LINEAR APPROXIMATIONS AND EXTENSIONS TO DISTANCE BASED MULTICRITERIA SORTING METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HASAN TAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

AUGUST 2022

Approval of the thesis:

LINEAR APPROXIMATIONS AND EXTENSIONS TO DISTANCE BASED MULTICRITERIA SORTING METHODS

submitted by **HASAN TAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** __________________

Prof. Dr. Esra Karasakal
Head of the Department, **Industrial Engineering** __________________

Prof. Dr. Esra Karasakal
Supervisor, **Industrial Engineering, METU** __________________

Examining Committee Members:

Prof. Dr. Cem İyigün
Industrial Engineering, METU __________________

Prof. Dr. Esra Karasakal
Industrial Engineering, METU __________________

Prof. Dr. Serhan Duran
Industrial Engineering, METU __________________

Assist. Pr. Dr. Ozgen Karaer
Industrial Engineering, METU __________________

Assist. Pr. Dr. Erdi Daşdemir
Industrial Engineering, Hacettepe University __________________

Date: 26.08.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Hasan Taş

Signature :

ABSTRACT

LINEAR APPROXIMATIONS AND EXTENSIONS TO DISTANCE BASED MULTICRITERIA SORTING METHODS

Taş, Hasan
Master of Science, Industrial Engineering
Supervisor : Prof. Dr. Esra Karasakal

August 2022, 159 pages

Multicriteria sorting is the assignment of alternatives to predefined preference ordered classes. In this thesis, linear approximations to nearest centroid and distance-based multicriteria sorting methods are studied. Three studies are conducted. The first study is the linearization of a nearest centroid based method. In the second study, the nearest centroid classifier method is investigated under monotonic centroids and a new linear programming model is developed based on the feasibility and redundancy conditions. In the third study, a new linear octagonal approximation for nonlinear oval contours of distance functions is developed and analyzed. It is shown that the new approximation is consistent with distance functions. Due to the elimination of nonlinearities in mathematical programs, solution time significantly decreases. It is also observed that the classification accuracy increased in the studied models.

Keywords: Multicriteria sorting, distance functions, distance based sorting, nearest centroid classifier, linear approximation

ÖZ

MESAFE FONKSİYONU BAZLI SIRALI SINIFLANDIRMA PROBLEMLERİ İÇİN DOĞRUSAL YAKLAŞIMLAR VE İLAVE YÖNTEMLER GELİŞTİRİLMESİ

Taş, Hasan
Yüksek Lisans, Endüstri Mühendisliği
Tez Yöneticisi: Prof. Dr. Esra Karasakal

Ağustos 2022, 159 sayfa

Çok kriterli sıralı sınıflandırma problemi alternatiflerin önceden tanımlanmış tercihe göre sıralı sınıflara atanmasıdır. Bu tezde en yakın merkez ve mesafe fonksiyonu bazlı çok kriterli sıralı sınıflandırma problemlerine doğrusal yaklaşımlar geliştirilmiştir. Bu bağlamda üç çalışma yapılmıştır. Birinci çalışmada en yakın merkez bazlı sınıflandırma problemine bir doğrusal yaklaşım geliştirilmiştir. İkinci çalışmada ilk çalışmanın monoton merkezli versiyonu incelenmiş ve olurluluk koşulları baz alınarak yeni bir doğrusal programlama modeli geliştirilmiştir. Üçüncü çalışmada doğrusal olmayan mesafe fonksiyonları sekizgen bir çerçeve ile yakınsanmış ve doğrusal bir mesafe fonksiyonu yaklaşımı geliştirilmiştir. Bu yaklaşım detaylı olarak incelenip mesafe fonksiyonları ile tutarlı olduğu gösterilmiştir. Matematiksel modellerde doğrusal olmayan formüllerin doğrusal yaklaşımları sayesinde çözüm süresinde önemli ölçüde iyileşmeler sağlanmıştır. Ayrıca çalışılan modellerde sınıflandırma kesinliğinin arttığı da gözlemlenmiştir.

Anahtar Kelimeler: Çok kriterli sıralama, mesafe fonksiyonları, mesafe bazlı sıralama, en yakın Merkez bazlı sınıflandırma, doğrusal yaklaşım

To my family and friends…

ACKNOWLEDGMENTS

I am grateful to my mother Nuray Taş, my father İbrahim Taş and my siblings Merve and Mehmet for their endless love and support. I would like to thank my supervisor Prof. Dr. Esra Karasakal for her support, patience and valuable guidance. Her encouraging speeches and guidance made me realize my potential during this study and established my visions and goals for my academic career. I will always remember the valuable lessons I have learned from her for my future life and career. It is a great pleasure to work with and learn from her.

I also offer my blessings to my fiancée Buse Ünlütürk and my close friends Dursen Deniz Poyraz and Dilay Özkan. I would like to thank my friends Melis Boran and Furkan Baysal for their supports.

I am also thankful to the professors of my department for the valuable information I have learned from them and in their lectures. All of the engineering courses in my department shaped my background and gave their fruits to my research.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

When there is a discrete set of alternatives that are evaluated under multiple criteria, there are three main problems in Multi-Criteria Decision Aid (MCDA) [1]. Those problems are the choice problem, ranking problem, and sorting problem.

1. Choice problem: a single best or a group of best alternatives is chosen.
2. Ranking problem: alternatives are ranked from best to worst according to a preference order.
3. Sorting problem: alternatives are assigned to predefined preference ordered classes.

The solution to each of these problems may require preference information from the decision maker. The preference information can be in the form of criterion weights, reference profiles, reference/preference alternatives (a previously ranked or sorted data), and/or method-specific parameters (e.g., preference, indifference, and veto thresholds). According to the timing of obtaining the preference information, MCDA methods can be categorized into three as follows [2].

1. A priori methods: The first category is "before" methods called a priori. Preference information is obtained before model construction and solution approach.
2. Interactive methods: The second category is "during" methods called interactive. Preference information is obtained in different phases of the solution approach. According to the information obtained in every step, the solution is updated and converged to a final state.
3. A posteriori methods: The third category is "after" methods. First, a method is applied, then alternative solutions of that method are evaluated according to the preference information obtained from the decision maker.

In this thesis, distance-based multicriteria sorting methods are studied where the preference information is obtained via a reference set. The reference set is available at the beginning of the solution process. Therefore, the proposed method is a priori method.

Multicriteria sorting is the assignment of alternatives to the predefined ordinal classes. The alternatives are compared to class representatives. The class representatives can be in the form of class thresholds, limiting profiles, central profiles, and centroids. Those class representatives are ordered as classes. An alternative is evaluated concerning multiple criteria that are maximization or minimization type. Evaluated alternatives are compared to class representatives. Evaluation of alternatives and comparison can be based on utility function [3], preference relations such as outranking degree [4] and value functions [5]. Since the class representatives are also ordered with respect to (w.r.t) the class order and the alternatives are compared to these class representatives, preference order of alternatives is implicitly or explicitly applied in sorting. The class assignment can be performed based on deterministic measures [3] or probabilistic measures [7].

The classification and sorting problems are different [5]. Classification methods are descriptive approaches that are utilized for detecting and characterizing the similarities within a set of data. Sorting is a prescriptive approach to aid the Decision Maker (DM) to make wise decisions. In multicriteria sorting, the preference of the DM is associated with example decisions or preference information while this feature is not employed in classification. The other difference of classification and sorting is in the definition of classes and criteria. In sorting, classes are in ordinal scale, ordered from best to worst (or vice versa) according to the preference of the DM, which is called preference order. In classification, classes are nominal.

In both classification and sorting, higher classification accuracy and shorter solution (or training) time is desirable. The classification accuracy is the percentage of alternatives that are assigned to their correct classes. The solution time is the time that is required to elicitate or learn the preferences of DM.

The aim of this study is to improve the existing distance-based sorting methods in terms of solution time and classification accuracy. In the problem setting, DM provides the preference information as example class assignments of the alternatives or historical data that the class assignments are performed in the past. The analysts apply a sorting method to this data to elicitate the preferences of the DM with highest classification accuracy. In general, this elicitation is performed with mathematical programs that maximize the classification accuracy (or minimize error). In distance-based sorting methods, the mathematical programs include distance functions in their constraints. The distance functions are nonlinear formulations in general. Therefore, they are nonlinear programming models that are computationally expensive to solve. Due to variability of distance functions, it is not clear to use which distance function. One other issue is that the class representatives can be formulated in different forms. For instance, the centroid choice of [6] is handled by arithmetic average but it is not a necessity to choose this formulation. In this thesis, three different methods are developed to overcome the computational burden and distance function choice. The distance function used in this thesis is Minkowski distance (L_p distance).

In this thesis, a study is defined as an analysis (or a series of analyses). Method is defined as a result of a study. The first study conducted in this thesis is based on nearest centroid type of sorting method. The nearest centroid classifier type of sorting method is studied for linearization and parameter selection to improve the solution time and classification accuracy. The distance function choice problem is handled with a parameter selection method in the literature. Based on the main characteristics of this nearest centroid classifier type method, another study is conducted. Based on the results of the study, a new linear programming is proposed for monotonic centroids case that is computationally less expensive than nonlinear programming. From a much wider perspective, a third study is conducted as linear approximations of all distance-based methods that are not restricted to multicriteria sorting. In the second and third studies, solution time is improved by the linear approximation. The distance function choice and classification accuracy improvement are also tied to the improvement in solution time by linearization.

Because a set of different linear programs with different distance functions can be solved within the time that is required to solve a single nonlinear program. The distance function resulting with the highest accuracy can be chosen from the solutions of this set of linear programs with different distance functions. In this thesis, the terms "linearization" and "linear approximation" are used interchangeably.

To summarize the studies conducted and methods proposed in this thesis, a list is presented as follows.

1. The first study: a nearest centroid type nonlinear programming sorting method is linearized, distance function and centroid selection is studied. Five methods are proposed in this study.
2. The second study: monotonically ordered centroids case of the first study is analyzed. It is proven that if the centroids are in monotonic order, there is a linear relationship between classification accuracy of a specific set of alternatives and centroids. The linear relationship between centroids and alternatives are used to construct a linear programming model.
3. The third study: a general linear approximation to distance functions is studied that is not restricted to multicriteria sorting.

In all of the three studies, experiments result in solution time and accuracy improvement. The improvements in the first study is due to linearization and distance function and centroid selection. In the second study, the linearization improves the solution time. In the third study, distance function linearization improves the solution time in multicriteria sorting methods significantly. This study finishes the discussions on the linearization of distance functions in this thesis. The linearization in the third study is recommended for all distance based mathematical programming settings.

Organization of this thesis is as follows. Literature review for the first and the second studies based on the nearest centroid type sorting method is presented in Chapter 2. In Chapter 3, proposed methods of the first study is presented. The second study that is monotonically ordered centroids case of the nearest centroid classifier is presented in Chapter 4. Experimental results of first and second studies are reported and discussed in Chapter 5. In Chapter 6, a new linear approximation to distance functions is developed. The related work, application technique, example applications of the approximation method to multicriteria sorting and alternative courses of actions are presented. Experimental results of the third study are reported and discussed in Chapter 7. In Chapter 8, a general discussion of the three studies and the experimental results are given. Results of experiments are associated with the related literature. Finally in Chapter 9, concluding remarks and potential future research directions are presented.

CHAPTER 2

LITERATURE REVIEW

In this chapter, literature review of the first two studies based on the nearest centroid type sorting method are presented. In Section 2.1, the related literature of multicriteria sorting, distance-based sorting methods and centroid-based methods are provided. In Section 2.2, theoretical background is given.

2.1 Literature Review

Ordinal classification methods can be categorized into three groups as statistical, non-parametric and multicriteria methods. Ordinal classification multicriteria methods are called multicriteria sorting.

The first group of methods are statistical methods. Early studies in this group are Linear Discriminant Analysis (LDA) [8] and Quadratic Discriminant Analysis (QDA) [9]. Statistical methods have two main disadvantages as the statistical assumptions and the parametric structure.

The second group of methods are non-parametric methods. Examples of non-parametric methods are K-Nearest Neighbor (KNN) [10] and Artificial Neural Networks (ANN) (e.g., [11]).

The third group of methods are multi-criteria methods. Multi-criteria methods can be classified into two groups as Direct Judgement (DJ) methods and Preference Disaggregation (PD) analysis methods. DJ methods require the preference information from DM to perform the class assignment. Preference information can be in the form of reference profiles or limiting profiles (class thresholds), value functions or preference functions, preference thresholds and criterion weights. The preference information is used to construct the model to perform the class assignment. Therefore, it is obligatory to specify the preference information in DJ methods. However, obtaining preference information from DM may require extra effort and/or cognitive load for DM. Examples of direct judgment methods are Evaluation based on Distance from Average Solution (EDAS) [12], ELimination and Choice Translating REality (ELECTRE-TRI) [13] and ELECTRE-TRI nC [14].

ELECTRE-TRI [13] is an outranking relation method. In outranking relation methods, outranking degree is determined for each alternative. The outranking degree of each alternative is determined based on the comparison of the alternative to a reference profile. If the outranking degree of an alternative is greater than a specified reference profile, then the alternative outranks the reference profile. Each ordinal class is separated by a reference profile. Sorting is performed based on the comparison of alternatives to the reference profile of each class. The need for outranking relation methods is due to the absence of incomparability of value function based methods and transitivity of indifference.

EDAS [12] is another DJ method. The reference profile is not required in EDAS. The required preference information are criterion weights and class cardinalities. A reference artificial alternative (average solution) is computed based on the arithmetic average of each criterion of all alternatives. Based on the comparison of each alternative to the average solution, Positive Deviation from Average (PDA) and Negative Deviation from Average (NDA) are computed for each criterion. PDA and NDA are aggregated with weighted sum and normalized. An Appraisal Score (AS) based on this criteria aggregation is computed. AS is a higher the better type of measure. The alternatives are ranked from best to worst in descending order of AS. Class assignment is performed based on class cardinalities. In EDAS [12], an inventory ABC classification is studied. The class assignment is performed based on the class cardinalities of the ABC classes. The class cardinality means the number of alternatives in the class.

PD sorting methods elicit DM’ s preference information from a set of example decisions of DM or historical data of past decisions. They minimize the effort that is due to obtaining the preference information from DM in DJ methods. In multicriteria sorting, the decision is to assign alternatives to predefined ordinal classes. Therefore, PD sorting methods elicit the preference information of DM from a set of example classifications or historical data. In this thesis, example classifications and historical data are referred as training data. In PD sorting methods, a criteria aggregation function can be used to elicitate the preference of the DM.

The criteria aggregation function can be a utility/value function or a distance function. An example of sorting methods with utility function based criteria aggregation is Utilities Additives DIScriminantes (UTADIS) [3]. In UTADIS, additive utility function is employed to represent preference information of DM and class thresholds are used to discriminate the classes. The additive utility function is formulated in a way that it represents the ordinal relation between the alternatives. The class thresholds are also ordered from best to worst. UTADIS is a Linear Programming (LP) approach that is used to find optimal criterion weights and class thresholds to minimize classification error. Classification error is minimized on the training data. The validity of the criterion weights and class thresholds are tested based on the test accuracy/error level that is calculated using the test data.

Distance-based PD methods are [5]–[7], [15]. To describe distance-based methods, the term Ideal Criterion Vector (ICV) (or ideal point) is explained. ICV is the best possible point in the criterion space. ICV is described as the best point in each criterion of the non-dominated alternatives. In studies of [5], [7], [15], criteria aggregation is formulated as the distance of alternatives to the ICV. In general, weighted L_p distance is employed as the distance function. Mathematical Programming (MP) is employed in [5]–[7], [15].

Chen et al. [5] and Chen et al. [15] develop squared a Euclidean distance-based criteria aggregation model. Centroid (arithmetic average) of the best class is assumed as the ICV. Each criterion of the alternatives and ICV is compared with the squared deviation. Then, criteria aggregation is performed with the weighted sum of the squared deviations. The criteria aggregation is compared with the class thresholds and the class assignment is performed. The total squared classification error is minimized in the objective function. The classification error, class thresholds and the criterion weights are the decision variables. When using squared Euclidean distance, the square of the criterion weighs is not taken. Therefore, the distance formulation is not a regular distance norm when weighted Euclidean distance is evaluated in this way [16]. Although the distance function formulation is linear, both methods are nonlinear quadratic models due to the minimization of total squared classification error. When the regular L_p distance is used, the distance-based sorting method is also a NonLinear Programming (NLP) model and the computational burden increase.

The distance-based sorting is extended to other L_p distances by Çelik et al. [7], [16]. Çelik et al. [7], [16] study Probabilistic Distance-based Sorting method (PDIS). They evaluate the alternatives with distance-based criteria aggregation with regular L_p distance and use MP approach. This makes it an NLP model. Their criteria aggregation function is also based on distance to ICV. Alternatives are evaluated with distance-based criteria aggregation function. As in [5], [15], the evaluation is compared with the class thresholds. In the experimental results, it can be observed that the accuracy performance measure highly deviates for different L_p distances. Robustness of the proposed method w.r.t different L_p distances can be questioned. Although it is not clear which specific L_p distance results in better classification accuracy, it is shown that low p values ($1 \leq p \leq 3$) result in better classification accuracy.

There are four main differences between [5], [15] and [7], [16]. The first difference is that [7], [16] use regular weighted L_p distance and their formulation allows the usage of other L_p distances in addition to Euclidean distance. The second difference is that the proposed method is probabilistic in [7], [16]. The class assignments are determined based on a probabilistic approach. The resulting class assignment is a conditional probability, and the probabilistic approach is fundamentally based on the Bayesian approach. For class assignment, uniform and triangular probability distributions are used. Since it is a multicriteria method, the probability distribution is a joint formulation. Criteria are assumed to be independent therefore the joint probability formulation is employed accordingly. The third difference is that the risk attitude of the DM is also considered in PDIS. Different class assignments for a risk-averse and a risk-seeking DM is shown in the study [16]. Also, optimistic and pessimistic class assignment procedures are developed in [7], [16]. The fourth difference is that the class assignment for the test data is performed by a MP. After thresholds and criterion weights are optimized, a new model is solved to perform the class assignments of the test data. In [7], [16], different accuracy measures are developed based on the class assignment probabilities of alternatives. Besides the criteria aggregation, in [5], [7], [15], [16], class thresholds are also ordered from best to worst as in UTADIS [3]. The main difference between UTADIS and [5], [7], [15], [16] is that the additive utility function is used in UTADIS, and the distance function is used in [5], [7], [15], [16].

Another distance-based multicriteria sorting method is DIstance-based Sorting WithOut class THresholds (DISWOTH) that is developed by Karasakal and Civelek [6], [17]. In DISWOTH, a class centroid (a class representative) is estimated for each ordinal class. Alternatives are evaluated based on their proximity (similarity) to each class centroid. The L_p distance function is employed for formulating the proximity or similarity of an alternative and a class centroid. They [17] also show how the L_∞ distance can be employed for DISWOTH with an LP model. Class assignment of an alternative is performed based on the evaluation of similarity between class centroids and the alternatives. An alternative is assigned to the class of the most similar (nearest) centroid. Therefore, DISWOTH is a Nearest Centroid classifier (NC) type of sorting method, which is similar to K-Means clustering method [18] in terms of cluster assignment. Class centroids are estimated with the arithmetic average of the alternatives of each class. As in PDIS [7], [16], the formulation allows the utilization of different L_p distances. NC type classification methods have roots in nominal classification, it is also called nearest centroid neighborhood [19] and Rocchio classification [20]. Therefore, the NC type formulation of DISWOTH also enables the method to handle the nominal classification.

Unlike PDIS [7], [16] and the study of Chen et al. [5], [15], ordering of classes and alternatives are ignored to improve the classification accuracy in DISWOTH. This enables DISOWTH to evaluate data with non-monotonic criteria that can be viewed as the flexibility of the method. Ignoring ordering (or monotonicity) to improve classification accuracy is discussed in the literature [21]. Findings of Ben-David et al. [21] show that there is no statistically significant difference of classification accuracy between the methods that consider ordering and the ones that ignore the ordering. Furthermore, it is discussed that adding monotonicity to the learning methods impair accuracy.

There are also statistical ordinal classification methods that are based on class centroids [22]–[26]. Liu et al. [23] and Sun et al. [24] develop ordinal classification methods based on LDA. The centroids are employed to find a projection that best discriminates the ordinal classes. Pelckmans et al. [26] develop Least-Square Support Vector Ordinal Regression (LS-SVOR) method that is based on Support Vector Machines (SVM) and LDA. The utilization of centroids in [26] is the same as in [23], [24]. A different nearest centroid-based statistical method is Ordinal Nearest Centroid Projection (OrNCP) that is developed by Tian and Chen [22]. They employ the total absolute deviation from class centroids that is basically an NC formulation with L_1 distance. From that perspective, DISWOTH is an extension of OrNCP that allows the usage of other L_p distances. In OrNCP, different from DISWOTH, ordering of classes is considered. In these statistical methods [22]–[26], centroids are also estimated with arithmetic average.

Recently a study is conducted by Tian et al. [25] that focus on centroid choice. Tian et al. develop a centroid estimation method based on L_p distances. In their study, they show that arithmetic average results in a centroid that minimizes total within class distance when the distance function is the Euclidean norm (that is L_2 distance). They argue that there must be different centroids for different L_p distances. Tian et al. [25] propose the L_p-Centroid method that gives different centroids for different L_p distances. The L_p-Centroids that is given by the method minimize within class distance when L_p distance is chosen. Therefore, the output of the method is a different centroid for an L_p distance, which we can call a centroid-distance pair.

To conclude the literature review, there are distance-based multicriteria methods that are NLP models [5]–[7], [15]–[17]. There are also centroid-based multicriteria sorting methods [6], [12], [17]. EDAS [12] is a DJ method and DISWOTH [6], [17] is a PD method and it is based on the NC formulation. There are also centroid-based statistical machine learning methods [22]–[26]. Except for [25], they use the arithmetic average as the centroid estimation method. Figure 2.1 summarizes the literature review of the multicriteria sorting methods. Figure 2.2 summarizes the centroid-based methods.

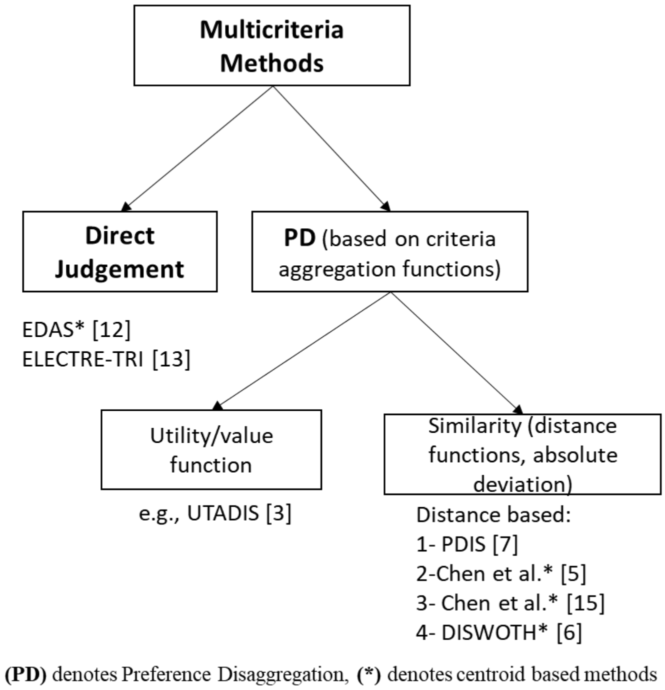


Figure 2.1 Categorization of multicriteria methods

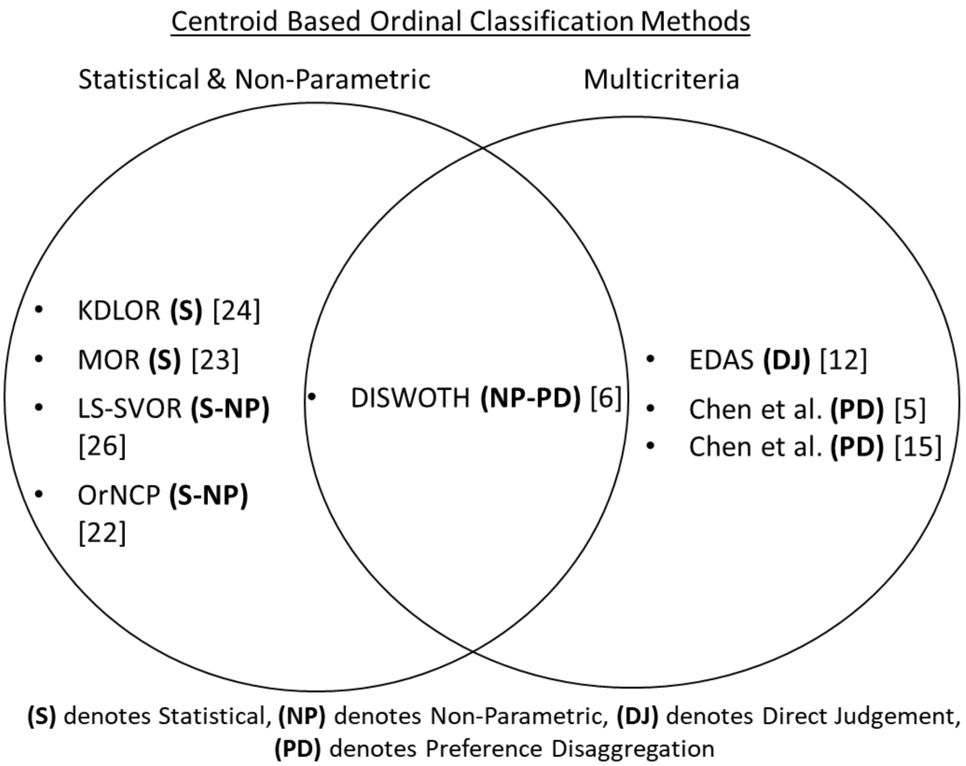


Figure 2.2 Categorization of centroid-based methods

In the literature review, it is observed that the centroid-based methods employ arithmetic average as the centroid estimation method and L_p-Centroid method is not adapted to NC formulations. Furthermore, NC based sorting method (DISWOTH) is an LP model when $p \in \{1, \infty\}$ and it is NLP model that is computationally expensive to solve when $p \notin \{1, \infty\}$. For NC and MP based methods (DISWOTH), our critics are as follows.

1. Only arithmetic average is employed as the centroid estimation method and L_p-Centroid method can be adapted.
2. It is not clear to use which L_p distance since it is not known in advance that which distance function results with better accuracy.
3. L_p distance is a nonlinear formulation. When used in MP, it is an NLP model. NLP models are computationally expensive to solve.

In addition to those three points, DISWOTH can be criticized due to ignoring the ordering of classes and the centroid choice for different L_p distances.

Based on these three points, in this chapter, four new NC based multicriteria sorting methods are developed. Developed methods are MP approaches as DISWOTH. Therefore, they can be viewed as extensions to DISWOTH. In the first method, NLP formulation of DISWOTH is linearized by employing binary variables. In the second method, L_p-Centroid method is adapted to DISWOTH to examine the effect of centroid and distance function choice to the accuracy. In the third method, first two methods are combined, and two extensions are developed to represent ordering of classes. Choosing a proper centroid and distance pair is handled with L_p-Centroid method in the second and third models. Ordering of classes is considered with compromise ranking and additive difference of utilities [27] as extensions to third model.

As mentioned, Ben-David et al. [21] discuss that adding monotonicity to learning models may impair the classification accuracy. Not to impair accuracy with monotonicity, the two extensions are considered with soft constraints that seek alternative optimal solutions to the best accuracy outcome.

2.2 Theoretical Background

In this section, criteria aggregation based on utility functions is exemplified with UTADIS method. Theoretical background for the distance functions and criteria aggregation based on distance functions are explained. Then, MP formulation of DISWOTH is presented. Lastly, L_p-Centroid method is explained.

Relevant notation for the multicriteria sorting methods is as follows. Index $i \in \{1,2,\cdots,n\}$ represents alternatives, $j \in \{1,2,\cdots,m\}$ represents criteria, and $q,\ r \in \{1,2,\cdots,Q\}$ represents ordinal classes. The ordering of the classes is presented such that class 1 is the worst class and class Q is the best class. ϵ is an infinitesimal positive scalar. A represents the set of alternatives and A_i represents i^{th} alternative. A_{ij}^q represents the j^{th} criterion evaluation of alternative i belonging to class q, $A_i^q = \{A_{i1}^q, A_{i2}^q, \dots, A_{im}^q\}$. C^q is the set of alternatives belonging to class q that is $A_i^q \in C^q$. An example setting of the reference set (training data) is illustrated in Table 2.1.

The alternatives and classes are defined as follows.

1. $C^q \cap C^r = \emptyset$ for $q \neq r$
2. $C^1 \cup C^2 \cup C^3 \cup \ldots \cup C^Q = A$
3. $C^q \neq \emptyset$ and $C^q \in A$

The preference and indifference settings for alternatives and ordinal classes are as follows. $\sim$ denotes indifference relationship (equally preferred entities) and $\gg$ denotes preference relationship. $A{\sim}B$ means DM is indifferent between alternatives/actions A and B. $A \gg B$ means DM prefers A to B.

1. If A_i and $A_{i'} \in C^q$, then $A_i{\sim}A_{i'}$.
2. If $A_i \in C^q$ and $A_i{\sim}A_{i'}$, then $A_{i'} \in C^q$.
3. $A_i \in C^q$ and $A_{i'} \in C^{q+1}$, then $A_{i'} \gg A_i$ for $q < Q$.

Table 2.1 Example setting for training data

Alternatives	*Criterion 1*	*Criterion 2*	…	*Criterion m*	*Class Label*
Alternative *1*	A_{11}^1	A_{12}^1	…	A_{1m}^1	*1*
Alternative *2*	A_{21}^2	A_{22}^2	…	A_{2m}^2	*r*
Alternative *i*	A_{i1}^q	A_{22}^q	…	A_{im}^q	*q*
…	…	…	…	…	…

In this thesis, it is assumed that each criterion is monotonic. The monotonic criterion is explained as follows. A criterion can be "*higher the better*" type (maximization type or benefit type) or it can be "*lower the better*" type (minimization type or cost type). Without loss of generality, terms maximization type and minimization type are used in this thesis. If a criterion is maximization type, then higher values of that criterion are preferred to lower values. If a criterion is minimization type, then lower values of that criterion are preferred to higher values. A criterion can also be non-monotonic, meaning that an intermediate value can be the most preferred value. This case is not considered in this thesis. It is assumed that the criteria are independent.

Based on assumptions, preference order of alternatives is determined based on the evaluation of monotonic criteria. The values of each alternative on each criterion can be categorical and numerical. Numerical criterion can be discrete or continuous. The trade-off between criteria is defined by criterion weights.

In PD methods, criteria aggregation is performed using criteria aggregation function and criterion weights. A composite indicator or a score is computed. Then, the score is compared with class thresholds to perform the class assignment. After monotonic criteria and ordinal class concepts are introduced, a class assignment example of multicriteria sorting is illustrated in Figure 2.3.

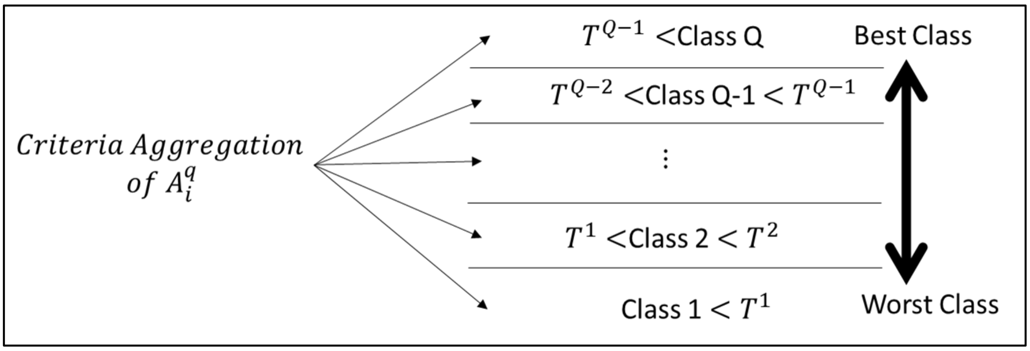


Figure 2.3 A class assignment example with criteria aggregation

2.2.1 UTADIS

In this section, utility function based criteria aggregation PD sorting method, namely UTADIS [3] is introduced. Additional notation for UTADIS method is as follows. Decision variable w_j is the weight of the j^{th} criterion. Decision variable T^q is the threshold separating classes q and $q+1$. $U(.)$ denotes the additive utility function and $u_j(.)$ is the marginal utility function. g_{j*} and g_j^* are the worst, and the best values for criterion j. Moreover, criterion j is divided into $\Lambda_j - 1$ intervals, the intervals are denoted by t ($\left[g_j^t, g_j^{t+1}\right], t = 1,2, \dots, \Lambda_j - 1$). The value of Λ_j is determined by DM. Λ_j is used to approximate the utility function by determining the number of marginal utility points u_j. Therefore, the larger Λ_j, the approximation becomes more precise. g_j^t is calculated as in equation (1). Equation (1) is a linear interpolation.

$$g_j^t = g_{j*} + \frac{t-1}{\Lambda_j - 1}(g_j^* - g_{j*}) \tag{1}$$

The aim of introducing the intervals is to calculate the marginal utility of alternatives in the interval $\left[g_j^t, g_j^{t+1}\right]$. For $A_{ij}^q \in \left[g_j^t, g_j^{t+1}\right]$, $u_j(A_{ij}^q)$ is calculated as in equation (2).

$$u_j\left(A_{ij}^q\right) = u_j\left(g_j^t\right) + \frac{A_{ij}^q - g_j^t}{g_j^{t+1} - g_j^t}\left[u_j\left(g_j^{t+1}\right) - u_j\left(g_j^t\right)\right], A_{ij}^q \in \left[g_j^t, g_j^{t+1}\right] \tag{2}$$

UTADIS respects the preference order of breakpoints of the intervals based on monotonicity. The preference order of marginal utilities is satisfied with constraint (3). By respecting the preference order of breakpoints of intervals, according to equation (2), it also orders alternatives in each criterion. When equations (2)-(3) are considered together, an alternative falling in a higher interval dominates the alternative falling in the lower interval.

$$\omega_{jt} = u_j\left(g_j^{t+1}\right) - u_j\left(g_j^t\right) \geq 0, \forall j, \forall t \leq \Lambda_j - 1 \tag{3}$$

ω_{jt} is the utility value of interval $\left[g_j^t, g_j^{t+1}\right]$. Therefore, $u_j\left(g_j^t\right)$ can be reformulated as equation (4) and equation (2) can be reformulated as equation (5). Additive utility function for an alternative is formulated as equation (6). $U\left(A_i^q\right)$ maps an m dimensional real numbered vector $A_i^q \in R^m$ to a single dimension R^1, $U(.): R^m \to R^1$. The marginal utility for each criterion is normalized as in equations (7)-(8).

$$u_j\left(g_j^{t'}\right) = \sum_{t=1}^{t'-1} \omega_{jt}, \forall j, t' = 1,2, \dots, \Lambda_j - 1 \tag{4}$$

$$u_j\left(A_{ij}^q\right) = \sum_{t=1}^{t'-1} \omega_{jt} + \frac{A_{ij}^q - g_j^t}{g_j^{t+1} - g_j^t}\omega_{jt'}, \forall j, t' = 1,2, \dots, \Lambda_j - 1 \tag{5}$$

$$U\left(A_i^q\right) = \sum_{j=1}^{m} u_j(A_{ij}^q) \tag{6}$$

$$\sum_{j=1}^{m} u_j(g_j^*) = 1 \tag{7}$$

$$\sum_{j=1}^{m} u_j(g_{j*}) = 0 \tag{8}$$

Classification of an alternative with UTADIS is performed as follows.

$$A_i \in C^1 \; if \, T^1 > U(A_i^1) \; \forall i \tag{9}$$

$$A_i \in C^Q \; if \, T^{Q-1} \leq U\left(A_i^Q\right) \; \forall i \tag{10}$$

$$A_i \in C^q \; if \, T^q > U(A_i^r) \geq T^{q-1} \; \forall i, 1 < q < Q \tag{11}$$

In UTADIS, a class assignment is accurate if the three conditions hold as follows.

