to take a part of them for computing the solution. In case (1), we can transfer only o_5 and o_7 to favorable regions, because o_6 and o_8 cannot be transferred since they contain the stable attribute, *age*.

6.4 Attribute-value Pair Reduction

The idea of attribute-value pair reduction is similar to attribute reduction. It can be implemented as rule simplification by removing redundant attribute-value pairs without sacrificing the classification power of the rules. Attribute-value pair reduction may reduce the cost of some actions. For example, suppose two equivalence classes $[x]$ and $[y]$ are from S_F and S_U, respectively, and two rules induced from them based on a reduct $\{bp, chol\}$ are $r_{[x]} : bp = normal \wedge chol = medium \Rightarrow diagnosis = +$ and $r_{[y]} : bp = high \wedge chol = high \Rightarrow diagnosis = -$, respectively. We have to change both the blood pressure and the cholesterol level for $[y]$ to transfer it to $[x]$. If we can remove the attribute-value pair $chol = medium$ from the rule $r_{[x]}$ without reducing any classification power, then the simplified rule $r_{[x]}$ leads to a shorter actionable rule $r_{[y]} \rightsquigarrow r_{[x]} : bp : high \rightsquigarrow normal \Rightarrow diagnosis : - \rightsquigarrow +$. The cost of this action is reduced because we do not have to pay the cost for changing the cholesterol level.

The decision matrix, a structure similar to the discernibility matrix, was introduced by Shan and Ziarko [85] for attribute-value pair reduction. In the context of actionable three-way decision, we provide a new definition as follows:

Definition 6.7 *Given a decision table S and an attribute reduct R, the* ***decision matrix*** $D = (d([x], [y]))$ *is an $m \times n$ matrix, in which $[x] \subseteq S_F$ and $[y] \subseteq S_U$ are equivalence classes on R, m and n are the numbers of equivalence classes in S_F and S_U, respectively, and the element $d([x], [y])$ is a set of attribute-value pairs defined by:*

$$d([x], [y]) = \{a = I_a(x) \mid a \in R, I_a(x) \neq I_a(y)\}. \tag{6.9}$$

The decision matrix of Table 6.1 is shown in Example 6.6. Every row of a decision table includes sufficient information to distinguish an equivalence class (row) from S_F and all equivalence classes (columns) from S_U. We can get a classification rule whose left hand side is a conjunction of all attribute-value pairs of that row. The task of this section is to find a subset of attribute-value pairs for each row that leads to a classification rule which further leads to a solution with lowest cost.

Example 6.6 *Based on a reduct* $R = \{chol, bp\}$, $S_F = [o_1] \cup [o_3] \cup [o_4] = \{o_1, o_2, o_3, o_4\}$, *and* $S_U = [o_5] \cup [o_6] \cup [o_8] = \{o_5, o_6, o_7, o_8\}$, *the decision matrix of Table 6.1 is computed in Table 6.9.*

Table 6.9: Decision matrix for favorable regions.

	$[o_5]$	$[o_6]$	$[o_8]$
$[o_1]$	$chol = low, bp = normal$	$bp = normal$	$chol = low$
$[o_3]$	$chol = medium, bp = low$	$chol = medium, bp = low$	$bp = low$
$[o_4]$	$chol = low, bp = low$	$bp = low$	$chol = low, bp = low$

The conjunction of the attribute-value pairs in the first row is: $chol = low \wedge bp = normal \wedge bp = normal \wedge chol = low$, *i.e.,* $chol = low \wedge bp = normal$. *Therefore, the classification rule of row* $[o_1]$ *is:*

$$r_{[o_1]} : chol = low \wedge bp = normal \Rightarrow diagnosis = +.$$

Each row of the decision matrix can be treated as a discernibility matrix, and each attribute-value pair in the row can be treated as an attribute. Then all properties and theorems for discernibility matrices are also satisfied for each row of the decision matrix. We provide the definition of attribute-value pair reduct in terms of the decision matrix as follows:

Definition 6.8 *Given a row* $d([x], [y_i]), i = 1, \ldots, n$ *of a decision matrix, let* $M = \{d([x], [y_i])\}$, *let* $AV = \bigcup_{i=1,\ldots,n} d([x], [y_i])$ *be the set of all attribute-value pairs in this* *row.* $R \subseteq AV$ *is an* ***attribute-value pair reduct*** *if it satisfies the following two conditions:*

$$\text{(s3)} \quad \forall d([x],[y_i]) \in M, \; R \cap d([x],[y_i]) \neq \emptyset;$$

$$\text{(n3)} \quad \forall a \in R, \exists d([x],[z]) \in M, \; (R - \{a\}) \cap d([x],[z]) = \emptyset.$$

The attribute-value pair reduct can be derived for each row of the decision matrix and treated as a simplified rule. As with attribute reduction, the result of attribute-value pair reduction is not unique. A reduction method based on the algorithm schema of Algorithm 5 can be applied to each row of the decision matrix for attribute-value pair reduction. In particular, the Algorithm 6 can be directly applied to each row of the decision matrix for attribute-value pair reduction. In this case, the symbols in Algorithm 6 are replaced as follows: M is a row of the decision matrix, CA is the set of all attribute-value pairs in the row, and $Group_M(a)$ is the set of attribute-value pairs containing attribute-value pair a. The algorithm must be performed separately on each row of a decision matrix.

As with attribute reduction, only the classification rules induced from favorable regions require simplification. We use Example 6.7 to demonstrate this procedure.

Example 6.7 *We continue the same setting of Example 6.5 and 6.6. Based on Table 6.9, it is easy to compute simplified rules for favorable regions and they are listed below:*

$r'_{[o_1]} : \; chol = low \wedge bp = normal \Rightarrow diagnosis = +,$

$r'_{[o_3]} : \; bp = low \Rightarrow diagnosis = +,$

$r'_{[o_4]} : \; bp = low \Rightarrow diagnosis = +.$

Based on these simplified rules, the benefits and the costs of some actions are reduced and we have a new list of desirable actions in the DES *shown in Table 6.10.*

According to the table, the maximum benefit under a limited cost of 8 is now 24, which is the upper bound in this case. For comparison, the cost was 10 to obtain the same benefit without rule simplification (calculated from Table 6.8). Obviously, the minimum cost for obtaining a desired benefit is lowered after rule simplification.

Table 6.10: All desirable actions in DES with costs and benefits after attribute reduction and rule simplification.

$[o_5]$	b_{ij}	c_{ij}
$r_{[o_5]} \rightsquigarrow r'_{[o_1]}$	6	5
$r_{[o_5]} \rightsquigarrow r'_{[o_3]}$	6	3
$r_{[o_5]} \rightsquigarrow r'_{[o_4]}$	6	3

$[o_6]$	b_{ij}	c_{ij}
$r_{[o_6]} \rightsquigarrow r'_{[o_1]}$	6	1
$r_{[o_6]} \rightsquigarrow r'_{[o_3]}$	6	3
$r_{[o_6]} \rightsquigarrow r'_{[o_4]}$	6	3

$[o_7]$	b_{ij}	c_{ij}
$r_{[o_7]} \rightsquigarrow r'_{[o_1]}$	6	5
$r_{[o_7]} \rightsquigarrow r'_{[o_3]}$	6	3
$r_{[o_7]} \rightsquigarrow r'_{[o_4]}$	6	3

$[o_8]$	b_{ij}	c_{ij}
$r_{[o_8]} \rightsquigarrow r'_{[o_1]}$	6	2
$r_{[o_8]} \rightsquigarrow r'_{[o_3]}$	6	1
$r_{[o_8]} \rightsquigarrow r'_{[o_4]}$	6	1

We now explain why attribute-value pair reduction should not be applied to unfavorable regions. For example, if the classification rule of $[o_5]$ in the unfavorable region, $r_{[o_5]}$: $chol = high \wedge bp = high \Rightarrow diagnosis = -$ is simplified into $r'_{[o_5]}$: $chol = high \Rightarrow diagnosis = -$, then we can transfer $[o_5]$ to only $[o_1]$, because we removed the attribute-value pair of the blood pressure that is required to transfer $[o_5]$ to $[o_3]$ or $[o_4]$.

To specify Algorithm 6 for attribute-value pair reduction, we only have to customize two fitness functions, i.e., σ and δ. For σ, we may use the following two types of heuristic information:

(i) The cost, i.e., the average sub-action cost of an attribute-value pair. It is the average cost of changing the attribute values of unfavorable objects. The average cost for attribute-value pair $a = I_a(x)$ is:

$$c(a = I_a(x)) = \begin{cases} \frac{1}{|S_U|} \sum_{y \in S_U} C_a(I_a(y), I_a(x)), & a \text{ is flexible;} \\ \lambda_s, & a \text{ is stable,} \end{cases} \tag{6.10}$$

where $\lambda_s \geq 0$ is a large, fixed value, e.g., $\lambda_s = 10^9$ or $\lambda_s = +\infty$.

(ii) The confidence, i.e., the probability that an object has value $I_a(x)$ for a if its decision attribute value is $I_d(x)$. An attribute-value pair with higher confidence is considered to have a higher coverage of the extension of a rule. The confidence of $a = I_a(x)$ is defined as:

$$p(d = I_d(x) \mid a = I_a(x)) = \frac{|m(a = I_a(x)) \cap m(d = I_d(x))|}{|m(a = I_a(x))|}, \tag{6.11}$$

where $m(\cdot)$ is a set of all objects satisfying the condition in the parenthesis, e.g., $m(d = I_d(x)) = \{y \in OB \mid I_d(y) = I_d(x)\}$.

We prefer a rule that consists of attribute-value pairs with lower costs and higher confidences. Therefore, the fitness function σ can be designed as the ratio of these two values:

$$\sigma(a = I_a(x)) = p(d = I_d(x) \mid a = I_a(x)) c(a = I_a(x))^{\lambda_\sigma}, \tag{6.12}$$

where $\lambda_\sigma \leq 0$. An attribute-value pair with the largest fitness value of σ will be checked first by the Algorithm 6.

For function δ, let $Group_M(a = I_a(x))$ be the set of $d([x], [y])$ containing $a = I_a(x)$. We use an equation similar in form to Equation (6.8) for any $g \in Group_M(a = I_a(x))$:

$$\delta(g) = |\{m \in M \mid m \cap g \neq \emptyset\}| \cdot \big(\frac{1}{|g|} \sum_{b \in g} c(b)\big)^{\lambda_\delta}, \tag{6.13}$$

where M is a row of the decision matrix, b is an attribute-value pair, $c(\cdot)$ is as given in Equation (6.10), and $\lambda_\delta \leq 0$. An element g with the smallest value for the fitness function δ will be chosen to shrink M.

6.5 Rule Reduction

Rule reduction is used to remove redundant rules produced by rule simplifications. A common approach, which is based on generality, is to search for a minimum subset of rules that covers OB or all classifiable objects [30, 118]. For example, suppose there are two rules: $r_1 : a = 1 \Rightarrow d = +$ and $r_2 : b = 2 \Rightarrow d = +$. If $m(a = 1) = \{o_1, o_2\}$ and $m(b = 2) = \{o_1, o_2, o_7\}$, i.e., the r_2 is more general, then r_1 can be removed.

In the context of actionable three-way decisions, costs and benefits are more important than generality. A less general classification rule is not redundant if it promotes an action with lower cost. Based on this idea, we provide a new definition of redundant rules in Definition 6.9.

Definition 6.9 *Given an equivalence class* $[x] \subseteq S_F$, $r_{[x]}$ *is a* ***redundant rule*** *if for any desirable action* $r_{[y_i]} \rightsquigarrow r_{[x]}, [y_i] \subseteq S_U$, *there exists a desirable action* $r_{[y_i]} \rightsquigarrow r_{[z]}, [z] \neq [x], [z] \subseteq S_F$, *such that the benefit of* $r_{[y_i]} \rightsquigarrow r_{[z]}$ *is greater than or equal to the benefit of* $r_{[y_i]} \rightsquigarrow r_{[x]}$ *and the cost of* $r_{[y_i]} \rightsquigarrow r_{[z]}$ *is less than or equal to the cost of* $r_{[y_i]} \rightsquigarrow r_{[x]}$.

According to the definition, any rule that may reduce the cost or increase the benefit is not redundant. Therefore, removing redundant rules will not affect the benefit or cost of a solution.