1. $T^1 > U(A_i^1)$.
2. $T^{Q-1} \leq U\left(A_i^Q\right)$.
3. $T^q > U\left(A_i^q\right) \geq T^{q-1} \; \forall i, \forall q \notin \{1, Q\}$.

If conditions 1-3 do not hold, then it is an inaccurate (erroneous) class assignment. Two error variables are used to define erroneous class assignments, $e_i^+$and e_i^-. e_i^- is the class assignment error of A_i^q to a worse class and e_i^+ is the class assignment error of A_i^q to a better class. $e_i^+$and e_i^- are formulated as equations (12)-(13).

$$e_i^- = max \, \{0, U\left(A_i^q\right) - T^{q-1}\} \tag{12}$$

$$e_i^+ = max \, \{0, T^q - U\left(A_i^q\right)\} \tag{13}$$

UTADIS minimizes class assignment errors. UTADIS model is as follows.

(UTADIS)

$$Minimize \sum_i e_i^+ + e_i^- \tag{14}$$

Subject to:

$$U\left(A_i^q\right) + e_i^- - \epsilon \geq T^r + \epsilon, \forall i, \forall r < Q \tag{15}$$

$$U\left(A_i^q\right) - e_i^+ + \epsilon \leq T^{r-1} + \epsilon, \forall i, \forall r > 1 \tag{16}$$

$$\sum_{j=1}^{m} \sum_{t=1}^{A_j - 1} \omega_{jt} = 1 \tag{17}$$

$$T^r - T^{r-1} \geq \epsilon, \forall r > 1 \tag{18}$$

$$\omega_{jt} \geq 0, \forall j, \forall t \tag{19}$$

$$e_i^+, e_i^- \geq 0, \forall i \tag{20}$$

Objective function (14) minimizes the total class assignment error. Constraints (15)-(16) perform class assignments by comparing the criteria aggregation and class thresholds. Constraint (17) is used to normalize the criterion weights w.r.t monotonic utility values of the predetermined criterion intervals. Constraint (18) orders class thresholds in strictly increasing order from worst to best as utility is a higher the better type of measure. Constraints (19)-(20) are sign constraints.

A global preference mechanism is modeled with criteria aggregation using additive utility functions in UTADIS. Then, criteria aggregation is compared with class thresholds that are class representatives. Class assignment with UTADIS is illustrated in Figure 2.4 for a three-class example.

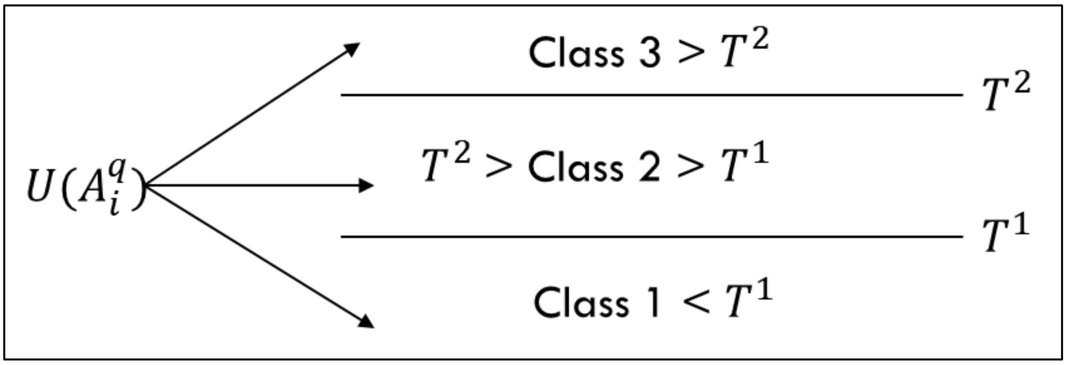


Figure 2.4 Class assignment illustration of UTADIS on a three-class example

2.2.2 Distance Functions and Distance-based Criteria Aggregation

Criteria aggregation can be formulated based on distance functions as well. Distance-based ordinal classification methods that use criteria aggregation are [5]–[7], [15], [22].

To describe distance-based methods, the term Ideal Criterion Vector (ICV) (or ideal point) is explained. ICV is the best possible point in the criterion space that is the best point in each criterion of the alternatives. In studies of [5], [7], [15] criteria aggregation is performed based on the distance of alternatives to the ICV. Weighted L_p distance is employed as the distance function that is formulated as equation (21). $\mathrm{L_p}$ distance is called Rectilinear (Manhattan or city block) distance when $\mathrm{p} = 1$ and Euclidean when $\mathrm{p} = 2$. A specific version of $\mathrm{L_p}$ distance is $\mathrm{p} = \infty$, which is called Tchebycheff distance.

[5], [15] use squared Euclidean (L_2^2) distance as the distance function. ICV is determined as the centroid of the best class. Çelik et al. [7] extend the formulation of

[5], [15] to all L_p distances and develop a probabilistic class assignment method, namely PDIS. As in UTADIS, criteria aggregation and class thresholds are used to discriminate the classes. In this thesis, $d_p^w(A, B)$ represents weighted L_p distance between two points A and B. $d_p^w(A, B)$ is formulated as in equations (21)-(23).

$$d_p^w(A, B) = \sqrt[p]{\sum_j w_j^p \left|A_j - B_j\right|^p} \tag{21}$$

$$\sum_j w_j = 1 \tag{22}$$

$$w_j \geq 0, \forall j \tag{23}$$

A distance function is called metric (norm) if it satisfies the following three properties. L_p distance is a metric when $p \geq 1$ and it is not a metric for $p < 1$. $d_p(A, B)$ denotes L_p distance and $A, B, C \in R^m$.

1. Positivity: $d_p(A, B) > 0$ and $d_p(A, B) = 0$ iff $A = B$.
2. Symmetry: $d_p(A, B) = d_p(B, A)$
3. Triangular inequality: $d_p(A, B) \leq d_p(A, C) + d_p(B, C)$

The criteria aggregation function employed in [7], [16] is as in equation (24). ICV is denoted by I and j^{th} criterion value of ICV is denoted by I_j. Since ICV is the best possible point, distance (or dissimilarity) to ICV is a lower the better type of measure. Class assignment with distance-based criteria aggregation function is illustrated in Figure 2.5 on a three-class example.

$$d_p^w(A_i^q, I) = \sqrt[p]{\sum_j w_j^p \left|A_{ij}^q - I_j\right|^p} \tag{24}$$

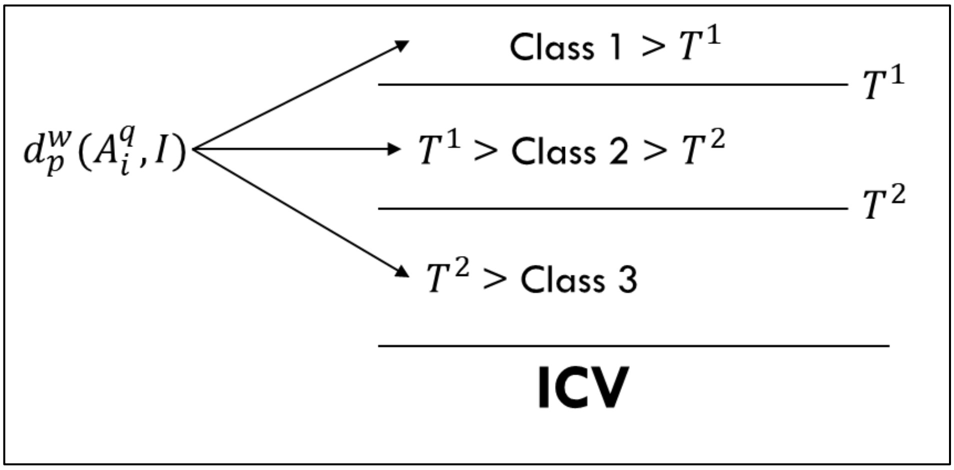


Figure 2.5 Class assignment illustration of distance-based sorting on a three-class example

The illustration in Figure 2.5 can be formulated as follows.

$$d_p^w(A_i, I) \leq T^{Q-1} \rightarrow A_i \in C^Q$$

$$T^{q+1} \leq d_p^w(A_i, I) \leq T^q \ \rightarrow A_i \in C^q \ \forall q < Q$$

$$d_p^w(A_i, I) \geq T^1 \rightarrow A_i \in C^1$$

Class thresholds are ordered in decreasing order from best to worst as opposed to UTADIS (compare Figures 2.4 and 2.5). This is because the utility function (equation (6)) is a higher the better and distance-based value function (equation (24)) is a lower the better type of measure.

To represent classification error, similar to UTADIS, equations (25)-(26) are used. A more detailed model is given in Section 6.3.1.

$$e_i^- = max\ \{0, T^{q-1} - d_p^w(A_i^q, I)\ \} \tag{25}$$

$$e_i^+ = max\ \{0, d_p^w\left(A_i^q, I\right) - T^q\} \tag{26}$$

Figure 2.5 is an illustration of the class assignment logic of distance-based sorting. In criterion space, a class threshold forms the contour of L_p distance chosen. An example illustration for two criteria and two class problem is given in Figure 10.2 in Appendix F.

2.2.3 DISWOTH and Nearest Centroid Classifier

DISWOTH is a NC type sorting method. In this setting, each class has a class centroid which can be defined as a typical artificial (or real) alternative of that class. DISWOTH and NC assign the alternatives to the class of the nearest centroid. Additional notation for DISWOTH and NC are as follows. μ_j^q is the j^{th} criterion value of the centroid of class q. Instead of e_i^+ and e_i^- of UTADIS, a single error variable is used as e_i.

For DISWOTH and NC, class centroids are calculated as in equation (27) as the arithmetic average of the alternatives of each class. However, it is not necessary to use arithmetic average as the class centroid. Class assignment is performed as in equation (28). Class assignment error is calculated as in equation (29) for NC and equation (30) for DISWOTH.

$$\mu_j^q = \frac{1}{|C^q|} \sum_{i|A_{i \in C^q}} A_{ij}^q, \forall q, \forall j \quad (27)$$

$$A_i \in C^q \ if\, q = argmin_r \{d_p^w(A_i, \mu^r)\} \quad (28)$$

For NC,

$$e_i = \begin{cases} 0 & if \ \ d_p^w(A_i^q, \mu^q) = min_r \{d_p^w(A_i^q, \mu^r)\} \\ 1 & otherwise \end{cases} \forall i \quad (29)$$

For DISWOTH,

$$e_i = max\{0, d_p^w(A_i^q, \mu^q) - d_p^w(A_i^q, \mu^r)\} \quad (30)$$

Formulation of DISWOTH is as follows.

(DISWOTH)

$$Minimize \ \sum_{i|A_i \in C^q} \frac{e_i}{|C^q|} \quad (31)$$

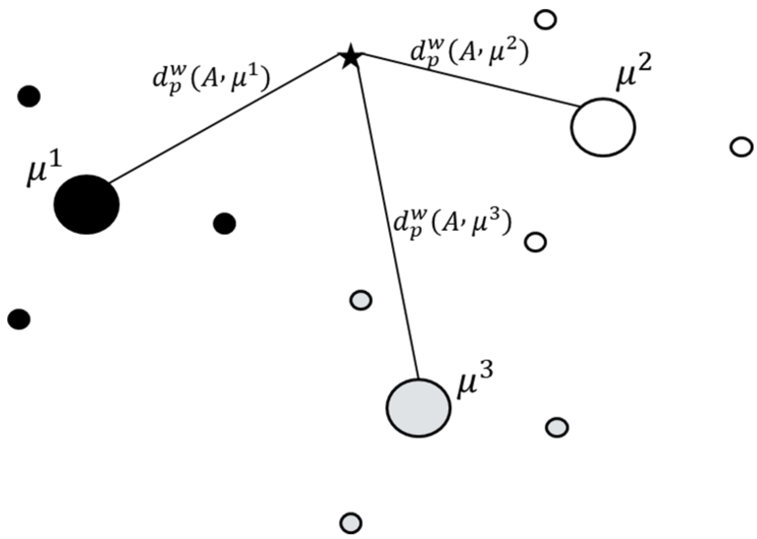
Subject to:

Constraints (22)-(23)

$$e_i - \epsilon \geq d_p^w(A_i^q, \mu^q) - d_p^w(A_i^q, \mu^r) \ \forall i, q \neq r \quad (32)$$

$$e_i \geq 0 \quad (33)$$

Objective function (31) minimizes class-weighted classification error. In objective function (31), $|C^q|$ is the cardinality of class q. Class cardinality weighted total classification error is minimized. Constraint (32) performs class assignments and computes the classification errors based on equation (30). The class assignment logic of DISWOTH is illustrated in Figure 2.6 for a three-class example. As it can be noticed from DISWOTH model, different from [3], [5], [15], [7], ordering of classes (or monotonicity) is ignored. This is done to improve the classification accuracy.



Small circles are alternatives from black, grey and white classes. Large circles are centroids of each class. Star is an alternative to be classified into one of three classes. Classification is to be performed based on equation (28)

Figure 2.6 Class assignment illustration of DISWOTH on a three-class example.

2.2.4 L_p-Centroid Method

Previous centroid-based studies employ arithmetic average as the centroid estimation method. A different centroid estimation method for centroid-based classifiers is developed by Tian et al. [25], namely L_p-Centroid method. Tian et al. [25] criticize the usage of arithmetic average (equation (27)) as the centroid estimation. They show that equation (27) can be obtained by minimizing the total squared Euclidean distance of all alternatives to a point as in equation (34). $||\mu^q - A_i^q||_2^2$ represents the squared Euclidean distance. Authors argue that, for each L_p distance, a different centroid should be estimated.

$$\frac{1}{|C^q|}\sum_{i|A_{i\in C^q}} A_{ij}^q = argmin_{\mu_j^q} \left\{\sum_{i|\, A_i\in C^q} ||\mu^q - A_i^q||_2^2\right\} \quad (34)$$

L_p-Centroid method is formulated as equation (35) and estimates different centroid for each L_p distance. $||\mu^q - A_i^q\ ||_p^p$ denotes the p^{th} power of L_p distance. (35) is solved with L_p-Centroid Algorithm [25]. $\mu_{L_p}^q$ is the L_p-Centroid of class q. Besides providing a different centroid for each L_p distance, the regularization effect of L_p-Centroid method is also discussed.

(L_p-Centroid)

$$\mu_{L_p}^q = argmin_{\mu^q}\ \{\sum_{i|\ A_i \in\ C^q} ||\mu^q - A_i^q\ ||_p^p\}. \quad (35)$$

2.2.5 Nearest Centroids and Nearest Central Profiles

In multicriteria sorting, the central profiles are also used as class representatives. The terms “central profiles” and “centroids” are often used interchangeably in the literature (e.g., [28], [29]). In this thesis, the term “central profile” is not used because the nearest centroid and nearest central profile approaches differ in comparison of the class representatives with alternatives.

When central profiles are employed in multicriteria sorting, a criteria aggregation is performed on alternatives and central profiles as the first step. That is both alternatives and central profiles are mapped into a single dimensional value space. Then, as the second step, the central profiles and alternatives are compared in the single dimension. Class assignments are performed accordingly. To clarify, a class assignment structure is used as follows. Let $U(.)$ be a criteria aggregation function.

$$U(A_i) \leq U(\mu^1) \rightarrow A_i \in C^1$$

$$U(\mu^q) \leq V(A_i) < U(\mu^{q+1})\ and\ U(\mu^{q+1}) - U(A_i) > U(A_i) - U(\mu^q) \rightarrow A_i \in C^q\ \forall q < Q$$

$$U(\mu^q) \leq U(A_i) < U(\mu^{q+1})\ and\ U(\mu^{q+1}) - U(A_i) < U(A_i) - U(\mu^q) \rightarrow A_i \in C^{q+1}\ \forall q < Q$$

$$U(A_i) > U(\mu^Q) \rightarrow A_i \in C^Q$$

This structure is not used in the nearest centroid type of classifier. Instead, the class assignment structure in DISWOTH is employed with equation (28).

To apply the class assignment structure in equation (28), alternative and the class representative is compared as the first step. Then, as the second step, the criteria aggregation is performed with a similarity measure. In DISWOTH, the L_p distance is used as the similarity measure. To sum up, the difference between the nearest central profile based methods and nearest centroid-based methods is that the first and the second steps are switched.

CHAPTER 3

EXTENSIONS TO DISWOTH METHOD

This chapter presents the first study conducted in this thesis. The base method, DISWOTH [6], that is a nearest centroid type of sorting method is extended to improve the classification accuracy and solution time. Five methods are proposed in this chapter.

The first method is a linear approximation of DISWOTH by employing a Mixed Integer Programming (MIP) approach. In the second method, L_p-Centroid method is employed. An algorithm is developed to choose a good L_p distance and L_p-Centroid pair as a heuristic approach. The third method is the combination of the first two. L_p-Centroid is adapted as in the second method and the formulation is linearized as MIP as in the first method. Two extensions to the third method are developed to reflect the ordering of classes in the model. These extensions are formulated in a way that they seek alternative solutions to the best accuracy solution. In the third method, objective function is changed so that the alternative solution seeking procedure is enabled. All of the three methods and the two extensions are based on DISWOTH. For the methods proposed in this chapter, it is assumed that the data and centroids are scaled to [0,1] range (explained in Chapter 5 equation (100)). Therefore, Big M values used in this section equal to 1.

After introducing the proposed methods, application procedure and their categorization in the literature is presented.

3.1 Linearization of DISWOTH with MIP, Bin-Dis Method

The first method is the Binary variable DISWOTH method, namely Bin-Dis. NLP DISWOTH method is converted into a Mixed Integer NonLinear Programming

(MINLP) model by formulating the classification error with a binary variable. Then, the MINLP model is linearized.

The main motivation behind the usage of binary variables can be explained by the means and ends objective approach [30]. The main aim and the fundamental objective are to maximize classification accuracy (minimize classification error). Note that maximizing classification accuracy (minimizing classification error) is directly formulated with maximizing (minimizing) the total "*number*" of accurate (inaccurate) class assignments. Minimizing the total classification error with a continuous variable does not necessarily minimize the total number of inaccurate class assignments. Therefore, it can be seen as a means objective that serves the aim. The ends objective in here is to minimize the total number of inaccurate class assignments. Summation of the binary error variables is exactly the ends objective here. With the introduction of binary variable classification error, the classification error of Bin-Dis is formulated as equation (29). To apply this adjustment, constraint (32) is changed as constraint (36) and constraint (33) is changed as constraint (37).

$$Me_i - \epsilon \geq d_p^w\left(A_i^q, \mu^q\right) - d_p^w\left(A_i^q, \mu^r\right) \; \forall i, q \neq r \tag{36}$$

$$e_i \in \{0,1\} \tag{37}$$

Big M in constraint (36) is a sufficiently large number. Constraint (36) can be linearized as follows. $e_i = 1$ iff, $\sqrt[p]{\sum_j w_j^p \left|A_{ij}^q - \mu_j^q\right|^p} > \sqrt[p]{\sum_j w_j^p \left|A_{ij}^q - \mu_j^r\right|^p}$ for some $q \neq r$. Relaxing the roots does not change the value of e_i. Therefore, the inequality can be rewritten as $\sum_j w_j^p \left|A_{ij}^{\mathrm{q}} - \mu_j^q\right|^p > \sum_j w_j^p \left|A_{ij}^{\mathrm{q}} - \mu_j^r\right|^p$. After relaxing the roots, power p of w_j^p can also be relaxed based on the decision boundary of classification. Relaxation of power p of w_j^p is as follows.

Let E denote the set of equidistant points to μ^q and μ^r for classes $q \neq r$. E satisfies equation (38) below. Equation (38) can be rearranged as equation (39). Assume a positive constant Ω as in equation (40).

$$\sum_j w_j^p \left|E_j - \mu_j^q\right|^p = \sum_j w_j^p \left|E_j - \mu_j^r\right|^p \tag{38}$$

$$\sum_j w_j^p (|E_j - \mu_j^q|^p - |E_j - \mu_j^r|^p) = 0 \tag{39}$$

$$\Omega \sum_j w_j^p (|E_j - \mu_j^q|^p - |E_j - \mu_j^r|^p) = 0 \tag{40}$$

Decision boundary equations (39) and (40) result with the same decision boundary. Because positive constant Ω can be cancelled due to zero in the RHS of (40). Therefore, if there exist v_j which equals Ωw_j^p ($v_j = \Omega w_j^p$) then it can be used instead of w_j^p to linearize constraint (36). Such v_j satisfies equation (41).

$$\frac{1}{\Omega} = \frac{v_1}{w_1^p} = \frac{v_2}{w_2^p} = \cdots = \frac{v_m}{w_m^p} \tag{41}$$

Based on equation (41), v_j can be represented as equation (42) for some index $k \in \{1,2,\dots,m\}/j$.

$$v_j = \frac{w_j^p}{w_k^p} v_k \forall j \tag{42}$$

To analyze whether equation (42) violates constraint (22) (weight normalization constraint), replace w_j in constraint (22) with v_j formulation in equation (42). (43) is the resulting equation. For simplicity, let $k = m$ in equation (42).

$$\frac{w_1^p}{w_m^p} v_m + \frac{w_2^p}{w_m^p} v_m + \cdots + \frac{w_m^p}{w_m^p} v_m = 1 \rightarrow v_m \sum_{j=1}^m \frac{w_j^p}{w_m^p} = 1 \rightarrow v_m = \frac{w_m^p}{||w||_p^p} \tag{43}$$

(43) shows that a linear substitute of w_j^p is obtained by the distributive normalization of w_j. Insert $\frac{w_j^p}{||w||_p^p}$ in constraint (22) to see that constraint (22) is not violated. This result is shown in equation (44).

$$\sum_{j=1}^m \frac{w_j^p}{||w||_p^p} = \frac{||w||_p^p}{||w||_p^p} = 1 \tag{44}$$

The linearization of criterion weights in weighted L_p distance is exemplified with a numerical example (see Appendix G).

As a result, there exist a linear variable as a substitute for optimal criterion weights w_j^*. By employing binary variable, it is shown that w_j^p in constraint (36) can be linearized and reformulated as MIP with constraint (45).

$$Me_i - \epsilon \geq \sum_j w_j \left|A_{ij}^q - \mu_j^q\right|^p - \sum_j w_j \left|A_{ij}^q - \mu_j^r\right|^p \; \forall i, q \neq r \tag{45}$$

Bin-Dis model is proposed as linearized version of DISWOTH. Bin-Dis model is as follows.

(Bin-Dis)

Objective function (31)

Subject to:

Constraints (22)-(23), (37) and (45)

3.2 L_p Centroid Induced DISWOTH, L_pDis Method

The second method is the L_p-centroid induced DISWOTH method, namely L_pDis. L_p-Centroid method [25] is adapted to DISWOTH to improve the classification accuracy.

To properly adapt L_p-Centroid method to DISWOTH, an algorithm is developed. Distance Choice (DC) algorithm solves the problem of distance function and centroid choice. It finds a distance-centroid pair to improve the classification accuracy.

DC algorithm:

Step 1: Initialize $p^* = 0,\ \mu_{L_{p*}} = [0]_{Qxm}\ and\ z^* = n$.

Step 2: Increment p by a "*small value*" and solve L_p-Centroid Model for L_p distance with L_p-Centroid Algorithm and find the resulting centroids μ_{L_p}= $[\mu_{L_p}^1, \mu_{L_p}^2, \dots, \mu_{L_p}^3]$.

Step 3: With μ_{L_p} and p, by L_p distance, calculate total classification accuracy, $z_p = \sum_i e_i$. Compute $e_i \in \{0,1\}$ using equation (29) with equal criterion weights.

Step 4: If $z_p \leq z^*$ then, $z^* = z_p$, $p^* = p$ and $\mu_{L_{p*}} = \mu_{L_p}$.

Step 5: If $p < p'$ then, return to Step 2. Else terminate.

Outputs: L_{p*} and $\mu_{L_{p*}}^q$ for each q

The small value in step 2 of the DC algorithm is chosen as 0.1 and the stopping condition p' is chosen as 10. Outputs of the DC algorithm are the p^* value to use as L_{p*} distance and the $\mu_{L_{p*}}^q$ to use in DISWOTH as centroid estimation. DISWOTH method with L_{p*} distance and $\mu_{L_{p*}}^q$ is named L_pDis Method. Although the star sign (*) is used, note that that p^* and the $\mu_{L_{p*}}$ do not mean an optimal distance and centroid pair. They are improved distance and centroid pair in terms of nearest centroid classification accuracy according to the DC algorithm. Therefore, the DC algorithm is a heuristic approach to determine improved p^* and the $\mu_{L_{p*}}$. L_pDis is an NLP model and formulated as follows.

(L_pDis)

Objective function (31)

Subject to:

Constraints (22)-(23), (33)

$$e_i - \epsilon \geq d_{p*}^w\left(A_i^q, \mu_{L_{p*}}^q\right) - d_{p*}^w\left(A_i^q, \mu_{L_{p*}}^r\right) \ \forall i, q \neq r \tag{46}$$

3.3 MIP L_p-Centroid Induced DISWOTH, Bin-L_pDis Method

The third method is Binary variable L_p-centroid induced DISWOTH, namely Bin-L_pDis method. Linearization made in Section 3.1 for Bin-Dis is also used in this method. As L_pDis in Section 3.2, $\mu^q_{L_{p*}}$ and L_{p*} are used. Constraint (46) is reformulated linearly as in constraint (47).

$$Me_i - \epsilon \geq \sum_j w_j \left|A^q_{ij} - \mu^q_{L_{p*}j}\right|^{p*} - \sum_j w_j \left|A^q_{ij} - \mu^r_{L_{p*}j}\right|^{p*} \; \forall i, q \neq r \tag{47}$$

(Bin-L_pDis)

$$Minimize \; \sum_{i=1}^{n} e_i \tag{48}$$

Subject to:

Constraints (22)-(23), (37) and (47)

3.4 Ordering of Classes

In multicriteria sorting problem, classes are ordered w.r.t a preference order. In DISWOTH, the ordering is ignored to improve classification accuracy. In this section, ordering of classes is applied to Bin-L_pDis method. It is applied to the proposed methods with soft constraints that seek alternative solutions of best accuracy outcome. Referring to the findings of Ben-David et al. [21], with this approach, the ordering of classes is considered without decreasing the classification accuracy. In both extensions, criterion-wise min-max feature scaling is applied to data. The data sets are scaled to [0,1] range (explained in Chapter 5 equation (100)).

3.4.1 Compromise Ranking Extension

In this extension, it is assumed that the class centroids are ordered according to a preference order as the classes. The ordering relation is formulated with distance to ICV. It is assumed that the centroid of a more preferred class should be closer to ICV than a less preferred class. This formulation is similar to the criteria aggregation function of Çelik et. al [7]. Additional notation is as follows. J^+ represents the set of maximization criteria and J^- represents the set of minimization criteria. The ICV, I is found as in equation (49).

$$I_j = \begin{cases} \max_j\{A_{ij}^q\} = 1 \; \forall j \in J^+ \\ \min_j\{A_{ij}^q\} = 0 \; \forall j \in J^- \end{cases} \tag{49}$$

The formulation for extension is given in constraint (50).

$$\sqrt[p]{\sum_j w_j^p \left| \mu_j^q - I_j \right|^p} > \sqrt[p]{\sum_j w_j^p \left| \mu_j^{q-1} - I_j \right|^p} \; \forall i \; \forall q > 1 \tag{50}$$

Inequality (50) can be linearized and simplified to constraint (51). Because the greater than operator is not affected by the p^{th} degree root. After relaxation of the root, same linearization approach in Bin-Dis can be applied and formulated linearly.

$$\sum_j w_j \left| \mu_j^{q-1} - I_j \right|^p - \sum_j w_j \left| \mu_j^q - I_j \right|^p > 0 \; \forall i \; \forall q > 1 \tag{51}$$

Constraint (51) may not be feasible always. A new free variable λ is introduced to make (51) a soft constraint as in (53). Objective function is updated as (52).

(Bin-L_pDis Com)

$$Minimize \; \sum_{i=1}^{n} e_i - \lambda \tag{52}$$

Subject to:

Constraints (22)-(23), (37) and (47)

$$\sum_j w_j (\left| A_{ij}^q - I_j \right|^p - \left| A_{ij}^{q-1} - I_j \right|^p) \geq \lambda \; \forall i, \forall q > 1 \tag{53}$$

$$\lambda \leq \epsilon \tag{54}$$

$$\lambda \text{ is } u.r.s \tag{55}$$

Lastly, due to the scaling of the data and centroids, weighted L_p distance formulation returns values in range [0,1]. Therefore, the difference of two such distance functions is in the range (-1,1) for L_p distances $p \geq 1$. In constraint (53), centroids are ordered with compromise ranking formulation. Due to constraints (54)-(55), constraint (53) is a soft constraint. λ is maximized in the objective to satisfy constraint (51). As e_i is a binary variable and λ is in the range of (-1,ϵ], employing objective function (52) provide an alternative solution of optimal of Bin-L_pDis model that satisfy constraint (51) as much as possible. Therefore, the solution of Bin-L_pDis Com model is the best accuracy obtained by Bin-L_pDis with ordered class centroids. The proof for finding alternative solutions with objective function (52) and constraints (53)-(55) is explained in Appendix A.

3.4.2 Additive Difference Model Extension

Additive Difference Model (ADM) [27] extension is applied to Bin-L_pDis method to reflect the ordering of classes. ADM is applied as a soft constraint as in Bin-L_pDis-Com. This extension is named Bin-L_pDis ADM. Additional notation for Bin-L_pDis ADM is as follows. $g_j(.)$ is evaluation function of criterion j. $g_j(.)$ is formulated such that $g_j(\mu_j^q) = \mu_j^q, \forall j \in J^+$ and $g_j(\mu_j^q) = -\mu_j^q,, \forall j \in J^-$. Assume linear utility function $U(x) = \sum_{j\in J^+} w_j g_j(x_j) + \sum_{j\in J^-} w_j g_j(x_j)$. For two centroids of two adjacent ordered classes, utility function $U(.)$ can be represented as constraint (56).

$$U(\mu^q - \mu^{q-1}) = \sum_{j\in J^+} w_j g_j(\mu_j^q - \mu_j^{q-1}) + \sum_{j\in J^-} w_j g_j(\mu_j^q - \mu_j^{q-1}) =$$
$$\sum_{j\in J^+} w_j(\mu_j^q - \mu_j^{q-1}) + \sum_{j\in J^-} w_j(\mu_j^{q-1} - \mu_j^q) > 0 \ \forall q > 1 \qquad (56)$$

Constraint (56) can be rewritten as (57) as a soft constraint. Definition of λ is as the same in Com extension.

$$\sum_{j\in J^+} w_j(\mu_j^q - \mu_j^{q-1}) + \sum_{j\in J^-} w_j(\mu_j^{q-1} - \mu_j^q) \geq \lambda \ \forall q > 1 \qquad (57)$$

(Bin-L_pDis ADM)

Objective Function (52)

Subject to:

Constraints (22)-(23), (37), (54)-(55) and (57)

3.4.3 Application Procedure and Categorization of the Proposed Methods

The categorization of the proposed methods in the literature is presented in Figure 3.1. For Bin-Dis, L_pDis and Bin-L_pDis methods, ordering of classes is ignored as in DISWOTH. Therefore, they can be categorized into the same group. However, for Com and ADM extensions, ordering of classes is a necessary information and the knowledge of the objective type of criteria is required (as maximization or minimization). Therefore, they are categorized as multicriteria methods.

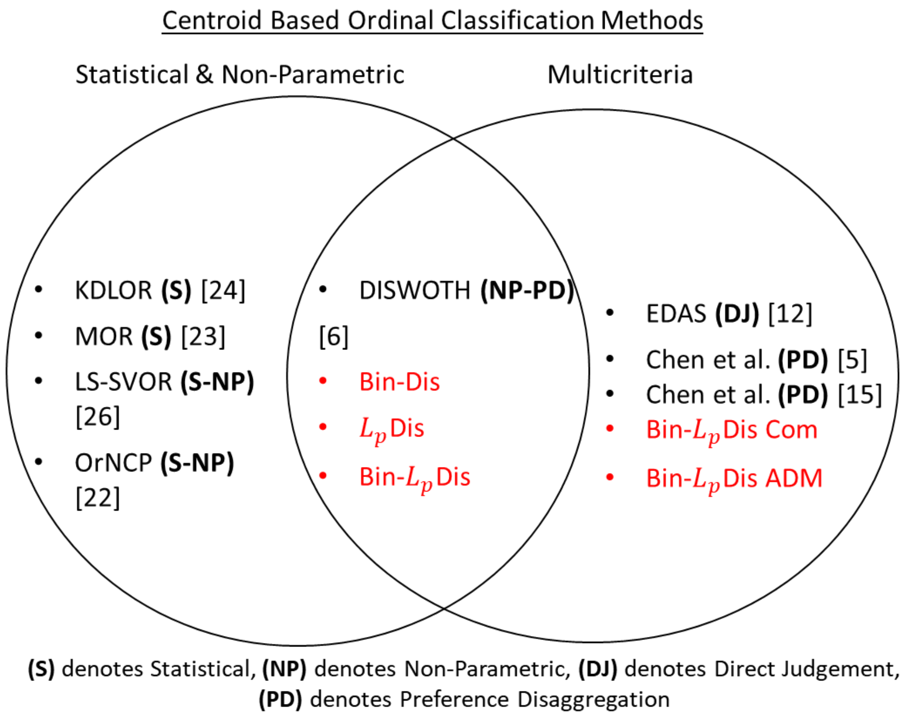


Figure 3.1 The categorization of proposed methods in the literature

The application procedure of the proposed methods is as follows.

Step 1: Solve the proposed model with the following inputs and the training data set:

1- A predetermined L_p distance
2- An estimated centroid
3- The training data

Obtain the outputs:

1- Training accuracy
2- Optimal criterion weights

Step 2: With optimal criterion weights, predetermined L_p distance, estimated centroid and test data, compute the test accuracy.

Obtain output:

1- Test accuracy

To compute the test accuracy, solve the proposed method with test data and optimal criterion weights obtained from step 1. Since the decision variable criterion weights are known, solving the model is not an optimization. Since all of the decision variables are known, it is a simple computation for error variables.

Formulations of training and test accuracy are explained in Chapter 5. In the next chapter, a specific case of DISWOTH is studied. Findings are reported.

CHAPTER 4

MONOTONICALLY ORDERED CENTROIDS CASE OF NEAREST CENTROID CLASSIFIER

This chapter presents the second study in this thesis. In this chapter, a specific case of DISWOTH is studied. It is proven that when the centroid estimations are monotonically ordered, there are redundant alternatives such that DISWOTH cannot change the class assignment of those alternatives.

In section 4.1, decision boundary characteristics of DISWOTH with monotonically ordered centroids is explained. Based on decision boundary characteristics, redundant alternatives are detected. It is shown that the conditions that satisfy the redundancy are linear expressions. In Section 4.2, redundancy conditions are formulated, and an LP model is developed for all L_p distances.

4.1 Theoretical Background of Redundancy Conditions

To analyze the decision boundary characteristics of DISWOTH, recall the equidistant point E presented in Section 3.1. The decision boundary of DISWOTH is given as equation (39) in Section 3.1. Let DBC denote Decision Boundary of Classification. DISWOTH with monotonic centroids is denoted as Monotonic NC.

$$\sum_j w_j^p\left(\left|E_j-\mu_j^q\right|^p-\left|E_j-\mu_j^r\right|^p\right)=0 \tag{39}$$

DBC with Euclidean distance function is exemplified in equations (58)-(59) and Figure 4.1. b denotes a positive scalar and a is a real numbered vector, $a \in R^m$.

$$\sum_j (\mu_j^q)^2-(\mu_j^r)^2-2E_j(\mu_j^r-\mu_j^q)=0 \tag{58}$$

$$a^T E+b=0 \tag{59}$$

$a_j = -2(\mu_j^r - \mu_j^q)$ and $b = \|\mu^q\|^2 - \|\mu^r\|^2$. Alternatively, $a_j = (\mu_j^r - \mu_j^q)$ and $b = -\frac{\|\mu^q\|^2 - \|\mu^r\|^2}{2}$. Equations (58)-(59) form a line between μ^q and μ^r.

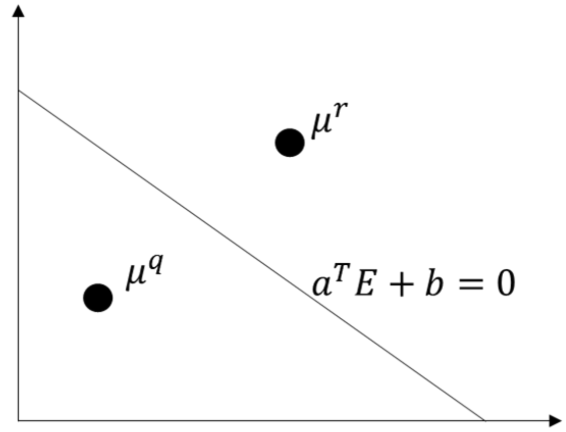


Figure 4.1 Decision boundary example of DISWOTH for Euclidean distance

The weighted distance with nonnegative weights rotates the DBC. The rotation is defined as a circular movement around a fixed point. The fixed point of rotation for the example DBC in Figure 4.1 can be found via formulations (60)-(65).

$$\sum_j \left|\mu_j^q - E_j\right|^2 - \left|\mu_j^r - E_j\right|^2 = \sum_j w_j \left|\mu_j^q - E_j\right|^2 - w_j \left|\mu_j^r - E_j\right|^2 \tag{60}$$

In equation (60), distance without weights and with weights are equated to find the midpoint of the rotation. It is simplified to equation (61).

$$\sum_j (1 - w_j)(\mu_j^r - \mu_j^q)(\mu_j^q + \mu_j^r - 2E_j) = 0 \tag{61}$$

$$\sum_j w_j = 1 \tag{62}$$

$$w_j \geq 0 \; \forall j \tag{63}$$

$$\mu_j^{q+1} \geq \mu_j^q \; \forall j \tag{64}$$

Constraint (64) is assumed for Monotonic NC. Equation (60) is simplified to equation (61). Regardless of values of w_j and $(\mu_j^r - \mu_j^q)$, the only condition that "*always*" satisfies the equation (61) is as follows.

$$E_j = \frac{\mu_j^q + \mu_j^r}{2} \tag{65}$$

Equation (65) is the fixed point of rotation and called midpoint in Euclidean geometry. Figure 4.2a and Figure 4.2b demonstrate four example rotations of DBC for four different weights.

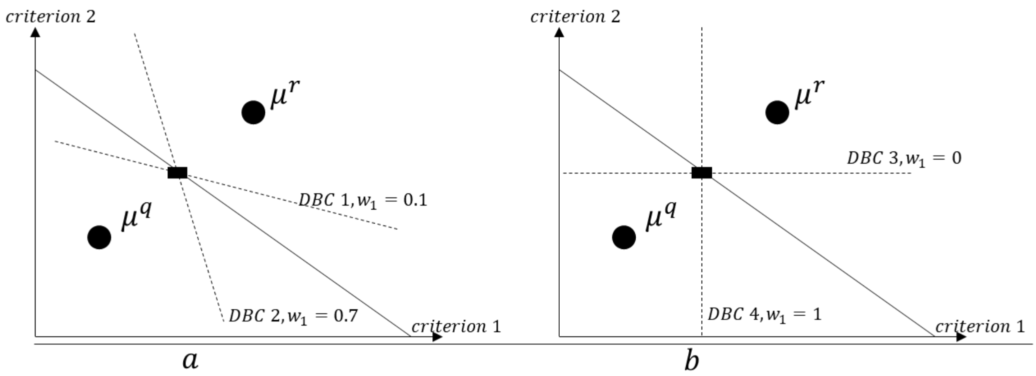


Figure 4.2 Example rotations of DBC

In Figure 4.2b where the extreme conditions on criterion weights are applied, there are regions that the DBC cannot reach. Those regions are shown in Figure 4.3.

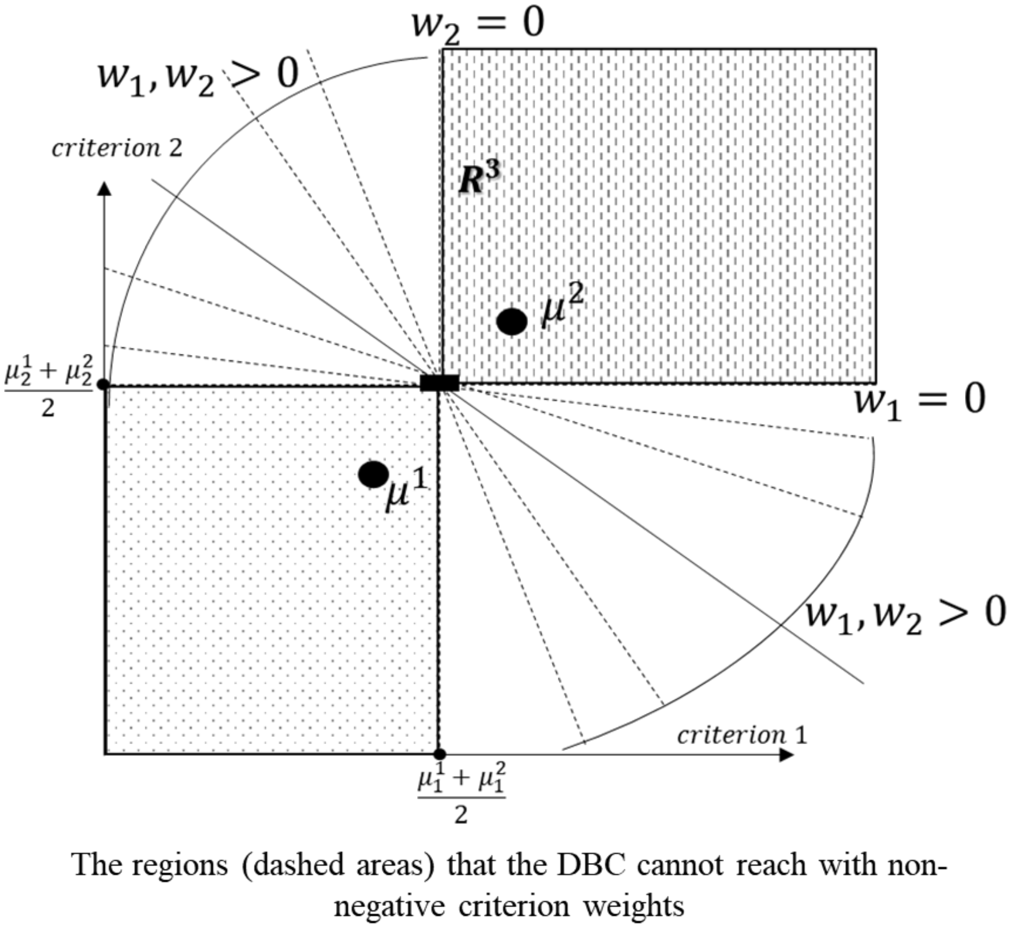


Figure 4.3 Illustration of regions that decision boundary of Monotonic NC cannot reach

Alternatives in those regions are out of class assignment initiative of Monotonic NC. Therefore, once the class centroids are determined, alternatives in dashed regions of Figure 4.3 are redundant for DISWOTH model if the centroids are in monotonic order. Until now, the equations and figures are used for exemplifying the rotations, redundancy regions and decision boundary characteristics. Theorems 1-3 provide formulations of redundancy regions.

Theorem 1: *An alternative A with criteria evaluations $\frac{\mu_j^q+\mu_j^{q-1}}{2} < A_j < \frac{\mu_j^q+\mu_j^{q+1}}{2}, \forall j$ is always assigned to class q and cannot be assigned to class $r \neq q$ by a Monotonic NC with non-negative weights and L_p distance.*

Assume there are two artificial centroids μ^{Q+1} and μ^0 such that $\mu^{Q+1} = \infty$ and $\mu^0 = -\infty$.

Proof: For $A_i \in C^q$ to be always true, (66) must hold.

$$\sum_j w_j |\mu_j^q - A_{ij}|^p - w_j |\mu_j^r - A_{ij}|^p < 0 \; \forall r \neq q \tag{66}$$

Rewrite (66) as (67)-(68):

$$\sum_j w_j |\mu_j^q - A_{ij}|^p - w_j |\mu_j^u - A_{ij}|^p < 0 \; \forall q < u \tag{67}$$

$$\sum_j w_j |\mu_j^q - A_{ij}|^p - w_j |\mu_j^l - A_{ij}|^p < 0 \; \forall q > l \tag{68}$$

(67) and (68) always hold in the case of $\frac{\mu_j^q + \mu_j^{q-1}}{2} < A_{ij} < \frac{\mu_j^q + \mu_j^{q+1}}{2}$. This is proven with (69)-(73). Superscripts u and l are not used for power operation. They are used for indexing. u and l are integers such that $u, l \in \{0,1,2,\ldots,Q,Q+1\}$. Due to constraint (64), $A_{ij} < \frac{\mu_j^q + \mu_j^{q+1}}{2}$ also satisfies $A_{ij} < \frac{\mu_j^q + \mu_j^u}{2}, \forall u > q$ and $A_{ij} > \frac{\mu_j^q + \mu_j^{q-1}}{2}$ satisfies $A_{ij} > \frac{\mu_j^q + \mu_j^l}{2}, \forall l < q$. $\mu_j^l < \mu_j^q < \mu_j^u$ is clear. Rewrite $A_{ij} > \frac{\mu_j^q + \mu_j^l}{2}$ and $A_{ij} < \frac{\mu_j^q + \mu_j^u}{2}$ as equations (69) and (70), respectively. β_j^u, β_j^l, x_j^u and x_j^l are positive values.

$$A_{ij} = \frac{\mu_j^q}{2} + \frac{\mu_j^u}{2} - \beta_j^u, \beta_j^u > 0. \tag{69}$$

$$A_{ij} = \frac{\mu_j^q}{2} + \frac{\mu_j^l}{2} + \beta_j^l, \beta_j^l > 0. \tag{70}$$

Due to (64), $\frac{\mu_j^{q+1}}{2} > \frac{\mu_j^q}{2}$. Rewrite (64) as:

$$\frac{\mu_j^u}{2} - \frac{\mu_j^q}{2} = x_j^u, \forall u > q \,,\; x_j^u > 0 \tag{71}$$

$$\frac{\mu_j^q}{2} - \frac{\mu_j^l}{2} = x_j^l, \forall l < q \,,\; x_j^l > 0 \tag{72}$$

Inequality (73) is used for proving Theorem 1.

$$w_j |a_j - b_j|^{\mathrm{p}} < w_j |-a_j - b_j|^p \text{ for } a_j, b_j > 0, w_j \geq 0 \; \forall j, p > 0 \tag{73}$$

(67) always holds due to (69) and (71) with the following formulation. $\sum_j w_j(\left|\frac{\mu_j^q}{2}-\frac{\mu_j^u}{2}+\beta_j^u\right|^p-\left|\frac{\mu_j^u}{2}-\frac{\mu_j^q}{2}+\beta_j^u\right|^p)=\sum_j w_j(\left|x_j^u+\beta_j^u\right|^p-\left|-x_j^u+\beta_j^u\right|^p)<0$ always holds because

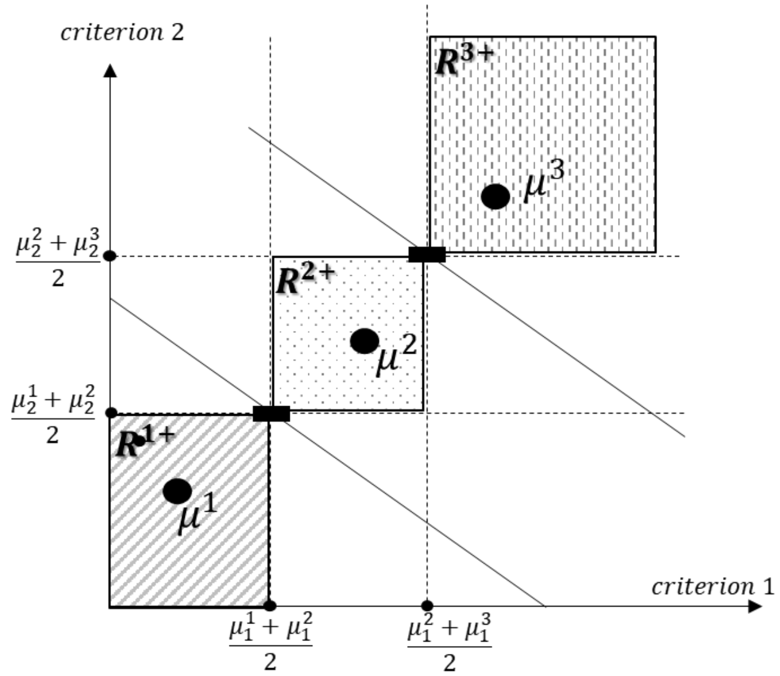
$\left|-x_j^u+\beta_j^u\right|^p-\left|-x_j^u-\beta_j^u\right|^p<0\ for\ x_j^u,\beta_j^u>0\ \forall j$ always holds due to (73).

(68) always holds due to (70) and (72) with the following formulation.

$\sum_j w_j(\left|\frac{\mu_j^q}{2}-\frac{\mu_j^l}{2}-\beta_j^l\right|^p-\left|\frac{\mu_j^l}{2}-\frac{\mu_j^q}{2}-\beta_j^l\right|^p)=\sum_j w_j(\left|x_j^l-\beta_j^l\right|^p-\left|-x_j^l-\beta_j^l\right|^p)<0$ always holds because $\left|x_j^l-\beta_j^l\right|^p-\left|-x_j^l-\beta_j^l\right|^p<0\ for\ x_j^l,\beta_j^l>0\ \forall j$ always holds due to (73).❑

Due to Theorem 1, by Monotonic NC, A_i^q is always accurately classified if $A_i^q\in R^{q+}$ where $R_j^{q+}=\left(\frac{\mu_j^q+\mu_j^{q-1}}{2},\ \frac{\mu_j^q+\mu_j^{q+1}}{2}\right)\forall j$. As seen from the formulation of R^{q+}, these redundancy regions are in shape of boxes.

R^{q+} regions are demonstrated for a three-class example in Figure 4.4.



Redundancy regions are in the form of boxes. Any point in R^{3+} can only be assigned to class three and cannot be assigned to other classes. Each of those boxes are separated by $\frac{\mu_j^q+\mu_j^{q-1}}{2}$ and $\frac{\mu_j^q+\mu_j^{q-1}}{2}$ for each criterion j.

Figure 4.4 A 3-class example of the redundancy regions

Another redundancy condition occurs depending on A_i^q and monotonic centroids as follows. A_i^q is always misclassified if $A_i^q \in R_l^{q-}$ or $A_i^q \in R_u^{q-}$ where $R_{lj}^{q-} = \left(-\infty, \frac{\mu_j^q+\mu_j^{q-1}}{2}\right)\ \forall j$ and $R_{uj}^{q-} = \left(\frac{\mu_j^q+\mu_j^{q+1}}{2}, \infty\right)\ \forall j$.

Theorem 2: *A_i^q cannot be accurately classified if $A_i^q \in R_l^{q-}$ and it is always classified to a class less than q $(C^r, r < q)$.*

Proof: (67) and (68) never holds for $A_i^q \in R_l^{q-}$ $\left(A_{ij}^q < \frac{\mu_j^q+\mu_j^{q-1}}{2}\right)$. Rewrite $A_{ij}^q < \frac{\mu_j^q+\mu_j^{q-1}}{2}$ as follows:

$$A_{ij}^q = \frac{\mu_j^q}{2} + \frac{\mu_j^{q-1}}{2} - \beta_j^{q-1}, \beta_j^{q-1} > 0 \tag{74}$$

Show that μ_j^{q-1} is always closer to A_{ij}^q than μ_j^q.

$\sum_j w_j(\left|\frac{\mu_j^q}{2} - \frac{\mu_j^{q-1}}{2} + \beta_j^{q-1}\right|^p - \left|\frac{\mu_j^{q-1}}{2} - \frac{\mu_j^q}{2} + \beta_j^{q-1}\right|^p) = \sum_j w_j \left(\left|x_j^{q-1} + \beta_j^{q-1}\right|^p - \left|-x_j^{q-1} + \beta_j^{q-1}\right|^p\right) > 0.$ $\left|x_j^{q-1} + \beta_j^{q-1}\right|^p - \left|-x_j^{q-1} + \beta_j^{q-1}\right|^p > 0, \quad for\ x_j^{q-1},$ $\beta_j^{q-1} > 0\ \forall j$ always holds due to (73). ❑

Theorem 3: *A_i^q cannot be accurately classified if $A_i^q \in R_u^{q-}$ and is always classified to a class greater than q.*

Proof: (67) and (68) never holds for $A_i^q \in R_u^{q-}$ ($A_{ij}^q > \frac{\mu_j^q + \mu_j^{q+1}}{2}$). Rewrite $A_{ij}^q > \frac{\mu_j^q + \mu_j^{q+1}}{2}$ as follows.

$$A_{ij}^q = \frac{\mu_j^q}{2} + \frac{\mu_j^{q+1}}{2} + \beta_j^{q+1}, \beta_j^{q+1} > 0 \tag{75}$$

Show that μ_j^{q+1} is always closer to A_{ij}^q than μ_j^q. Thus, NC cannot classify A_{ij}^q to class q.

$\sum_j w_j(\left|\frac{\mu_j^q}{2} - \frac{\mu_j^{q+1}}{2} - \beta_j^{q+1}\right|^p - \left|\frac{\mu_j^{q+1}}{2} - \frac{\mu_j^q}{2} - \beta_j^u\right|^p) = \sum_j w_j \left(\left|-x_j^{q+1} - x_j^{q+1}\right|^p - \left|x_j^{q+1} - \beta_j^{q+1}\right|^p\right) > 0.$ $\left|-x_j^{q+1} - \beta_j^{q+1}\right|^p - \left|x_j^{q+1} - \beta_j^{q+1}\right|^p > 0\ for\ x_j^{q+1}, B_j^{q+1} > 0\ \forall j$ always hold due to (73). ❑

Due to Theorems 2 and 3, and alternative A_i satisfying $A_i^q \in R_l^{q-}$ or $A_i^q \in R_u^{q-}$ cannot be accurately classified and is a redundant alternative.

Let us categorize alternative $A_i^q \in R^{q+}$ as Accurately Redundant (AR) and $A_i^q \in R_l^{q-}$ or $A_i^q \in R_u^{q-}$ as Inaccurately Redundant (IR). An example of R^{q+}, $R_l^{q-}$and R_u^{q-} regions is illustrated in Figure 4.5. Intuitively, a proposed method to maximize (minimize) the number of AR (IR) alternatives enlarges (diminishes) the $R^{q+}(R^{q-})$ area(s) in Figure 4.5.

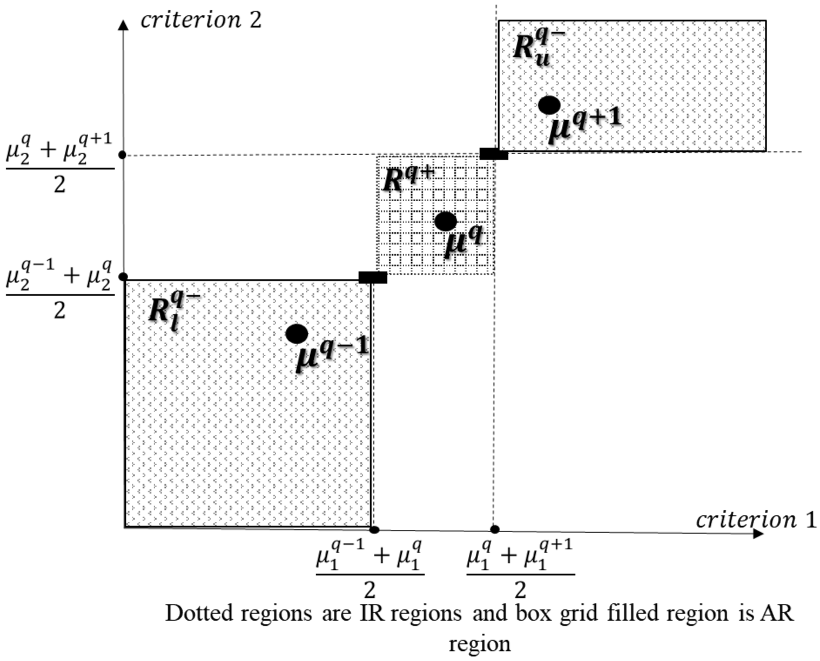


Figure 4.5 A representation of AR and IR regions.

To conclude this section; it is shown that when centroids are in monotonic order, there are redundancy regions. Those regions are classified into two as AR and IR regions. In the AR region of class q, any alternative belonging to class q (A_i^q) is always accurately classified. There is no positive weight set that can violate this condition. If an alternative is in the IR region of class q (R_l^- and R_u^-), it cannot be accurately classified and is always classified to some other class. There is no positive weight set that can violate this condition too. The formulations of redundancies are linear expressions.

Examples of AR and IR alternatives are illustrated in Figure 4.6. Small shapes are alternatives and large shapes are centroids. Triangles are from class one; black circles are from class two and white circles are from class three. Examples are illustrated for alternatives of class two. Alternative one from class two (A_1^2) and alternative four from class two (A_4^2) are IR alternatives since $A_1^2 \in R_u^{2-}$ and $A_4^2 \in R_l^{2-}$. Alternative three from class two is an AR alternative since $A_3^2 \in R^{2+}$. Alternative two from class two A_2^2 is not an AR or IR alternative. Class assignment of A_2^2 is based on the decision boundary of classification (DBC2). If criterion weights change, then it may be assigned to different classes. Therefore, it is not a redundant alternative.

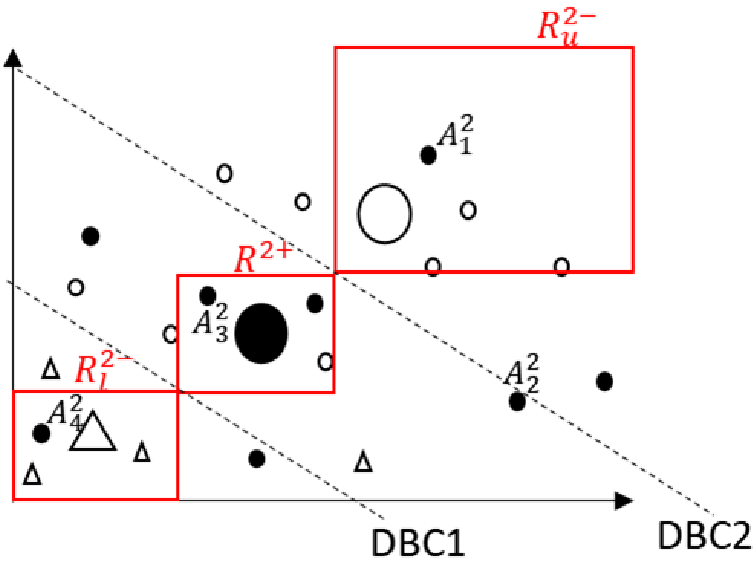


Figure 4.6 An example illustration for alternatives in AR and IR regions

4.2 Proposed Model

In this section, using Theorems 1, 2 and 3; an LP model is developed to maximize the accurate redundancies and minimize inaccurate redundancies. For formulation of R^{q+}, R_l^{q-} and R_u^{q-} in MP, new decision variables λ_{ij}^l and λ_{ij}^u are introduced. In the following set of constraints, $\lambda_{ij}^l, \lambda_{ij}^u = 0\ \forall j$ if $A_i^q \in R^{q+}$ which is desired to improve accuracy. Because $A_i^q \in R^{q+}$ cannot be misclassified. New notation is as follows.

ϕ_i is the weight of alternative i. This is used in the computation of centroids as a weighted sum of alternatives. ϕ_{min} is the minimum weight of alternatives. δ_j^q separates the class centroids. ϕ_{min} and δ_j^q are artificial variables to avoid trivial solutions. The explanation of trivial solutions are given and illustrated in Appendix H. Constraint formulation of R^{q+}, R_l^{q-} and R_u^{q-} is as follows.

$$A_{ij}^q + \lambda_{ij}^l \geq \frac{\mu_j^q + \mu_j^{q-1}}{2} + \epsilon\ \forall j,\ \forall i,\ \forall q > 1 \qquad (76)$$

$$A_{ij}^{q} - \lambda_{ij}^{u} \leq \frac{\mu_j^q + \mu_j^{q+1}}{2} - \epsilon \ \forall j,\ \forall i,\ \forall q < Q \tag{77}$$

$$\mu_j^{q+1} \geq \mu_j^q + \delta_j^q \ \forall j,\ \forall q < Q \tag{78}$$

$$\mu_j^q = \sum_i A_{ij}^q \phi_i \ \ \forall q, \forall j \tag{79}$$

$$\sum_{i \in C^q} \phi_i = 1 \ \forall q \tag{80}$$

$$\phi_{min}^q \leq \phi_i \ \forall i, \forall q \tag{81}$$

$$\lambda_{ij}^u, \lambda_{ij}^l \geq 0 \ \ \forall i, \forall j \tag{82}$$

$$\mu_j^q \geq 0 \ \forall q, \forall j \tag{83}$$

$$\phi_i \geq 0 \ \forall i \tag{84}$$

$$\phi_{min}^q \geq 0 \tag{85}$$

$$\delta_j^q \geq \epsilon \ \forall q \ \forall j \tag{86}$$

In constraint (76), it is checked whether $\lambda_{ij}^l > 0$ for any j. In this case, $A_i^q \in R^{q+}$ condition is violated. If $\lambda_{ij}^l > 0$ for all j then, $A_i^q \in R_l^{q-}$. The same applies to constraint (77) such that if $\lambda_{ij}^u > 0$ for any j then, $A_i^q \in R^{q+}$ condition is violated. If $\lambda_{ij}^u > 0$ for all j then, $A_i^q \in R_u^{q-}$. Constraint (78) orders centroids from the best to worst class that provides the monotonic centroids. δ_j^q in constraint (78) is maximized in the objective function to avoid trivial solutions and it is a strictly positive variable. Constraints (79)-(80) ensure that the centroid of class q is estimated as the convex combination of alternatives of class q. With these constraints it is guaranteed that the centroid is an interior point of the class that it represents. Constraint (81) is used to find the alternative that contributes the least in the computation of the centroids and is maximized in the objective. Constraints (82)-(86) are sign constraints. $\lambda_{ij}^u + \lambda_{ij}^l$ are minimized, δ_j^q and ϕ_{min} are maximized in the objective functions. Maximization of δ_j^q and ϕ_{min} in objective function does not serve the optimization of redundant alternatives. They are used to avoid trivial solutions. Therefore, they are regularization variables. These objective functions are formulated as equations (87)-(89). z_1 objective is to be minimized and z_2 and z_3 objectives are to be maximized. All of the objectives are scaled to [0,1] range in the objective function[1].

$$z_1 = \sum_q \sum_{i \in C^q} \sum_j \frac{\lambda_{ij}^u + \lambda_{ij}^l}{|C^q| m} \tag{87}$$

$$z_2 = \frac{\sum_q \sum_j \delta_j^q}{Qm} \tag{88}$$

$$z_3 = \frac{\sum_q \phi_{min}^q * |C^q|}{Q} \tag{89}$$

To obtain efficient solution from constraints (76)-(86) and objective functions (87)-(89), scalarizing function is used. As scalarizing function formulation, Augmented Tchebycheff [31] program is used. Accurate and Inaccurate Redundancies Optimization (AIRO) model with three-objectives is as follows.

(AIRO)

$$Minimize\ z_\infty + \rho(z_1 - z_1^{**} + z_2^{**} - z_2 + z_3^{**} - z_3) \tag{90}$$

Subject to:

$$z_\infty \geq V_1(z_1 - z_1^{**}) \tag{91}$$

$$z_\infty \geq \frac{(1-V_1)}{2}(z_2^{**} - z_2) \tag{92}$$

$$z_\infty \geq \frac{(1-V_1)}{2}(z_3^{**} - z_3) \tag{93}$$

Constraints (76)-(86)

Equations (87)-(89)

---

[1] z_1 is divided by $|C^q|m$ (class cardinality times number of criteria) to scale this objective to [0,1] range. Because the highest value that $\lambda_{ij}^u + \lambda_{ij}^l$ can take is 1 due to normalization of data. z_1 is divided by Qm because the largest value δ_j^q can take is 1. The highest value ϕ_{min}^q can take is $\frac{1}{|C^q|}$. Therefore, it is multiplied by $|C^q|$ and divided by Q.

z_1^{**}, z_2^{**} and z_3^{**} are predetermined parameters, namely utopian points. Utopian point is a point that is too good that it is impossible to achieve and formulated as $z_1^{**} = z_1^* - \epsilon$, $z_2^{**} = z_2^* + \epsilon$ and $z_3^{**} = z_3^* + \epsilon$. z_1^*, z_2^* and z_3^* denotes optimal values of z_1, z_2 and z_3. V_1 is used to define the projection direction of the closest efficient solution from the utopian point. It is defined by the user. The Augmented Tchebycheff program is explained in Chapter 6 (Section 6.1.2). AIRO model is presented for a sorting problem with maximization criteria. For minimization criteria, model is modified as follows:

Modify constraint (78) as (94).

$$\mu_j^{q+1} \geq \mu_j^q + \delta_j^q \ \forall j, \ \forall q < Q \tag{78}$$

$$\mu_j^{q+1} \leq \mu_j^q - \delta_j^q \ \forall j, \ \forall q < Q \tag{94}$$

Modify constraints (76)-(77) as (95)-(96)

$$A_{ij}^q + \lambda_{ij}^l \geq \frac{\mu_j^q + \mu_j^{q-1}}{2} + \epsilon \ \forall j, \ \forall i, \ \forall q > 1 \tag{76}$$

$$A_{ij}^q - \lambda_{ij}^u \leq \frac{\mu_j^q + \mu_j^{q+1}}{2} - \epsilon \ \forall j, \ \forall i, \ \forall q < Q \tag{77}$$

$$A_{ij}^q + \lambda_{ij}^l \geq \frac{\mu_j^q + \mu_j^{q+1}}{2} + \epsilon \ \forall j, \ \forall i, \ \forall q > 1 \tag{95}$$

$$A_{ij}^q - \lambda_{ij}^u \leq \frac{\mu_j^q + \mu_j^{q-1}}{2} - \epsilon \ \forall j, \ \forall i, \ \forall q < Q \tag{96}$$

No matter what criterion weights (as long as positive) and L_p distances are used, it is shown that the redundancy conditions are valid. To perform NC class assignment, a distance function and criterion weights are also needed.

To obtain and evaluate criterion weights, two different approaches are used. The first approach is developed based on the weight estimation LP in the study of Korhonen et al. [32]. The second approach is the equal weight case. This approach is named, AIRO-Equal Criterion Weights (AIRO-ECW) The second approach is applied to analyze whether the weight estimation LP improves the classification accuracy or not.

Instead of linear value function used by Korhonen et al. [32], distance-based criteria aggregation function is used to be consistent with NC method that is also the distance-based method. A new unrestricted in sign $(u.r.s)$, Θ^q, is introduced which takes negative value if the preference relationship between alternatives of two classes are violated. Weight Estimation (WE) LP model is as follows.

(WE)

$$Maximize\ \sum_q \Theta^q \tag{97}$$

Subject to:

$$\sum_j w_j \left|A_{ij}^q - I_j\right|^p - \sum_j w_j \left|A_{i'j}^{q+1} - I_j\right|^p \geq \Theta^q\ \forall q < Q, \forall i \in C^q, \forall i' \in C^{q+1}$$

(98)

$$\Theta^q urs \tag{99}$$

Constraints (22)-(23)

Referring to the relationship between alternatives and classes, alternatives should be consistent with the order of classes that is $A_i \in C^q$ and $A_{i'} \in C^{q+1}$, then $A_{i'} \gg A_i$ for $q < Q$.

Constraint (98) of WE measures the consistency of alternatives of ordered classes. It should be satisfied that the alternatives of a better class are preferrable to the alternatives of a worse class. Θ^q is maximized in the objective function (97) to satisfy consistency of alternatives of ordered classes. Constraint (99) is the sign constraint. When WE weights are applied to AIRO, it is named AIRO-WE.

Application procedure of the proposed method for monotonically ordered centroids is as follows.

AIRO Method

Step 1: Solve AIRO model for the training data and obtain monotonic centroids.

In solution of AIRO model, Augmented Tchebycheff program is used to avoid weakly efficient solutions. The objective weight, V_1 is determined by empirical study based on preliminary experiments. More explanation about the experiment setting of AIRO model is given in Section 5.3.

Step 2: Determine p value of L_p distance, solve WE model for training data and obtain the criterion weights.