Unfortunately, checking whether a rule is redundant requires the calculation of all its actions with their costs and benefits. This checking may be infeasible in practice. A special case of redundant rules is duplicate rules, which have the same left hand sides and the same right hand sides. Due to computational cost, we suggest only dealing with this special case by keeping one rule and remove all its duplicates. Since an algorithm for removing such duplicate rules is straightforward, we omit it.

We provide an example illustrating the main idea of rule reduction in Example 6.8.

Example 6.8 *Based on the simplified rule list from Example 6.7,* $r'_{[o_3]}$ *and* $r'_{[o_4]}$ *are identical, which means one of them can be removed without affecting the solution. Suppose we remove* $r'_{[o_4]}$. *We obtain the following list of rules in favorable regions:*

$r'_{[o_1]} : chol = low \wedge bp = normal \Rightarrow diagnosis = +,$

$r'_{[o_3]}$: $bp = low \Rightarrow diagnosis = +$.

As before, we get all desirable actions listed in Table 6.11. Therefore, the maximum benefit under a limited cost of 8 is 24 and the minimum cost for obtaining the upper bound of benefit is 8. The results are the same as in Example 6.7.

Table 6.11: All desirable actions in DES with costs and benefits after rule reduction.

$[o_5]$	b_{ij}	c_{ij}
$r_{[o_5]} \rightsquigarrow r'_{[o_1]}$	6	5
$r_{[o_5]} \rightsquigarrow r'_{[o_3]}$	6	3

$[o_6]$	b_{ij}	c_{ij}
$r_{[o_6]} \rightsquigarrow r'_{[o_1]}$	6	1
$r_{[o_6]} \rightsquigarrow r'_{[o_3]}$	6	3

$[o_7]$	b_{ij}	c_{ij}
$r_{[o_7]} \rightsquigarrow r'_{[o_1]}$	6	5
$r_{[o_7]} \rightsquigarrow r'_{[o_3]}$	6	3

$[o_8]$	b_{ij}	c_{ij}
$r_{[o_8]} \rightsquigarrow r'_{[o_1]}$	6	2
$r_{[o_8]} \rightsquigarrow r'_{[o_3]}$	6	1

Rule reduction is only required in favorable regions. Because rule simplification is not applied to unfavorable regions, there are no duplicated rules in unfavorable regions. The removal of duplicate rules will not reduce the cost or improve the benefit. However, for some data sets, a great number of the redundant rules can be removed, which reduces the computational cost.

6.6 Action Reduction

The last step in the R4 framework is action reduction, which removes redundant actions from the set of desirable actions DES. There may be many actions for transferring an equivalence class from S_U to S_F, but only one or none of them will be taken. Intuitively, actions with low cost or high benefit are preferred, while actions with both relatively high cost and low benefit may never been taken.

With respect to an equivalence class in S_U, some of its actions are comparable and redundant. For example, in Table 6.11, after the above three steps of reductions, $[o_5]$ has two actions, i.e., $r_{[o_5]} \rightsquigarrow r'_{[o_1]}$ and $r_{[o_5]} \rightsquigarrow r'_{[o_3]}$. Both have the same benefit, but $r_{[o_5]} \rightsquigarrow r'_{[o_1]}$ has a higher cost, which makes it redundant because it will never been taken. However, if one action has both higher benefit and higher cost than another action, then the two actions are incomparable. We provide a formal definition of a redundant action in Definition 6.10.

Definition 6.10 *Given an action* $r_{[x]} \rightsquigarrow r_{[y]}$ *that transfers* $[x]$*, its cost and benefit are* c *and* b*, respectively.* $r_{[x]} \rightsquigarrow r_{[y]}$ *is a* ***redundant action*** *if*

$$\exists r_{[x]} \rightsquigarrow r_{[y_i]}, c \geq c_i \text{ and } b \leq b_i, \tag{6.14}$$

where c_i *and* b_i *are the cost and benefit of* $r_{[x]} \rightsquigarrow r_{[y_i]}$*, respectively.*

Algorithm 7: An action reduction algorithm.

Input: A set of all desirable actions DES.
Output: A reduced desirable action list DES′.

compute SOURCE;
let $\text{DES}' = \emptyset$;
foreach $[x_i] \in \text{SOURCE}$ **do**
    let $A = \{r_{[x_i]} \rightsquigarrow r_{[y]} \mid r_{[x_i]} \rightsquigarrow r_{[y]} \in \text{DES}\}$;
    let $R = \emptyset$;
    foreach action $a \in A$ **do**
        let c and b be the cost and benefit of a, respectively;
        let $bRedundant =$ **false**;
        foreach $r_{[x_i]} \rightsquigarrow r_{[y_j]} \in R$ **do**
            let $c_{ij} = C_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$, $b_{ij} = B_{r_{[x_i]} \rightsquigarrow r_{[y_j]}}$;
            if $c \geq c_{ij}$ **and** $b \leq b_{ij}$ **then** // action a is redundant
                let $bRedundant =$ **true**;
                break;
            endif
        end
        if $bRedundant =$ **false then**
            let $R = R \cup \{a\}$;
        endif
    end
    let $\text{DES}' = \text{DES}' \cup R$;
end
return DES′;

Algorithm 7 is designed for action reduction. The algorithm consists of three nested loops that need $|\text{SOURCE}||A||R|$ comparisons, where $1 \leq |R| \leq |A| \leq |\text{SOURCE}|$. In the worst case, $|\text{SOURCE}| = |OB|$, i.e., every equivalence class contains one object. Therefore, the overall time complexity of the algorithm is $O(|OB|^3)$.

The effect of action reduction is similar to rule reduction because it improves computation time but does not improve the cost or benefit. Nonetheless, the search for solutions of actionable models can be significantly accelerated. Let us continue using the previous examples to demonstrate the idea of action reduction.

Example 6.9 *Continuing from Example 6.8, we apply action reduction to the* DES *set given in Table 6.11. According to the definition of a redundant action, actions* $r_{[o_5]} \rightsquigarrow r'_{[o_1]}$, $r_{[o_6]} \rightsquigarrow r'_{[o_3]}$, $r_{[o_7]} \rightsquigarrow r'_{[o_1]}$, *and* $r_{[o_8]} \rightsquigarrow r'_{[o_1]}$ *are redundant with regard to equivalence classes* $[o_5]$, $[o_6]$, $[o_7]$, *and* $[o_8]$, *respectively. Therefore, we can remove them from* DES. *The reduced* DES *is shown in Table 6.12.*

Table 6.12: All desirable actions in DES after action reduction.

$[o_5]$	b_{ij}	c_{ij}
$r_{[o_5]} \rightsquigarrow r'_{[o_3]}$	6	3

$[o_6]$	b_{ij}	c_{ij}
$r_{[o_6]} \rightsquigarrow r'_{[o_1]}$	6	1

$[o_7]$	b_{ij}	c_{ij}
$r_{[o_7]} \rightsquigarrow r'_{[o_3]}$	6	3

$[o_8]$	b_{ij}	c_{ij}
$r_{[o_8]} \rightsquigarrow r'_{[o_3]}$	6	1

The maximum benefit under a limited cost of 8 is still 24, which is also the upper bound of the benefit. The results are not changed with respect to Example 6.8.

Chapter 7

EXPERIMENTAL EVALUATIONS

This chapter provides experimental results for evaluating the actionable rule mining algorithms, the R4 reduction framework, and comparing with some existing methods.

7.1 Effectiveness of Four Actionable Models

We experimented with several data sets from UCI Machine Learning Repository [17] and received similar results. In this section, we provide the experimental results of our proposed four actionable models on a single data set. The experiments in this section are completed on Matlab 2013a running on a Mac OS 10.9.5 with a dual core Intel Core i5 2.4 GHz CPU and 8 GB 1600 MHz DDR3 RAM.

We use the Heart Disease Cleveland data set [25], which has 303 people, 13 symptoms, and one diagnosis. Three attributes, *age*, *sex*, and *ca* (i.e., number of major vessels) are recognized as stable attributes, while the others are flexible. The values of some attributes' values are grouped and reassigned as follows. Age is categorized into 5 groups, i.e., 0-19, 20-39, 40-59, 60-79, and 80+, they are reassigned to values 1, 2, 3, 4, and 5, respectively. Cholesterol is categorized into 3 groups: 0-199, 200-239, and 240+, they are reassigned to values 1, 2, and 3, respectively. Blood pressure is categorized into 3 groups: 0-89, 90-139, and 140+, they are reassigned to 1 to 3 as well. Maximum heart rate is categorized into 3 groups: 0-149, 150-209, and 210+, they are reassigned to 1, 2, and 3, respectively. For simplicity, all missing values are filled in with highest frequent values of corresponding attribute. The decision attribute has 5 categories, valued from 0 to 4, in which only the value 0 means healthy. Therefore, we construct three regions, namely POS, BND, and NEG to approximate the concept of healthy people $X = \{x \in OB \mid I_d(x) = 0\}$. A misclassification cost matrix in Table 7.1 is used to compute the quality of three regions.

To demonstrate how our methods work, we use a simple cost function $C_f(v_1, v_2) = scale * |v_1 - v_2|$ for all flexible attributes. The parameter *scale* in this function is a scale for sub-action costs and controls the magnitude of action cost during experiments. By using different values of *scale*, we are able to show the difference between the ceil and floor versions of Algorithm 3 and their differing computation times.

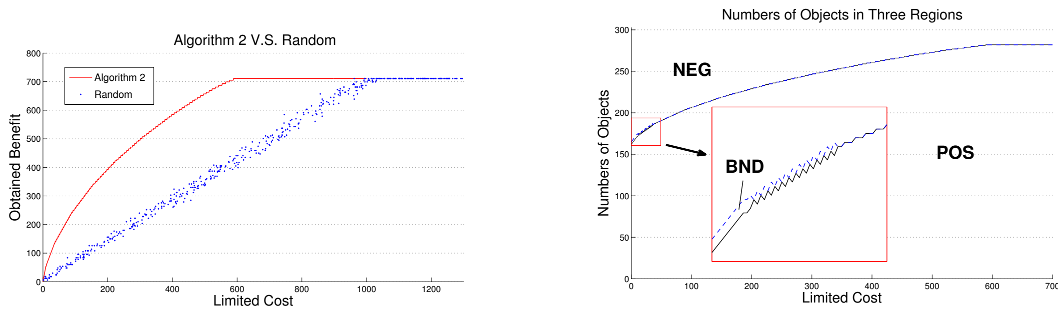
Table 7.1: Cost matrix for experiments.

	POS	BND	NEG
X	2	3	6
X^C	12	9	8

7.1.1 Evaluations of Algorithm 1 and Algorithm 2

Based on this setting, the upper bounds $\bar{B} = 711$ and $\bar{C} = 589$ are easily found by Algorithm 1 when $scale = 1$. Two experiments of Algorithm 2 are studied when $scale = 1$, one is to compare the performances of our algorithm and the random-action-select method, the other is to show the relation between cost and the number of transferred objects. The experimental results are shown in Figure 7.1.

In Figure 7.1(a), the solid line shows the maximum benefit obtained by Algorithm 2 under limited cost and every dot shows the obtained benefit by randomly



(a) The performance of Algorithm 2.

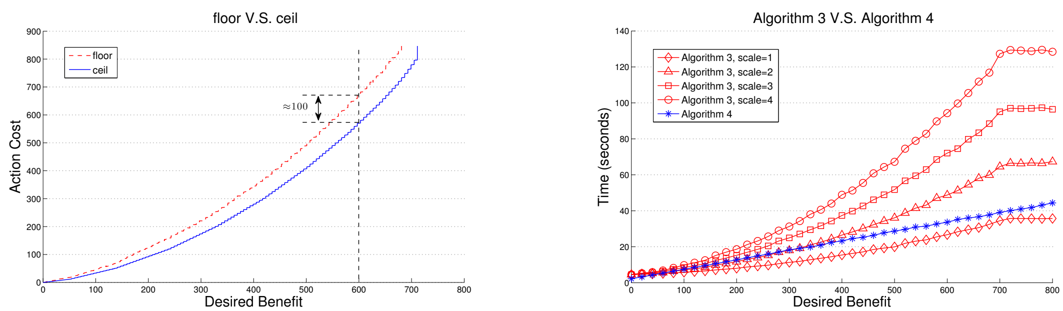
(b) Numbers of objects in regions.