Step 3: Using the test data, with the L_p distance used in Step 2 and the criterion weights obtained in step 2 and the centroids obtained in step 1 compute test accuracy by computing errors using DISWOTH model. Solving DISWOTH model is not an optimization since the criterion weights are inputs to the model.

Step 4: If more L_p distances are to be evaluated, change the L_p distance to be evaluated and return to step 2. Otherwise, terminate.

Step 5: Calculate test accuracy for all V values and L_p distances used in AIRO. Choose the best test accuracy among them.

In step 1 of the solution procedure, the values of objectives do not have any economic meaning for the decision maker. Therefore, it may be beneficial that a set of V_1 values are used in the step 1. For application of AIRO, different L_p distances are evaluated with step 4. The output of application procedure of AIRO is a single best test accuracy of a set of test accuracy outcomes of L_p distances with different p values.

Since AIRO is an LP model, different solutions for different L_p distances can be explored by solving many computationally inexpensive LP models.

Proposed methods are listed in Table 4.1.

Table 4.1 List of proposed methods

Proposed Method	**Developed Based on**	**Centroid Choice**	**Distance Function**	**Model Type**
L_pDis	DISWOTH and L_p −Centroid	L_p −Centroid ($\mu^q_{L_{p*}}$)	Weighted L_{p*}	NLP
Bin-Dis	DISWOTH	Arithmetic Average	Weighted L_p	MIP
Bin-L_pDis	L_pDis and Bin-Dis	L_p −Centroid ($\mu^q_{L_{p*}}$)	Weighted L_{p*}	MIP
Bin-L_pDis COM	Bin-L_pDis and Compromise Ranking	L_p −Centroid ($\mu^q_{L_{p*}}$)	Weighted L_{p*}	MIP
Bin-L_pDis ADM	Bin-L_pDis and ADM	L_p −Centroid ($\mu^q_{L_{p*}}$)	Weighted L_{p*}	MIP
AIRO	DISWOTH	Monotonic centroids based on AIRO	Any Weighted L_p	LP

CHAPTER 5

EXPERIMENTS OF DISWOTH EXTENSIONS AND MONOTONICALLY ORDERED CENTROIDS CASE

5.1 Experiment Settings

Performance of proposed methods are compared with DISWOTH [6] and UTADIS [3]. 5 intervals are used in each criterion of UTADIS. Each data set is partitioned into two as training and test data. Training data includes approximately 80 of whole data containing 80 of each class. Remaining 20 is taken as test data containing 20 of each class. All models are solved for training data, and the optimal criterion weights are used for the classification of test data. All the missing information in data sets are eliminated before experiments. For all data sets, whole data is normalized to $[0,1]$ range using criterion wise Min-Max Feature Scaling. Normalization procedure is given below. ϵ is chosen as 10^{-6} in all of the models. In preliminary experiments, it is observed that $\epsilon \geq 10^{-5}$ decreases the classification accuracy and values lower than 10^{-6} does not change the classification accuracy.

$$Normalized\ A_{ij} = \frac{A_{ij} - min_i\{A_{ij}\}}{max_i\{A_{ij}\} - min_i\{A_{ij}\}} \tag{100}$$

The experiments are conducted on 9 data sets from different application areas. They are obtained from UCI Machine Learning Repository [33], WEKA [34] and study of Amine Lazouni et al. [35]. Details about the data sets are explained in the next section.

5.1.1 Data Sets

The data sets are chosen from different application areas to show the applicability of the proposed methods to different areas. The application areas are automotive industry, health and medical area, employee selection, construction, and hardware performance. The data sets are as follows.

Automotive Industry

Car Evaluation Data Set (CAR*)*: This data set consists of 1728 rows (alternatives). There six categorical criteria and four ordinal classes. The criteria used for sorting and the criterion type are as follows.

Table 5.1 Details about CAR data set

Criteria	**Values**	**Type**
Buying Price	Low (1), Medium (2), High (3), Very High (4)	Minimization
Maintenance Cost	Low (1), Medium (2), High (3), Very High (4)	Minimization
Number of Doors	2 (1), 3 (2), 4 (3), 5 and more (4)	Maximization
Number of People	2 (1), 4 (2), 5 and more (3)	Maximization
Luggage Boot	Small (1), Medium (2), Big (3)	Maximization
Safety Score	Low (1), Medium (2), High (3)	Maximization
Car Acceptability* (Class labels)	Unacceptable (1), Acceptable (2), Good (3), Very good (4)	-

Automobile Fuel Consumption Miles/Gallon Data Set (AUTOMPG): AUTOMPG data set contains fuel consumption data of different automobile models in miles/gallon. Data set originally consists of 8 criteria each defining a different spec of automobiles. Car name is not predictive and not used in classification. Cylinder, displacement, horsepower, weight, acceleration, model year and origin criteria are used for classification. Acceleration, model year and origin are maximization criteria and others are minimization. There are 392 rows. MPG column is continuous and binarized from median, lower than median being class 1 and remaining being class 2. (Available at UCI Repository)

Table 5.2 Details about AUTOMPG data set

Criteria	**Values**	**Type**
Cylinder	4, 6, 8 Cylinders	Minimization
Displacement	Integer values varying between 68 and 455	Minimization
Horsepower	Integer values varying between 46 and 230	Minimization
Weight	Integer values varying between 1613 and 5140	Minimization
Acceleration	Values between 8 and 24.8	Maximization
Model Year	Years between 70 (1970) and 82 (1982)	Maximization
Origin	Integer values varying between 1 and 3	Maximization
Miles Per Gallon* (MPG) (Class labels)	Values between 9 and 46.6 (Binarized from median)	-

Health and Medical Areas

Breast Cancer Data Set (BC): BC data set contains 286 rows, 9 columns 2 classes. Breast and breast-quad columns are non-predictive, and they are excluded. Remaining 7 criteria are used for sorting. Age and menopause state criteria are minimization criteria and others are maximization. (Available at UCI Repository)

Table 5.3 Details about BC data set

Criteria	**Values**	**Type**
Age	Ordinal categorical data 10-19 (1), 20-29 (2), …, 60-69 (6)	Minimization
Menopause	It40 (0), ge40 (1), premeno (2)	Minimization
Tumor Size	Integer values varying between 1 and 11	Maximization
Inv Node	Integer values varying between 1 and 7	Maximization
Node Caps	Yes (0), No (1)	Maximization
Deg-Malign	1, 2, 3	Maximization
Irradiate	Yes (0), No (1)	Maximization
Recurrence* (Class labels)	1, 2	-

Mammographic Mass Data Set (MMG): MMG data set contains 961 different breast cancer screening information. All criteria are used for sorting and all of them are maximization type. Severity column is to be predicted.(Available at UCI Repository)

Table 5.4 Details about MMG data set

Criteria	**Values**	**Type**
BI-RADS Assessment	Ordinal categorical data between 1 and 5	Maximization
Age	Integer values varying between 18 and 96	Maximization
Shape	Integer values varying between 1 and 4	Maximization
Margin	Integer values varying between 1 and 5	Maximization
Density	Integer values varying between 1 and 4	Maximization
Severity* (Class Label)	1, 2	-

American Society of Anesthesiologists Scores Data Set (ASA): ASA data set [35] has 16 criteria and 898 rows divided into 2 classes. All 16 criteria are used for sorting. Bradycardia, cardiac steadiness, spo2 and hypoglycemia are maximization criteria and others are minimization criteria.

Table 5.5 Details about ASA data set

Criteria	Values	Type
Age	2 months to 105 years old	Minimization
Hypertension	No (0), Yes (1)	Minimization
Diabetes	No (0), Yes (1)	Minimization
Respiratory Failure	No (0), Yes (1)	Minimization
Hearth Failure	No (0), Yes (1)	Minimization
Brady Cardia (Hearth Rate in bpm)	Integer values between 58 and 70	Maximization
Tachycardia (Hearth Rate in bpm)	Integer values between 58 and 70	Minimization
Steadiness of Heart rate	No (0), Yes (1)	Maximization
Pacemaker	No (0), Yes (1)	Minimization
Atrioventricular Block	No (0), Yes (1)	Minimization
Left Ventricular Hypertrophy	No (0), Yes (1)	Minimization
Oxygen Saturation	Integer values between 43 and 100	Maximization

Table 5.5 Continued

Hypoglycemia (Glucose level as lower)	Values between 0.7 and 0.92	Maximization
Hyperglycemia (Glucose level as upper)	Values between 0.92 and 3.8	Minimization
Systole	Values between 9 and 20.5	Minimization
Diastole	Values between 5 and 13	Minimization
ASA Class* (Class Label)	1, 2	-

Employee Selection and Performance Evaluation

Employee Selection Data Set (ESL): ESL data set contains evaluations of expert psychologists about 488 applicants. Data set consists of 488 rows and 4 ordinal criteria. All criteria are maximization. Name of the criteria are not given by the donator of the data set. Applicants are evaluated by psychologists of a recruiting company in an ordinal scale from 1 to 9 points based on psychometric test results. Evaluations are binarized by dividing from 6, employees with larger than 6 points being class 2 and others 1. (Available at WEKA)

Table 5.6 Details about ESL data set

Criteria	Values	Type
Criteria 1	Ordinal categorical data between 1 and 9	Maximization
Criteria 2	Ordinal categorical data between 1 and 9	Maximization
Criteria 3	Ordinal categorical data between 1 and 8	Maximization
Criteria 4	Ordinal categorical data between 1 and 8	Maximization
Class Labels*	1, 2, …, 9 (Binarized by cutting from 6 points. 1-6 (1) and 7-9 (2))	-

Lecturer Evaluation Data Set (LEV): LEV data set contains evaluations of students about lecturers. LEV data set consists of 1000 rows and 4 criteria. All criteria are maximization. Name of the criteria are not provided by the donator. Each criterion is evaluated with categorical ordinal scores between 1 and 4. Outcome column is to be predicted having 5 ordinal integer values between 1 and 5. Values 4 and 5 are assumed to be class 2 while others are class 1. (Available at WEKA)

Table 5.7 Details about LEV data set

Criteria	**Values**	**Type**
Criteria 1	Ordinal categorical data between 1 and 4	Maximization
Criteria 2	Ordinal categorical data between 1 and 4	Maximization
Criteria 3	Ordinal categorical data between 1 and 4	Maximization
Criteria 4	Ordinal categorical data between 1 and 4	Maximization
Class Labels*	1, 2, …, 5 (Binarized by cutting from 3 points. 1-3 (1) and 4-5 (2))	-

Material Science and Construction

Concrete Compressive Strength Data Set (CCS): CCS data set contains information about different types of concrete where concrete compressive strength is the outcome. CCS data consists of 1030 rows and 8 predictive criteria. All 8 criteria are used for sorting. Fly ash, water, coarse aggregate, and fine aggregate are minimization criteria and others are maximization. Concrete Compressive Strength (in MPa) column is to be predicted. CCS column is binarized from median as classes 1 and 2. (Available at UCI Repository)

Table 5.8 Details about CCS data set

Criteria	Values	Type
Cement (component 1)(kg in a m^3 mixture)	Values between 102 and 540	Maximization
Blast Furnace Slag (component 2)(kg in a m^3 mixture)	Values between 0 and 359.4	Maximization
Fly Ash (component 3)(kg in a m^3 mixture)	Values between 0 and 200.1	Minimization
Water (component 4)(kg in a m^3 mixture)	Values between 121.8 and 247	Minimization
Superplasticizer (component 5)(kg in a m^3 mixture)	Values between 0 and 32.2	Maximization
Coarse Aggregate (component 6)(kg in a m^3 mixture)	Values between 801 and 1145	Minimization
Fine Aggregate (component 7)(kg in a m^3 mixture)	Values between 594 and 992.6	Minimization
Age (day)	Integer values between 1 and 365	Maximization
Concrete Compressive Strength (CCS: MPa, megapascals)* (Class Label)	Values between 2.3 and 82.6 (Binarized from median)	-

Hardware Performance

Computer Hardware Data Set (CPU): CPU data set contains information about different computer hardware and their estimated relative performance. CPU data originally consists of 209 rows, 9 criteria and 6 of which are predictive while other three criteria are non-predictive. Cycle time, min memory, max memory, cache memory, minimum channels and maximum channels criteria are used for sorting. Cycle time is minimization criterion and others are maximization. Estimated Relative Performance (ERP) attribute is to be predicted. ERP column is binarized from median as classes 1 and 2. (Available at UCI Repository)

Table 5.9 Details about CPU data set

Criteria	**Values**	**Type**
Machine Cycle Time in Nanoseconds	Integer values between 17 and 1500	Minimization
Minimum Main Memory in Kilobytes	Integer values between 64 and 32000	Maximization
Maximum Main Memory in Kilobytes	Integer values between 64 and 32000	Maximization
Cache Memory in Kilobytes	Integer values between 0 and 256	Maximization
Minimum Channels in Units	Integer values between 0 and 52	Maximization
Maximum Channels in Units	Integer values between 0 and 176	Maximization
Relative Performance (published)	Integer values between 6 and 1150	Minimization
Relative Performance (estimated)* (Class Label)	Integer values between 15 and 1238 (Binarized from median)	-

Discretization of continuous class columns are performed similar to [4], [36]. A summary related to data sets and their p^* values are given in Table 5.10.

Table 5.10 Brief information about data sets

Application Area	**Data Sets**	**No. Of Alternatives**	**No. Of Criteria**	**No. Of Classes**	$\boldsymbol{p^*}$ **value**[1]
Automotive Industry	CAR	1728	6	4	4.3
	AUTOMPG	392	7	2	0.2
Health and Medical	BC	286	7	2	1.4
	MMG	961	5	2	2.1
	ASA	898	16	2	2.6
HR	ESL	488	4	2	4.2
	LEV	1000	4	2	1.2
Construction	CCS	1030	8	2	1.5
Hardware Performance	CPU	209	6	2	0.7

[1]p^* values are obtained with DC algorithm.

Evaluations of criteria as maximization and minimization is an assumption. This assumption is based on Pearson Correlation Coefficient. The correlation coefficient between each criteria and the class label column is computed. If the resulting coefficient is positive, then the criteria is assumed to be a maximization type. Otherwise, it is assumed to be minimization type.

5.1.2 Hardware Setting and Performance Measures

To model the proposed methods, (General Algebraic Modeling System release 23.9. 5.) GAMS modeling language is used. To solve MIP and LP models, IBM ILOG CPLEX (version 12.4.0.1) is used, to solve NLP models, BARON Solver (version 11.5.2) is used. Solver settings are as follows: Nodetable limit (Nodlim=1E+9), time limit (Reslim=14400, 4 hours), optimality gap (Optcr=1E-9), integer tolerance (Epint=1E-9). The hardware setting is 8 GB RAM, Intel(R) Core (TM) i7-8550U @1.80GHz, Windows 10 PC.

Performance of models is evaluated using three performance measures, namely training accuracy, test accuracy and solution time. Training and test accuracies are the percentages of correct class assignments in training and test data, respectively. Solution time is the elapsed time of the solver. Further analyses on solution time and accuracy are conducted to determine the best method. A time versus test accuracy trade-off matrix is constructed to compare the methods in terms of both accuracy and time as a single measure. Also, TOPSIS [37] is used to rank the methods from best to worst based on time and test accuracy performance. Performance of the models are compared with UTADIS [3] and DISWOTH [6] with L_1, L_2, L_3, L_{p*} distances. Different from L_pDis, DISWOTH with L_{p*} denotes the DISWOTH model with L_{p*} distance as distance function and arithmetic average as centroid. DISWOTH with L_{p*} distance is compared with L_pDis to examine the effect of using L_p-Centroid, $\mu^q_{L_{p*}}$. Total accuracy ($Ac(.)$) is calculated for different models with equations (101)-(103). $\delta(.)$ is an indicator function which returns 1 if the condition in the parentheses is true and 0 if false. $\delta(.)$ is used for counting 0/1 loss for UTADIS, DISWOTH and L_pDis. n_r (n_s) represents number of alternatives in training (test) data. i_r (i_s) represent the alternatives in the training (test) data. $i_r \in \{1,2, \dots, n_r\}$ and $i_s \in \{1,2, \dots, n_s\}$.

For Bin-Dis, Bin-L_pDis, Bin-L_pDis COM and Bin-L_pDis ADM:

$$Ac(.) = (1 - \sum_{i_r} \frac{e_{i_r}}{n_r}) * 100 \tag{101}$$

For DISWOTH and L_pDis:

$$Ac(.) = (1 - \sum_{i_r} \frac{\delta(e_{i_r} > 0)}{n_r}) * 100 \tag{102}$$

For UTADIS:

$$Ac(.) = (1 - \sum_{i_r} \frac{\delta(e_{i_r}^{+} > 0 \; or \; e_{i_r}^{-} > 0)}{n_r}) * 100 \tag{103}$$

Computation of e_i and total accuracy is the same for both training and test data. For test accuracy calculations, replace i_r and n_r with i_s and n_s in equations (101)-(103). To compute the test accuracy same mathematical models are used with test data input and optimal criterion weights (and class thresholds for UTADIS). Accuracy is a larger-the-better type of performance measure.

$$\Delta(m1, m2) = \frac{Time(m1) - Time(m2)}{Ac(m1) - Ac(m2)} \tag{104}$$

$\Delta(m1, m2)$calculates average amount of seconds required to improve 1 accuracy given that accuracy of $m1$ is larger than that of $m2$. $\Delta(m1, m2) \in (-\infty, \infty)$ is a smaller-the-better type of performance measure. If $\Delta(m1, m2) \leq 0$, then $m1$ dominates $m2$ according to the specific experiments. The time vs. test accuracy trade-off is evaluated with the average time and average test accuracy. Average accuracy and the time are calculated using experimental results of 9 data sets for each method.

Solution time is the elapsed time. A time limit of 14400 seconds (4 hours) is established to solve each model. Time efficiency of the models is compared based on the time vs. test accuracy trade-off matrix using a trade-off variable $\Delta(m1, m2)$. $m1$ and $m2$ are the two models to be compared such that $Ac(m1) > Ac(m2)$. $\Delta(m1, m2)$ is calculated as follows.

$$\Delta(m1, m2) = \frac{Time(m1) - Time(m2)}{Ac(m1) - Ac(m2)} \tag{105}$$

$\Delta(m1, m2)$calculates average amount of seconds required to improve 1 accuracy given that accuracy of $m1$ is larger than that of $m2$. $\Delta(m1, m2) \in (-\infty, \infty)$ is a smaller-the-better type of performance measure. If $\Delta(m1, m2) \leq 0$, then $m1$ dominates $m2$ according to the specific experiments. The time vs. test accuracy trade-off is evaluated with the average time and average test accuracy. Average accuracy and the time are calculated using experimental results of 9 data sets for each method.

5.2 Experiments of DISWOTH Extensions

Training and test accuracy (in percentages) and solution time (in seconds) results of the experiments for 9 data sets and 5 methods (13 methods with different model inputs) are presented in Tables 5.11-5.14. L_pDis, Bin-Dis, Bin-L_pDis, Bin-L_pDis Com and Bin-L_pDis ADM are compared with UTADIS and DISWOTH. In Table 5.12, Mean Absolute Deviation (MAD) row is used to examine the tendency of the models to overfit to the given training data set. MAD between test and training accuracy is computed for each method as follows.

$$MAD(method) = \frac{1}{9}\sum_{t=1}^{9}|Training\ Accuracy(method,t) - Test\ Accuracy(method,t)| \qquad (106)$$

where t represents each data set.

DISWOTH with L_{p*} improves the average training accuracy DISWOTH with L_1, L_2 and L_3 by 2.65 , 1.57 and 2.35, respectively (see Table 5.11). Moreover, L_pDis improves the average training accuracy of DISWOTH with L_{p*} by 0.33. The improvement provided by $\mu^q_{L_{p*}}$ is not significant in the training accuracy.

Bin-Dis increases the training accuracy of DISWOTH for each data set as reported in Table 5.11. Bin-Dis improves the average training accuracy of the DISWOTH with L_1, L_2, L_3 and L_{p*} by 3.88, 3.51, 4.33 and 3.37, respectively. Bin-Dis improves the average training accuracy of the DISWOTH more than DISWOTH with L_{p*} and L_pDis. Bin-L_pDis and the two extensions give better training accuracy compared to UTADIS, DISWOTH and L_pDis methods. Bin-L_pDis improves training accuracy of L_pDis by 3.76. Bin-L_pDis method results in higher average training accuracy than other methods. In 4 out of 9 experiments, training accuracies of Bin-Dis with L_{p*} are the best among all methods. Bin-L_pDis method provides the best results in 5 out of 9 experiments. Note that the monotonicity restriction does not reduce the training accuracy since the models are primarily designed for seeking the alternative solution with the best accuracy.

Test results are reported in Table 5.12. DISWOTH with L_{p*} model improves the average test accuracy of DISWOTH with L_1, L_2 and L_3 by 3.15, 4.76 and 2.47, respectively. L_pDis improves the average test accuracy of DISWOTH with L_1, L_2, L_2 and L_{p*} by 9.37, 10.99, 8.70 and 6.23, respectively. But it improves training accuracy of DISWOTH with L_{p*} only by 0.33. L_{p*} with $\mu^q_{L_{p*}}$ yields a significant improvement on the average test accuracy compared to DISWOTH with L_{p*}.

Bin-Dis extension increases test accuracy of DISWOTH in 31 out of 36 experiments as given in Table 5.12. The remaining 5 exceptions are observed in the experiments of ESL, BC, and LEV data sets. Test accuracy of ESL data set decreases for Bin-Dis with L_1 and L_2 compared to DISWOTH with L_1 and L_2. Test accuracy of LEV data set decreases for Bin-Dis with L_2 and L_{p*} compared to DISWOTH with L_2 and L_{p*}. Test accuracy of BC data set decreases for Bin-Dis with L_3 compared to DISWOTH with L_3. Bin-Dis improves the average test accuracy of the DISWOTH with L_1, L_2, L_3 and L_{p*} by 10.09, 11.50, 8.37 and 8.39, respectively. Using binary variable instead of continuous error variable constitutes a significant improvement similar to using L_p-Centroid. Bin-L_pDis and its extensions gives better test accuracies than UTADIS and DISWOTH methods. Bin-Dis provides a better test accuracy only for CCS and CAR data sets compared to Bin-L_pDis.

Bin-L_pDis method improves average test accuracy of L_pDis by 4.19. It provides higher average test accuracy than other methods. In 5 out of 9 experiments, test accuracies of Bin-L_pDis method are the best among all methods. For all Bin-Dis methods, 8 out of 36 test accuracy results are the best among all. The best average test accuracy result is obtained with ADM extension. However, the average test accuracy improvement is only 0.12 when compared with Bin-L_pDis method that does not consider monotonicity.

Table 5.11 Training accuracy results in percentages

TRAINING													
	UTADIS	DISWOTH				Bin-Dis				L_pDis	Bin-L_pDis	Bin-L_pDis ADM	Bin-L_pDis COM
		L_1	L_2	L_3	L_{p^*}	L_1	L_2	L_3	L_{p^*}	L_{p^*}, μ_{Lp^*}			
AUTOMPG	80.71	89.71	89.07	88.42	90.03	**93.25**[2]	92.60	90.03	92.60	90.03	**93.25**	**93.25**	**93.25**
CPU	73.49	86.14	82.53	83.13	84.34	**89.16**	87.95	87.95	89.16	86.75	88.55	88.55	88.55
BC	54.71	70.85	69.06	69.06	68.61	**80.27**	72.20	70.40	76.68	68.16	78.03	78.03	78.03
ESL	85.71	93.11	93.62	93.37	93.88	93.37	**94.13**	**94.13**	**94.13**	93.62	**94.13**	**94.13**	**94.13**
CAR	56.11	61.53	73.68	79.90	82.28	66.31	77.87	83.88	**85.90**	82.65	85.68	85.68	85.68
CCS	62.39	82.22	81.86	76.42	84.04	85.85	88.39	86.82	88.03	86.46	**88.63**	**88.63**	**88.63**
LEV	66.96	75.94	74.44	69.45	75.81	80.92	80.17	81.80	79.55	79.30	**84.04**	**84.04**	**84.04**
ASA	96.52	92.77	96.38	96.80	96.80	95.82	98.47	**98.61**	**98.61**	95.69	98.33	98.33	98.33
MMG	84.33	84.33	85.67	82.69	84.63	**86.57**	86.12	84.63	86.12	80.75	**86.57**	**86.57**	**86.57**
Average[3]	73.44	81.85	82.92	82.14	84.49	85.72	86.44	86.47	87.86	84.82	**88.58**	**88.58**	**88.58**
APO	73.44	81.85	83.17	78.98	84.19	73.23	87.38	86.79	87.84	81.96	**88.99**	**88.99**	**88.99**
SD	14.45	10.61	9.30	9.69	8.68	16.70	8.39	7.99	6.89	8.35	**6.04**	**6.04**	**6.04**

[2] Best results are demonstrated in boldface.

[3] Average row represents the arithmetic average of all results, APO row represents Average of Proven Optimal results within 4 hours limit, SD row represents the Standard Deviation of results.

Generalization of the models are measured with MAD given in Table 5.12. For all DISWOTH models, MAD measure is above 11. MAD significantly decreases to the range between 6.134-8.14 for Bin-Dis and L_pDis methods. MAD is decreased to 5.39 for Bin-L_pDis. Compromise ranking extension increases MAD whereas ADM extension improves MAD by only 0.12. The monotonicity extensions do not provide a significant improvement on MAD for Bin-L_pDis model.

Robustness of the proposed models w.r.t different L_p distances can be examined in Tables 5.11 and 5.12. In training results of Bin-Dis, the worst average accuracy is observed with L_1 as 85.72 and best average accuracy is observed with L_{p*} as 87.76. In test results, the worst average accuracy is observed with L_3 as 78.67 and the best average accuracy is observed with L_{p*} as 81.16. The range of average accuracy for Bin-Dis is less than 2.5 for different L_p distances in both training and test results. These results show that Bin-Dis is a robust method on accuracy performance measure w.r.t different L_p distances. Similarly for Bin-L_pDis and its extensions, the range of test accuracy (between 83.07 and 83.31) is less than 0.25 although different monotonicity constraints are used.

Table 5.12 Test accuracy results in percentages

TEST													
	UTADIS	DISWOTH				Bin-Dis				L_pDis	Bin-L_pDis	Bin-L_pDis ADM	Bin-L_pDis COM
		L_1	L_2	L_3	L_{p*}	L_1	L_2	L_3	L_{p*}	L_{p*}, μ_{Lp*}			
AUTOMPG	77.78	79.01	80.25	76.54	80.25	**91.36**[4]	86.42	88.89	87.65	80.25	**91.36**	**91.36**	**91.36**
CPU	76.74	81.4	81.4	79.07	81.4	83.72	**86.05**	83.72	83.72	81.4	**86.05**	**86.05**	**86.05**
BC	50	62.07	46.55	48.28	58.62	70.69	58.62	44.83	68.97	65.52	**75.86**	74.14	74.14
ESL	79.17	87.5	90.63	89.58	90.63	86.46	87.5	91.67	**92.71**	90.63	90.63	**92.71**	90.63
CAR	57.39	61.74	60.87	79.42	82.61	64.06	76.23	82.9	**84.93**	84.06	84.06	84.35	83.77
CCS	37.44	48.28	56.16	59.11	61.08	66.5	**74.38**	59.61	62.07	67.49	63.05	64.04	64.04
LEV	68.69	79.8	74.75	68.69	78.28	82.83	73.23	**83.33**	77.27	77.27	**83.33**	**83.33**	**83.33**
ASA	93.85	**94.97**	92.18	93.3	92.74	**94.97**	94.41	**94.97**	94.41	94.41	**94.97**	94.41	**94.97**
MMG	76.25	31.88	29.38	38.75	29.38	76.88	78.75	78.13	78.75	70	**79.38**	**79.38**	**79.38**
Average	68.59	69.63	68.02	70.3	72.78	79.72	79.51	78.67	81.16	79	83.19	**83.31**	83.07
APO	68.59	69.63	69.54	73.01	72.46	79.72	80.71	80.79	83.55	78.7	**85.94**	85.91	85.69
SD	17.32	20.29	21.26	18.4	19.98	10.89	10.55	16.25	10.64	10.02	9.64	9.72	**9.57**
MAD[5]	6.24	13.61	14.98	11.83	12.34	6.43	6.92	8.14	6.7	6.13	5.39	**5.27**	5.51

[4]Best results are demonstrated in boldface.

[5]MAD row represents the Mean of Absolute Deviations between test accuracy and training accuracy.

Solution times are reported in Table 5.13. Bin-Dis models improve the average solution time of DISWOTH with L_2, L_3 and L_{p*} distances by 2494.61 seconds (43.57), 5010.20 seconds (59.92) and 5746.41 seconds (77.15), respectively. The improvement on L_pDis model provided with linearization (Bin-L_pDis) is 4890.31 seconds (60.39). Bin-Dis with L_1 performs worse than DISWOTH with L_1 by 686.381 seconds. The result is expected since Bin-Dis with L_1 is an MIP model while DISWOTH with L_1 is an LP model.

CCS data set solutions are not solved within time limit. Thus, they are not proven optimal for both NLP and MIP models except for Bin-Dis with L_1. CAR data set is solved with proven optimal solution with all NLP and LP models whereas a proven optimal solution is provided by only Bin-Dis with L_1 and L_{p*} among the MIP models. Solution time of Bin-Dis with L_1 is the shortest of all NLP and MIP models on average.

UTADIS and DISWOTH with L_1 are LP models. These LP models are solved within the shortest solution times when compared with MIP (Bin-Dis and Bin-L_pDis and its extensions) and NLP (DISWOTH with L_2, L_3 and L_{p*}) models as expected.

Solution times of Bin-Dis models are at least as short as their NLP versions for 23 out of 27 experiments. The remaining four exceptions are observed in the experiments of MMG and CAR data sets. The solution times of DISWOTH with L_2 and L_{p*} increase with binary variable adjustment for the MMG data set. The solution times of DISWOTH with L_2 and L_3 increase with binary variable adjustment for the CAR data set. Respecting the monotonicity of centroids improves the solution time less than 1 second on the average for both ADM and compromise ranking extensions. There are 11 experiments that are not solved within time limit for NLP DISWOTH models while there are 5 such experiments for Bin-Dis. For instance, DISWOTH with L_2, L_3 and L_{p*} are not optimally solved on ASA data set while Bin-Dis is solved. There are 5 experiments for L_pDis that are not optimally solved in time limit while there are only two for Bin-L_pDis, Bin-L_pDis COM and Bin-L_pDis ADM. Although time improvement is not guaranteed in conversion of NLP to MIP models, solution times significantly improve on average.

The solution time improvements in the proven optimal experiments of NLP models are more significant. Bin-Dis models improve the APO solution times of DISWOTH with L_2, L_3 and L_{p*}distances by 1349.52 seconds (96.73), 3335.97 seconds (94.50) and 1772.12 seconds (93.94), respectively. The improvement for proven optimal solutions of Bin-L_pDis on L_pDis model provided with linearization is 95.66 with 210.36 seconds. On the proven optimal solutions, linearization with binary variables gives a better average solution time performance with improvement larger than 90.

Table 5.13 Solution time results in seconds

SOLUTION TIME													
	UTADIS	DISWOTH				Bin-Dis				L_pDis	Bin-L_pDis	Bin-L_pDis ADM	Bin-L_pDis COM
		L_1	L_2	L_3	L_{p^*}	L_1	L_2	L_3	L_{p^*}	L_{p^*}, μ_{Lp^*}			
AUTOMPG	0.12	**0.10**	2478.56	-[6]	-	1.07	0.54	1.12	1.45	-	1.09	1.09	1.00
CPU	0.11	**0.07**	59.61	197.65	39.55	0.27	0.37	0.26	0.27	12.47	0.24	0.35	0.24
BC	0.10	**0.08**	-	10759.28	-	0.53	25.42	65.75	16.46	858.13	3.75	2.21	3.09
ESL	0.17	**0.09**	5.19	9.10	5.09	0.31	0.32	0.32	0.21	6.67	0.33	0.33	0.22
CAR	0.61	**0.19**	5767.61	6640.97	9350.85	4378.27	-	-	606.86	-	-	-	-
CCS	0.87	**0.10**	-	-	-	1302.11	-	-	-	-	-	-	-
LEV	0.21	**0.09**	38.60	42.93	6.98	0.61	0.80	0.56	1.10	2.33	0.89	0.84	0.90
ASA	0.53	**0.13**	-	-	-	494.55	3.93	4.55	3.67	-	9.01	14.41	13.24
MMG	0.19	**0.12**	20.99	-	29.70	0.67	287.67	1285.56	284.50	-	52.46	41.49	39.90
Average	0.32	**0.11**	5730.06	8361.10	7448.02	686.49	3235.45	3350.90	1701.61	8097.73	3207.42	3206.75	3206.51
APO	0.32	**0.11**	1395.09	3529.98	1886.43	686.49	45.58	194.02	114.31	219.90	9.54	8.67	8.37
SD	0.28	**0.04**	6763.80	6709.82	7225.17	1452.06	6330.39	6278.14	4766.42	7478.11	6345.62	6345.99	6346.13

[6] dash (-) signs represent the solutions that are not proven optimal within 14400 secs. solution time limit.

To compare the methods based on the solution time and accuracy together, a trade-off matrix is used. Trade-off matrix results are reported in Table 5.14. If accuracy of method in column is larger than that of the method in row, the time (in secs.) required to get 1 more test accuracy on average is given in Table 5.14. Bin-Dis with $L_1$and L_{p*}, Bin-L_pDis and Bin-L_pDis COM dominate six of the methods according to the average results of these specific experiments. DISWOTH and Bin-Dis with L_1, Bin-Dis with L_{p*} and Bin-L_pDis ADM are not dominated. Bin-L_pDis ADM is non-dominated and dominates eight of the methods. Other positive entries indicate that there is a time-accuracy trade-off between the methods according to the average results of the experiments.

To rank the methods, TOPSIS [37] is used. TOPSIS is a multi-criteria ranking method used for ranking alternatives from best to worst. All methods are ranked for each data set considering time and test accuracy criteria. Then average ranks for 9 data sets are evaluated. W_{Ac} and W_{Time} denote the weights of test accuracy and model solution time, respectively. TOPSIS is applied with 7 different TOPSIS distances $(L_1, L_2, L_3, L_4, L_5, L_{100}, L_\infty)$ and three different performance measure weights such that $W_{Time} \in \{0.1, 0.5, 0.9\}$ and $W_{Ac} = 1 - W_{Time}$. Since TOPSIS is a distance-based ranking method, to avoid scaling effect, performance measures are scaled to [0,1] range. To analyze whether a solution time favoring or a test accuracy favoring method is ranked better, L_∞ distance is included in the analyses since as p of L_p distance increases, larger weighted differences become more dominant [6]. The analysis provides intuitions about methods such that when L_∞ distance is used and weight of time is higher ($W_{Time} = 0.9$ and $W_{Ac} = 0.1$), a time effective method should be ranked better. When the weight of test accuracy is higher ($W_{Time} = 0.1$ and $W_{Ac} = 0.9$), an accuracy effective method should be ranked better. Ranks of the methods for different TOPSIS distance functions and weights are reported in Table 5.15. For all TOPSIS distances and criterion weights, the Bin-L_pDis, Bin-L_pDis COM and Bin-L_pDis ADM are in the best three methods. Bin-Dis with L_{p*} is ranked as the fourth best method. DISWOTH with L_3 is ranked the fifth based on average ranks of 7 distance functions. To summarize, Bin-L_pDis is shown to be both time and accuracy favoring method among all methods.

Table 5.14 The trade-off matrix

		UTADIS	DISWOTH				Bin-Dis				L_pDis	Bin-L_pDis	Bin-L_pDis COM	Bin-L_pDis ADM	No. of negatives[8]
			L_1	L_2	L_3	L_{p*}	L_1	L_2	L_3	L_{p*}					
UTADIS		-	**-0.2**[1]	-	4877	1780	62	296	332	135	778	220	221	218	1
DISWOTH	L_1	-	-	-	12332	2365	68	327	370	147	864	237	238	234	0
	L_2	**-99897**	**-3559**	-	1150	361	**-431**	**-217**	**-223**	**-306**	216	**-166**	**-168**	**-165**	9
	L_3	-	-	-	-	**-370**	**-815**	**-557**	**-599**	**-613**	**-30**	**-400**	**-404**	**-396**	9
	L_{p*}	-	-	-	-	-	**-974**	**-625**	**-695**	**-685**	104	**-407**	**-412**	**-403**	7
Bin-Dis	L_1	-	-	-	-	-	-	-	-	702	-	727	751	702	0
	L_2	-	-	-	-	-	**-12308**	-	-	-928	-	**-8**	**-8**	**-8**	5
	L_3	-	-	-	-	-	**-2545**	**-137**	-	**-661**	14349	**-32**	**-33**	**-31**	6
	L_{p*}	-	-	-	-	-	-	-	-	-	-	745	789	703	0
L_pDis	L_{p*}	-	-	-	-	-	**-10347**	**-9549**	-	**-2957**	-	**-1168**	**-1201**	**-1136**	6
Bin-L_pDis	L_{p*}, μ_{L_p}	-	-	-	-	-	-	-	-	-	-	-	-	**-8**	1
Bin-L_pDis COM		-	-	-	-	-	-	-	-	-	-	6	-	**-1**	1
Bin-L_pDis ADM		-	-	-	-	-	-	-	-	-	-	-	-	-	0
No. of negatives[9]		1	2	0	0	1	6	5	3	6	1	6	6	8	

[7]Negative entries in cells represent that the method in column dominates the method in row. For instance, DISWOTH with L_2 is dominated by UTADIS. Values are rounded to nearest integer except for **-0.2** value under DISWOTH with L_1.

[8]Number of negatives in the last column represents the number of methods that dominates the method in the row. For instance, UTADIS is dominated by only 1 method.

[9]Number of negatives in the last row represent the number of methods dominated by the method in the column. For instance, UTADIS dominates only 1 method.

Table 5.15 Average TOPSIS ranks

	UTADIS	DISWOTH				Bin-Dis				LpDis	Bin-LpDis	Bin-LpDis Com	Bin-LpDis ADM
		L_1	L_2	L_3	L_{p*}	L_1	L_2	L_3	L_{p*}	L_{p*}, μ_{Lp*}			
	W_{Time}=0.1 W_{Ac}=0.9												
L_1	11.22	8.11	10.89	10.89	8.67	6.00	7.00	5.56	4.89	7.67	3.44	3.33	**3.00**
L_2	11.22	8.11	10.78	10.67	8.56	6.33	6.89	5.56	4.78	7.78	3.56	3.44	**3.00**
L_3	11.00	7.78	10.89	10.67	8.78	6.33	7.00	5.67	4.78	7.78	3.56	3.44	**3.00**
L_4	10.78	7.33	11.00	10.67	9.00	6.44	7.11	5.56	4.89	7.78	3.67	3.44	**3.00**
L_5	10.33	6.44	11.00	10.78	9.11	6.33	7.33	5.78	4.89	8.00	3.78	3.67	**3.22**
L_{100}	8.67	5.89	10.44	11.00	9.33	5.67	7.67	6.11	5.00	8.78	4.00	4.11	**3.67**
L_∞	11.44	8.56	10.00	10.67	8.33	5.78	6.44	6.00	4.89	6.56	2.44	2.67	**2.22**
Average	10.67	7.46	10.71	10.76	8.83	6.13	7.06	5.75	4.87	7.76	3.49	3.44	**3.02**
SD	0.95	0.97	0.37	0.13	0.34	0.31	0.38	0.23	**0.08**	0.65	0.50	0.43	0.43
RANK	11	8	12	13	10	6	7	5	4	9	3	2	**1**

Table 5.15 Continued

	UTADIS	DISWOTH				Bin-Dis				LpDis	Bin-LpDis	Bin-LpDis Com	Bin-LpDis ADM
		L_1	L_2	L_3	L_{p*}	L_1	L_2	L_3	L_{p*}	L_{p*}, μ_{Lp*}			
	W_{Time}=0.5 W_{Ac}=0.5												
L_1	8.56	5.78	10.44	10.89	9.44	5.67	7.67	6.33	5.00	8.89	4.22	4.11	**3.67**
L_2	8.67	5.89	10.44	11.00	9.33	5.67	7.67	6.22	5.00	8.78	4.22	4.11	**3.67**
L_3	8.67	5.89	10.44	11.00	9.33	5.78	7.67	6.11	5.00	8.78	4.22	4.11	**3.67**
L_4	8.67	5.89	10.44	11.00	9.33	5.78	7.67	6.11	5.00	8.78	4.22	4.11	**3.67**
L_5	8.67	5.89	10.44	11.00	9.33	5.78	7.67	6.11	5.00	8.78	4.22	4.11	**3.67**
L_{100}	8.67	5.89	10.44	11.00	9.33	5.67	7.67	6.11	5.00	8.78	4.00	4.11	**3.67**
L_∞	8.67	5.89	10.44	11.00	9.33	5.67	7.67	6.22	5.00	8.78	4.22	4.11	**3.67**
Average	8.65	5.87	10.44	10.98	9.35	5.71	7.67	6.17	5.00	8.79	4.19	4.11	**3.67**
SD	0.04	0.04	**0.00**	0.04	0.04	0.06	**0.00**	0.09	**0.00**	0.04	0.08	**0.00**	**0.00**
RANK	9	6	12	13	11	5	8	7	4	10	3	2	**1**

Table 5.15 Continued

	UTADIS	DISWOTH				Bin-Dis				LpDis	Bin-LpDis	Bin-LpDis Com	Bin-LpDis ADM
		L_1	L_2	L_3	L_{p*}	L_1	L_2	L_3	L_{p*}	L_{p*}, μ_{Lp*}			
	W_{Time}=0.5 W_{Ac}=0.5												
L_1	8.00	5.44	10.22	11.22	9.44	5.56	7.67	6.56	5.33	9.22	4.22	4.11	**3.67**
L_2	8.56	5.78	10.22	11.11	9.56	5.56	7.67	6.33	5.00	8.89	4.22	4.11	**3.67**
L_3	8.56	5.89	10.22	11.11	9.56	5.56	7.67	6.33	5.00	8.78	4.22	4.11	**3.67**
L_4	8.56	5.89	10.22	11.11	9.56	5.56	7.67	6.33	5.00	8.78	4.22	4.11	**3.67**
L_5	8.67	5.89	10.22	11.11	9.56	5.56	7.67	6.22	5.00	8.78	4.22	4.11	**3.67**
L_{100}	8.67	5.89	10.44	11.00	9.33	5.67	7.67	6.11	5.00	8.78	4.00	4.11	**3.67**
L_∞	8.44	5.67	10.22	11.11	9.56	5.78	7.44	6.33	4.89	9.00	4.22	4.44	**3.56**
Average	8.49	5.78	10.25	11.11	9.51	5.60	7.63	6.32	5.03	8.89	4.19	4.16	**3.65**
SD	0.23	0.17	0.08	0.06	0.09	0.09	0.08	0.13	0.14	0.17	0.08	0.13	**0.04**
RANK	9	6	12	13	11	5	8	7	4	10	3	2	**1**

5.3 Experiments of Monotonically Ordered Centroids Case

The data sets, training-test partitioning and the hardware setting used for AIRO, and WE models are the same as in the experiments of UTADIS, DISWOTH in Section 5.1. The assumptions on the criteria, normalization and the ICV are also the same.

The objective weights of AIRO model are determined by empirical study. To examine the change in the test accuracy w.r.t changing V_1 values, 5 different values are used such that $V_1 \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ based on preeliminary experiments. To evaluate whether WE model estimates weights that improve classification accuracy or not, the equal criterion weights (AIRO-ECW) case is also evaluated. The distance functions used in the experiments are the same as in experiments of DISWOTH.

Accuracy calculation of AIRO model is the same as in DISWOTH. To ease the comparison, best test accuracy obtained by the AIRO and DISWOTH models are reported in Table 5.16. Also, UTADIS results are presented in the table. Detailed results are reported in Appendix B. Test accuracy of AIRO-WE and AIRO-ECW are reported in Appendix B, Tables 10.1-10.9.

Table 5.16 Comparison of AIRO with DISWOTH and UTADIS

	UTADIS	Best of DISWOTH	AIRO-WE	AIRO-ECW
AUTOMPG	77.78	80.25	91.36	92.59
CPU	76.74	81.40	90.70	93.02
BC	50.00	62.07	62.07	67.24
ESL	79.17	90.63	84.38	87.50
CAR	57.39	82.61	46.38	60.00
CCS	37.44	61.08	66.01	59.11
LEV	68.69	79.80	80.30	72.73
ASA	93.85	94.97	71.51	81.01
MMG	76.25	38.75	76.88	82.50
Average	68.59	74.62	74.40	77.30
SD	17.32	17.56	14.62	13.12

In Table 5.16, AIRO-ECW results are better than UTADIS in 8 out of 9 experiments. When it is compared with DISWOTH, 4 out of 9 experiments are better than DISWOTH. AIRO-WE results are better than UTADIS in 7 out of 9 experiments. When it is compared with DISWOTH, 6 out of 9 results are better than DISWOTH. In comparison of number of better/worse accuracy results, there is not advantage of AIRO-WE over AIRO-ECW. Both are advantageous when compared with UTADIS as the majority of experiment results are better.

The average test accuracy does not improve significantly when DISWOTH results are compared with AIRO-WE. This result can be interpreted as that criterion weights obtained by WE model do not improve the classification accuracy. This is consistent with findings of [38] which makes the results intuitive. Average test accuracy of AIRO-WE is 0.22 worse than DISWOTH and 5.81 better than UTADIS. Average test accuracy of AIRO-ECW is 2.68 better than DISWOTH and 8.82 better than UTADIS. The improvement obtained by considering the monotonicity does not bring a significant improvement of accuracy which is an intuitive result that is also reported by [21].

Solution time results are reported in Table 5.17. The time performance in this section is the total time to solve all models for all selected distance measures. For DISWOTH, it is the sum of solution times of DISWOTH with four distance functions $p \in \{1, 2, 3, p^*\}$. For AIRO, it is the sum of solution times of four distance functions and the solution time of WE model. With 5 different V_1 values and four distance functions, solution time reported for AIRO is the sum of 20 solution times. Solution time reported for AIRO-ECW is the solution time of AIRO-WE minus solution time of WE model.

Table 5.17 Total solution time comparison

	UTADIS	Total solution time of DISWOTH	AIRO-WE	AIRO-ECW
AUTOMPG	0.12	30478.66	5.65	1.27
CPU	0.11	296.88	2.06	0.94
BC	0.10	38759.35	2.92	0.87
ESL	0.17	19.47	3.89	0.98
CAR	0.61	21759.61	122.53	1.66
CCS	0.87	42000.10	426.03	2.05
LEV	0.21	88.60	16.00	0.70
ASA	0.53	42000.13	193.90	4.50
MMG	0.19	14050.81	16.56	0.91
Average	0.32	21050.40	87.73	1.54
SD	0.28	18173.41	143.80	1.19

UTADIS solution time is the shortest among all. In Table 5.17, it can be observed that addition of WE model to the AIRO significantly increases the solution time. Instead, simply using equal criterion weights is more efficient than adding the WE model to classification. As a result of comparison with DISWOTH, solving a limited set of LP models (twenty models for AIRO-ECW and twenty four models for AIRO-WE) take shorter time than solving NLP models with the same data size. The same comment is also valid for solution time comparison of UTADIS (solving one LP model) and AIRO.

Trade-off matrix used in Section 5.1 is also constructed and evaluated in this section. Trade-off matrix is tabulated and reported in Table 5.18. According to trade-off matrix, AIRO-ECW dominate DISWOTH and AIRO-WE. Although AIRO-ECW dominate UTADIS, only 0.14 seconds are required to obtain 1 more accuracy with AIRO-ECW. 0.14 seconds can be considered as a negligible difference for 1 test accuracy. According to trade-off matrix, there is a significant gain of using AIRO-ECW instead of DISWOTH.

Table 5.18 The trade-off matrix

	UTADIS	DISWOTH	AIRO-WE	AIRO-ECW	Nr. of Negatives
UTADIS	-	-	-	-	0
DISWOTH	3490.89	-	95284.86	-	0
AIRO-WE	15.04	-	-	-	0
AIRO-ECW	0.14	**-7854.05**	**-29.72**	-	2
Nr. of negatives	0	1	1	0	

Footnotes for Table 5.14 are also valid for this table.

TOPSIS ranking method is also used to compare the methods with the same parameters as in Section 5.1. Results are reported in Table 5.19. According to TOPSIS rankings, AIRO is the best method for all TOPSIS distance parameters and performance measure weights we used.

Table 5.19 Average TOPSIS ranks

	UTADIS	**Best of DISWOTH**	**AIRO-WE**	**AIRO-ECW**
W_{Time}=0.1 W_{Ac}=0.9				
L_1	3.00	3.33	2.11	**1.56**
L_2	3.00	3.33	2.11	**1.56**
L_3	3.00	3.33	2.11	**1.56**
L_4	3.00	3.33	2.11	**1.56**
L_5	3.00	3.33	2.11	**1.56**
L_{100}	3.00	3.22	2.22	**1.56**
L_∞	3.00	3.33	2.11	**1.56**
Average	3.00	3.32	2.13	**1.56**
SD	**0.00**	0.04	0.04	**0.00**
RANK	3	4	2	**1**

Table 5.19 Continued

W_{Time}=0.5 W_{Ac}=0.5				
L_1	3.00	3.22	2.22	**1.56**
L_2	3.11	3.11	2.22	**1.56**
L_3	3.11	3.11	2.22	**1.56**
L_4	3.11	3.11	2.22	**1.56**
L_5	3.11	3.11	2.22	**1.56**
L_{100}	3.00	3.22	2.22	**1.56**
L_∞	3.00	3.22	2.22	**1.56**
Average	3.06	3.16	2.22	**1.56**
SD	0.06	0.06	0.00	**0.00**
RANK	3	4	2	**1**
W_{Time}=0.9 W_{Ac}=0.1				
L_1	3.33	2.56	2.33	**1.78**
L_2	3.33	2.67	2.33	**1.67**
L_3	3.33	2.67	2.33	**1.67**
L_4	3.22	2.78	2.33	**1.67**
L_5	3.11	3.00	2.22	**1.67**
L_{100}	3.11	3.11	2.22	**1.56**
L_∞	2.74	2.66	1.97	**1.39**
Average	3.17	2.78	2.25	**1.63**
SD	0.21	0.20	0.13	**0.12**
RANK	4	3	2	**1**

,

CHAPTER 6

LINEAR APPROXIMATION OF L_p DISTANCE BASED ON AUGMENTED TCHEBYCHEFF PROGRAM

This chapter presents the third study conducted in this thesis. In this chapter, a new linear approximation to the L_p distance function is presented. To approximate all L_p distances, a single formulation is developed based on the formulations of Augmented Tchebycheff program [31] and Chaudhuri et al. [39]. The new formulation is affine combination of L_1 and L_∞ distances. Metricity conditions of the new formulation is analyzed and shown that it is consistent with L_p distances. Important characteristics of the new formulation are analyzed and a complete guideline for MP usage is presented. It is shown that by employing formulations of Charnes et al. [40] and Kelley [41], it can be adapted to LP which is computationally inexpensive. An algorithm is developed for approximating a set of L_p distances progressively, to solve the problem of determining a proper L_p distance. The LP formulation and the algorithm are combined as a method and applied to the distance-based multicriteria sorting methods (that are NLP models) to improve the solution time. Based on the new method, three alternative courses of actions are developed for implementation.

Organization of this chapter is as follows. In Section 6.1, related work of linear L_p distance approximation and L_p distance formulations for MP are explained. The base of new approximation, Augmented Tchebycheff Program is explained. In Section 6.2, the new approximation formulation is presented. Empirical and theoretical foundations about the new approximation is reported. Those foundations are also used as a guideline for usage of the new approximation method. The MP formulation of the new approximation method is formulated in Section 6.3, and it is applied to two distance-based sorting methods that use NLP model. These NLP models are linearized with the new approximation. To solve the new LP approximation an algorithm is introduced. A heuristic algorithm is developed for appropriate implementation of approximation. In Section 6.4, to fully benefit from the new approximation method, alternative courses of actions are presented. In Chapter 7, experiments are conducted on two distance-based sorting methods. The results of approximations are compared with the original methods. Results and improvements are reported and discussed.

6.1 Related Work

In this section, theoretical background is presented with literature review. Augmented Tchebycheff program is explained. A research question is asked in this section and answered in the next section.

6.1.1 Theoretical Background on Distance Functions

Consider two variables x_j and y_j where $j \in \{1,2,\dots,m\}, 1 < m < \infty$ and $a_j = |x_j - y_j|$. The L_p distance of vector a is formulated as $||a||_p = \sqrt[p]{\sum_j a_j^p}$ for $0 < p < \infty$. A specific case of L_p distance is L_∞ distance. It is formulated as $||a||_\infty = \sqrt[\infty]{\sum_j a_j^\infty} = \max_j\{a_j\}$ and called Tchebycheff distance. Contours of L_p distances for $p \in \{0.5, 1, 2, 4, 10, \infty\}$ are illustrated in Figure 6.1. A distance measure, denoted with $||a||$, is a norm (metric) if it satisfies three properties below. These three properties are also presented in Chapter 2. But in this chapter, second property is renamed as homogeneity to be consistent with the studies presented in the literature review [39], [42]–[45].

1. Nonnegativity and definiteness: $||a|| > 0$ and $||a|| = 0$ iff $a = 0$.
2. Homogeneity: $||ka|| = |k|\ ||a||$.
3. Triangular inequality: $||a + b|| \leq ||a|| + ||b||$.

To ease the usage of formulation and conveying ideas, L_p distance is denoted as a^p in this chapter. It satisfies all of three properties of metricity for $p \geq 1$ and does not satisfy triangular inequality if $0 < p < 1$. Therefore, a^p is a metric for $p \geq 1$ and not a metric for $p < 1$ due to violation of triangular inequality.

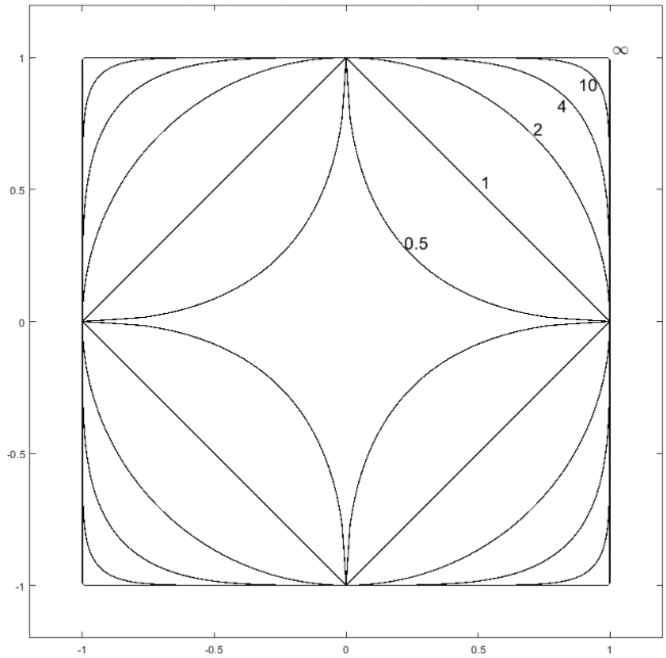


Figure 6.1 L_p distance contour examples for $p \in \{0.5, 1, 2, 4, 10, \infty\}$

In MP setting, various L_p distance models are studied. L_1 distance model is first studied by Charnes et al. [40] on an LP model as a goal programming approach to estimate executive compensation as in MP1.

(MP1)

$$minimize \sum_j u_j + v_j \tag{105}$$

Subject to:

$$x_j - y_j + u_j - v_j = 0 \; \forall j \tag{106}$$

$$u_j, v_j \geq 0 \; \forall j \tag{107}$$

u_j and v_j are nonnegative variables and they are used to formulate the absolute deviation. L_∞ distance formulation in MP setting is studied by Kelley [41]. Kelley [41] formulates L_∞ distance as an LP model as in MP2.

(MP2)

$$minimize\ u \tag{108}$$

Subject to:

$$u \geq x_j - y_j\ \forall j \tag{109}$$

$$u \geq y_j - x_j\ \forall j \tag{110}$$

The general L_p distance modelling problem is formulated in [46] as an NLP model for $p \notin \{1, \infty\}$. In [47] an NLP model is formulated for p^{th} power L_p distance approximation. Various other NLP formulations are proposed in the literature. Interested readers may refer to [48] and [49] and Chapter 1 of Gonin & Arthur [50] (for a review).

A linear approximation for the nonlinear L_2 distance (Euclidean distance) is first studied by Chaudhuri et al. [39] by using the convex combination of L_1 and L_∞ distances as in (109). Rhodes [42], [43] and [45] improve the approximation error of Chaudhuri et al.' s [39] formulation. In those studies, formulation is rewritten as (110). A positive weight $V \in (0{,}0.5)$ is defined to approximate L_2 distance. [43] states that any L_p distance can be approximated by linear combinations of other L_p distances. Distance approximation formulations in (111)-(112) form an octagonal contour as in Augmented Tchebycheff program (explained in Section 6.1.2). However, the presented formulation is not extended for any MP and other L_p distances. To our knowledge, there is no study in the literature that formulates the general L_p distance approximation for LP model for $p \notin \{1, \infty\}$. For the following formulation assume $a_{j^+} = a^\infty = \max_j\{|a_j|\}$. $\widehat{a^2}$ denotes the approximation function of the Euclidean distance.

$$\widehat{a^2} = a_{j^+} + \frac{1}{m-\left\lceil\frac{m-2}{2}\right\rceil}\sum_{j\neq j^+} a_j \tag{111}$$

$$\widehat{a^2} = (1-V)a_{j^+} + Va^1, V \in [0.5,1) \tag{112}$$

It is worth noting that L_p distance approximation makes the MP a multi-objective optimization problem when used with other constraints and objective functions in the model that depend on the distance functions. For such an example, see L_∞ distance approximation of Karasakal & Civelek [6]. However, to make this study self-explanatory, an example can be given with MP3 by employing formulation in MP2 of Kelley [41]. Objective function (113) is a multi-objective formulation. X is a nonempty feasible region that is constructed by a set of known constraints. In such models, choice of coefficient C_∞ is important since a large C_∞ value may cause a trade-off between objective function $z(x,y)$ and u. On the other hand, a small C_∞ may result with an incorrect L_∞ approximation resulting in $u > \max_j\{|x_j - y_j|\}$. Choosing a suitable C_∞ value is handled by empirical study in the study of Karasakal & Civelek [6].

(MP3)

$$minimize\ z(x,y) + C_\infty u \tag{113}$$

Subject to:

Constraints (109)-(110) and $x, y \in X$

6.1.2 Augmented Tchebycheff Program

Optimizing multiple conflicting objectives is handled with multi-objective optimization. Many methods are employed to optimize multiple conflicting objectives. Several examples to these methods can be weighted sum method, e-constraint method, goal programming method and other scalarizing functions etc. For scalarizing functions, Tchebycheff Program and Augmented Tchebycheff

Program are frequently used [31]. [31] also shows how L_p distances can be used in multi-objective Optimization.

In general, the aim is to find a non-dominated (efficient) outcome (and the solution) that is preferable for the DM between possibly infinitely many outcomes. Such efficient solutions are sought on the non-dominated frontier of the objective space. Finding a non-dominated solution is also problematic because a non-dominated solution can be a weakly efficient solution. To clarify the concept of weakly efficient solutions, the objective space and non-dominated solutions consider MP4 with k conflicting maximization objectives, $z_k(x)$. A two-objective example of MP4 with $z_1(x)$ and $z_2(x)$ is illustrated in Figure 6.2. Z is the objective space that is the objective function outcomes of all feasible solutions. The thick black line is the efficient frontier that is non-dominated, and the dashed line is the weakly efficient outcomes that are weakly dominated by the outcome z' on the objective z_1.

(MP4)

$$maximize\ z_k(x)\ \forall k \tag{114}$$

$$x \in X$$

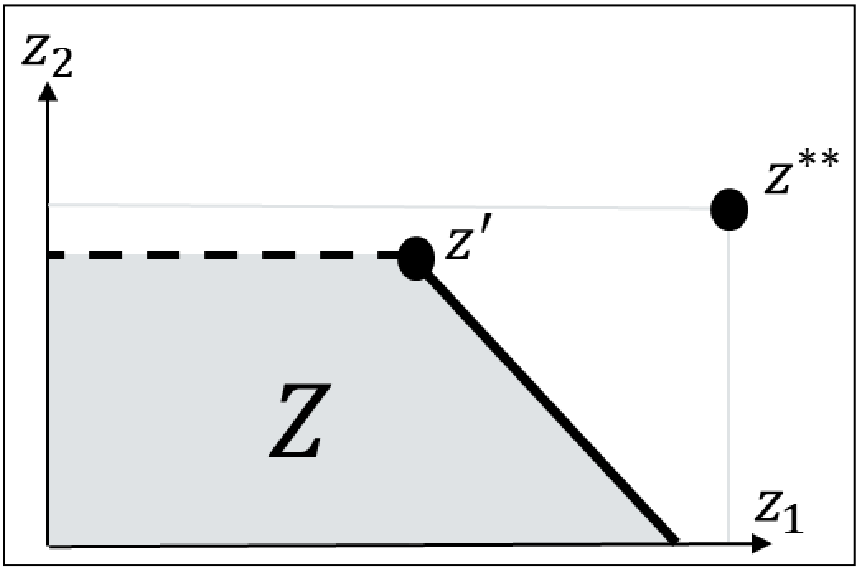


Figure 6.2 An example illustration of objective space

Different solutions from the efficient frontier can be found by many different methods but this study is interested in Tchebycheff program and the Augmented Tchebycheff program. Tchebycheff program projects a given reference point (z^{**} in Figure 6.2) to the non-dominated solutions by minimizing the Tchebycheff distance (L_∞ distance) between the reference point and z_1 and z_2. Geometrically, Tchebycheff program form a rectangular contour (Figure 6.3a and Figure 6.3b) around the reference point to reach an efficient solution. The reached efficient solution is the one with minimum L_∞ distance from the reference point. Tchebycheff program for MP4 is formulated as in MP5.

(MP5)

$$Minimize\ \alpha^\infty \tag{115}$$

Subject to:

$$\alpha^\infty \geq \lambda_k(z^{**} - z_k(x))\ \forall k \in \{1,2\} \tag{116}$$

$$x \in X$$

However, Tchebycheff program may result in a weakly efficient solution. To overcome this problem, Augmented Tchebycheff program is introduced. A small augmentation is provided to the L_∞ contour with the addition of L_1 distance with a small positive coefficient ρ. Augmented Tchebycheff program form an octagonal contour around the reference point to reach the efficient frontier. Augmented Tchebycheff program can be formulated as MP6. To illustrate how the efficient frontier is found by the two methods, Figure 6.3 is presented.

(MP6)

$$Minimize\ \alpha^\infty + \rho \sum_{k\in\{1,2\}}(z^{**} - z_k(x)) \tag{117}$$

Constraint (116) and $x \in X$

In Figure 6.3a, an efficient solution from efficient frontier is obtained with the Tchebycheff program. In Figure 6.3b, the problem of obtaining a weakly efficient solution with Tchebycheff program is illustrated. In Figure 6.3c, an efficient solution found by the Augmented Tchebycheff program is illustrated. In Figure 6.3a, z'' is reached by the rectangular contour of Tchebycheff program and not dominated by any outcome in objective space. In Figure 6.3b, z'' is reached by the rectangular contour and it is weakly dominated by the outcome z'. In Figure 6.3c, $z''$is reached by the octagonal contour of the Augmented Tchebycheff program and it is not dominated by any other outcome.

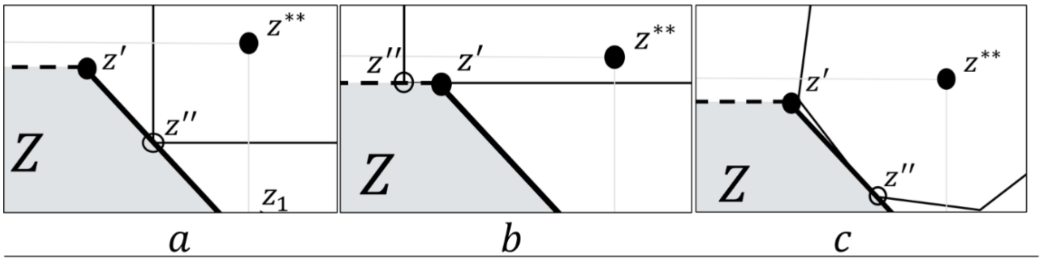


Figure 6.3 An example illustration of how the efficient frontier is reached by Tchebycheff and Augmented Tchebycheff program

It should be noted that, both formulation of Chaudhuri et al. [39] in (111)-(112) and Augmented Tchebycheff formulation (117) in MP6 form octagonal contours. In Augmented Tchebycheff, the shape of contour depends on the parameter ρ. In formulation of Chaudhuri et al., the shape of the octagon depends on the coefficient that is multiplied with $\sum_{j \neq j^+} a_j$.

Here we question "*Can the L_p distance contours illustrated in Section 6.1 (Figure 6.1) be approximated by the octagonal contour provided by Augmented Tchebycheff program (Figure 6.3c) by working on the formulation given in objective function (116) and equations (111)-(112)*?". In the next section, this question is answered with empirical and theoretical foundations.

6.2 Proposed Approximation Method

In this section, first, new notation for discrete MCDM problem setting and the proposed formulation is presented. Empirical foundations are illustrated with figures in Section 6.2.1. Then, a suitable error function is set up for the new approximator in Section 6.2.2. The proposed formulation is examined w.r.t approximation error and approximation parameters. According to the theoretical foundations in Section 6.2.2, important characteristics of the approximation method is presented. The characteristics presented in Section 6.2.2 are also a complete guideline for the usage of the approximator. Relevant notation used in the previous chapters are revisited and the new notation is presented as follows.

Notation:

$j \in \{1,2,\dots,m\}$ denotes the criteria where $m \in \{2,\dots\}$ and w_j denotes the criterion weights. Formulation of a^p is presented as weighted L_p distance as in equation (117). V denotes the weight for the affine combination of L_1 and L_∞. The approximator, Augmented Tchebycheff L_p distance Approximation method (ATLA), is denoted with $\widehat{a^p}(V)$ and formulated as the affine combination of Tchebycheff (L_∞) and Rectilinear (L_1) distances in (118). Criterion weights are normalized in (119)-(120).

$$a^p = \sqrt[p]{\textstyle\sum_j w_j^p a^p} \tag{117}$$

$$\widehat{a^p}(V) = Va^\infty + (1-V)a^1,\ -\infty < V \leq 1 \tag{118}$$

$$\textstyle\sum_j w_j = 1 \tag{119}$$

$$w_j \geq 0\ \forall j \tag{120}$$

It should be noted that equation (120) is a modified version of equation (111) and objective function (116). Equation (111) is not able to approximate all L_p distances for the assumed range of V, especially when $p \leq 1$. This formulation enables approximator to approximate all L_p distances with different V values. This formulation is similar to (112), but it is not necessarily a positive linear combination as in [43]. To present the capabilities and characteristics of $\widehat{a^p}(V)$, empirical foundations about the $\widehat{a^p}(V)$ contour and theoretical foundations related to properties of $\widehat{a^p}(V)$ are reported in Sections 6.2.1 and 6.2.2, respectively.

6.2.1 Empirical Foundations

In this section, examples of contours that are obtained with the ATLA and a^p are illustrated and compared. First, let us examine the four of example illustrations given in Figure 6.4 for different p and V values. As it is illustrated in Section 6.1, a^p contours can be closely approximated by piecewise linear octagonal contours of new approximator in equation (118). Criterion weights are equal in this illustration.

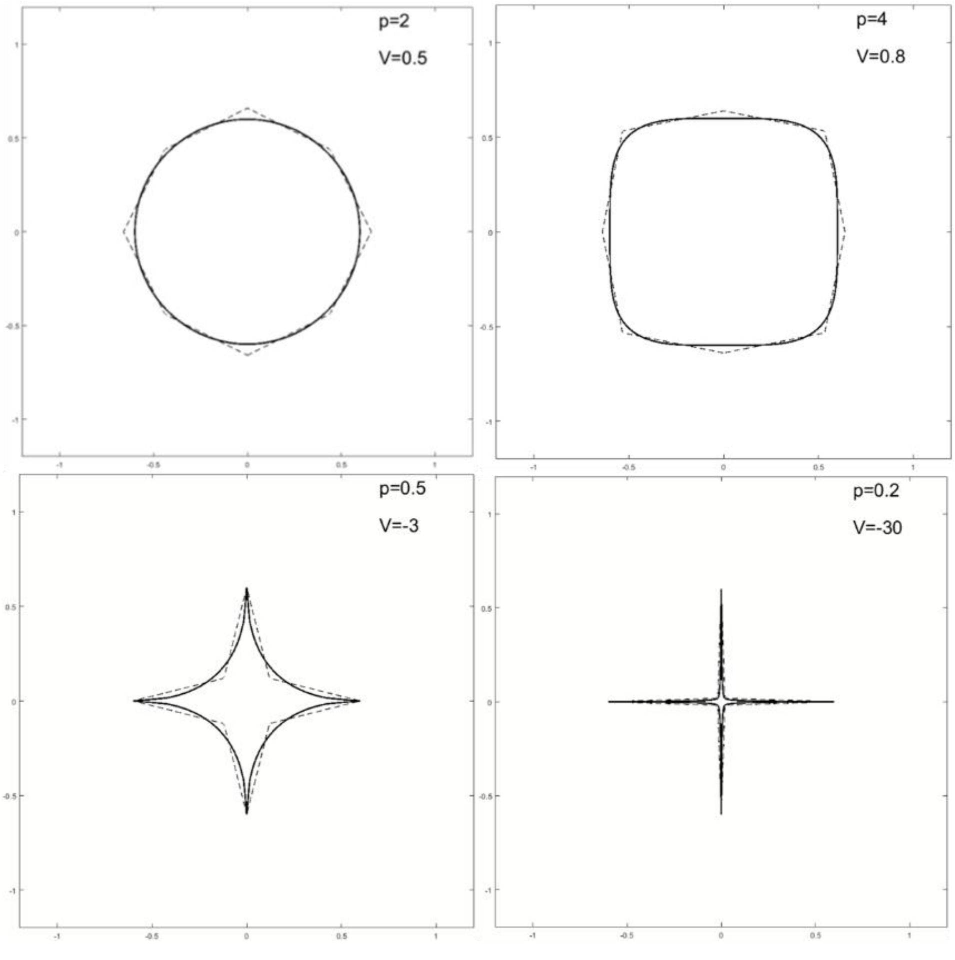


Figure 6.4 Example illustration of $\widehat{a^p}(V)$ (dashed lines) and a^p (solid curves) contours

From Figure 6.4, it is observed that negative V values can approximate a^p contours for $p < 1$ and positive V values can approximate a^p contours for $p > 1$. And as it is clear from equation (118), $V = 0$ gives a^1 distance and $V = 1$ gives a^∞ distance. In Figure 6.4, V and p increase together, which forms a close approximation.