Figure 7.1: Results of two experiments of Algorithm 2.

choosing actions under a limited cost. Obviously, our algorithm has consistently the same or better results. The algorithm reaches the maximum benefit 711 when cost is 589, while the random method requires almost twice the cost, about 1000 to obtain it. Figure 7.1(b) shows the numbers of objects in three regions when we have different limited costs. In this figure, the solid and dashed lines indicate the numbers of objects in $\mathrm{POS}(X')$ and $\mathrm{POS}(X') \cup \mathrm{BND}(X')$, respectively, and the region above the dashed line in the figure is $\mathrm{NEG}(X')$. By increasing the cost, two lines climb higher indicating that objects are sequentially transferred from the negative and boundary regions to the positive region. When the limited cost is about greater than 40, all objects in the boundary region will be transferred into the positive region. That is, the two lines are combined into one. When the cost reaches the upper bound, the three regions become stable and no more object will be transferred, because the rest objects in the negative region are non-actionable. In this scenario, the $\mathrm{POS}(X')$, $\mathrm{BND}(X')$, and $\mathrm{NEG}(X')$ regions have 282, 0, and 21 objects, respectively. Most objects in unfavorable regions are transferred into the favorable region.

7.1.2 Evaluations of Algorithm 3 and Algorithm 4

The third experiment shown in Figure 7.2(a) illustrates the difference between floor and ceil operators on Algorithm 3. In this experiment, we use $scale = 1.3527$ for all cost functions of flexible attributes. The number 1.3527 is randomly chosen, it is one of many numbers that may scale action cost to a non-integer value, such that the floor and ceil operators can produce different values. Inherently, to obain any desired benefit, the cost of the floor version of Algorithm 3 is higher than that of the ceil version and the difference between them gradually increases as the desired benefit increases. As shown in Figure 7.2(a), when the desired benefit is 600, the difference in cost between the two solutions is about 100. When $scale = 1$, the floor and ceil operators have the same result, which means that the two lines are identical.



(a) Algorithm 3 with floor and ceil.

(b) Computation times of Algorithm 3 and Algorithm 4.

Figure 7.2: Experiments on Algorithm 3 and Algorithm 4.

The fourth experiment compares the computation times between Algorithm 3 and Algorithm 4. The result is shown in Figure 7.2(b). In this figure, there are four lines representing Algorithm 3 with different scales for cost functions. The line with stars indicates the computation time of Algorithm 4. The computation time of Algorithm 4 only depends on the desired benefit, which in turn relies on the misclassification cost matrix. Its computation time is not affected by different scales. Therefore, we draw one line for Algorithm 4. When the desired benefits exceed the maximum value ($b_l = 711$), the computation times of Algorithm 3 stabilize. It is obvious that the computation times needed by Algorithm 3 are longer than the time needed by Algorithm 4 when $scale > 1$.

7.2 Effectiveness of R4 Reduction Framework

In this section, we evaluate the effectiveness of the R4 reduction framework by some experimental results. We use Algorithm 1, Algorithm 2, and Algorithm 4 to generate the results for four actionable models. These algorithms are reimplemented in C++ by Xcode 6.2 running on the same computer, i.e., Mac OS 10.9.5 with an Intel Core i5 2.4 GHz dual core CPU and 8 GB 1600 MHz DDR3 RAM. The implemented code uses only one core of the CPU for computation. For attribute reduction, we apply Equations (6.5) and (6.6) to Equation (6.3), and set $\lambda_\sigma = -1$, $\lambda_s = +\infty$, and $\lambda_\delta = -1$. For attribute-value pair reduction, we use Equation (6.10), and set $\lambda_s = +\infty$, $\lambda_\sigma = -1$, and $\lambda_\delta = -1$. The movement patterns are: $S_F = \text{POS}(X)$, $S_U = \text{NEG}(X) \cup \text{BND}(X)$, and $S_I = \emptyset$. We use the same configurations as the last section for experiments. That is, we use cost functions $C_f(v_1, v_2) = |v_1 - v_2|$ for each flexible attribute and the misclassification cost matrix shown in Table 7.1.

We use 9 data sets from the UCI Machine Learning Repository [17] for experiments in this section. Table 7.2 lists these data sets with their preprocessing. In Table 7.2, Shuttle, CMC, and TAE stand for the Statlog (Shuttle) data set (test set), Contraceptive Method Choice data set, and the Teaching Assistant Evaluation data set, respectively, and the Heart Disease data set is the same data set used in the last section. Column A_s denotes the indices (starting from 1) of stable attributes appearing in the corresponding preprocessed data set. We designate a subset A_s of the attributes as stable by our understanding, but they may be categorized differently by other people. Column X denotes the objective class, which consists of objects with the indicated decision attribute values according to the data descriptions. The Acute data set has two decision attributes and we consider that the objective class consists of objects having a *no* value for both decision attributes, which means healthy. Column preprocessing shows the detail preprocessing for these data sets.

Table 7.2: Data sets information.

Data set	Domain	$\|OB\|$	$\|A_s \cup A_f\|$	A_s	X	Preprocessing
Hayes-Roth	Social	160	4	$\{3\}$	$d = 1$	The first attribute *name* is not used.
Heart Disease	Life	303	13	$\{1, 2, 12\}$	$d = 0$	Preprocessed according to [22].
Breast Cancer	Life	699	9	$\{2, 3, 5\}$	$d = 2$	The first attribute, i.e., *Sample code number*, is not used. Missing attribute values are filled with most frequent value for that attribute.
Acute	Life	120	6	$\emptyset$	$d_1 = no$ and $d_2 = no$	All attribute values of *no* and *yes* are replaced as values 0 and 1, respectively. The first attribute temperature is grouped as follows: 0-36.5, 36.6-37.3, and 37.4+, and reassigned to values 1 to 3.
CMC	Life	1473	9	$\{1, 4, 5, 9\}$	$d = 1$	Attribute *age* is categorized into 5 groups: 0-19, 20-39, 40-59, 60-79, and 80+, and reassigned to values 1 to 5.
Haberman	Life	306	6	$\{1, 3\}$	$d = 1$	Attribute *age* and *operation age* are categorized into 5 groups: 0-19, 20-39, 40-59, 60-79, and 80+, and reassigned to values 1 to 5.
Shuttle	Physical	14500	9	$\emptyset$	$d = 1$	No preprocessing.
TAE	Education	151	5	$\{1, 2\}$	$d = 3$	Attribute *class size* is categorized into 4 groups: 0-15, 16-30, 31-45, and 46+, and reassigned to values 1 to 4.
Car	Business	1728	6	$\emptyset$	$d = vgood$	For each condition attribute, its non-numerical values are replaced with integers starting from 1 according to their order of appearance in the data set description. Attribute values 5*more* and *more* are replaced to value 5.

7.2.1 Evaluations of R4 on Model (I) and Model (II)

In the first experiment, we compare the results of model (i) and model (ii) before and after R4 reductions. The results are shown in Table 7.3, where $\bar{B}$ and $\bar{C}$ represent the upper bounds of benefit and cost, computed without the R4 reductions, and $\bar{B}'$ and $\bar{C}'$ are those computed after the R4 reductions. Columns AVPs, Rules, RRules, Actions, and RActions denote the average number of attribute-value pairs of rules induced from S_F, the number of rules in S_F after R4, the number of reduced rules, the number of actions after R4, and the number of reduced actions, respectively. Bold face is used to indicate cases where R4 improved.

Table 7.3: Comparison before and after reductions on model (i) and model (ii).

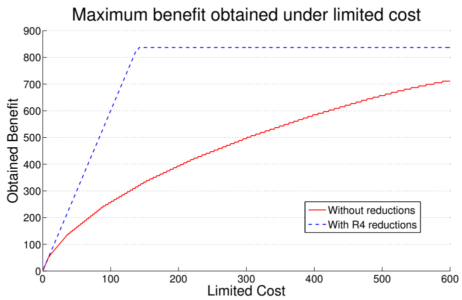
Data set	$\bar{B}$	$\bar{C}$	$\bar{B}'$	$\bar{C}'$	$\|R\|$	AVPs	Rules	RRules	Actions	RActions
Hayes-Roth	525	154	525	**137**	3	3	12	0	49	131
Heart Disease	711	589	**837**	**142**	11	4.87	97	43	135	9876
Breast Cancer	138	374	**1446**	**576**	4	2.22	51	56	238	11900
Acute	540	241	540	**109**	2	2	1	0	11	0
CMC	5414	1988	**5492**	**1178**	9	4.16	245	154	541	36548
Haberman	142.02	42	**178.13**	**49**	3	2.05	35	2	12	1
Shuttle	18132	280545	18132	**8152**	4	1.92	686	4096	3022	2070070
TAE	608	494	608	**165**	5	2.38	23	3	60	600
Car	9978	6168	9978	6168	6	5.38	35	30	1663	56542

As shown in Table 7.3, the R4 reduction framework can decrease the upper bound of cost or increase the upper bound of benefit effectively. Specifically, both the benefit and cost values for the Heart Disease, Breast Cancer, CMC, and Haberman data sets are improved. The upper bound of cost in the Haberman data set increases after the R4 reduction, but this higher cost also increases the upper bound of benefit. If we desire to obtain a benefit of 142.02, the required cost is decreased to 34, which is lower than the original cost of 42, and thus we highlight the number in the table as well. The upper bound of benefit or cost is not improved on the Car data set, because no attributes can be removed and most rules in S_F cannot be simplified. From this table, we can also gain a basic understanding of the effectiveness of the last two steps of R4: (1) in the rule reduction step, a remarkable number of rules can be removed, and (2) in the action reduction step, the majority of the actions can be removed. These reductions imply that the required storage can be decreased significantly.

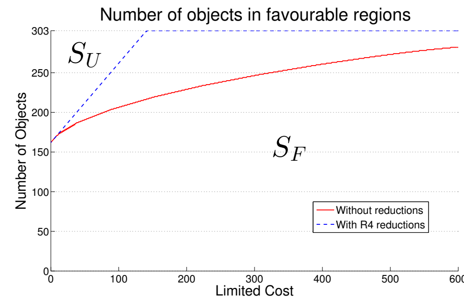
7.2.2 Evaluations of R4 on Model (III) and Model (IV)

The second experiment checked the improvements expected from Algorithm 6 for attribute reduction and attribute-value pair reduction in models (iii) and (iv). Since the experimental results on these data sets are similar, we show only the results of Heart Disease and CMC in Figure 7.3 and Figure 7.4, respectively.

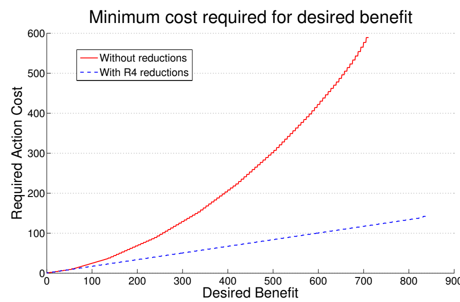
Figure 7.3(a) and Figure 7.4(a) show that the maximum benefit obtained after R4 is greater than or equal to that obtained without R4, and Figure 7.3(c) and Figure 7.4(c) show that for any desired benefit, the required minimum cost after R4 is less than or equal to that without R4. For example, the upper bound of the benefit without reduction in the Heart Disease data set is 711 and the cost is 589, while with R4, the upper bound of the benefit is increased to 837 and the cost is decreased to 142. Figure 7.3(b) and Figure 7.4(b) show that applying R4 reductions transfers more objects when there is a limit on cost. Additionally, because some stable attributes



(a) Results of model (iii).



(b) Numbers of transferred objects.

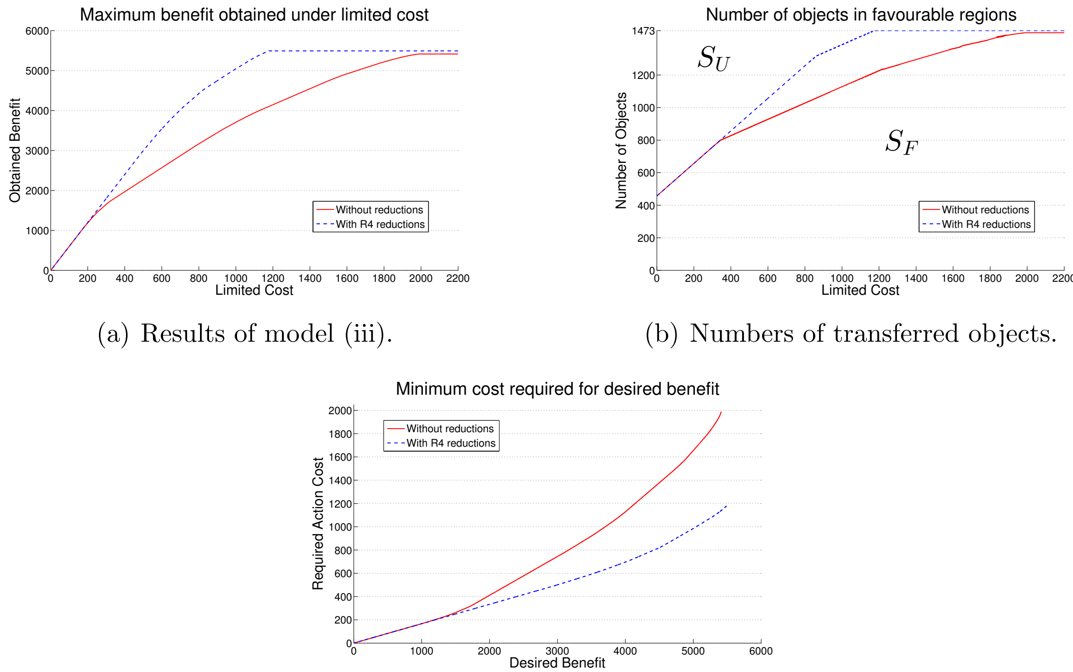


(c) Results of model (iv).

Figure 7.3: Experiments on actionable models (iii) and (iv) on the Heart Disease data set.

are removed by R4 reductions, all objects in unfavorable regions in these two data sets can be transferred to favorable regions when cost is sufficient, whereas without R4, some objects could not be transferred.

There is a general tendency for the number of transferred objects to increase as the cost is increased, but the tendency may not be monotonic. This differs from the tendency towards higher benefit, which is monotonic. This property is apparent in the solid line in Figure 7.3(b) where the value on x-axis fluctuates from 10 to 40. We can explain this phenomenon by a simple example. Suppose we have two actions: a_1 with a cost of 10 and a benefit of 20 and a_2 with a cost of 5 and a benefit of 10. Actions a_1 and a_2 can transfer 2 and 3 objects, respectively. When cost is limited to 8, only a_2 can be selected and it transfers 3 objects. When the limited cost is increased to 10, a_2 is selected instead of a_1 due to its higher benefit, but only 2 objects are transferred.



(a) Results of model (iii).

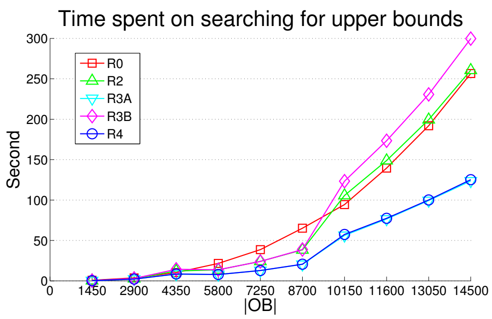
(b) Numbers of transferred objects.

(c) Results of model (iv).

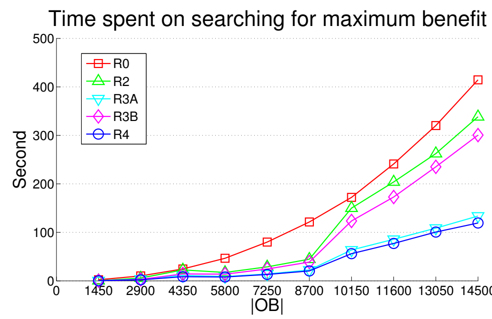
Figure 7.4: Experiments on actionable models (iii) and (iv) on the CMC data set.

7.2.3 Evaluations of Computation Time

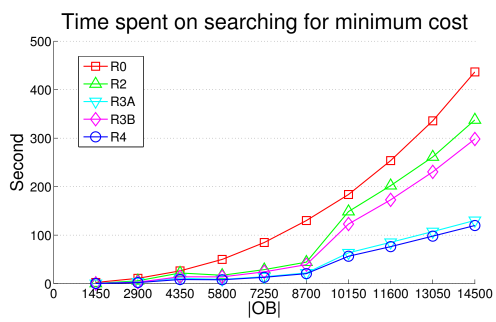
In the third experiment, we check the computation time on different sizes of data sets. On most data sets in Table 7.2, this experiment can be completed in a few seconds. Some smaller data sets can be processed in one second, but the larger Shuttle data set takes longer to search for a solution. Therefore, we artificially create differently sized data sets from the Shuttle data set for demonstration. The Shuttle data set is evenly split into 10 parts, each having 1450 objects. The first data set consists of the first 1450 objects, the second consists of the first 2900 objects, and so on. We experimented on these 10 artificial data sets for four actionable models. The results are shown in Figure 7.5, where R0 uses no reductions, R2 uses only attribute and attribute-value pair reductions, R3A uses attribute, attribute-value pair, and rule reductions (no action reduction), R3B uses attribute, attribute-value pair, and action reductions (no rule reduction), and R4 uses all four reductions. The Figure 7.5(b) displays the time spent on searching for the maximum benefit with a cost limit of 1000 and the Figure 7.5(c) displays the time spent on searching for the minimum cost for a desired benefit of 5000. As shown in Figures 7.5(b) and 7.5(c), the evaluations of computation time for model (iii) and that for model (iv) are similar.



(a) Results on model (i) and model (ii).



(b) Results on model (iii).



(c) Results on model (iv).

Figure 7.5: Time spent on four actionable models on different sizes of the Shuttle data set.

In Figure 7.5, we see that the R4 requires the least computation time for each actionable model. In the computation of model (i) and model (ii), R2 and R3B require more time than R0 when the size of data set is larger than about 10000. In the computation of model (iii) and model (iv), any experiment with reductions is faster than R0. An interesting finding is that the R3A only requires a little more computation time than R4, because rule reduction removes many classification rules, which significantly reduces the time required to construct actions. This experiment confirms that R4 effectively reduces computation time, especially for model (iii) and model (iv).

7.2.4 Comparisons with Existing Methods

The last experiment compared the proposed method and existing methods. Each of the attribute reduction and attribute-value pair reduction steps can follow one of three strategies, i.e., the Addition strategy, the Addition-deletion strategy, or the Deletion strategy. The algorithms based on the Addition-deletion and Deletion strategies are from [127], but we adapt them by computing the fitnesses of attributes prior to the while loop for consistency with Algorithm 6. Therefore, we have nine combinations for the first two steps of R4. We use A, Ad, and D to denote the Addition strategy, the Addition-deletion strategy, and the Deletion strategy, respectively. Thus, a combination AD means that the first step (attribute reduction) uses the Addition strategy and the second step (attribute-value pair reduction) uses the Deletion strategy. Other combinations follow the same pattern. We use the same fitness functions and experiment settings as in the above experiments for each of the three reduction algorithm strategies. The LEM2 [28] rule induction algorithm does not explicitly need a fitness function.

The results of all nine combinations and LEM2 on actionable models (i) and (ii) are shown in Table 7.4, where the first column of each data set denotes the upper bound of benefits, the second column denotes the upper bound of costs, and bold face indicates the best performances. In Table 7.4, the AA method, which uses both attribute and attribute-value pair reductions based on the Addition strategy, has the best outcomes, i.e., the upper bound of benefit obtained by AA is greater than or equal to that obtained by any other method. However, it should be noted that for most data sets, all methods yield the same benefits. Also, for any desired benefit, the corresponding cost required by AA is less than or equal to that required by any other method. A comparison between the first three rows shows that when attribute reduction is executed using the Addition strategy, the attribute-value pair reduction using the Addition strategy offers a best or equal results on both benefit and cost.

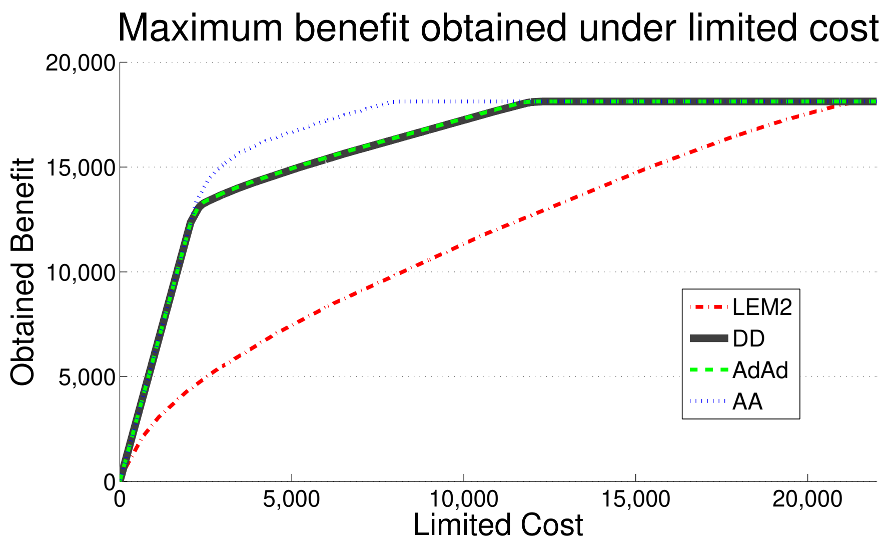
The same is true of the second group (rows 4, 5, and 6) and the third group (rows 7, 8, and 9) based on the same attribute reduct found by the Addition-deletion or the Deletion strategy. The Addition strategy for attribute-value pair reduction provides better or equal results than any other strategy on all data sets. Rows 1, 4, and 7 all use the Addition strategy for attribute-value pair reduction, but different strategies for attribute reduction, and AA offers the best result of cost on the Shuttle data set. Rows 1, 5, and 9 all use the same strategy for both attribute and attribute-value pair reductions, and the results also show that the Addition strategy has the best or equal outcomes. Additionally, the Addition-deletion strategy has better results than the Deletion strategy in all cases, and the Deletion strategy has the worst results in all nine combinations.

Table 7.4: Comparison between different methods on model (i) and model (ii).

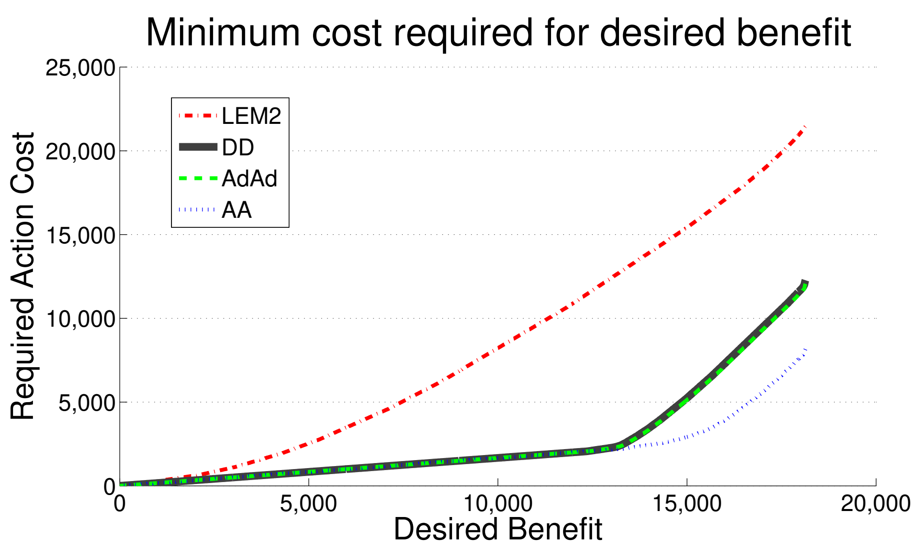
#	Method	Hayes-Roth	Heart Disease	Breast Cancer	Acute	CMC	Haberman	Shuttle	TAE	Car
1	AA	**525**, **137**	**837**, **142**	**1446**, **576**	**540**, **109**	**5492**, **1178**	**178.13**, 49	**18132**, **8152**	**608**, **165**	**9978**, **6168**
2	AAd	**525**, **137**	**837**, 154	**1446**, 742	**540**, **109**	**5492**, 1196	171.02, 48	**18132**, 9719	**608**, 179	**9978**, **6168**
3	AD	**525**, **137**	**837**, 178	**1446**, 868	**540**, **109**	**5492**, 1416	142.02, **42**	**18132**, 10970	**608**, 217	**9978**, **6168**
4	AdA	**525**, **137**	**837**, **142**	**1446**, **576**	**540**, **109**	**5492**, **1178**	**178.13**, 49	**18132**, 11055	**608**, **165**	**9978**, **6168**
5	AdAd	**525**, **137**	**837**, 154	**1446**, 742	**540**, **109**	**5492**, 1196	171.02, 48	**18132**, 12059	**608**, 179	**9978**, **6168**
6	AdD	**525**, **137**	**837**, 178	**1446**, 868	**540**, **109**	**5492**, 1416	142.02, **42**	**18132**, 12271	**608**, 217	**9978**, **6168**
7	DA	**525**, **137**	**837**, **142**	**1446**, **576**	**540**, **109**	**5492**, **1178**	**178.13**, 49	**18132**, 11055	**608**, **165**	**9978**, **6168**
8	DAd	**525**, **137**	**837**, 154	**1446**, 742	**540**, **109**	**5492**, 1196	171.02, 48	**18132**, 12059	**608**, 179	**9978**, **6168**
9	DD	**525**, **137**	**837**, 178	**1446**, 868	**540**, **109**	**5492**, 1416	142.02, **42**	**18132**, 12271	**608**, 217	**9978**, **6168**
10	LEM2	**525**, **137**	**837**, 162	**1446**, 815	**540**, 139	**5492**, 1318	141.77, 43	**18132**, 21442	**608**, 403	**9978**, **6168**