Studying $V > 1$ is out of scope since ATLA formulation will return a negative value in this case. A negative output cannot be returned by a distance function. As V increases, the value returned by ATLA decreases which is a consistent finding of previous presumption (V and p increase together) since it is known that $a^\infty < a^p < a^1$ for $p \geq 1$. The relationship between V and p are analyzed in Section 6.2.2. L_p and $\widehat{a^p}(V)$ are also illustrated in Figure 6.5.

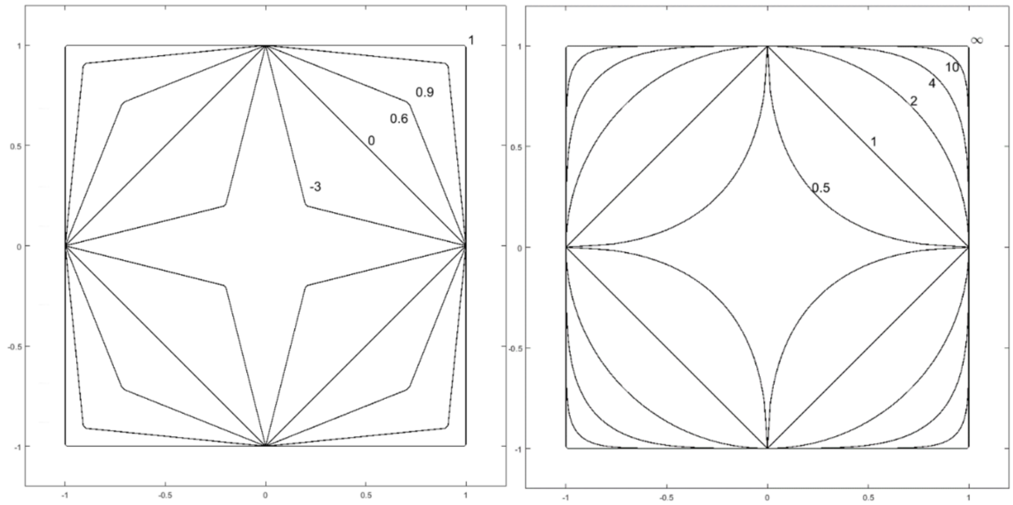


Figure 6.5 Illustration of $\widehat{a^p}(V)$ (on the left) and a^p (on the right) with different V and p values, respectively.

Response of contours to changing criterion weights is illustrated in Figure 6.6. The change in the contours of both ATLA and a^p is similar to the same criterion weights. This is expected since ATLA itself is a function of two L_p distances that are L_1 and L_∞ and response of L_p distances to criterion weights are the same. The response of L_p contours to criterion weights is as follows. The contour extends on along the axis of lower criterion weights and squeezed along the axis of higher criterion weights.

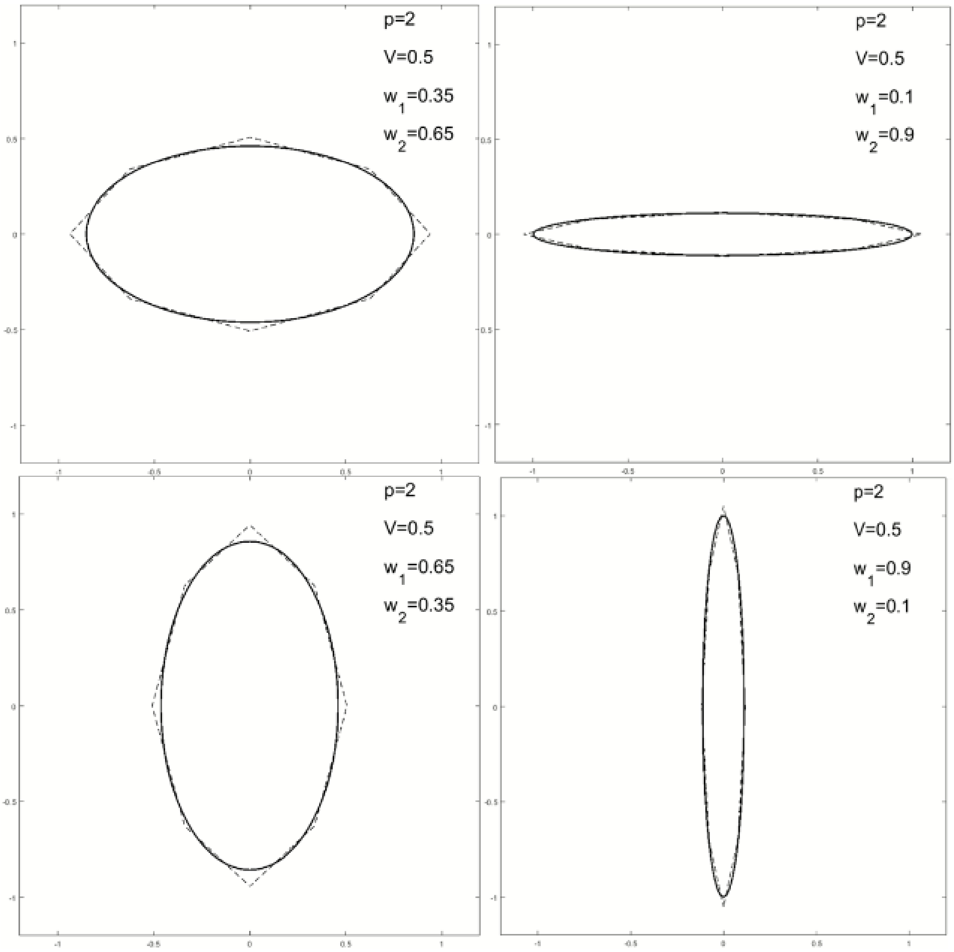


Figure 6.6 An example of the response given by ATLA (dashed lines) and a^p (solid curves) contours to criterion weights

6.2.2 Theoretical Foundations

In this section, ATLA is analyzed for its theoretical properties. The analyses in this section are as follows.

- Approximation error
- Optimal approximation parameter and specific cases
- Verification and approximation conditions
- Characteristics of the approximator
- Consistency of metricity conditions

To analyze the approximator, let us define an error function $error(V)$. Find the least square error (LSE) between a^p and $\widehat{a^p}(V)$ with (121)-(122).

(LSE)

$$minimize\ error(V)^2 \tag{121}$$

Subject to:

$$error(V) = \widehat{a^p}(V) - a^p \tag{122}$$

LSE is a convex programming model as shown in inequality (123). Therefore, equation (124) is applied to find the optimal approximation parameter V^* with $\frac{\delta error(V)}{\delta V} = 0$.

$$\frac{\delta^2 error(V)}{\delta V^2} = 2\left({a^\infty}^2 - 2a^1 a^\infty + {a^1}^2\right) = 2(a^1 - a^\infty)^2 > 0 \tag{123}$$

$$V^* = \frac{(a^\infty - a^1)(a^p - a^1)}{(a^\infty - a^1)^2} = \frac{a^1 - a^p}{a^1 - a^\infty} \tag{124}$$

Undefined Case $\mathrm{a}^1 = \mathrm{a}^\infty$

V^* is undefined for $a^1 = a^\infty$ due to $a^1 - a^\infty$ in denominator of equation (124). There are two cases of a where $a^1 = a^\infty$.

Case 1: $a = \bar{0}$. In this case, $a^p = 0\ \forall p$ and the approximation can be performed with $error(V^*) = 0$.

Case 2: a has only one non-zero entry. In this case, $a^p = a^q = a^\infty = a^1\ \forall q \neq p$. Again, the approximation can be performed with $e(V^*) = 0$.

To sum up, the undefined cases result with zero error ($e(V^*) = 0$). As a result, it can be said that $e(V^*) = 0$ if there are at most one non-zero entry in a and choice of V is irrelevant.

Error of Approximation of Parameter V^*

To find the level of error, use V^* in $error(V)$ as follows.

$$error(V^*) = \left[\frac{a^1 - a^p}{a^1 - a^\infty} a^\infty + \frac{a^p - a^\infty}{a^1 - a^\infty} a^1\right] - a^p = \frac{a^1 a^\infty - a^p a^\infty + a^1 a^p - a^1 a^\infty}{a^1 - a^\infty} - a^p = a^p - a^p = 0 \quad (125)$$

Theoretically, the ATLA can approximate any L_p distance with zero error for a given a. Note that this does not mean an octagon can approximate an oval shape with zero error. Because the contour of a^p distance is formed by infinitely many different a vectors.

Verification

For $p = 1$, function returns $V^* = 0$ (a^1). For $p = \infty$, function returns $V^* = 1$ (a^∞) as expected. From (118), it is clear that $\widehat{a^p}(V') \leq \widehat{a^p}(V)$ for $V' \geq V$, since $a^1 \geq a^\infty$.

Condition 1: For $p \geq 1$, $a^p \leq a^1$. Thus, the range $0 \leq V^* \leq 1$ can be applied since $a^\infty \leq \widehat{a^p}(V) \leq a^1 \; \forall V \in [0,1]$.

Condition 2: For $p < 1$, $a^p \geq a^1$. Thus, the range $-\infty < V^* < 0$ can be applied as $\widehat{a^p}(V) \geq a^1 \; \forall V \in (-\infty, 0)$

Conditions 1 and 2 are also proven in Theorem 4 in this section. The range of parameter V is established with the verification based on Conditions 1 and 2 as $V \in (-\infty, 1]$ which is supported with Theorem 4.

From equations (124)-(125), Conditions 1 and 2, for a given a, it can be inferred that all L_p distances can be found using equation (118) with $V \in (-\infty, 1]$. In practice, V^* is not used since it is a function of a^p and if a^p is calculated then V^* is not needed.

Characteristics of V^*

Equations (124)-(125) show that there is a V^* value for each p for a given a. Therefore, we can denote V^* as $V(p|a)$. $V(p|a)$ is linear w.r.t a^p as shown below.

$$\frac{\delta V(p|a)}{\delta a^p} = \frac{1}{a^\infty - a^1} < 0 \; where \; a^\infty \neq a^1 \quad (126)$$

$$\frac{\delta^2 V(p|a)}{\delta {a^p}^2} = 0 \quad (127)$$

Since a^p is decreasing in p, $V(p|a)$ is increasing in p due to (126)-(127). This validates the intuitions stated in empirical foundations.

Theorem 4: *$V(p|a)$ is a concave and asymptotic function of p.*

Proof: Since monotonicity of $V(p|a)$ w.r.t. p is shown, if (128) holds then it is concave. (128) is simplified to (129) and then (130) after elementary operations.

$$V(hp_1 + (1-h)p_2) \geq hV(p_1) + (1-h)V(p_2) \; h \in [0,1] \tag{128}$$

$$\frac{a^1 - a^{hp_1+(1-h)p_2}}{a^1 - a^\infty} \geq h\frac{a^1 - a^{p_1}}{a^1 - a^\infty} + (1-h)\frac{a^1 - a^{p_2}}{a^1 - a^\infty} \tag{129}$$

$$a^{hp_1+(1-h)p_2} = a^{p_1 h} a^{p_2(1-h)} \leq ha^{p_1} + (1-h)a^{p_2} \tag{130}$$

Using logarithm to relax the exponents h and $(1-h)$, we can obtain (129).

$$hlog(a^{p_1}) + (1-h)\, log(a^{p_2}) \leq log(ha^{p_1} + (1-h)a^{p_2}) \tag{131}$$

(131) always holds since the logarithm is a concave function. Moreover, $V(p|a)$ is asymptotic when $p \to \infty$ and $p \to 0^+$ as shown below.

$$\lim_{p\to\infty} a^p = a^\infty \text{ and } V(p|a) = \frac{a^1 - a^p}{a^1 - a^\infty} \to 1 \tag{132}$$

$$\lim_{\mathrm{p}\to 0^+} a^p = \infty \text{ and } V(p|a) = \frac{a^1 - a^p}{a^1 - a^\infty} \to -\infty \tag{133}$$

(132) is a trivial result. (133) can be proven by using exp of ln(.) ($x = e^{\ln x}$): $\lim_{\mathrm{p}\to 0^+} a^p = e^{\lim_{\mathrm{p}\to 0^+} \ln(a^p)} = e^{\lim_{\mathrm{p}\to 0^+} \frac{1}{\mathrm{p}} \ln(a_1^p + \cdots + a_m^p)} = e^{\lim_{\mathrm{p}\to 0^+} \frac{1}{\mathrm{p}} (\lim_{\mathrm{p}\to 0^+} \ln(a_1^p + \cdots + a_m^p))} =$ $e^{\ln(m)\infty} = \infty$, since $\infty > m \geq 2$ and $\ln(m) > 0$. ❑

Plot of $V(p|a)$ w.r.t. p for a given a is shown in Figure 6.7. Due to (121)-(133), a^p function with $0 < p \leq \infty$ can be approximated with $V \in (-\infty, 1]$.

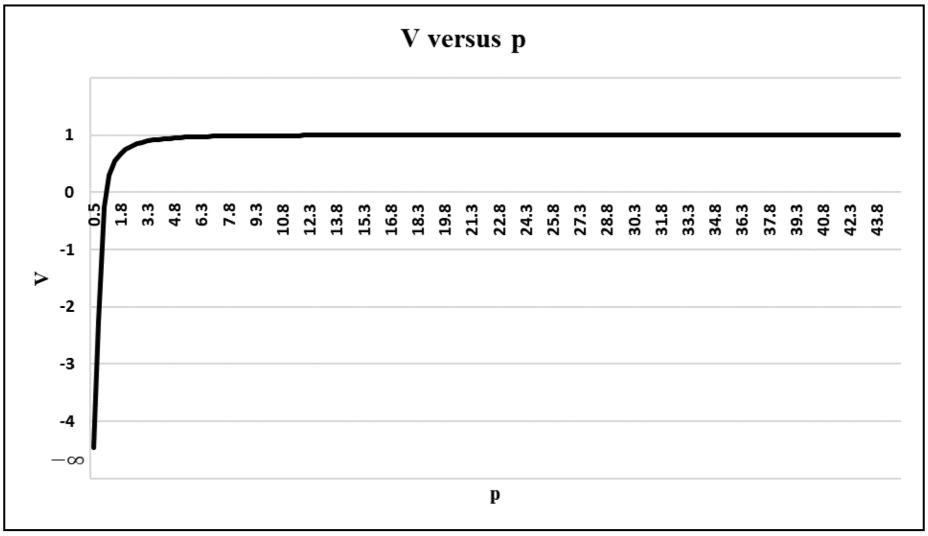


Figure 6.7 Plot of V versus p

Metricity Conditions of $\widehat{\mathrm{a^p}}(\mathrm{V})$

As a reminder, L_p (or a^p) distance function is a metric when $p \geq 1$ and is not a metric when $p < 1$ due to the violation of triangular inequality.

Theorem 5: $\widehat{\mathrm{a^p}}(V)$ *is a metric for* $\mathrm{V} \geq 0$ *which approximates* $\mathrm{L_p}$ *distance with* $\mathrm{p} \geq 1$ *which is also a metric.*

Proof: (134)-(135) are used when showing the triangular inequality.

$$(\mathrm{a}+\mathrm{b})^1 = \mathrm{a}^1 + \mathrm{b}^1 \tag{134}$$

$$(\mathrm{a}+\mathrm{b})^\infty \leq \mathrm{a}^\infty + \mathrm{b}^\infty, \text{ since } \max_{\mathrm{j}}\{\mathrm{a_j}+\mathrm{b_j}\} \leq \max_{\mathrm{j}}\{\mathrm{a_j}\} + \max_{\mathrm{j}}\{\mathrm{b_j}\} \tag{135}$$

1. $Va^\infty + (1-V)a^1 > 0$ always hold for since the formulation is a convex combination of two positive numbers for $V \geq 0$. Also, $Va^\infty + (1-V)a^1 = 0 \; iff \quad a = 0$.

2. $V \max_j\{\,|k|a_j\} + (1-V)\sum_j |k|a_j = V|k| \max_j\{a_j\} + (1-V)|k|\sum_j a_j$ holds due to absolute function in a^1 and a^∞ distances.

3. $\mathrm{V}(\mathrm{a}+\mathrm{b})^{\infty}+(1-\mathrm{V})(\mathrm{a}+\mathrm{b})^{1} \leq \mathrm{V}(\mathrm{a}^{\infty}+\mathrm{b}^{\infty})+(1-\mathrm{V})(\mathrm{a}^{1}+\mathrm{b}^{1})$. This inequality is simplified to $\mathrm{V}(\mathrm{a}+\mathrm{b})^{\infty} \leq \mathrm{V}(\mathrm{a}^{\infty}+\mathrm{b}^{\infty})$ due to (134). $V(\mathrm{a}+\mathrm{b})^{\infty} \leq \mathrm{V}(\mathrm{a}^{\infty}+\mathrm{b}^{\infty})$ always hold due to (135). ❑

In [45], it is stated that when $\mathrm{V} < 0$**,** $\widehat{\mathrm{a}^{\mathrm{p}}}(\mathrm{V})$ formulation may not satisfy the metricity conditions, but it is not proven. In this study, it is proven by Theorem 6.

***Theorem 6*:** $\widehat{\mathrm{a}^{\mathrm{p}}}(\mathrm{V})$ *is not a metric for* $\mathrm{V} < 0$ *which approximates* L_p *distance function with* $0 < \mathrm{p} < 1$ *which is also not metric . Metricity is violated by triangular inequality as in* L_p *with* $0 < \mathrm{p} < 1$**.**

Proof:

1. $Va^{\infty}+(1-V)a^{1} > 0$ always holds for since $a^{1} > a^{\infty}$. $Va^{\infty}+(1-\mathrm{V})a^{1} = 0$ iff $a = 0$.

2. $V \max_{j}\{\,|k|a_j\}+(1-V)\sum_j |k|a_j = V\,|k|\max_{j}\{a_j\}+(1-V)|k|\sum_j a_j$ holds due to absolute function.

3. $\mathrm{V}(\mathrm{a}+\mathrm{b})^{\infty}+(1-\mathrm{V})(\mathrm{a}+\mathrm{b})^{1} \leq \mathrm{V}(\mathrm{a}^{\infty}+\mathrm{b}^{\infty})+(1-\mathrm{V})(\mathrm{a}^{1}+\mathrm{b}^{1})$. This expression is simplified to $V(\mathrm{a}+\mathrm{b})^{\infty} \leq V(\mathrm{a}^{\infty}+\mathrm{b}^{\infty})$ due to (134) and the result does not hold due to (135) and since $\mathrm{V} < 0$, except for the equality case. The equality case occurs when the $\mathrm{a}^{\infty} = \mathrm{a}_{\mathrm{j}'}$ and $\mathrm{b}^{\infty} = \mathrm{b}_{\mathrm{j}''}$ and $\mathrm{j}' = \mathrm{j}''$. ❑

In Theorems 4-6, it is shown that the approximator $\widehat{\mathrm{a}^{\mathrm{p}}}(\mathrm{V})$ is not only consistent with L_p due to Conditions 1 and 2 but also consistent due to metricity conditions. Four corollaries are presented as follows.

Corollary 1: There is a V value for each p of $\mathrm{L_p}$ distance. All $\mathrm{L_p}$ distances can be calculated with $\widehat{\mathrm{a}^{\mathrm{p}}}(\mathrm{V})$ for a given a.

***Corollary 2*:** $V \in [0,1]$ approximates the L_p distance with $p \geq 1$ and $V \in (-\infty, 0)$ approximates the L_p distance with $0 < p < 1$.

***Corollary 3*:** $\widehat{a^p}(V)$ is a metric when L_p distance is a metric ($p \geq 1$ *and* $1 \geq V \geq 0$) and $\widehat{a^p}(V)$ is not a metric when L_p distance is not a metric ($1 > p > 0$ and $0 > V > -\infty$).

Corollary 4: $\widehat{a^p}(V)$ is concave, monotonically increasing, and asymptotic w.r.t p such that $V \to -\infty$ when $p \to 0^+$ and $V \to 1$ when $p \to \infty$.

It may not be precisely known which V approximates which L_p distance. However, a set of V values can be supplied to ATLA to approximate a set of L_p distances. ATLA is illustrated with two different distance-based multicriteria sorting methods in the next section.

6.3 Application of ATLA in MP: ATLAS Algorithm for Multicriteria Sorting

Consider following nonlinear mathematical program, NLP1 where $z(x, y)$ is a linear objective function and $f(.)$ is a linear function. The nonlinearity is caused by the $a^p = \sqrt[p]{\sum_j w_j^p |x_j - y_j|^p}$ in the constraint (137). RHS is a known parameter.

(NLP1)

$$Minimize\ z(x, y) \tag{136}$$

Subject to:

$$f(a^p) = RHS \tag{137}$$

$$x, y \in X$$

Constraints (119)-(120)

where a^p contains decision variables w, x and y and it is an NLP formulation for $p \notin \{1, \infty\}$. NLP1 can be approximated as LP1 by employing formulations of Charnes et al. [40], Kelley [41] in objective function (138) and constraints (139)-(142).

(LP1)

$$Minimize\ z(x,y) + C_1 \sum_j a_j + C_\infty a^\infty \tag{138}$$

Subject to:

$$f\left(V a^\infty + (1-V)\sum_j a_j\right) = RHS, V \in (-\infty, 1] \tag{139}$$

$$a_j \geq x_j - y_j\ \forall j \tag{140}$$

$$a_j \geq y_j - x_j\ \forall j \tag{141}$$

$$a^\infty \geq a_j\ \forall j \tag{142}$$

Formulations of Charnes et al. [40] and Kelly [41] are minimization of distances. Therefore, in their problem environment they are exact models. In multicriteria sorting, the objective is to minimize error. In our case, those formulations are approximation. Because in this study, aim is not to minimize the distance of an alternative to a reference point. The aim is to find the exact values of the L_1 and L_∞ distance functions in an MP that are formulated with greater and equal to constraints. In LP1, nonlinearity due to the L_p distance is eliminated with ATLA as an approximation of the original NLP1. Choice of coefficients C_1 and C_∞ are important since the small coefficients may not properly approximate the distance functions and large coefficients may cause a significant trade-off between the objective function $z(x,y)$ and distance function approximations $C_1 \sum_j a_j + C_\infty a^\infty$. C_1 and C_∞ can be decided via empirical study as in [6]. But to construct a well-defined approximation method, in the next section, a heuristic algorithm is developed to find small coefficients and it is applied to two distance-based sorting models as an example.

6.3.1 Application to Distance-based Sorting Method

In this section, a new approximation is applied to distance-based multicriteria sorting. A distance-based multicriteria sorting formulation is constructed based on [7] and [5]. The sorting model is named Distance-based Sorting (DS).

Let us briefly revisit the notation and present the new relevant notation for DS. $i \in \{1,2,\ldots n\}$ stands for the alternatives. Ordinal classes are denoted by $q \in \{1,2,\ldots Q\}$ where class Q is the best class and class 1 is the worst class. C^q is the group of alternatives in class q. A_{ij}^q stands for the criterion evaluation of alternative i from class q on criterion j. I denotes ICV and I_j denotes the j^{th} element of I. Class thresholds are denoted by T^q. The L_p distance, $\left|\left|A_i^q - I\right|\right|_p$, is the distance-based criteria aggregation function. Class assignment errors are determined by comparing the criteria aggregation with the class thresholds of adjacent classes. e_i^+ and e_i^- represents the error of class assignment due to comparison of criteria aggregation to worse class and better class thresholds, respectively.

DS model assigns the alternatives to the ordinal classes based on a criteria aggregation function and class thresholds. Criteria aggregation of alternatives are performed based on their distances to the ICV. The class thresholds are in monotonic order increasing from most preferred (Q) to the least preferred class (1).

DS for a predetermined L_p distance is as follows.

(DS)

$$Minimize\ z = \sum_q \frac{\sum_{i \in C^q} e_i^+ + e_i^-}{|C^q|} \tag{143}$$

Subject to:

$$\left|\left|A_i^q - I\right|\right|_p - e_i^+ \leq T^{q-1}\ \ \forall q > 1 \tag{144}$$

$$\left|\left|A_i^q - I\right|\right|_p + e_i^- \geq T^q\ \forall q < Q \tag{145}$$

$$\left|\left|A_i^q - I\right|\right|_p = \sqrt[p]{\sum_j w_j^p \left|A_{ij}^q - I_j\right|^p}\ \ \forall i, \forall q \tag{146}$$

$$T^{q-1} \geq T^q,\ 1 < q < \infty \tag{147}$$

$$\sum_j w_j = 1 \tag{148}$$

$$w_j \geq 0 \; \forall j \tag{149}$$

$$T^q \geq 0 \; \forall q < Q \tag{150}$$

$$e_{i_r}^+, e_{i_r}^- \geq 0 \; \forall i \tag{151}$$

The criteria aggregation is as in equation (146). Class assignments are performed, and assignment errors are computed in constraints (144)-(145). Class thresholds are ordered in constraint (147) and criterion weights are normalized in constraint (148). Class weighted total error is minimized in the objective function (143).

L_p distance is used as criteria aggregation function in DS. Therefore, ATLA method can be applied to the criteria aggregation function $\left|\left|A_i^q - I\right|\right|_p$.

To approximate the criteria aggregation function, the formulation is updated as follows. (146) is replaced by (152) and $\left|\left|A_i^q - I\right|\right|_p$ in (144)-(145) is replaced by the right hand side of equation (152). A new constraint (153) is added to approximate the a^∞ distance. Objective function is updated as (154). This version of the DS is named Approximated DS.

$$\left|\left|A_i^q - I\right|\right|_p \cong V a_i^\infty + (1 - V) \sum_j w_j \left|A_{ij}^q - I_j\right| \tag{152}$$

$$a_i^\infty \geq w_j \left|A_{ij} - I_j\right| \; \forall j, \forall i \tag{153}$$

$$Minimize\ z = \sum_q \frac{\sum_{i \in C^q} e_i^+ + e_i^- + C_{\infty,V} a_i^\infty}{|C^q|} \tag{154}$$

Since $\left|A_{ij}^q - I_j\right|$ term is a parameter, rectilinear distance is not approximated in this model. $C_{\infty,V}$ in the objective function (154) should be decided properly. If it is a high coefficient there may exist a trade-off that decreases the accuracy by increasing the error. If it is low, then an a_i^∞value may be determined erroneously, that is $a_i^\infty > w_j \left|A_{ij} - I_j\right|$. Therefore $C_{\infty,V}$ must be low to maximize accuracy and sufficiently high to correctly approximate the L_∞ (a_i^∞) distance. To approximate a set of L_p distances and determine a proper $C_{\infty,V}$ coefficient, ATLA Sorting (ATLAS) algorithm is developed.

New notation for ATLAS is as follows. $T_rA(V), T_sA(V), w^*(V)$ and $T^*(V)$ denote the training accuracy, test accuracy, optimal weights and thresholds that are obtained for a predetermined V, respectively. Feasibility Condition (FC) is developed for checking if the a_i^∞ approximations are correct. FC is formulated as follows:

$$FC = \sum_i \left| a_i^\infty - \max_j \{ w_j |A_{ij} - I_j| \} \right| \tag{155}$$

$FC = 0$ means that all a_i^∞ approximation are correct and $FS > 0$ means that a_i^∞ approximations are incorrect for at least one i. The accuracy (for both training and test) is computed as follows. n_r and n_s denote the size of training and test data respectively, $i_r \in \{1,2,\dots,n_r\}$ and $i_s \in \{1,2,\dots,n_s\}$. $\delta(.)$ is an indicator function, returns 1 if the expression in the parenthesis is true and returns 0 if it is false.

$$T_rA(V) = \frac{1}{n_r} \sum_{i_r} \delta(e_{i_r}^+ + e_{i_r}^- = 0) \tag{156}$$

$$T_sA(V) = \frac{1}{n_s} \sum_{i_s} \delta(e_{i_r}^+ + e_{i_r}^- = 0) \tag{157}$$

To compute $T_sA(V)$, find the error variables of the test data by supplying parameters $V, w^*(V), T^*(V)$ to TestCalculationModel-ApproximatedDS in Appendix E. $w^*(V)$ and $T^*(V)$ pair is the optimal solution of ApproximatedDS model for a predetermined V. TestCalculationModel-ApproximatedDS is not an optimization problem, since $V, w^*(V), T^*(V)$ are known parameters except for $e_{i_S}^+$ and $e_{i_S}^-$. It is only used to calculate $e_{i_S}^+$ and $e_{i_S}^-$. $Cloop$ and $Vloop$ are indexes used for looping through a set of $C_{\infty,V}$ and V values.

ATLAS algorithm supplies a set of V values to the approximated sorting model, sequentially. For each V value an LP is solved, and feasibility check is done for the approximation of a^∞ with FC formulation. The LP that is solved is the ApproximatedDS. $C_{\infty,V}$ values are systematically increased from a small value to a higher value until $FC = 0$ is satisfied to find a small coefficient to approximate a^∞ distance in formulation of ATLA.

ATLAS Algorithm:

Step 1: Determine a set of V values, $\bar{V} = [\ldots]$ ordered in ascending order of $|V|$. Set $Vloop = 1,\ Cloop = 1, C_{\infty,V-1} = 0$.

Step 2: Solve "Approximated DS" for training data with:

$V = \bar{V}[Vloop]$, Determine $C_{\infty,\bar{V}[Vloop]}(Cloop)$ such that $C_{\infty,\bar{V}[Vloop]}(Cloop) > C_{\infty,\bar{V}[Vloop-1]}(Cloop)$, go to Step 3.

Step 3: Check feasibility of a^∞:

If $FC = 0$, Feasible a_i^∞:

Reset $Cloop = 1$, record $T_rA(V)$, $w^*(V)$ and $T^*(V)$. Go to Step 4.

Else, Infeasible a_i^∞:

Update $Cloop = Cloop + 1$. Go to step 2.

Step 4: Check termination condition:

If $Vloop < |\overline{V}|$, $Vloop = Vloop + 1$, go to step 2.

Else Terminate.

Outputs: $T_rA(\overline{V}), w^*(\overline{V})\ and\ T^*(\overline{V})$

In the first step of the algorithm, the sets for performing two loops are initialized. The first loop is for supplying a set of V values and the second loop is for finding a small $C_{\infty,V}$ threshold. $|\overline{V}|$ is the cardinality of $\overline{V}$. $Cloop$ is used to find a small coefficient of approximation to avoid trade-off between the original objective function of the model (minimizing total error) and distance approximation. In the second step, the sorting model is solved with the V and C_∞ ($C_{\infty,V}$) values. C_∞ is recorded as $C_{\infty,V}$ and aggressively increased to speed up the algorithm as in equation (158). In empirical studies, it is observed that this kind of aggressive increase in C_∞

do not decrease the solution quality but improves the solution time significantly. In this study, $C_{\infty,V}$ is determined such that:

$$C_{\infty,V}(Cloop) = \frac{C_{\infty,V-1} + Cloop * |\bar{V}[Vloop]| * 100}{10^2} \tag{158}$$

This formulation is determined by empirical study. In step 3, the feasibility of L_∞ distance is checked. If $FC = 0$ condition is true, then with a small $C_{\infty,V}$ coefficient the correct a_i^∞ values are found, and the algorithm can iterate to the next V value. If the feasibility condition is false, then the $C_{\infty,V}$ value is increased as in step 2 until a sufficiently large $C_{\infty,V}$ is found. However, different $C_{\infty,V}$ formulations can be used based on the choice of analyst or DM. In this way, a set of different L_p distances are approximated and solutions for those L_p distances are explored.

ATLAS Algorithm solves two important problems addressed in Chapter 1. Firstly, it explores a set of L_p distances iteratively by solving a number of computationally inexpensive LP models instead of computationally expensive NLP models. This is done by looping through a set of V values. By looping through the $C_{\infty,V}$ coefficients from low to high values, it finds a small coefficient of weighted Tchebycheff distance that is sufficiently large to satisfy $a_i^\infty = \max_j\{w_j|A_{ij} - I_j|\}\ \forall i.$ Computationally expensive NLP models are approximated as LP models and this can reduce the solution time, significantly. ATLAS explores solutions of a set of L_p distance approximations. Therefore, obtaining the solutions of different approximated L_p distances in a short time is a solution to the problem of determining which L_p distance to use. Still, which V value approximates which L_p distance is not known. This problem is handled with one of alternative courses of actions, namely BALA in Section 6.4.

6.3.2 Application to DISWOTH

ATLAS is also applied to DISWOTH [6] that is a nearest centroid-based classifier. Nearest centroid-based classifiers also require the usage of distance functions. In

DISWOTH, class centroids are employed to represent the classes. Class assignment of DISWOTH method is performed based comparison of alternatives to the class centroids. An alternative is assigned to class q if the centroid of class q is the closest centroid to the alternative.

Let us briefly recall the notation related to DISWOTH method. μ_j^q represents the j^{th} element of centroid of class q. μ_j^q is estimated with arithmetic average as in equation (159). ϵ_i is the error of class assignment. e_i returns zero if the closest centroid to A_i^q is μ_j^q and otherwise it returns a positive value. o is an infinitesimal positive scalar.

$$\mu_j^q = \frac{1}{|C^q|} \sum_{i \in C^q} A_{ij}^q \;\; \forall j, \forall q \tag{159}$$

DISWOTH model for a predetermined L_p distance is as follows.