The LEM2 method aims to produce a set of simplified rules with maximal generality. Therefore, it is not a good choice for actionable three-way decision problems. LEM2 has two parts, rule induction and rule reduction. Rule induction produces simplified rules with an addition-deletion strategy and rule reduction removes less general rules with a deletion strategy. Thus, it is similar to the AdD method. However, LEM2 does not consider the cost factor, which excludes its use in cost-sensitive applications.

To check the effects of restricted costs and desired benefits in models (iii) and (iv), we show the experimental results of AA, AdAd, DD, and LEM2 on the Shuttle data set in Figure 7.6. The graphs clearly show that in cases where the strategies differ in effectiveness, the Addition strategy offers the greatest or equal benefit at any cost limit and requires the lowest or equal cost to obtain any desired benefit. The DD and AdAd have no significant difference; they are almost overlapped in the figure.



(a) Results on model (iii).



(b) Results on model (iv).

Figure 7.6: Comparison between methods for actionable models (iii) and (iv) on the Shuttle data set.

Chapter 8

CONCLUSION AND FUTURE WORKS

Three-way decisions play an important role in many real world decision-making problems. This chapter concludes the major contributions of this thesis and presents some future research topics.

8.1 Summary of Contributions

The trisecting-and-acting three-way decision model has two steps, i.e., trisecting and acting. This thesis extends the trisecting-and-acting three-way decision model by proposing the actionable three-way decision framework. In the trisecting step of the framework, we statistically interpret the three regions and provide a chi-square method for determining the optimal trisection. In the acting step of the framework, we propose four actionable models by adopting the concept of the actionable rule to transfer objects from unfavorable regions to favorable regions. Further, we propose the R4 reduction framework to reduce the action costs and increase benefits. In the following, we conclude these works.

8.1.1 An Actionable Three-way Decision Framework and Four Models

The existing studies on three-way decisions focus mainly on the trisecting step and relatively little on the acting step. We adopt actionable rules in the acting step to promote the movement of objects from unfavorable regions to favorable regions. Such movement leads to a new trisection and can improve the quality of the trisection.

We propose a general framework of actionable three-way decisions. Four actionable models are introduced through a cost-benefit analysis of actions, and four algorithms are designed to mine optimal actions for different models. The algorithms have a polynomial time complexity. The experimental results on the Heart Disease data set show that the algorithms have promising outcomes and objects can be effectively moved between regions.

8.1.2 Two Statistical Interpretations for Trisecting

Evaluation based three-way decisions use an evaluation function and a pair of thresholds to divide a universal set into three pair-wise disjoint regions. Statistical interpretations of three-way decisions construct three regions based on an understanding that the middle region M consists of normal or typical instances from a population, while regions L and R consist of, respectively, abnormal or atypical instances. Two special cases of statistical interpretation are given in this thesis. One is a set of non-numeric values and the other is a set of numeric values. These interpretations are widely used in many applications, such as boxplots, IQ score classifications, and blood pressure classifications.

The two statistical interpretations for trisecting use mean, standard deviation, median, and percentile. One may explore other types of statistical information, such as mode, average absolute deviation, median absolute deviation, and z-score to interpret three regions.

8.1.3 A Statistical Objective Function for Determining Optimal Trisection

The chi-square statistic is widely used in independence tests of a contingency table. In the context of trisecting, it measures the correlation between the actual classification $\{X, X^C\}$ and a three-way approximation $\pi_{(\alpha,\beta)}(X)$. Therefore, the chi-square statistic can be used as an objective function to quantify the goodness of a three-way approximation. According to the meaning of chi-square statistic, the greatest chi-square statistic suggests a high probability of correlation between $\{X, X^C\}$ and $\pi_{(\alpha,\beta)}(X)$. An optimal pair of (α, β) is determined by maximizing the statistic.

The maximally selected χ^2 statistic method for three-way decisions can be easily extended to other applications. For example, the area around the decision hyperplane in any classifier has higher impurity, i.e., it includes instances from different classes that are difficult to distinguish. The suggested method in this thesis provides an option to abstract a boundary region between two hyperplanes located on both sides of and parallel with the decision hyperplane for further analysis. These two hyperplanes are determined by a pair of distances from the decision hyperplane and can be found by maximizing χ^2 statistic.

8.1.4 The R4 Reductions Framework for Reducing Action Cost and Increasing Benefit

To reduce the cost and increase the benefit of actionable models, we redefine the concepts of reductions in the context of actionable three-way decisions and introduce the R4 reduction framework. The R4 framework consists of four steps to remove redundant attributes, attribute-value pairs, rules, and actions. The first two steps of R4 can reduce the action cost and increase the benefit of a solution, and the last two steps of R4 can reduce the computation time of a solution.

In R4, the Addition strategy for reduction is adapted and its correctness is proven. A schema of Addition strategy for reduct construction is provided for designing a variety of reduction algorithms. Based on this schema, an algorithm instance for attribute and attribute-value pair reduction is designed. The experimental results show that the R4 can not only decrease action cost but can also increase benefit in a reduced computation time. The proposed Algorithm 6 can produce the best or equal results in many data sets, when compared to the LEM2 algorithm and the Deletion and Addition-deletion strategy based algorithms.

8.2 Future Research Topics

In the future, we may study some problems of actionable three-way decisions in both the trisecting and acting aspects.

For trisecting, we will examine the process of constructing an evaluation function and study the methods for computing the pair of thresholds for trisection. We will also study trisection to distinguish between normal and abnormal, and between typical and atypical instances by using specific classes of distributions. The criteria to determine the pair of percentiles and the pair of thresholds for two interpretations can be studied based on concrete applications. Several additional topics may be discussed, such as analyzing thresholds in 2 × 2 contingency table by combining two columns together, developing a heuristic algorithm to find the thresholds, considering Fisher's exact test [20] if the magnitudes of some cells are less than 5, using likelihood ratio statistic or phi coefficient instead of chi-square statistic, and using log-linear model to determine the pair of thresholds.

For acting, we will research the following topics: (1) correlation between actions and between sub-actions; (2) adapting the R4 framework to multi-objective problems; (3) adapting the R4 framework to a sequential and dynamic scenario; (4) applying utility theory to the actionable models; (5) designing a parallel reduct construction algorithm for dealing with big data. For example, in real practice, we may have limited money, time, and personnel, and each must be accounted for in a solution; an action to be taken may be determined based on the results of the previous action. We may consider relaxing the assumptions used in this thesis and make use of sets UND and IDF.

Appendix A

PROOFS

A.1 Proof of Proposition 3.1

Without loss of generality, we only prove the case that $[x] \subseteq \text{NEG}(X)$ and $[y] \subseteq \text{POS}(X)$. Other cases can be proved in similar way.

After taking action $r_{[x]} \rightsquigarrow r_{[y]}$, the POS($X$) and NEG($X$) regions will be changed to POS(X') and NEG(X'), respectively. We have $\text{POS}(X') = \text{POS}(X) \cup [x]$ and $\text{NEG}(X') = \text{NEG}(X) - [x]$. BND($X$) region will not change, i.e., BND(X')=BND(X). We have $|X \cup [x]| = |X' \cup [x]|$, because, the different objects between X and X' are all in $[x]$. Therefore, we use b and a giving in Equation (3.9) to get $|X'|$:

$$\begin{aligned} |X'| &= b - a + |X| - |X \cup [x]| + |X' \cup [x]| \\ &= |X| - a + b, \end{aligned} \tag{A.1}$$

and we also have

$$\begin{aligned} |X'^{C} \cap [x]| &= |(OB - X') \cap [x]| \\ &= |(OB \cap [x]) - (X' \cap [x])| \\ &= |[x] - (X' \cap [x])| \end{aligned}$$

$$= |[x]| - |[x] \cap X' \cap [x]|$$
$$= |[x]| - b. \tag{A.2}$$

The new quality of positive region is:

$$Q(\text{POS}(X')) = |X' \cap \text{POS}(X')|\lambda_{PP} + |X'^C \cap \text{POS}(X')|\lambda_{PN}, \tag{A.3}$$

where the first term in the right hand side of equation (A.3) is:

$$\begin{aligned} |X' \cap \text{POS}(X')| &= |X' \cap (\text{POS}(X) \cup [x])| \\ &= |(X' \cap \text{POS}(X)) \cup (X' \cap [x])| \\ &= |X' \cap \text{POS}(X)| + |X' \cap [x]| - |(X' \cap \text{POS}(X)) \cap (X' \cap [x])| \\ &= |X \cap \text{POS}(X)| + b. \end{aligned} \tag{A.4}$$

We prove that the last equal sign in Equation (A.4) is satisfied, i.e., $|X' \cap \text{POS}(X)| = |X \cap \text{POS}(X)|$. Because $X' \subseteq X \cup [x]$ and $X \subseteq X' \cup [x]$. Therefore, we have

$$\begin{aligned} X' \cap \text{POS}(X) &\subseteq (X \cup [x]) \cap \text{POS}(X) \\ &= (X \cap \text{POS}(X)) \cup ([x] \cap \text{POS}(X)) \\ &= X \cap \text{POS}(X). \end{aligned} \tag{A.5}$$

Similarly,

$$\begin{aligned} X \cap \text{POS}(X) &\subseteq (X' \cup [x]) \cap \text{POS}(X) \\ &= (X' \cap \text{POS}(X)) \cup ([x] \cap \text{POS}(X)) \\ &= X' \cap \text{POS}(X). \end{aligned} \tag{A.6}$$

According to (A.5) and (A.6), we have $X' \cap \text{POS}(X) = X \cap \text{POS}(X)$, so that their cardinalities are the same. Therefore, (A.4) is satisfied.

The second term in the right hand side of Equation (A.3) is:

$$\begin{aligned} |X'^C \cap \mathrm{POS}(X')| &= |X'^C \cap (\mathrm{POS}(X) \cup [x])| \\ &= |(X'^C \cap \mathrm{POS}(X)) \cup (X'^C \cap [x])| \\ &= |X'^C \cap \mathrm{POS}(X)| + |X'^C \cap [x]| \\ &\quad - |(X'^C \cap \mathrm{POS}(X)) \cap (X'^C \cap [x])| \\ &= |X^C \cap \mathrm{POS}(X)| + |[x]| - b, \end{aligned} \tag{A.7}$$

where the last equal sign can be proved by the same principle of above.

Similarly, we can compute the quality of $\mathrm{NEG}(X')$:

$$Q(\mathrm{NEG}(X')) = |X' \cap \mathrm{NEG}(X')|\lambda_{NP} + |X'^C \cap \mathrm{NEG}(X')|\lambda_{NN}, \tag{A.8}$$

where the first term in the right hand side is:

$$\begin{aligned} |X' \cap \mathrm{NEG}(X')| &= |X' \cap (\mathrm{NEG}(X) - [x])| \\ &= |(X' \cap \mathrm{NEG}(X)) - (X' \cap [x])| \\ &= |X' \cap \mathrm{NEG}(X)| - |(X' \cap \mathrm{NEG}(X)) \cap (X' \cap [x])| \\ &= |X'| + |\mathrm{NEG}(X)| - |X' \cup \mathrm{NEG}(X)| - |X' \cap [x]| \\ &= |X| - a + b + |\mathrm{NEG}(X)| - |X' \cup \mathrm{NEG}(X)| - b \\ &= |X| + |\mathrm{NEG}(X)| - |X \cup \mathrm{NEG}(X)| - a \\ &= |X \cap \mathrm{NEG}(X)| - a, \end{aligned} \tag{A.9}$$

and where the second last equal sign is satisfied because $X' \cup \mathrm{NEG}(X) = X \cup \mathrm{NEG}(X)$, which can be proved by following:

$$\begin{aligned} X' \cup \text{NEG}(X) &\subseteq (X \cup [x]) \cup \text{NEG}(X) \\ &= X \cup \text{NEG}(X) \end{aligned} \tag{A.10}$$

and

$$\begin{aligned} X \cup \text{NEG}(X) &\subseteq (X' \cup [x]) \cup \text{NEG}(X) \\ &= X' \cup \text{NEG}(X). \end{aligned} \tag{A.11}$$

The second term of equation (A.8) is:

$$\begin{aligned} |X'^C \cap \text{NEG}(X')| &= |X'^C \cap (\text{NEG}(X) - [x])| \\ &= |(X'^C \cap \text{NEG}(X)) - (X'^C \cap [x])| \\ &= |(X'^C \cap \text{NEG}(X))| - |X'^C \cap \text{NEG}(X) \cap X'^C \cap [x]| \\ &= |(X'^C \cap \text{NEG}(X))| - |X'^C \cap [x]| \\ &= |X'^C| + |\text{NEG}(X)| - |X'^C \cup \text{NEG}(X)| - |(OB - X') \cap [x]| \\ &= |OB| - |X'| + |\text{NEG}(X)| - |X'^C \cup \text{NEG}(X)| \\ &\quad - |(OB \cap [x]) - (X' \cap [x])| \\ &= |OB| - (|X| - a + b) + |\text{NEG}(X)| - |X'^C \cup \text{NEG}(X)| \\ &\quad - |[x] - (X' \cap [x])| \\ &= (|OB| - |X|) + |\text{NEG}(X)| - |X'^C \cup \text{NEG}(X)| + a - b \\ &\quad - (|[x]| - |[x] \cap (X' \cap [x])|) \\ &= |X^C| + |\text{NEG}(X)| - |X^C \cup \text{NEG}(X)| + a - b \\ &\quad - (|[x]| - |X' \cap [x]|) + |X^C \cup \text{NEG}(X)| - |X'^C \cup \text{NEG}(X)| \\ &= |X^C \cap \text{NEG}(X)| + a - b - (|[x]| - b) + |X^C \cup \text{NEG}(X)| \end{aligned}$$

$$- |X'^C \cup \text{NEG}(X)|$$

$$= |X^C \cap \text{NEG}(X)| - |[x]| + a \tag{A.12}$$

The last equal sign is satisfied because $|X^C \cup \text{NEG}(X)| = |X'^C \cup \text{NEG}(X)|$ and we prove it as follows:

$$\begin{aligned} |X^C \cup \text{NEG}(X)| &= |(OB - X) \cup \text{NEG}(X)| \\ &= |(OB \cup \text{NEG}(X)) - (X \cup \text{NEG}(X))| \\ &= |OB| - |(OB \cup \text{NEG}(X)) \cap (X \cup \text{NEG}(X))| \\ &= |OB| - |X \cup \text{NEG}(X)| \end{aligned} \tag{A.13}$$

and

$$\begin{aligned} |X'^C \cup \text{NEG}(X)| &= |(OB - X') \cup \text{NEG}(X)| \\ &= |(OB \cup \text{NEG}(X)) - (X' \cup \text{NEG}(X))| \\ &= |OB| - |(OB \cup \text{NEG}(X)) \cap (X' \cup \text{NEG}(X))| \\ &= |OB| - |X' \cup \text{NEG}(X)| \\ &= |OB| - |X \cup \text{NEG}(X)|. \end{aligned} \tag{A.14}$$

Therefore, $|X^C \cup \text{NEG}(X)| = |X'^C \cup \text{NEG}(X)|$.

Based on Equation (A.4), (A.7), (A.9), and (A.12), the quality of trisection π' after taking action $r_{[x]} \rightsquigarrow r_{[y]}$ is:

$$\begin{aligned} Q(\pi') &= Q(\text{POS}(X')) + Q(\text{BND}(X')) + Q(\text{NEG}(X')) \\ &= w_P\big[(|X \cap \text{POS}(X)| + b)\lambda_{PP} + (|X^C \cap \text{POS}(X)| + |[x]| - b)\lambda_{PN}\big] + \\ &\quad w_B Q(\text{BND}(X)) + \\ &\quad w_N\big[(|X \cap \text{NEG}(X)| - a)\lambda_{NP} + (|X^C \cap \text{NEG}(X)| - |[x]| + a)\lambda_{NN}\big]. \end{aligned}$$

Therefore, the benefit of action $r_{[x]} \rightsquigarrow r_{[y]}$ is computed by:

$$\begin{aligned} B_{r_{[x]} \rightsquigarrow r_{[y]}} &= Q(\pi) - Q(\pi') \\ &= w_P\big[-b\lambda_{PP} - (|[x]| - b)\lambda_{PN}\big] + w_N\big[a\lambda_{NP} + (|[x]| - a)\lambda_{NN}\big]. \end{aligned}$$

□

A.2 Proof of Property P(5)

Because $M' = \{m - \text{USELESS} \mid m \in M\}$, i.e., $\forall m' \in M'$, $m' = m - \text{USELESS}, m \in M$, and thus $m' \subseteq m$.

$\text{RED}(M') \subseteq \text{RED}(M)$: Assume $R \in \text{RED}(M')$. By condition (s2) of Theorem 6.1, $R \cap m' \neq \emptyset$. Since $m' \subseteq m$, we get $R \cap m \neq \emptyset$. Thus, R satisfies condition (s2) of Theorem 6.1 for M. Consider an attribute $a \in R$. By the condition (n2) of Theorem 6.1 for M', we conclude that there exists at least one $m' \in M'$ such that $(R - \{a\}) \cap m' = \emptyset$. According to (P3), we know $a \notin \text{USELESS}$. Thus, $(R - \{a\}) \cap m = \emptyset$. That is, R satisfies condition (n2) of Theorem 6.1 for M. Therefore, $R \in \text{RED}(M)$.

$\text{RED}(M) \subseteq \text{RED}(M')$: Assume $R \in \text{RED}(M)$. By condition (s2) of Theorem 6.1, we have $R \cap m \neq \emptyset$ for all $m \in M$. By (P3), $R \cap \text{USELESS} = \emptyset$. We can conclude that $R \cap m' = R \cap (m - \text{USELESS}) \neq \emptyset$ for all $m' \in M'$. Thus, condition (s2) of Theorem 6.1 holds for R with respect to M'. Since R satisfies condition (n2) of Theorem 6.1 for M, by $m' \subseteq m$, thus, R satisfies condition (n2) for Theorem 6.1 for M'. This means that $R \in \text{RED}(M')$. □

A.3 Proof of Lemma 6.2

$\bigcup M^* \supseteq \text{USEFUL}$: Assume $a \in \text{USEFUL}$. By Definition 6.3, we know $a \in \bigcup \text{RED}(M)$. Since $\text{RED}(M^*) = \text{RED}(M)$, we obtain $a \in \bigcup \text{RED}(M^*) \subseteq \bigcup M^*$.

$\bigcup M^* \subseteq$ USEFUL: Assume $a \in \bigcup M^*$. Since M^* is an absorbed discernibility matrix, $\forall m \in (M^* - Group_{M^*}(a))$, $\forall g \in Group_{M^*}(a)$, $m \nsubseteq g$, i.e., $m - g \neq \emptyset$. By expressing $g = \{a\} \cup A$, we get $m - A \neq \emptyset$. By Lemma 6.1, we get $\text{RED}(M_1) \subseteq \text{RED}(M^*)$, where $M_1 = \{m - A \mid m \in M^*\}$. After removing A from M^*, we obtain M_1 and the original element $g = \{a\} \cup A$ becomes a singleton element $\{a\}$. By using $\{a\}$ to absorb elements of $Group_{M^*}(a)$ in M_1, all elements of $Group_{M^*}(a)$ can be absorbed by $\{a\}$. After this absorption, we denote the resulting discernibility matrix as M_2. Since absorption will not change any reduct, we have $\text{RED}(M_2) = \text{RED}(M_1)$.

Let M_2 be the union of two parts: $M_2 = P_1 \cup P_2$, in which each element in P_1 contains attribute a and each element in P_2 does not contains a. In P_1, there is only one element, $\{a\}$. In P_2, all elements have been shrunk by A, i.e., $P_2 = \{m - A \mid m \in (M^* - Group_{M^*}(a))\}$. Since $\forall m \in (M^* - Group_{M^*}(a))$ does not contain a and $g = \{a\} \cup A$, it follows $m - A = m - g$. Therefore, $P_2 = \{m - A \mid m \in (M^* - Group_{M^*}(a))\} = \{m - g \mid m \in (M^* - Group_{M^*}(a))\}$. Thus, $M_2 = P_1 \cup P_2 = \{\{a\}\} \cup P_2$. It follows that $\text{RED}(\{\{a\}\} \cup P_2) = \text{RED}(M_2) = \text{RED}(M_1) \subseteq \text{RED}(M^*) = \text{RED}(M)$. Because $\{a\}$ is a singleton element in $(\{\{a\}\} \cup P_2)$, then a is a core attribute in $(\{\{a\}\} \cup P_2)$, i.e., $a \in$ CORE. Since CORE $\subseteq$ USEFUL, a is useful, i.e., $a \in$ USEFUL.
□

A.4 Proof of Lemma 6.3

$\Leftarrow$: If $\exists g \in Group_M(a)$ such that g cannot be absorbed by any other elements in M, then $g \in M^*$. By Lemma 6.2, $a \in$ USEFUL.

$\Rightarrow$: Assume $a \in$ USEFUL. By Lemma 6.2, $a \in \bigcup M^*$. Therefore, $\exists g \in M^*$ and $g \in Group_{M^*}(a)$. Since M^* is already absorbed, g cannot be absorbed by any other elements in M. □

A.5 Proof of Theorem 6.2

The proof uses similar approach in the proof of Lemma 6.2. Because a is useful, according to Lemma 6.3, $\exists g \in Group_M(a)$ such that $\forall m \in M, g \not\subseteq m$, i.e., $m - g \neq \emptyset$. Let $g = \{a\} \cup A$, we have $\forall m \in M, m - A \neq \emptyset$. By Lemma 6.1, we have $\text{RED}(M_1) \subseteq \text{RED}(M)$, where $M_1 = \{m - A \mid m \in M\}$. After removing A from M, $g \in M$ becomes $\{a\}$ in M_1. Let M_2 denote the discernibility matrix obtained by using $\{a\}$ to absorb M_1. Therefore, $M_2 = \{\{a\}\} \cup \{m - A \mid m \in (M - Group_M(a))\}$. Because every element $m \in (M - Group_M(a))$ does not contain a, we have $m - A = m - g$. Thus, $\{m - A \mid m \in (M - Group_M(a))\} = \{m - g \mid m \in (M - Group_M(a))\} = M'$. Therefore, $M_2 = \{\{a\}\} \cup \{m - g \mid m \in (M - Group_M(a))\} = \{\{a\}\} \cup M'$. Because any absorption will not change the reducts, i.e., $\text{RED}(M_2) = \text{RED}(M_1)$, we have $\text{RED}(\{\{a\}\} \cup M') = \text{RED}(M_2) = \text{RED}(M_1) \subseteq \text{RED}(M)$. $\square$

REFERENCES

[1] N. Azam, J.T. Yao. Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets. International Journal of Approximate Reasoning, 55(1): 142-155, 2014.