(DISWOTH)

$$Minimize\ z = \sum_q \frac{\sum_{i \in C^q} e_i}{|C^q|} \tag{160}$$

Subject to:

$$e_i - \epsilon \geq \left|\left|A_i^q - \mu^q\right|\right|_p - \left|\left|A_i^q - \mu^r\right|\right|_p \; \forall q \neq r, \forall i \tag{161}$$

$$\left|\left|A_i^q - \mu^r\right|\right|_p = \sqrt[p]{\sum_j w_j^p \left|A_{ij}^q - \mu_j^r\right|^p} \;\; \forall i, \forall r \tag{162}$$

Constraints (148)-(149)

$$e_i \geq 0 \; \forall i \tag{163}$$

Objective function (160) minimizes the class weighted classification error. Constraint (161) performs the class assignment and constraint (163) is the sign constraint of e_i. $\left|\left|A_i^q - \mu^q\right|\right|_p$ and $\left|\left|A_i^q - \mu^r\right|\right|_p$ in constraint (161) can be approximated with equation (164). Equation (162) is replaced with equation (164). a_i^∞ can be approximated with constraint (165) and objective function is updated as (166). This version of the DISWOTH is named ApproximatedDISWOTH.

$$\left|\left|A_i^q - \mu^q\right|\right|_p \cong Va_i^\infty + (1-V)\sum_j w_j\left|A_{ij}^q - \mu_j^q\right| \;\forall q, \forall i \tag{164}$$

$$a_i^\infty \geq w_j\left|A_{ij}^q - \mu_j^r\right| \;\forall j, \forall r, \forall i \tag{165}$$

$$Minimize\ z = \sum_q \frac{\sum_{i\in C^q} e_i + C_{\infty,V} a_i^\infty}{|C^q|} \tag{166}$$

Approximation of DISWOTH method with ATLAS is the same as in DS. "*ApproximatedDS"* expression in step 2 of ATLAS is replaced by "*ApproximatedDISWOTH"*. Every explanation on the ATLAS algorithm for DS is also valid for DISWOTH. There is no $T^*(V)$ in ATLAS for DISWOTH. For computation of $T_rA(V)$ and $T_sA(V)$, $e_{i_r}^+, e_{i_r}^-$ and $e_{i_s}^+, e_{i_s}^-$ expressions of DS are replaced by e_{i_r} and e_{i_s}, respectively. To compute $T_sA(V)$, find the error variables of the test data by supplying parameters V and $w^*(V)$ to TestCalculationModel-ApproximatedDISWOTH in Appendix E for ApproximatedDISWOTH model. Solving TestCalculationModel-ApproximatedDISWOTH is not an optimization problem, since $V, w^*(V)$ are known parameters except for e_{i_s}. It is only used for computation of e_{i_s}.

6.4 Alternative Courses of Actions and Implementation Plan

To present a full guideline on the efficient usage of ATLAS, three alternative courses of actions related to implementation are presented. The alternative courses of actions are also used to avoid from overfitting. The actions are solely based on training accuracy of the ATLAS outputs. The test accuracy is computed based on the alternative courses of actions.

To be able to clearly explain the application of alternative courses of actions, a numerical example is given. Consider the following example solution of ATLAS algorithm that is for DS model for a hypothetical data set in Table 6.1. ATLAS algorithm is run for $V \in \{0.1, 0.5, 0.9\}$. The hypothetical data set has two classes and three criteria. Numerical examples are given using this example for alternative courses of actions. Computations in the alternative courses of actions are performed after ATLAS is applied.

Table 6.1 A numerical example for the alternative courses of actions

V Value	$w^*(V)$ for 3 criteria	$T^*(V)$ for 2 classes	$T_rA(V)$
0.1	$0.2, 0.1, 0.7$	3	78
0.5	$0.4, 0.2, 0.4$	2	79
0.9	$0.2, 0.3, 0.5$	5	72

Best of All Action (BA)

The first action is the Best of all Action (BA). In BA, the output of the ATLAS algorithm is the best training accuracy giving V value. Therefore, in implementation for DS to determine which $w^*(V) and\, T^*(V)$ are to be used (only $w^*(V)$ for DISWOTH), $T_rA(V)$ is used. BA action is applied as follows. First, V' value that satisfies $V' = arg\, \max_V \{T_rA(V)\}$ is chosen. Then, $w^*(V')$ and $T^*(V')$ are used in test accuracy calculation and $T_sA(V')$ is reported to DM. This action can be a greedy approach and may result in a poor test accuracy ($T_sA(V')$). For BA, on the numerical example, $T_sA(V)$ is computed using $V' = 0.5$, $w^*(V') = [0.4, 0.2, 0.4]$ and $T^*(V') = 2$.

Smoothing Action (SA)

The second action is the Smoothing Action (SA). In SA, the output of the ATLAS algorithm is the training accuracy weighted V' value. It results in aggregated weight (w) and threshold (T) instead of the best V, $w^*(V)$ and $T^*(V)$ in BA.

$T_sA(V')$ is computed with V', w and T. This action is developed to smooth out overfitting results. To find training accuracy weighted V' value, training accuracy output of each V value is normalized. $\varphi(V')$ denotes the normalized $T_rA(V')$ and formulated as $\varphi(V') = \frac{T_rA(V\prime)}{\sum_V T_rA(V)}$.

The criterion weights and thresholds are computed as $w = \sum_V \varphi(V)w^*(V)$ and $T = \sum_V \varphi(V)T^*(V)$ and $V' = \sum_V \varphi(V)V$. Then, by using w, T and V', compute the test accuracy to be reported to DM ($T_sA(V')$). For DISWOTH, to compute the $T_sA(V')$, T is not needed. On the numerical example, $\varphi(0.1) = \frac{78}{78+79+72} = 0.3406$, $\varphi(0.5) = \frac{79}{78+79+72} = 0.345$, $\varphi(0.9) = \frac{72}{78+79+72} = 0.3144$.

For SA, parameters that are used to compute the $T_sA(V')$ are computed as follows.

Update V' value as $V' = 0.3406 * 0.1 + 0.345 * 0.5 + 0.3144 * 0.9 = 0.489$.

Update Criterion weights as $w = 0.3406 * [0.2, 0.1, 0.7] + 0.345 * [0.4, 0.2, 0.4] + 0.3144 * [0.2, 0.3, 0.5] = [0.269, 0.229, 0.532]$.

Update Class threshold as $T = 0.3406 * 3 + 0.345 * 2 + 0.3144 * 5 = 3.2838$.

Best Accuracy* $\mathrm{L_p}$ *Approximation Action (BALA)

The third action is Best Accuracy L_p approximation Action (BALA). In BALA, the outputs of the ATLAS algorithm are used with the original NLP sorting models. $w^*(V)$ and $T^*(V)$ are supplied as parameters into DS (or DISWOTH) models (originals models, not approximated ones) with different L_p distances. $e^+_{i_r}$ and $e^-_{i_r}$ are computed for given $w^*(V)$ and $T^*(V)$ for L_p. Training accuracy for each V, $w^*(V)$ and $T^*(V)$ and L_p distance is computed with (156). Not that this step does not solve computationally expensive NLP models because the decision variables of those NLP models are given as parameters to compute error variables.

$w^*(V)$ and $T^*(V)$ outputs of each V value may result in different training accuracy with each L_p distance. The training accuracy table $T_rA(V, p)$ is obtained. For each L_p distance, the highest training accuracy giving V' is obtained as $(V', p) = \underset{V}{argmax}\{T_rA(V, p)\}$.

After obtaining (V', p) pair for all p, $w^*(V')$ and $T^*(V')$ are used for computing the test accuracy for the L_p distance. The result is $T_sA(p)$ table BALA can be recommended when L_p distance has a meaning for DM or the analyst conducting the study. BALA is explained as follows for DS model.

BALA Action:

Step 1: Apply ATLAS algorithm to obtain $w^*(\bar{V})$ and $T^*(\bar{V})$.

Step 2: For all p and V, supply $w^*(V)$, $T^*(V)$ to L_p distance-based sorting model and solve for $e_{i_r}^+$ and $e_{i_r}^-$ (error of training data). It should be noted that this step is not an optimization since all decision variables of the DS model are supplied as parameters except for errors. This step is just computation of errors w.r.t given $p, w^*(V')$ and $T^*(V')$. Obtain $T_rA(V, p)$ table.

Step 3: To obtain (V', p) pairs, apply $(V', p) = \underset{V}{argmax}\{T_rA(V, p)\}$ for all p. Use $w^*(V')$ and $T^*(V')$ in test accuracy computation to obtain $e_{i_s}^+$ and $e_{i_s}^-$ (error of test data). Calculate the test accuracy of L_p distance-based original NLP model using (157).

Step 4: Construct $T_sS, A(p)$ table to present to DM.

After constructing $T_sA(p)$ table, the best test accuracies for different L_p distances can be obtained. The accuracy values in this table can also be interpreted as test accuracy of heuristic solutions to the original model.

A new numerical example for BALA method is as follows. On the numerical example, assume $p \in \{1, 2, 3\}$ are to be used. Supply $w^*(0.1), w^*(0.5), w^*(0.9)$ and $T^*(0.1), T^*(0.5), T^*(0.9)$ to DS model with $p \in \{1, 2, 3\}$. Solve DS model with these weights for each p value to calculate $e_{i_r}^+$ and $e_{i_r}^-$. Obtain 3x3 $T_rA(V, p)$ matrix as in Table 6.2.

Table 6.2 An example $T_r A(V, p)$ table

Inputs\\p values	$p = 1$	$p = 2$	$p = 3$
$V = 0.1$, $w^*(0.1), T^*(0.1)$	**82**	71	70
$V = 0.5$, $w^*(0.5), T^*(0.5)$	70	**81**	75
$V = 0.7$, $w^*(0.7), T^*(0.7)$	68	72	**73**

Based on Table 6.2, the (V', p) pair can be found as in Table 6.3 as follows.

Table 6.3 Example of (V', p) pairs based on the example $T_r A(V, p)$

V'	p	$T_s A(p)$ is found with
0.1	1	$w^*(0.1), T^*(0.1)$
0.5	2	$w^*(0.5), T^*(0.5)$
0.9	3	$w^*(0.7), T^*(0.7)$

From $T_r A(V, p)$ table (Table 6.2), (V', p) is obtained as in Table 6.3. To compute $T_s A(p)$, use $w^*(0.1)$ and $T^*(0.1)$ for $p = 1$, use $w^*(0.5)$ and $T^*(0.5)$ for $p = 2$, use $w^*(0.9)$ and $T^*(0.9)$ for $p = 3$.

Implementation of ATLAS algorithm is based on those three alternative courses of actions. One can apply one of them. All of them can be applied and between the test results of three alternative courses of actions, the method with the best test accuracy can be chosen.

CHAPTER 7

EXPERIMENTS OF ATLAS METHOD

In this chapter, DS and DISWOTH model are solved for 10 different L_p distances for $p \in \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$. To approximate these distances, twenty-four different V values are used as $\bar{V} = [0,\ 0.05,\ 0.1,\ 0.15, \dots, 0.95, 1, -3, -6, -9]$ based on the empirical and theoretical foundations in Section 6.2.1. The original NLP DS and DISWOTH results are compared with the ATLAS results based on the test accuracy and the solution time performance measures.

Software and Hardware Setting

The software and the hardware setting are the same as in Chapter 5. Due to constraint tolerance of the software, feasibility check step of ATLAS (step 3) is performed using $FC < 10^{-9}$ instead of $FC = 0$.

Datasets

The data sets used to evaluate the performance of methods in this study and assumptions for maximization and minimization criteria are the same as in Section 5.1.1.

7.1 Experiments of DS Model

Test accuracy for each p and total training time for all selected p values are reported in Table 7.1. Also, average test accuracy of all p values for each data set and average training time are calculated. On average, it takes 23794.03 seconds to train DS model for a data set for 10 different p values. Training time of each data set for each p value for DS model is reported in Table 10.10 in Appendix C.

Table 7.1 DS model results for each p value

p	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**	**3.5**	**4**	**4.5**	**5**	**Average Accuracy**	**Training Time**
AUTOMPG	80.25	86.42	87.65	**88.89**[11]	88.89	86.42	85.19	83.95	83.95	82.72	85.96	46373.81
BC	91.38	**94.83**	70.69	70.69	70.69	70.69	67.24	70.69	70.69	70.69	70.26	20534.67
CAR	82.61	85.80	89.57	91.88	92.46	94.20	94.49	94.49	**95.07**	94.49	93.33	4169.66
CCS	75.86	79.31	77.83	74.88	78.33	79.80	79.31	**81.28**	80.30	80.30	79.00	105017.30
CPU	**86.05**	86.05	86.05	86.05	86.05	86.05	86.05	86.05	86.05	86.05	86.05	34226.58
ESL	94.79	**95.83**	93.75	93.75	93.75	90.63	89.58	91.67	90.63	91.67	91.93	210.34
LEV	**84.34**	83.84	78.79	78.28	76.26	77.78	73.74	75.25	74.24	74.75	76.14	412.04
ASA	**97.77**	97.77	97.77	97.77	97.77	97.77	97.77	97.77	97.77	97.77	97.77	1.76
MMG	50.00	50.00	76.88	77.50	**79.38**	78.75	78.75	78.13	78.13	78.13	78.20	3200.95
Average	**82.56**	**84.43**	**84.33**	**84.41**	**84.84**	**84.68**	**83.57**	**84.36**	**84.09**	**84.06**	**84.13**	**23794.03**

[11] Boldface entries are the highest test accuracy observed for the data set in the same row.

ATLAS results for BA and SA

Experimental results of ATLAS method are reported in Table 7.2. On average, it takes 6.2 seconds to train ApproximatedDS model with ATLAS for a data set for 24 different V values. *Maximum* total training time of ATLAS for DS is 11.48 seconds (CAR data set) while *minimum* total training time for DS is 210.34 seconds (ESL data set). Training time of ATLAS for all V values and number of iterations to satisfy FC condition are reported in Table 10.12 in Appendix D. A significant improvement in training time is observed.

The average test accuracy for DS is 84.13 as reported in Table 7.1. In Table 7.2, BA and SA approaches result in 85.60 and 80.45 average test accuracy, respectively. Average test accuracy is better than original DS for BA action and worse than original DS for SA action. SA approach may decrease the test accuracy in some cases (e.g., BC data set test accuracy result). It is specifically designed for overfitting issue. As an example, test accuracy and BA result of MMG data set can be seen. Test accuracy of MMG significantly increase when SA is applied instead of BA. For the results similar to MMG, SA may be used instead of BA. More than 99.9 average time improvement is observed with 1.47 (3.68) improvement (loss) in average test accuracy for BA (SA). In 5 (two) out of 9 experiments, BA (SA) results are better than the best test accuracy obtained by DS.

Table 7.2 ATLAS algorithm results for Approximated DS

Approximated DS	**BA**	**SA**	**Total Training Time**[12]
AUTOMPG	86.42[13]	90.12*[14]	6.95
BC	100.00*	62.07	4.64
CAR	96.23*	83.48	11.48
CCS	79.31	72.41	4.95
CPU	86.05*	86.05*	4.35
ESL	97.92*	87.50	4.63
LEV	88.89*	83.84	5.83
ASA	97.77	96.65	1.88
MMG	50.00	78.13	6.77
Average	85.60*	80.45	5.72

[12] Training Time column is the total time to solve all LP models for 24 V values.

[13] Red colored entries are better than average DS results for each data set.

[14] Entries with "*" are greater than highest test accuracy observed with DS model.

ATLAS results for BALA

BALA results are reported in Table 7.3. Table 7.3 is a $T_sA^*(p)$ table for DS. In the experiments of AUTOMPG and BC data sets, BALA results with better accuracy 9 out of 10 experiments. Except for CAR data set, BALA results in better test accuracy for at least one distance function. 24 out of 90 experiments (the ones with the * sign), BALA accuracies are better than the highest accuracy obtained by DS. On the average test accuracy, for all p values (Average row), BALA results with better accuracies for six different p values. On the average test accuracy of data sets (Average column), BALA results with better average accuracy for four out of 9 data sets. On average, BALA test accuracy is 83.50 while DS test accuracy is 84.13. Less than 1 loss is observed in average test accuracy. This is a promising result since the time improvement is more than 99.9.

Table 7.3 BALA experimental results of ATLAS Algorithm for DS model

p	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	Average
AUTOMPG	92.59*[15,16]	86.42	86.42	90.12*	90.12*	86.42	90.12*	87.65	90.12*	90.12*	89.01*
BC	82.76	94.83*	91.38	91.38	91.38	91.38	91.38	91.38	91.38	91.38	90.86
CAR	77.68	85.22	86.09	86.38	87.25	88.41	88.70	88.12	88.12	88.12	86.41
CCS	46.80	79.31	82.27	66.01	75.86	75.86	63.05	72.91	74.88	72.91	70.99
CPU	79.07	86.05	83.72	74.42	83.72	81.40	67.44	81.40	55.81	62.79	75.58
ESL	76.04	97.92*	87.50	86.46	89.58	88.54	89.58	89.58	84.38	88.54	87.81
LEV	80.30	83.84	82.83	83.84	80.30	78.28	76.26	73.74	73.74	73.74	78.69
ASA	97.77*	97.77*	97.77*	97.77*	97.77*	97.77*	97.77*	97.77*	97.77*	97.77*	97.77*
MMG	82.50*	50.00	76.88	76.88	77.50	75.63	77.50	75.00	75.00	76.88	74.38
Average	79.50	84.60	86.10*	83.70	85.94*	84.85*	82.42	84.17	81.24	82.47	83.50

[15] Red colored entries are better than average DS results for each data set.

[16] Entries with “ *” are greater than highest test accuracy observed with DS model

7.2 Experiments of DISWOTH Model

DISWOTH model is also solved for 9 data sets and 10 different p values as DS. Test accuracy for each p and total training time for all selected p values are reported in Table 7.4. Also, average test accuracy of all p values for each data set and average training time are calculated to compare ATLAS with average accuracy value and average time. On average, it takes 77605.08 seconds to train DISWOTH model for a data set for 10 different p values. Training time of each data set for each p value for DISWOTH model is reported in Table 10.11 Appendix C.

Table 7.4 DISWOTH model results for each p value

p	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**	**3.5**	**4**	**4.5**	**5**	**Average Accuracy**	**Training Time**
AUTOMPG	72.84	79.01	75.31	**80.25**[17]	80.25	76.54	79.01	79.01	79.01	80.25	77.91	68205.71
BC	37.93	62.07	58.62	46.55	46.55	48.28	46.55	44.83	44.83	44.83	48.10	61032.58
CAR	41.16	61.74	70.72	60.78	77.39	79.42	80.00	81.45	82.90	**83.77**	71.93	122053.10
CCS	48.28	48.28	64.04	56.16	**67.49**	59.11	64.53	62.56	61.58	59.11	59.11	100814.38
CPU	79.07	**81.40**	79.07	81.40	79.07	79.07	81.40	81.40	81.40	81.40	80.47	70389.99
ESL	83.33	87.50	90.63	90.63	90.63	89.58	**92.71**	90.63	92.71	92.71	90.11	57858.25
LEV	68.69	**79.80**	78.28	74.75	71.21	69.69	67.68	68.18	68.69	69.19	71.62	2011.80
ASA	90.50	**94.97**	92.18	92.18	93.30	93.30	93.30	93.30	92.18	92.18	92.74	129600.13
MMG	**76.25**	31.88	29.38	29.38	74.38	38.75	74.38	73.75	73.75	74.38	57.63	86479.80
Average	**66.45**	**69.63**	**70.91**	**66.48**	**75.59**	**70.42**	**75.51**	**75.01**	**75.23**	**75.31**	**72.18**	**77605.08**

[17] Boldface entries are the highest test accuracy observed for the data set in the same row.

ATLAS results for BA and SA

Experimental results of ATLAS method are reported in Table 7.5. On average, it takes 50.99 seconds to train Approximated DISWOTH model with ATLAS for a data set for 24 different V values. Number of iterations to satisfy FC condition are reported in Table 10.13 in Appendix D. The average test accuracy for DISWOTH is 72.18 as reported in Table 7.5. BA and SA approach result in 74.8 and 78.12 test accuracy, respectively. Average test accuracy is better than DISWOTH for both BA and SA actions. More than 99.9 improvement in average solution time is observed with 2.62 (5.94) improvement is observed with BA (SA) action. In DISWOTH experiments, accuracy loss is not observed in average test accuracy results.

Table 7.5 ATLAS algorithm results for Approximated DISWOTH

Approximated DISWOTH	BA	SA	Total Training Time[18]
AUTOMPG	83.95*[19, 20]	86.42*	8.16
BC	70.69*	79.31*	7.82
CAR	68.99	60.58	378.1
CCS	72.91*	75.86*	14.05
CPU	81.40	83.72*	3.36
ESL	89.58	85.42	3.39
LEV	78.79	82.32*	11.51
ASA	94.97*	73.18	28.6
MMG	31.88	76.25	3.93
Average	74.80	78.12	50.99

[18] Training Time column is the total time to solve all LP models for 24 V values.

[19] Red colored entries are better than average DISWOTH results for each data set.

[20] Entries with "*" are higher than highest test accuracy observed with DISWOTH model

ATLAS results for BALA

BALA results are reported in Table 7.6. Table 7.6 is a $T_s A^*(p)$ table for DISWOTH. For BC data set, BALA always results better test accuracy than DISWOTH. In the experiments of CCS and MMG data sets, BALA results with better accuracies 9 out of 10 p values. Except for CAR and ASA data sets, BALA results in better test accuracy for at least 5 out of 10 p values. 52 out of 90 experiments (the ones with the * sign), BALA accuracies are better than the highest accuracy obtained by DISWOTH. On the average test accuracy, for all p values (Average row), BALA always results with better accuracies. 9 out of 10 results, average BALA accuracies are better than the best average result obtained by DISWOTH. On the average test accuracy of data sets (Average column), BALA results with better average accuracy for six out of 9 data sets. On average, BALA test accuracy is 77.27 while DISWOTH test accuracy is 72.18. More than 99.9 average time improvement is observed with more than 5 gain in average test accuracy.

Table 7.6 BALA experimental results of ATLAS Algorithm for DISWOTH model

p	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	Average Accuracy
AUTOMPG	80.25*[20,21]	87.65*	87.65*	82.72*	81.48*	81.48*	81.48*	80.25*	77.78	79.01	81.98*
BC	74.14*	72.41*	77.59*	72.41*	72.41*	67.24*	72.41*	65.52*	65.52*	65.52*	70.52*
CAR	50.72	61.74	66.67	68.99	71.88	71.88	71.59	70.72	70.72	70.72	67.57
CCS	66.50	48.28	76.85*	77.34*	77.34*	75.86*	70.94*	70.94*	71.43*	70.94*	70.64*
CPU	81.4*	81.4*	83.72*	83.72*	83.72*	83.72*	81.4*	79.07	79.07	81.40	81.86*
ESL	84.38	89.58	89.58	87.50	92.71*	92.71*	92.71*	92.71*	89.58	89.58	90.10
LEV	79.8*	83.84*	77.27	73.74	72.73	72.73	71.21	71.72	69.70	68.18	74.09
ASA	89.39	94.97	91.06	93.30	93.85	94.97*	93.85	92.74	93.30	93.85	93.85
MMG	77.5*	31.88	76.25*	76.25*	76.25*	76.25*	76.25*	76.25*	76.25*	76.25*	71.94*
Average	76.01*	76.01*	72.42	80.74*	79.55*	80.26*	79.65*	79.09*	77.77*	77.04*	77.27*

[20] Red colored entries are better than average DISWOTH results for each data set.

[21] Entries with " *" are higher than highest test accuracy observed with DISWOTH model.

7.3 Comparison of ATLAS with UTADIS

The UTADIS results in Chapter 5 are also used in this section. ATLAS for DS and DISWOTH are compared with UTADIS. To compare BALA with UTADIS, average test accuracy of each data set is used (Average column in Tables 7.3 and 7.6). Test accuracy and training time of UTADIS are reported in Table 7.7. To ease the comparison, BA, SA, and BALA (average column) results are added to Table 7.7. There are only six test accuracy results in three data sets that are worse than UTADIS. The training time is worse than UTADIS for all data sets. This is expected since only a single LP model is solved for UTADIS while 24 of LP models are solved for ATLAS. Although the solutions that are obtained with ATLAS are results of approximations, the test accuracy is significantly higher than UTADIS.

Table 7.7 Comparison of ATLAS with UTADIS

	UTADIS		ATLAS results of DS				ATLAS results of DISWOTH			
Data sets	**Test Accuracy**	**Training Time**	**BA**	**SA**	**BALA**	**Training Time**	**BA**	**SA**	**BALA**	**Training Time**
AUTOMPG	77.78	0.12	86.42	90.12	89.01	6.95	83.95	86.42	81.98	8.16
BC	50.00	0.10	100.00	62.07	90.86	4.64	70.69	79.31	70.52	7.82
CAR	57.39	0.61	96.23	83.48	86.41	11.48	68.99	60.58	67.57	378.10
CCS	37.44	0.87	79.31	72.41	70.99	4.95	72.91	75.86	70.64	14.05
CPU	76.74	0.11	86.05	86.05	**75.58**[22]	4.35	81.40	83.72	81.86	3.36
ESL	79.17	0.17	97.92	87.50	87.81	4.63	89.58	85.42	90.10	3.39
LEV	68.69	0.21	88.89	83.84	78.69	5.83	78.79	82.32	74.09	11.51
ASA	93.85	0.53	97.77	96.65	93.85	1.88	94.97	**73.18**	93.85	28.60
MMG	76.25	0.19	**50.00**	78.13	**74.38**	6.77	**31.88**	76.25	**71.94**	3.93
Average	68.59	0.32	86.95	82.25	83.07	5.72	74.80	78.12	78.06	50.99

[22] The results that are worse than UTADIS is shown with boldface.

CHAPTER 8

DISCUSSION

In this chapter, the proposed methods and the results are discussed. Three different topics are studied in this thesis. Comments and discussion on the topics studied are as follows.

8.1 Discussions on Linearization of DISWOTH and L_p-Centroid

The first study is the linearization of DISWOTH with MINLP approximation and improving accuracy with L_p-Centroid. It is proven that it can also be converted to MIP. The MIP model is named Bin-Dis. Bin-Dis is advantageous in terms of both time and training accuracy. The time improvement is intuitive due to linearization. The accuracy improvement is also an intuitive result. Because the error formulation of DISWOTH is continuous (a nonnegative continuous variable). However, the definition of an erroneous class assignment is binary. In Bin-Dis, the error variable is defined as a binary variable as accurate and inaccurate. It can be inferred from the formulations of DISWOTH and Bin-Dis that minimizing the error with a continuous variable is an indirect way to minimize the number of errors. Bin-Dis minimizes the number of errors directly. This can be interpreted in terms of means and ends objective.

The maximum accuracy is formulated as the minimum the number of errors in Chapter 5 equations (101)-(103). Minimizing the continuous error variables in DISWOTH is a means objective to reach maximum accuracy. Minimizing the number of errors in Bin-Dis is ends objective to maximize accuracy and also the fundamental objective.

L_p-Centroid [25] is employed as a centroid formulation. The problem of choosing an appropriate centroid-distance pair is also solved with a heuristic algorithm (DC algorithm). Employing the L_p-Centroid, improved the test accuracy. This is also an intuitive result based on the observations of Tian et al. [25]. The method is designed to regularize the centroid-based learning methods (regularization is done by improving the test accuracy). It improves the test accuracy of DISWOTH as it is intended.

When L_p-Centroid and Bin-Dis are used together, namely, Bin-L_pDis, the results are the best on average. Considering ordering of classes provides an insignificant benefit to the accuracy. This result is also intuitive based on the study of Ben-David et al. [21].

8.2 Discussions on Monotonically Ordered Centroids Case

For the DISWOTH method, when the centroids are monotonically ordered, it is proven that there are redundant alternatives. It is shown that these redundancy relations can be formulated with linear expressions. These linear expressions are functions of centroids, and they work for all L_p distances. With a reverse engineering, a new LP model (AIRO) is developed to find monotonically ordered centroids that works for all L_p distances.

If the centroids were decision variables in DISWOTH, it would be highly nonlinear. Solving such model is computationally expensive. The AIRO model is an LP model that results in monotonic centroids for the DISWOTH (or NC) method. The fact that the formulation is an LP allows the analyst or DM to analyze the method for different L_p distances in very short amount of time.

The economic interpretation of the monotonic order is related to the preference-order. Since the classes are in ordinal scale and preference ordered, ordering the class representatives that are so-called class centroids is a strong assumption. The monotonic order is an indication of strict dominance. In any type of preference function, a strictly monotonic relationship is a direct indication of the preference. If the alternative A strictly dominates another alternative B, on any occasion, A is preferred to B.

The economic interpretation of monotonic centroids is intuitive. The classification accuracy is not as high as other methods that are proposed in this thesis (e.g., Bin-L_pDis). The weight estimation model (WE) may not be a good or correct way of estimating criterion weights. However, the benefit of linearization of DISWOTH is observed in the experiments. According to trade-off table, AIRO-ECW results in better accuracy and solution time compared with DISWOTH. Results of AIRO-WE is worse than the AIRO-ECW. This means that there may be a better way of estimating the criterion weights that maximizes the classification accuracy.

8.3 Discussions on the ATLAS method

The third topic studied in this thesis is the linearization of all L_p distances with a single formula that is an approximation. ATLAS is an octagonal LP approximation to NLP distance-based MP approaches. Although examples and experiments are restricted with MP-based multicriteria sorting models, it can be used in any kind of distance-based method that requires the usage of L_p distance.

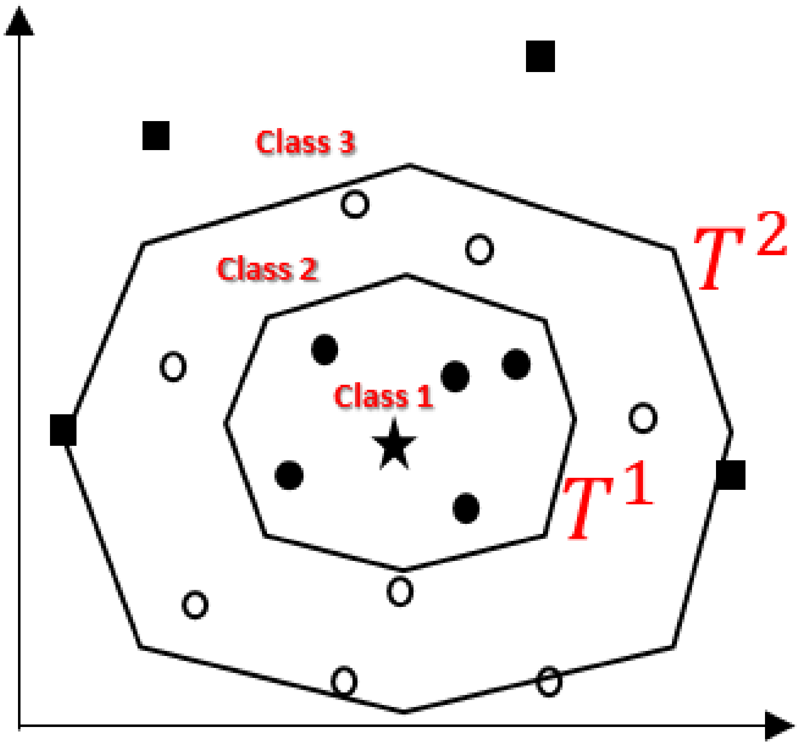
The analyses on the new approximation are conducted with empirical and theoretical studies. An analysis is conducted to examine the characteristics of the approximation method. The Augmented Tchebycheff formulation can be converted to L_p distance approximation with little effort due to its simplicity. Although the formulation is quite simple, the benefits are noticeable. Besides, the characteristics of the approximation used in the ATLAS method is consistent with L_p distances. A set of ATLAS parameters can be supplied to the method and a set of L_p distances can be approximated in a noticeably short time.

As the benefit of linearization, solution time decreases significantly. This is an intuitive result. Besides the time improvement, test accuracy of the new approximation is also higher than the original methods in most cases.

The outputs of the ATLAS method can be interpreted in different ways. In the experiments, three different interpretations of outputs are introduced as alternative courses of actions. Based on the interpretation, purposeful actions or formulations can be studied. In our example, we have developed a greedy approach (BA), a conservative aggregated approach (SA) and another approach that can be interpreted as a heuristic approach to distance-based models (BALA).

As an example, to many possible extensions, a voting mechanism can be developed based on the outputs of the ATLAS method. In our experiments, since 24 different V values are used, there are 24 outputs of the ATLAS method. One output (i.e., say the outcome of $V = 0.1$) can classify an alternative into class 1 and another output can classify the same alternative into class 2 (i.e., say the outcome of $V = 0.3$). This class assignments can be considered as votes. Out of 24 outputs, the number of times that a specific alternative is voted for a specific class can be counted. This is the same as counting the number of votes for the class assignment of the alternative. DM can be informed about the possible class assignments (number of votes for each class) for that alternative. Informing DM about the possible outcomes provides a clear perspective on the valuation of the alternative.

Other than being an approximation method, ATLAS can also be considered as the first Augmented Tchebycheff program based multicriteria sorting method in the literature. The unique property of this method is that it can draw an octagonal classification decision boundary around a reference point. The classification decision boundary of ATLAS is illustrated in criterion space in Figure 8.1 for a two criteria problem and three-class problem.



Black circles, white circles and squares are alternatives of classes one, two and three, respectively. The star on the center is a ICV. T^1 and T^2 are the class thresholds that separate classes one-two and two-three, respectively.

Figure 8.1 Decision boundary of classification of ATLAS

CHAPTER 9

CONCLUSION

In this thesis, three studies are conducted for distance-based multicriteria sorting. Based on the studies, new methods are proposed. Experiments are conducted on different application areas to examine the applicability of the new methods in different areas.

The first study is based on DISWOTH which is a nearest centroid type of sorting method. In this study, three new nearest centroid-based multi-criteria sorting methods are developed as extensions of DISWOTH. The proposed methods are linearized to improve solution time and classification accuracy. Linearization is based on the MINLP formulation of the existing NLP DISWOTH model. L_p-Centroid is employed to improve the classification accuracy. Compromise Ranking and Additive Difference Model constraints are also added to assure monotonicity of class centroids. The models are regularized with monotonicity constraints that seek monotonic alternative solutions. The models are solved for training data and alternatives of test data are assigned to the class of the closest predetermined class centroid by using optimal criterion weights of training data.

Model performance is evaluated over the solution time and test accuracy results. Experiments are conducted on 9 data sets from different application areas. Additional tests and evaluations are performed to rank the methods from best to worst. Bin-Dis, L_pDis and Bin-L_pDis methods are compared with UTADIS and DISWOTH with L_1, L_2, L_3 and L_{p*}.

Results indicate that the solution times of NLP models significantly decrease after linearization with binary variables in addition to improvement in all of training accuracies and average test accuracies. Bin-L_pDis and its extensions have the highest average training accuracy and test accuracy. Experiments show that the Bin-L_pDis method is both time and accuracy effective method when compared with UTADIS and DISWOTH methods. TOPSIS ranked Bin-L_pDis as the best method based on average rankings considering both classification accuracy and solution time performance criterion.

For future work, linearization of DISWOTH with a binary variable can be extended to all p norms, and similar nearest centroid-based NLP models can be considered with binary variables to benefit from linearization. Bin-L_pDis can be studied further for different data sets with different sampling techniques and training sizes and can be applied to real life problems.

The second study focuses on the monotonically ordered centroids case of the first study. It is proven that there are conditions that a limited set of alternatives can be redundant when the centroids are monotonically ordered. The redundancy formulation is linear. A linear programming model is developed based on the redundancy formulation. The new method is compared with DISWOTH and UTADIS as in the first study. In the experiments and discussion, it is reported that considering monotonicity (strict dominance) does not improve the classification accuracy. The weight estimation method results in worse test accuracy than the equal weights case. However, linearization benefits the solution time. The effect of this benefit is observed with the accuracy-time trade-off and multicriteria ranking. Also, economic interpretation of the monotonic order is given. This study can be extended by applying different mathematical programs based on AIRO and different weight estimation techniques.

In the third study, a new linear L_p distance approximation method is developed based on Augmented Tchebycheff program and Chaudhuri et al.' s formulation. The proposed method is analyzed to explain the characteristics. Metricity conditions are presented and shown that they are consistent with L_p distance. The analyses provide a full guideline for the user. It is shown that the new method can be adapted to mathematical programming. The proposed method is adapted to distance-based multicriteria sorting via an algorithm, namely ATLAS. ATLAS algorithm is developed for application of the new approximator to Multicriteria Sorting problems that are based on mathematical programming and distance functions. Although it is applied to multicriteria sorting as an example, it can also be applied to other mathematical programming and multicriteria decision making settings. Examples can be distance-based ranking methods, multi-objective optimization, and data mining methods. ATLAS provides a linear approximation for the distance-based nonlinear programming models. Three alternative courses of actions are developed to fully benefit from outputs of the method. One of the actions (SA) are specifically designed to protect decision maker from overfitting issue.

Experiments are conducted to compare the original distance-based sorting methods with their approximations based on test accuracy and training time performance measures. When compared with the first two studies, more distance functions are used in experiments. Experimental results show that ATLAS is a time effective method as it is computationally inexpensive. On average, test accuracy results of the ATLAS method are better than the results of original distance-based NLP sorting models. To sum up, the new linear approximation and ATLAS significantly decrease the training time of distance-based nonlinear programming and increase the average test accuracy. Based on the outputs of the ATLAS, new alternative courses of actions can be developed. Test accuracy results of new alternative courses of actions can be examined. The parameter of approximation (V value) can be further analyzed.

To sum up, in all of three studies, improvements in computation times are obtained as a result of linearization according to the experimental results.

REFERENCES

[1] C. Zopounidis and M. Doumpos, “Multicriteria classification and sorting methods: A literature review,” *European Journal of Operational Research*, vol. 138, no. 2, pp. 229–246, 2002.

[2] M. Luque, F. Ruiz, and R. E. Steuer, “Modified interactive Chebyshev algorithm (MICA) for convex multiobjective programming,” *European Journal of Operational Research*, vol. 204, no. 3, pp. 557–564, 2010.

[3] J. M. Devaud, G. Groussaud, and E. Jacquet-Lagreze, “UTADIS: Une méthode de construction de fonctions d’utilité additives rendant compte de jugements globaux,” *European Working Group on Multicriteria Decision Aid, Bochum*, vol. 94, 1980.

[4] O. Sobrie, V. Mousseau, and M. Pirlot, “Learning monotone preferences using a majority rule sorting model,” *International Transactions in Operational Research*, vol. 26, no. 5, pp. 1786–1809, 2019.

[5] Y. Chen, K. W. Hipel, and D. M. Kilgour, “Multiple-criteria sorting using case-based distance models with an application in water resources management,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 5, pp. 680–691, 2007.

[6] E. Karasakal and M. Civelek, “A distance based multiple criteria sorting method without class thresholds,” *Journal of Multi-Criteria Decision Analysis*, vol. 28, no. 3–4, pp. 134–143, 2021.

[7] B. Çelik, E. Karasakal, and C. İyigün, “A probabilistic multiple criteria sorting approach based on distance functions,” *Expert Systems with Applications*, vol. 42, no. 7, pp. 3610–3618, 2015.

[8] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Ann Eugen*, vol. 7, no. 2, pp. 179–188, 1936.

[9] C. A. B. Smith, “Some examples of discrimination,” *Annals of Eugenics*, vol. 13, no. 1, pp. 272–282, 1946.

[10] E. Fix and J. L. Hodges, “Discriminatory analysis. Nonparametric discrimination: Consistency properties,” *International Statistical Review/Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.

[11] A. Srinivasulu, T. Barua, U. Neelakantan, and S. Nowduri, “Early Prediction of COVID-19 Using Modified Convolutional Neural Networks,” in *Machine Learning and Internet of Things for Societal Issues*, Springer, 2022, pp. 69–82.

[12] M. Keshavarz Ghorabaee, E. K. Zavadskas, L. Olfat, and Z. Turskis, “Multi-criteria inventory classification using a new method of evaluation based on distance from average solution (EDAS),” *Informatica*, vol. 26, no. 3, pp. 435–451, 2015.

[13] W. Yu, “ELECTRE TRI(aspects méthodologiques et manuel d’utilisation),” *Document- Université de Paris-Dauphine, LAMSADE*, 1992.

[14] M. Doumpos and J. R. Figueira, “A multicriteria outranking approach for modeling corporate credit ratings: An application of the Electre Tri-nC method,” *Omega (Westport)*, vol. 82, pp. 166–180, 2019.

[15] Y. Chen, K. W. Li, D. M. Kilgour, and K. W. Hipel, “A case-based distance model for multiple criteria ABC analysis,” *Computers & Operations Research*, vol. 35, no. 3, pp. 776–796, 2008.

[16] B. Çelik, “A multiple criteria sorting approach based on distance functions,” Middle East Technical University, 2011.

[17] M. Civelek, “A Mathematical programming evaluation approach for multiple criteria sorting problems,” Middle East Technical University, 2019.

[18] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A k-means clustering algorithm,” *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[19] Q.-B. Gao and Z.-Z. Wang, “Center-based nearest neighbor classifier,” *Pattern Recognition*, vol. 40, no. 1, pp. 346–349, 2007.

[20] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*, vol. 39. Cambridge University Press Cambridge, 2008.

[21] A. Ben-David, L. Sterling, and T. Tran, “Adding monotonicity to learning algorithms may impair their accuracy,” *Expert Systems with Applications*, vol. 36, no. 3, pp. 6627–6634, 2009.

[22] Q. Tian and S. Chen, “A novel ordinal learning strategy: Ordinal nearest-centroid projection,” *Knowl Based Syst*, vol. 88, pp. 144–153, 2015.

[23] Y. Liu, Y. Liu, and K. Chan, “Ordinal regression via manifold learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2011, vol. 25, no. 1, pp. 398–403.

[24] B.-Y. Sun, J. Li, D. D. Wu, X.-M. Zhang, and W.-B. Li, “Kernel discriminant learning for ordinal regression,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 6, pp. 906–910, 2009.

[25] Q. Tian, W. Zhang, L. Wang, S. Chen, and H. Yin, “Robust ordinal regression induced by lp-centroid,” *Neurocomputing*, vol. 313, pp. 184–195, 2018.

[26] K. Pelckmans, P. Karsmakers, and J. A. K. Suykens, “Ordinal Least Squares Support Vector Machines-A Discriminant Analysis Approach,” in *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, 2006, pp. 247–252.

[27] A. Tversky, “Intransitivity of preferences.,” *Psychol Rev*, vol. 76, no. 1, p. 31, 1969.

[28] A. Ishizaka and M. Gordon, “MACBETHSort: a multiple criteria decision aid procedure for sorting strategic products,” *Journal of the Operational Research Society*, vol. 68, no. 1, pp. 53–61, 2017.

[29] P. Nemery and C. Lamboray, "$\mathscr{F}$low $\mathcal {S} $ ort: a flow-based sorting method with limiting or central profiles," *Top*, vol. 16, no. 1, pp. 90–113, 2008.

[30] R. L. Keeney, "Developing a foundation for strategy at Seagate Software," *Interfaces (Providence)*, vol. 29, no. 6, pp. 4–15, 1999.

[31] A. P. Wierzbicki, "On the completeness and constructiveness of parametric characterizations to vector optimization problems," *Operations-Research-Spektrum*, vol. 8, no. 2, pp. 73–87, 1986.

[32] P. J. Korhonen, K. Silvennoinen, J. Wallenius, and A. Öörni, "Can a linear value function explain choices? An experimental study," *European Journal of Operational Research*, vol. 219, no. 2, pp. 360–367, 2012.

[33] A. Asuncion and D. Newman, "UCI machine learning repository." Irvine, CA, USA, 2007.

[34] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[35] M. el Amine Lazouni, M. el Habib Daho, N. Settouti, M. A. Chikh, and S. Mahmoudi, "Machine learning tool for automatic ASA detection," in *Modeling Approaches and Algorithms for Advanced Computer Applications*, Springer, 2013, pp. 9–16.

[36] A. Fallah Tehrani, W. Cheng, K. Dembczyński, and E. Hüllermeier, "Learning monotone nonlinear models using the Choquet integral," *Machine Learning*, vol. 89, no. 1, pp. 183–211, 2012.

[37] C.-L. Hwang and K. Yoon, "Methods for multiple attribute decision making," in *Multiple attribute decision making*, Springer, 1981, pp. 58–191.

[38] T. Pajala, P. Korhonen, and J. Wallenius, "Judgments of importance revisited: What do they mean?," *Journal of the Operational Research Society*, vol. 70, no. 7, pp. 1140–1148, 2019.

[39] D. Chaudhuri, C. A. Murthy, and B. B. Chaudhuri, “A modified metric to compute distance,” *Pattern Recognition*, vol. 25, no. 7, pp. 667–677, 1992.

[40] A. Charnes, W. W. Cooper, and R. O. Ferguson, “Optimal estimation of executive compensation by linear programming,” *Manage Sci*, vol. 1, no. 2, pp. 138–151, 1955.

[41] J. Kelley James E, “An application of linear programming to curve fitting,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 6, no. 1, pp. 15–22, 1958.

[42] F. Rhodes, “On the metrics of Chaudhuri, Murthy and Chaudhuri,” *Pattern Recognition*, vol. 28, no. 5, pp. 745–752, 1995.

[43] C. Seol and K. Cheun, “A low complexity Euclidean norm approximation,” *IEEE transactions on signal processing*, vol. 56, no. 4, pp. 1721–1726, 2008.

[44] J. Mukherjee, “Linear combination of norms in improving approximation of Euclidean norm,” *Pattern Recognition Letters*, vol. 34, no. 12, pp. 1348–1355, 2013.

[45] J. Mukherjee, “On approximating Euclidean metrics by weighted t-cost distances in arbitrary dimension,” *Pattern Recognit Lett*, vol. 32, no. 6, pp. 824–831, 2011.

[46] I. Barrodale and F. D. K. Roberts, “Applications of mathematical programming to ℓp approximation,” in *Nonlinear programming*, Elsevier, 1970, pp. 447–464.

[47] H. Ekblom, “Calculation of linear bestL p-approximations,” *BIT Numerical Mathematics*, vol. 13, no. 3, pp. 292–300, 1973.

[48] H. Ekblom, “L p-methods for robust regression,” *BIT Numerical Mathematics*, vol. 14, no. 1, pp. 22–32, 1974.

[49] J. Fischer, “An algorithm for discrete linearL p approximation,” *Numerische Mathematik*, vol. 38, no. 1, pp. 129–139, 1982.

[50] R. Gonin and A. H. Money, *Nonlinear Lp-norm estimation*. CRC Press, 2017.

APPENDICES

A. Proof of Ordered Alternative Solutions

To illustrate the alternative ordered solutions in Com and ADM extensions, let $z^* = \sum_i \alpha_i^*$ where $\alpha_i \in \{0,1\}$ as in Bin-L_pDis and the two extensions. z^* denotes optimal total classification error and α_i^* denotes the optimal class assignment error of A_i^q. Let $z^A, z^B, z^C \in N^+$ in Figure 10.1, denote three alternative solutions of minimum number of incorrect assignments that is $z^A = z^B = z^C = z^*$. Let λ^A, λ^{B*} and λ^C be the λ values obtained in nodes A, B and C, respectively. Variable λ is the alternative solution seeking variables in constraints (53) and (57). Assume $\lambda^A > \lambda^B > \lambda^C$ in Figure 10.1. Then, $z^A - \lambda^A < z^B - \lambda^B < z^C - \lambda^C$ (objective function (51)) holds and $z^A - \lambda^A$ is chosen as the optimal solution. Since $\lambda^A, \lambda^B, \lambda^C \in (-1, \epsilon]$ and $z^A - \lambda^A < z^* + 1$, $z^B - \lambda^B < z^* + 1$, $z^C - \lambda^C < z^* + 1$, ordering of classes extensions do not decrease classification accuracy.

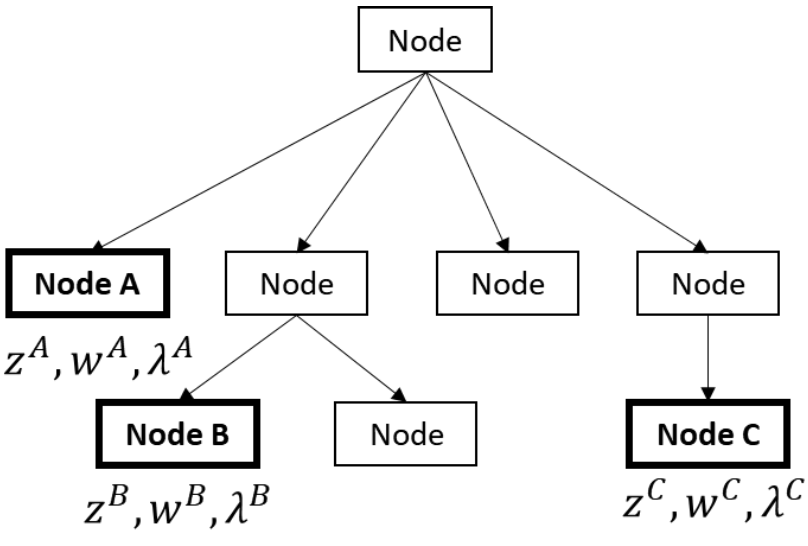


Figure 10.1 Example illustration of alternative solution of best accuracy outcomes

Without decreasing accuracy, alternative solution in Node A is selected which satisfy ordering the most for Bin-L_pDis Com extension as follows.

- $d_p^{w^A}(\mu^q, I) - d_p^{w^A}(\mu^{q+1}, I) > d_p^{w^B}(\mu^q, I) - d_p^{w^B}(\mu^{q+1}, I)$
- $d_p^{w^A}(\mu^q, I) - d_p^{w^A}(\mu^{q+1}, I) > d_p^{w^C}(\mu^q, I) - d_p^{w^C}(\mu^{q+1}, I)$

Same applies to Bin-L_pDis ADM extension as follows.

- $U_{w^A}(\mu^q - \mu^{q-1}) > U_{w^B}(\mu^q - \mu^{q-1})$
- $U_{w^A}(\mu^q - \mu^{q-1}) > U_{w^C}(\mu^q - \mu^{q-1})$

B. AIRO Experiments

Table 10.1 AIRO test accuracy results of AUTOMPG data set

AIRO-WE				
$V \backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	81.48	91.36	91.36	81.48
0.6	62.96	91.36	91.36	54.32
0.7	62.96	91.36	91.36	54.32
0.8	62.96	91.36	91.36	54.32
0.9	77.78	91.36	91.36	51.85
AIRO-ECW				
$V \backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	48.15	70.37	69.14	80.25
0.6	48.15	69.14	92.59	60.49
0.7	48.15	69.14	92.59	60.49
0.8	48.15	69.14	92.59	60.49
0.9	48.15	82.72	75.31	71.60

Table 10.2 AIRO test accuracy results of CPU data set

AIRO-WE				
$V \backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	86.05	81.40	88.37	86.05
0.6	88.37	90.70	88.37	83.72
0.7	88.37	90.70	88.37	83.72
0.8	88.37	90.70	88.37	83.72
0.9	81.40	81.40	90.70	83.72
AIRO-ECW				
$V \backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	74.42	72.09	72.09	74.42
0.6	90.70	93.02	93.02	93.02
0.7	90.70	93.02	93.02	93.02
0.8	90.70	93.02	93.02	93.02
0.9	90.70	90.70	93.02	90.70

Table 10.3 AIRO test accuracy results of BC data set

AIRO-WE				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	62.07	62.07	62.07	62.07
0.6	62.07	62.07	62.07	62.07
0.7	62.07	62.07	62.07	62.07
0.8	62.07	62.07	62.07	62.07
0.9	62.07	62.07	62.07	62.07
AIRO-ECW				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	67.24	62.07	58.62	62.07
0.6	53.45	27.59	27.59	27.59
0.7	27.59	27.59	27.59	27.59
0.8	27.59	27.59	27.59	27.59
0.9	27.59	27.59	27.59	27.59

Table 10.4 AIRO test accuracy results of ESL data set

AIRO-WE				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	68.75	75.00	84.38	62.50
0.6	55.21	70.83	56.25	52.08
0.7	55.21	70.83	56.25	52.08
0.8	55.21	70.83	56.25	52.08
0.9	75.00	70.83	84.38	75.00
AIRO-ECW				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	87.50	85.42	82.29	82.29
0.6	87.50	73.96	73.96	75.00
0.7	87.50	73.96	73.96	75.00
0.8	87.50	73.96	73.96	75.00
0.9	84.38	73.96	73.96	75.00

Table 10.5 AIRO test accuracy results of CAR data set

AIRO-WE				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	24.06	28.99	46.38	28.99
0.6	35.94	16.81	42.90	29.28
0.7	35.36	27.25	39.42	10.43
0.8	19.71	28.99	46.38	16.23
0.9	43.19	29.28	34.49	12.46
AIRO-ECW				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	37.97	28.12	27.25	25.22
0.6	47.54	56.52	58.84	60.00
0.7	46.38	56.52	58.84	60.00
0.8	46.38	56.52	58.84	60.00
0.9	46.38	56.52	58.84	60.00

Table 10.6 AIRO test accuracy results of CCS data set

AIRO-WE				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	59.11	42.36	58.62	50.25
0.6	54.19	61.58	66.01	45.81
0.7	54.19	61.58	66.01	45.81
0.8	56.65	39.41	58.62	51.23
0.9	56.16	42.36	60.10	50.74
AIRO-ECW				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	52.71	58.13	59.11	54.19
0.6	52.71	54.19	54.68	51.72
0.7	52.71	54.19	54.68	51.72
0.8	52.71	54.19	54.68	51.72
0.9	52.71	54.19	54.68	51.72

Table 10.7 AIRO test accuracy results of LEV data set

AIRO-WE				
$V \backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	79.80	79.80	79.80	69.19
0.6	79.80	79.80	79.80	79.80
0.7	79.80	79.80	79.80	79.80
0.8	79.80	79.80	61.62	53.54
0.9	79.80	57.58	61.62	80.30
AIRO-ECW				
$V \backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	71.72	71.21	70.20	65.66
0.6	69.19	66.16	65.15	65.66
0.7	69.70	68.18	67.17	67.17
0.8	69.70	68.18	67.17	64.14
0.9	72.73	71.21	68.18	71.21

Table 10.8 AIRO test accuracy results of ASA data set

AIRO-WE				
$V \backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	69.27	69.27	71.51	69.27
0.6	67.60	57.54	57.54	45.25
0.7	67.60	57.54	57.54	45.25
0.8	69.27	69.27	69.27	69.27
0.9	69.83	40.78	27.93	36.31
AIRO-ECW				
$V \backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	78.77	78.21	81.01	80.45
0.6	78.21	77.09	78.77	78.77
0.7	78.21	77.09	78.77	78.77
0.8	78.21	77.09	78.77	78.77
0.9	78.21	77.09	78.77	78.77

Table 10.9 AIRO test accuracy results of MMG data set

AIRO-WE				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	46.88	5.00	4.38	46.88
0.6	46.88	46.88	76.88	46.88
0.7	46.88	46.88	76.88	46.88
0.8	46.88	46.88	46.88	46.88
0.9	46.88	46.88	46.88	46.88
AIRO-ECW				
$V\backslash p$	L_1	L_2	L_3	L_{p^*}
0.5	82.50	81.88	80.63	81.25
0.6	82.50	81.88	80.63	81.25
0.7	82.50	81.88	80.63	81.25
0.8	82.50	81.88	80.63	81.25
0.9	82.50	82.50	81.88	81.88

C. Training Time of DS and DISWOTH models for 10 different p values

Table 10.10 Training time of DS model for 10 different p values

p	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	Total Training Time
AUTOMPG	325.5	0	230.86	401.67	532	753.3	930.45	14400	14400	14400	**46373.81**
BC	7.58	0	40.39	116.55	293.44	569.53	955.73	1266.5	2884.9	14400	**20534.67**
CAR	249.73	2.3	181	254.23	234.53	281.14	284.48	395.66	570.59	1716	**4169.66**
CCS	14400	0.1	2185.7	4966.8	11464.8	14400	14400	14400	14400	14400	**105017.28**
CPU	2815.3	0	644.02	1355.1	5497.67	1759.5	2128.4	1800.6	14400	3825.9	**34226.58**
ESL	68.77	0	15.39	13.27	16.45	20.11	16.42	19.13	16.36	24.41	**210.34**
LEV	83.77	0.8	14.23	16.92	24.3	48.98	40.89	45.81	55.09	81.25	**412.04**
ASA	0.11	0.1	0.11	0.81	0.09	0.09	0.13	0.09	0.09	0.11	**1.76**
MMG	14.56	0.1	24.16	91.48	198.97	205.13	503.03	551.66	641.45	970.42	**3200.95**
Average	**1996.15**	**0.39**	**370.65**	**902.00**	**2029.13**	**2004.20**	**2139.95**	**3653.28**	**5263.17**	**5535.34**	**23794.03**

Table 10.11 Training time of DISWOTH model for 10 different p values

p	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**	**3.5**	**4**	**4.5**	**5**	**Total Training Time**
AUTOMPG	1382.05	0.10	274.55	2478.56	2151.34	14400	4319.11	14400	14400	14400	**68205.71**
BC	1603.30	0.07	2842.86	59.61	5254.61	197.65	7874.48	14400	14400	14400	**61032.58**
CAR	14400	0.08	11763.94	14400	14400	10759.28	13129.80	14400	14400	14400	**122053.10**
CCS	14400	0.09	14400	5.19	14400	9.10	14400	14400	14400	14400	**100814.38**
CPU	176.53	0.19	40.94	5767.61	163.75	6640.97	14400	14400	14400	14400	**70389.99**
ESL	36.81	0.10	6.11	14400	9.67	14400	22.09	183.47	14400	14400	**57858.25**
LEV	128.78	0.09	36.41	38.60	57.08	42.93	85.75	215.91	81.91	1324.34	**2011.80**
ASA	14400	0.13	14400	14400	14400	14400	14400	14400	14400	14400	**129600.13**
MMG	40.77	0.12	17.92	20.99	14400	14400	14400	14400	14400	14400	**86479.80**
Average	**5174.25**	**0.11**	**4864.75**	**5730.06**	**7248.49**	**8361.10**	**9225.69**	**11244.38**	**12809.10**	**12947.15**	**77605.08**

D. Number of Iterations to Satisfy the FC Condition

Table 10.12 Number of iterations it takes to satisfy the FC condition for ATLAS with DS

V Values	AUTOMPG	BC	CAR	CCS	CPU	ESL	LEV	ASA	MMG
-9	3	3	3	14	3	3	3	3	3
-6	2	3	2	10	2	2	2	2	2
-3	2	2	2	6	2	2	2	2	2
0	1	1	1	1	1	1	1	1	1
0.05	3	3	3	3	3	4	3	1	3
0.1	2	2	2	2	2	1	2	1	2
0.15	3	3	3	3	3	4	3	1	3
0.2	2	3	2	2	2	1	2	1	2
0.25	4	5	4	4	4	5	4	1	4
0.3	2	2	2	2	2	1	2	1	4
0.35	4	4	4	4	4	5	4	1	2
0.4	2	3	2	2	2	1	2	1	4
0.45	4	3	4	4	4	5	4	1	2
0.5	2	3	2	2	2	1	2	1	4
0.55	4	3	4	4	4	5	4	1	2
0.6	2	3	2	2	2	1	2	1	4
0.65	4	3	4	4	4	5	4	1	2
0.7	2	3	2	2	2	1	2	1	4
0.75	4	3	4	4	4	5	4	1	2
0.8	2	3	2	2	2	1	2	1	4
0.85	4	3	4	4	4	5	4	1	2
0.9	2	3	2	2	2	1	2	1	4
0.95	4	3	4	4	4	5	4	1	2
1	2	1	2	2	2	1	2	1	4
Total	66	68	66	89	66	66	66	28	68

Table 10.13 Number of iterations it takes to satisfy the FC condition for ATLAS with DISWOTH

V Values	AUTOMPG	BC	CAR	CCS	CPU	ESL	LEV	ASA	MMG
-9	2	2	1	2	2	2	2	3	2
-6	2	2	2	2	2	2	2	2	2
-3	3	3	3	3	3	3	3	2	3
0	1	1	1	1	1	1	1	1	1
0.05	3	6	15	3	3	4	9	6	3
0.1	2	3	6	2	2	2	4	2	2
0.15	3	7	18	3	4	4	10	7	4
0.2	3	3	7	3	2	2	5	3	2
0.25	3	8	19	3	4	5	11	7	4
0.3	3	3	8	3	2	2	5	3	2
0.35	3	8	19	3	4	5	12	8	4
0.4	3	3	9	3	2	2	5	3	2
0.45	3	9	19	3	4	5	12	8	4
0.5	3	3	9	3	2	2	5	3	2
0.55	3	9	20	3	4	5	12	8	4
0.6	3	3	9	3	2	2	5	3	2
0.65	3	9	20	3	4	5	13	8	4
0.7	3	3	9	3	3	2	5	3	3
0.75	3	9	20	3	4	5	13	8	4
0.8	3	3	63	3	3	2	5	3	3
0.85	3	9	26	3	4	5	13	8	4
0.9	3	3	64	3	3	2	5	3	3
0.95	3	9	26	3	4	5	13	8	4
1	3	3	64	3	3	2	5	3	3
Total	67	121	457	67	71	76	175	113	71

E. Error Calculation Models of ATLAS

(TestCalculationModel-ApproximatedDS)

$$Minimize\ \sum_{i_s} \epsilon_{i_S}^{+} + \epsilon_{i_S}^{-} \tag{E.1}$$

Subject to:

$$Va_{i_s}^{\infty} + (1-V)\sum_j w^*(V)\ \left|A_{i_S j} - I_j\right| - e_{i_S}^{+} \leq T^*(V)\ \ \forall q > 1\ \forall i_S \in C^q$$
(E.2)

$$Va_{i_s}^{\infty} + (1-V)\sum_j w^*(V)\ \left|A_{i_s j} - I_j\right| + e_{i_S}^{-} \geq T^*(V)\ \ \forall q < Q\ \forall i_S \in C^q$$
(E.3)

$$a_{i_s}^{\infty} = \max_j\{w_j^*(V)\left|A_{i_s j} - I_j\right|\}\ \forall i_S \tag{E.4}$$

$$e_{i_S}^{+}, e_{i_S}^{-} \geq 0\ \forall i_s \tag{E.5}$$

(TestCalculationModel-ApproximatedDISWOTH)

$$Minimize\ \sum_{i_s} e_{i_s} \tag{E.6}$$

Subject to:

$$e_{i_s} - \epsilon \geq Va_{i_s,q}^{\infty} + (1-V)\sum_j w^*(V)\left|A_{i_s j} - \mu_j^q\right| - Va_{i_s,r}^{\infty} - (1 - V)\sum_j w^*(V)\left|A_{i_s j} - \mu_j^r\right|\ \forall q \neq r, \forall i_s \tag{E.7}$$

$$a_{i_s,q}^{\infty} = \max_j\{w_j^*(V)\left|A_{i_s j} - \mu_j^q\right|\}\ \forall i_S, \forall q \tag{E.8}$$

$$e_{i_s} \geq 0\ \forall i_s \tag{E.9}$$

F. Decision Boundaries of Classification for Distance-based Sorting Method

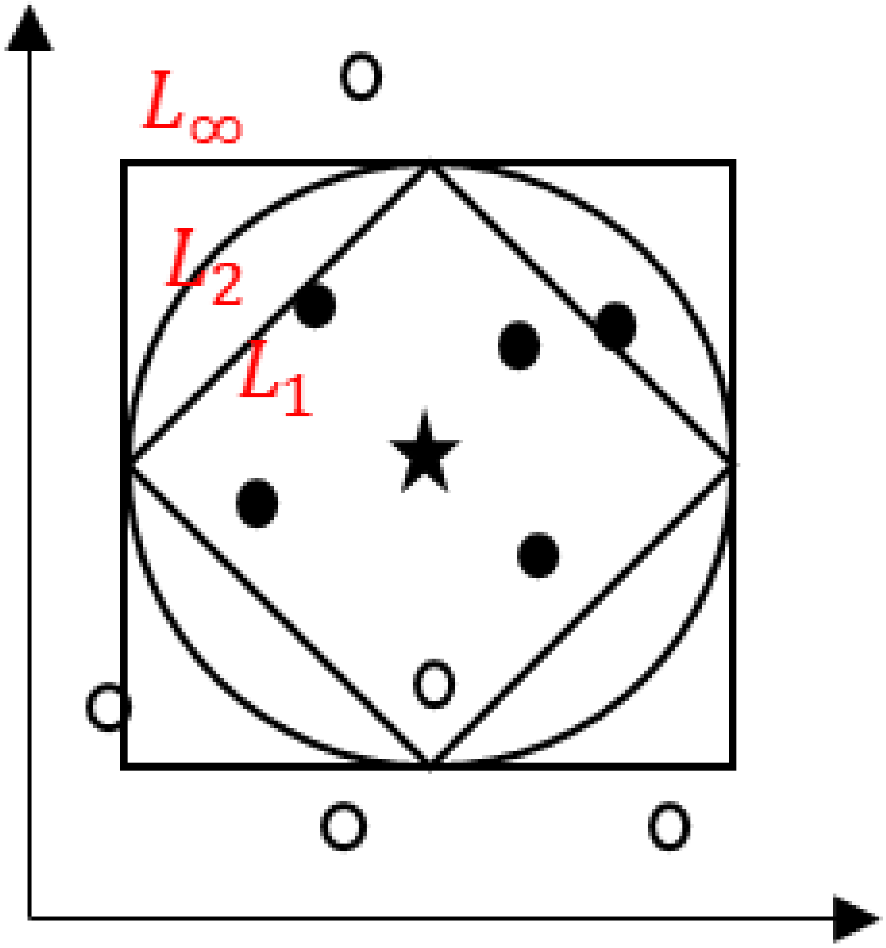


Figure 10.2 Decision boundaries of classification when distance-based sorting method is used.

In Figure 10.2, black circles are from class one and white circles are from class two. Star shape is the ICV. Interior of L_p distance contours (L_1, L_2, L_∞) is the region of class one and exterior is the region of class two.

G. Numerical Example of Weight Linearization

Consider simple example with two criteria, two classes (two centroids) and Euclidean distance is used. Assume that the model is solved and $w_1^* = 0.3$, $w_2^* = 0.7$ are obtained ($w_1^{2*} = 0.09$, $w_2^{2*} = 0.49$ in the distance calculation). The exact same decision boundary can be obtained with $v_1^* = \frac{0.3^2}{0.7^2+0.3^2} = 0.155$ and $v_2^* = \frac{0.7^2}{0.7^2+0.3^2} = 0.845$. If it is further examined, $\frac{0.155}{0.09} = \frac{0.845}{0.49} \approx 1.724$ (It is approximately 1.724 due to rounding). It is clear that $\Omega = 1.724$ and $0.3 + 0.7 = 0.155 + 0.845 = 1$. The resulting decision boundaries are $0.09|\mu_1 - E_1|^2 - 0.49|\mu_1 - E_1|^2 = 0$ and $0.155|\mu_1 - E_1|^2 - 0.845|\mu_1 - E_1|^2 = 0$ which are the same decision boundaries. Because if the second one is divided by 1.724, $0.09|\mu_1 - E_1|^2 - 0.49|\mu_1 - E_1|^2 = 0$ is obtained. Due to zero in RHS, multiplications and divisions by constants results in the same equation.

Instead of Euclidean distance if L_5 distance is used, again assuming the $w_1^* = 0.3$, $w_2^* = 0.7$ are the same ($w_1^{5*} = 0.00243$, $w_2^{5*} = 0.16807$ are used in distance calculation). In linearized form, $v_1^* = \frac{0.3^5}{0.7^5+0.3^5} = 0.0143$ and $v_2^* = \frac{0.7^5}{0.7^5+0.3^5} = 0.9857$. If it is further examined, $\frac{0.0143}{0.00243} = \frac{0.9857}{0.16807} \approx 5.88 = \Omega$ (It is approximately 5.88 due to rounding).

H. The Trivial Solution Examples of Monotonically Ordered Centroids Case

Minimizing only λ_{ij}^u and λ_{ij}^l variables maximize AR alternatives and minimize IR alternatives. However, a single DBC may be formed by infinitely many different class centroids. Those class centroids may not be interior to the convex hull formed by the alternatives of each class. This case is illustrated in Figure 10.3 below.

In Figure 10.3, there are alternatives of two classes that are illustrated with red and yellow regions. The decision boundary of classification is illustrated with a solid line. There are three examples of centroids that result in a single decision boundary of classification. When the term “centroid” is used, points similar to triangles are considered. But if only λ_{ij}^{u} and λ_{ij}^{l} are minimized without any regularization, large black and red circles denoted by μ^1 and μ^2 can also be obtained at monotonic class centroids. Those centroids are not even interior to the alternatives of each class. These are the trivial solutions of the AIRO model, regularized with additional objective functions.

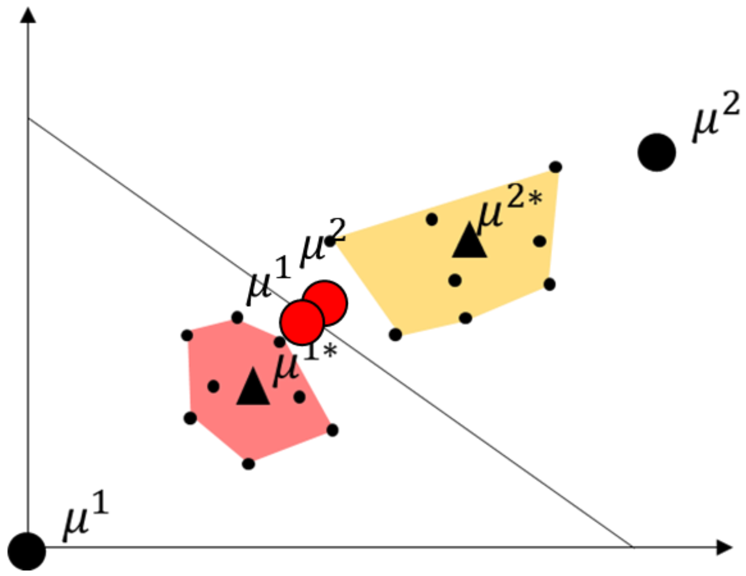


Figure 10.3 Examples of trivial solutions