[2] N. Azam, J.T. Yao. Game-theoretic rough sets for recommender systems. Knowledge-Based Systems, 72: 96-107, 2014.

[3] N. Azam, J.T. Yao. Multiple criteria decision analysis with game-theoretic rough sets. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 399-408, 2012.

[4] N. Azam, J.T. Yao. Variance based interpretation of three-way decisions using probabilistic rough sets. In Proceedings of the International Joint Conference on Rough Sets, pp. 209-218, 2016.

[5] N. Azam, Y. Zhang, J.T. Yao. Evaluation functions and decision conditions of three-way decisions with game-theoretic rough sets. European Journal of Operational Research, 261(2): 704-714, 2017.

[6] Y. Baram. Partial classification: the benefit of deferred decision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8): 769-776, 1998.

[7] A.L. Boulesteix. Maximally selected chi-square statistics and binary splits of nominal variables. Biometrical Journal, 48(5): 838-48, 2006.

[8] A.L. Boulesteix, C. Strobl. Maximally selected chi-square statistics and non-monotonic associations: an exact approach based on two cutpoints. Computational Statistics and Data Analysis, 51(12): 6295-6306, 2007.

[9] F. Cabitza, D. Ciucci, A. Locoro. Exploiting collective knowledge with three-way decision theory: cases from the questionnaire-based research. International Journal of Approximate Reasoning, 83: 356-370, 2017.

[10] Z.X. Chen, J. Tang, A.W.-C. Fu. Modeling and efficient mining of intentional knowledge of outliers. In Proceedings of the International Database Engineering and Applications Symposium, pp. 44-53, 2003.

[11] D. Ciucci. Orthopairs: a simple and widely used way to model uncertainty. Fundamenta Informaticae, 108(3-4): 287-304, 2011.

[12] V. Czitrom, P.D. Spagon. Statistical Case Studies for Industrial Process Improvement. SIAM, Philadelphia, 1997.

[13] S. DAniello, A. Gaeta, M. Gaeta, S. Tomasiello. Self-regulated learning with approximate reasoning and situation awareness. Journal of Ambient Intelligence and Humanized Computing, 9(1): 151-164, 2018.

[14] X.F. Deng. Three-Way Classification Models (Ph.D. Dissertation). University of Regina, Department of Computer Science, 2015.

[15] X.F. Deng, Y.Y. Yao. A multifaceted analysis of probabilistic three-way decisions. Fundamenta Informaticae, 132(3): 291-313, 2014.

[16] X.F. Deng, Y.Y. Yao. Decision-theoretic three-way approximations of fuzzy sets. Information Sciences, 279: 702-715, 2014.

[17] D. Dua, E. Karra Taniskidou. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, Irvine, School of Information and Computer Sciences, 2017.

[18] Y. Elovici, D. Braha. A decition-theoretic approach to data mining. IEEE Transaction on Systems, Man and Cybernetics, Part A, 33(1): 42-51, 2003.

[19] B.S. Everitt. The Analysis of Contingency Tables. Wiley, New York, 1977.

[20] R.A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of P. Journal of the Royal Statistical Society, 85(1): 87-94, 1922.

[21] H. Fujita, T.R. Li, Y.Y. Yao. Advances in three-way decisions and granular computing. Knowledge-Based Systems, 91: 1-3, 2016.

[22] C. Gao, Y.Y. Yao. Actionable strategies in three-way decisions. Knowledge-Based Systems, 133: 141-155, 2017.

[23] C. Gao, Y.Y. Yao. An addition strategy for reduct construction. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 535-546, 2014.

[24] C. Gao, Y.Y. Yao. Determining thresholds in three-way decisions with chi-square statistic. In Proceedings of the International Joint Conference on Rough Sets, pp. 272-281, 2016.

[25] J.H. Gennari, P. Langley, D. Fisher. Models of incremental concept formation. Artificial Intelligence, 40(1-3): 11-61, 1989.

[26] R. Goudey. Do statistical inferences allowing three alternative decision give better feedback for environmentally precautionary decision-making?. Journal of Environmental Management, 85(2): 338-344, 2007.

[27] S. Greco, B. Matarazzo, R. Slowinski. Rough membership and Bayesian confirmation measures for parameterized rough sets. In Proceedings of the International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, pp. 314-324, 2005.

[28] J. W. Grzymala-Busse. Rule induction. In: O. Maimon, L. Rokach (eds) Data Mining and Knowledge Discovery Handbook, pp. 249-265. Springer, Boston, 2010.

[29] J.W. Grzymala-Busse, P.G. Clarka, M. Kuehnhausena. Generalized probabilistic approximations of incomplete data. International Journal of Approximate Reasoning, 50(1): 180-196, 2014.

[30] H.J. Hamilton, N. Shan, N. Cercone. RIAC: a rule induction algorithm based on approximate classification. Technical Report CS 96-09, University of Regina, 1996.

[31] C. Hao, J.H Li, F. Min, W.Q. Liu, E.C.C. Tsang. Optimal scale selection in dynamic multi-scale decision tables based on sequential three-way decisions. Information Sciences, 415: 213-232, 2017.

[32] T. Hothorn, A. Zeileis. Generalized maximally selected statistics. Biometrics, 64(4): 1263-1269, 2008.

[33] B.Q. Hu. Three-way decisions space and three-way decisions. Information Sciences, 281: 21-52, 2014.

[34] B.Q. Hu, H. Wong, K.F.C. Yiu. On two novel types of three-way decisions in three-way decision spaces. International Journal of Approximate Reasoning, 82: 285-306, 2017.

[35] C.C. Huang, J.H. Li, C.L. Mei, W.Z. Wu. Three-way concept learning based on cognitive operators: An information fusion viewpoint. International Journal of Approximate Reasoning, 83: 218-242, 2017.

[36] K.V. Iserson, J.C. Moskop. Triage in medicine, part I: Concept, history, and types. Annals of Emergency Medicine, 49(3): 275-281, 2007.

[37] X.Y. Jia, W.H. Liao, Z.M. Tang, L. Shang. Minimum cost attribute reduction in decision-theoretic rough set models. Information Sciences, 219: 151-167, 2013.

[38] X.Y. Jia, L. Shang, X.Z. Zhou, J.Y. Liang, D.Q. Miao, G.Y. Wang, T.R. Li, Y.P. Zhang. Theory of Three-way Decisions and Applications (in Chinese). Nanjing University Press, Nanjing, 2012.

[39] X.Y. Jia, Z.M. Tang, W.H. Liao, L. Shang. On an optimization representation of decision-theoretic rough set model. International Journal of Approximate Reasoning, 55: 156-166, 2014.

[40] X.Y. Jia, K. Zheng, W.W. Li, T.T. Liu, L. Shang. Three-way decisions solution to filter spam email: an empirical study. In Proceedings of the International Conference on Rough Sets and Current Trends in Computing, pp. 287-296, 2012.

[41] J.-J. Jonker, N. Piersma, D.V.d. Poel. Joint optimization of customer segmentation and marketing policy to maximize long-term profitability. Expert System with Application, 27(2): 159-168, 2004.

[42] H.R. Ju, X.B. Yang, H.L. Yu, T.J. Li, D.-J. Yu, J.Y. Yang. Cost-sensitive rough set approach. Information Sciences, 355-356: 282-298, 2016.

[43] H. Kaur. Actionable rules: issues and new directions. In Proceedings of the World Enformatika Conference, pp. 61-64, 2005.

[44] E.M. Knorr, R.T. Ng. Finding intensional knowledge of distance-based outliers. In Proceedings of the International Conference on Very Large Data Bases, pp. 211-222, 1999.

[45] H.X. Li, L.B. Zhang, B. Huang, X.Z. Zhou. Sequential three-way decision and granulation for cost-sensitive face recognition. Knowledge-Based Systems, 91: 241-251, 2016.

[46] H.X. Li, L.B. Zhang, X.Z. Zhou, B. Huang. Cost-sensitive sequential three-way decision modeling using a deep neural network. International Journal of Approximate Reasoning, 85: 68-78, 2017.

[47] H.X. Li, X.Z. Zhou, B. Huang, D. Liu. Cost-sensitive three-way decision: a sequential strategy. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 325-337, 2013.

[48] J.H. Li, C.C. Huang, J.J. Qi, Y.H. Qian, W.Q. Liu. Three-way cognitive concept learning via multi-granularity. Information Sciences, 378: 244-263, 2017.

[49] D.C. Liang, D. Liu. Deriving three-way decisions from intuitionistic fuzzy decision-theoretic rough sets. Information Sciences, 300: 28-48, 2015.

[50] D.C. Liang, W. Pedrycz, D. Liu, P. Hu. Three-way decisions based on decision-theoretic rough sets under linguistic assessment with the aid of group decision making. Applied Soft Computing, 29: 256-269, 2015.

[51] B. Liu, W. Hsu, Y. Ma. Identifying non-actionable association rules. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 329-334, 2001.

[52] D. Liu, D.C. Liang, C.C. Wang. A novel three-way decision model based on incomplete information system. Knowledge-Based Systems, 91: 32-45, 2016.

[53] D. Liu, T.R. Li, D.C. Liang. Incorporating logistic regression to decision-theoretic rough sets for classifications. International Journal of Approximate Reasoning, 55: 197-210, 2014.

[54] D. Liu, T.R. Li, D.C. Liang. Three-way government decision analysis with decision-theoretic rough sets. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 20: 119-132, 2012.

[55] D. Liu, T.R. Li, D.Q. Miao, G.Y. Wang, J.Y. Liang. Three-way Decisions and Granular Computing (in Chinese). Science Press, Beijing, 2013.

[56] D. Liu, D.C. Liang, C.C. Wang. A novel three-way decision model based on incomplete information system. Knowledge-Based Systems, 91: 32-45, 2016.

[57] X.A. Ma, G.Y. Wang, H. Yu. Decision region distribution preservation reduction in decision-theoretic rough set model. Information Sciences, 278: 614-640, 2014.

[58] X.A. Ma, G.Y. Wang, H. Yu. Monotonic uncertainty measures for attribute reduction in probabilistic rough set model. International Journal of Approximate Reasoning, 59: 41-67, 2015.

[59] D.Q. Miao, Y. Zhao, Y.Y. Yao, H.X. Li, F.F. Xu. Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. Information Sciences, 179: 4140-4150, 2009.

[60] R. Miller, D. Siegmund. Maximally selected chi square statistics. Biometrics 38(4): 1011-1016, 1982.

[61] F. Min, H.P. He, Y.H. Qian, W. Zhu. Test-cost-sensitive attribute reduction. Information Sciences, 181: 4928-4942, 2011.

[62] N. Mishra, D. Ron, R. Swaminathan. A new conceptual clustering framework. Machine Learning, 56: 115-151, 2004.

[63] C. Pater. The blood pressure "uncertainty range" - a pragmatic approach to overcome current diagnostic uncertainties (II). Current Controlled Trials in Cardiovascular Medicine, 6(1): 5, 2005.

[64] Z. Pawlak. Rough classification. International Journal of Man-Machine Studies, 20(5): 469-483.

[65] Z. Pawlak. Rough sets. International Journal of Computer Information and Science, 11(5): 341-356, 1982.

[66] Z. Pawlak. Rough Sets: Theoretical Aspects Of Reasoning About Data. Kluwer Academic Publishers, Norwell, 1992.

[67] Z. Pawlak, S.K.M. Wong, W. Ziarko. Rough sets: probabilistic versus deterministic approach. Information Science, 29: 81-95, 1988.

[68] W. Pedrycz. Shadowed sets: representing and processing fuzzy sets. IEEE Transactions on System, Man and Cybernetics, 28: 103-109, 1998.

[69] W. Pedrycz, A. Skowron, V. Kreinovich. Handbook of Granular Computing. John Wiley & Sons Ltd, Chichester, 2008.

[70] J.F. Peters, S. Ramanna. Proximal three-way decisions: theory and applications in social networks. Knowledge-Based Systems, 91: 4-15, 2016.

[71] D. Pisinger. Algorithms for Knapsack Problems (Ph.D. thesis). University of Copenhagen, Department of Computer Science, 1995.

[72] J.J. Qi, T. Qian, L. Wei. The connections between three-way and classical concept lattices. Knowledge-Based Systems, 91: 143-151, 2016.

[73] J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, 1993.

[74] Z.W. Ras, L.S. Tsay. Discovering extended action-rules (system DEAR). In Proceedings of the Intelligent Information Processing and Web Mining, pp. 293-300, 2003.

[75] Z.W. Ras, A. Wieczorkowska. Action rules: how to increase profit of a company. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, pp. 587-592, 2000.

[76] R.S. Ren, L. Wei. The attribute reductions of three-way concept lattices. Knowledge-Based Systems, 99: 92-102, 2016.

[77] R.S. Ren, L. Wei, Y.Y. Yao. An analysis of three types of partially-known formal concepts. International Journal of Machine Learning and Cybernetics, 1-17, 2017. DOI: 10.1007/s13042-017-0743-z

[78] P.J. Rousseeuw, I. Ruts, J. W. Tukey. The bagplot: a bivariate boxplot. The American Statistician, 53: 382-387, 1999.

[79] D.H. Sanders, R.K. Smidt, A. Adatia, G.A. Larson. Statistics: A First Course. McGraw-Hill Ryerson, Toronto, 2001.

[80] Y.L. Sang, J.Y. Liang, Y.H. Qian. Decision-theoretic rough sets under dynamic granulation. Knowledge-Based Systems, 91: 84-92, 2016.

[81] J.M. Sattler. Assessment of Children's Intelligence. W.B. Saunders Company, Philadelphia, 1975.

[82] C.B. Schechter. Sequential analysis in a Bayesian model of diastolic blood pressure measurement. Medical Decision Making, 8(3): 191-196, 1988.

[83] H. Schofield. Assessment and Testing: An Introduction. Allen & Unwin, London, 1972.

[84] A. Shakiba, M.R. Hooshmandasl. S-approximation spaces: A three-way decision approach. Fundamenta Informaticae, 139(3): 307-328, 2015.

[85] N. Shan, W. Ziarko. Data-based acquisition and incremental modification of classification rules. Computational Intelligence, 11: 357-370, 1995.

[86] Y.H. She. On determination of thresholds in three-way approximation of many-valued NM-Logic. In Proceedings of the International Conference on Rough Sets and Current Trends in Computing, pp. 136-143, 2014.

[87] Y.-D. Shen, Z. Zhang, Q. Yang. Objective-oriented utility-based association mining. In Proceedings of the IEEE International Conference on Data Mining, pp. 426-433, 2002.

[88] R. Shivhare, A.K. Cherukuri. Three-way conceptual approach for cognitive memory functionalities. International Journal of Machine Learning and Cybernetics, 8(1): 21-34, 2016.

[89] A. Silberschatz, A. Tuzihilin. On subjective measures of interestingness in knowledge discovery. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 275-281, 1995.

[90] A. Silberschatz, A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. IEEE Transactions on Knowledge and Data Engineering, 8(6): 970-974, 1996.

[91] P.K. Singh. Three-way fuzzy concept lattice representation using neutrosophic set. International Journal of Machine Learning and Cybernetics, 8(1): 69-79, 2017.

[92] A. Skowron, C. Rauszer. The discernibility matrices and functions in information systems. Intelligent Decision Support, 11: 331-362, 1992.

[93] D. Slezak, W. Ziarko. Bayesian rough set model. In Proceedings of the International Workshop on Foundation of Data Mining, pp. 131-135, 2002.

[94] P. Su, W.J. Mao, D. Zeng, H.M. Zhao. Mining actionable behavioral rules. Decision Support Systems, 54(1): 142-152, 2012.

[95] P. Verdecchia, C. Porcellati, G. Schillaci, C. Borgioni, A. Ciucci, M. Battistelli, M. Guerrieri, C. Gatteschi, I. Zampi, A. Santucci, C. Santucci, G. Reboldi. Ambulatory blood pressure. An independent predictor of prognosis in essential hypertension.. Hypertension, 24(6): 793-801, 1994.

[96] A. Wald. Sequential Analysis. Dover Publications Inc., New York, 1947.

[97] J. Wang, J. Wang. Reduction algorithms based on discernibility matrix: the ordered attributes method. Journal of Computer Science and Technology, 16(6): 489-504, 2001.

[98] K. Wang, S.Q. Zhou, J.W. Han. Profit mining: from patterns to actions. In Proceedings of the International Conference on Extending Database Technology, pp. 70-87, 2002.

[99] L. Wei, H.R. Li, W.X. Zhang. Knowledge reduction based on the equivalence relations defined on attribute set and its power set. Information Sciences, 177: 3178-3185, 2007.

[100] S.K.M. Wong, W. Ziarko. A probabilistic model of approximate classification and decision rules with uncertainty in inductive learning. Technical Report CS 85-23, University of Regina, 1985.

[101] Q. Yang, H. Cheng. Mining case bases for action recommendation. In Proceedings of the IEEE International Conference on Data Mining, pp. 522-529, 2002.

[102] Q. Yang, J. Yin, C. Ling, R. Pan. Extracting actionable knowledge from decision trees. IEEE Transactions on Knowledge and Data Engineering, 19(1): 43-56, 2007.

[103] X. Yang, T.R. Li, H. Fujita, L. Liu, Y.Y. Yao. A unified model of sequential three-way decisions and multilevel incremental processing. Knowledge-Based Systems, 134: 172-188, 2017.

[104] X. Yang, T.R. Li, D. Liu, H.M. Chen, C. Luo. A unified framework of dynamic three-way probabilistic rough sets. Information Sciences, 420: 126-147, 2017.

[105] J.T. Yao, N. Azam. Web-based medical decision support systems for three-way medical decision making with game-theoretic rough sets. IEEE Transactions on Fuzzy Systems, 23: 3-15, 2015.

[106] J.T. Yao, J.P. Herbert. A game-theoretic perspective on rough set analysis. Journal of Chongqing University of Posts and Telecommunications, 20: 291-298, 2008.

[107] Y.Y. Yao. An outline of a theory of three-way decisions. In Proceedings of the International Conference on Rough Sets and Current Trends in Computing, pp. 1-17, 2012.

[108] Y.Y. Yao. Decision-theoretic rough set models. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 1-12, 2007.

[109] Y.Y. Yao. Granular computing and sequential three-way decisions. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 16-27, 2013.

[110] Y.Y. Yao. Interval sets and three-way concept analysis in incomplete contexts. International Journal of Machine Learning and Cybernetics, 8(1): 3-20, 2017.

[111] Y.Y. Yao. Interval-set algebra for qualitative knowledge representation. In Proceedings of the International Conference on Computing and Information, pp. 370-374, 1993.

[112] Y.Y. Yao. Perspectives of granular computing. In Proceedings of the IEEE International Conference on Granular Computing, pp. 85-90, 2005.

[113] Y.Y. Yao. Probabilistic rough set approximations. International Journal of Approximate Reasoning, 49(2): 255-271, 2008.

[114] Y.Y. Yao. Rough sets and three-way decisions. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 62-73, 2015.

[115] Y.Y. Yao. The superiority of three-way decisions in probabilistic rough set models. Information Sciences, 181: 1080-1096, 2011.

[116] Y.Y. Yao. Three-way decision: an interpretation of rules in rough set theory. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 642-649, 2009.

[117] Y.Y. Yao. Three-way decisions and cognitive computing. Cognitive Computation, 8(4): 543-554, 2016.

[118] Y.Y. Yao, R. Fu. The concept of reducts in Pawlak three-step rough set analysis. Transactions on Rough Sets 16: 53-72, 2013.

[119] Y.Y. Yao, C. Gao. Statistical interpretations of three-way decisions. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 309-320, 2015.

[120] Y.Y. Yao, S. Wang, X.F. Deng. Constructing shadowed sets and three-way approximations of fuzzy sets. Information Sciences, 412-413: 132-153, 2017.

[121] Y.Y. Yao, S.K.M. Wong. A decision theoretic framework for approximating concepts. International Journal of Man-Machine Studies, 37(6): 793-809, 1992.

[122] Y.Y. Yao, S.K.M. Wong, P. Lingras. A decision-theoretic rough set model. In Proceedings of the International Symposium on Methodologies for Intelligent Systems, pp. 17-25, 1990.

[123] Y.Y. Yao, H. Yu. An introduction of three-way decisions. In: H. Yu, G.Y. Wang, T.R. Li, J.Y. Liang, D.Q. Miao, Y.Y. Yao (eds) Three-way Decisions: Methods and Practices for Complex Problem Solving (in Chinese), pp. 1-19. Science Press, Beijing, 2015.

[124] Y.Y. Yao, X.Y. Zhang. Class-specific attribute reducts in rough set theory. Information Sciences, 418-419: 601-618, 2017.

[125] Y.Y. Yao, Y. Zhao. Attribute reduction in decision-theoretic rough set models. Information Sciences, 178: 3356-3373, 2008.

[126] Y.Y. Yao, Y. Zhao. Discernibility matrix simplification for constructing attribute reducts. Information Sciences, 179: 867-882, 2009.

[127] Y.Y. Yao, Y. Zhao, J. Wang. On reduct construction algorithms. Transactions on Computational Science, 2: 100-117, 2008.

[128] Y.Y. Yao, B. Zhou. Naive Bayesian rough sets. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 719-726, 2010.

[129] C.D. Yenigun, G.J. Szekely, M.L. Rizzo. A test of independence in two-way contingency tables based on maximal correlation. Communication in Statistics - Theory and Methods, 40(12): 2225-2242, 2011.

[130] H. Yu, P. Jiao, Y.Y Yao, G.Y Wang. Detecting and refining overlapping regions in complex networks with three-way decisions. Information Sciences, 373: 21-41, 2016.

[131] H. Yu, Z.G. Liu, G.Y. Wang. An automatic method to determine the number of clusters using decision-theoretic rough set. International Journal of Approximate Reasoning, 55(1): 101-115, 2014.

[132] H. Yu, T. Su, X.H. Zeng. A three-way decisions clustering algorithm for incomplete data. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 765-776, 2014.

[133] H. Yu, G.Y. Wang, T.R. Li, J.Y. Liang, D.Q. Miao, Y.Y. Yao. Three-Way Decisions: Methods and Practices for Complex Problem Solving (in Chinese), Science Press, Beijing, 2015.

[134] H. Yu, C. Zhang, G.Y. Wang. A tree-based incremental overlapping clustering method using the three-way decision theory. Knowledge-Based Systems, 91: 189-203, 2016.

[135] H. Zhang, B. Padmanabhan, A. Tuzhilin. On the discovery of significant statistical quantitative rules. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 374-383, 2004.

[136] H.R. Zhang, F. Min. Three-way recommender systems based on random forests. Knowledge-Based Systems, 91: 275-286, 2016.

[137] H.R. Zhang, F. Min, B. Shi. Regression-based three-way recommendation. Information Sciences, 378: 444-461, 2017.

[138] H.Y. Zhang, S.Y. Yang, J.M. Ma. Ranking interval sets based on inclusion measures and applications to three-way decisions. Knowledge-Based Systems, 91: 62-70, 2016.

[139] Q.H. Zhang, C.X. Lv, Y.H. Chen, G.Y. Wang. A dynamic three-way decision model based on the updating of attribute values. Knowledge-Based Systems, 142: 71-84, 2018.

[140] Y. Zhang. Optimizing Gini coefficient of probabilistic rough set regions using game-theoretic rough sets. In Proceedings of the Canadian Conference of Electrical And Computer Engineering, pp. 1-4, 2013.

[141] Y. Zhang, J.T. Yao. Gini objective functions for three-way classifications. International Journal of Approximate Reasoning, 81: 103-114, 2017.

[142] K. Zhao, J. Wang. A reduction algorithm meeting users' requirements. Journal of Computer Science and Technology, 17: 578-593, 2002.

[143] B. Zhou. Multi-class decision-theoretic rough sets. International Journal of Approximate Reasoning, 55(1): 211-224, 2014.

[144] B. Zhou, Y.Y. Yao, J.G. Luo. Cost-sensitive three-way email spam filtering. Journal of Intelligent Information Systems, 42(1): 19-45, 2014.

[145] W. Ziarko. Variable precision rough set model. Journal of Computer and System Sciences, 46(1): 39-59, 1993.

[146] W. Ziarko, N. Shan. A method for computing all maximally general rules in attribute-value systems. Computational Intelligence, 12(2): 223-234, 1996.