June 2024

WORKING PAPER SERIES

2024-EQM-05

A Pairwise-Frontier-based Classification Method for Two-Group Classification

Qianying Jin
College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China. qianying.jin@nuaa.edu.cn, Corresponding author.

Kristiaan Kerstens
Univ. Lille, CNRS, IESEG School of Management, UMR 9221 - LEM - Lille Économie Management, F-59000, Lille, France. k.kerstens@ieseg.fr

Ignace Van de Woestyne
KU Leuven, Research Centre for Operations Research and Statistics (ORSTAT), Brussels Campus, Warmoesberg 26, B-1000 Brussels, Belgium, ignace.vandewoestyne@kuleuven.be

Zhongbao Zhou
School of Business Administration, Hunan University, Changsha, 410082, China. Z.B.Zhou@hnu.edu.cn

IÉSEG School of Management Lille Catholic University 3, rue de la Digue F-59000 Lille Tel: 33(0)3 20 54 58 92
www.ieseg.fr

A Pairwise-Frontier-based Classification Method for Two-Group Classification[*]

Qianying Jin[†] Kristiaan Kerstens[‡] Ignace Van de Woestyne[§]

Zhongbao Zhou[¶]

31st May 2024

Abstract

Mathematical programming-based methods are widely used to generate separating boundaries in two-group classification problems. Nonlinear separating boundaries may have better classification performance than linear separating boundaries, but these require a pre-specification of a nonlinear functional form. This contribution proposes a novel pairwise-frontier-based classification (PFC) method to approximate nonlinear separating boundaries, without predetermining a nonlinear functional form. It consists of two steps that explicitly consider and focus on overlap. The first step is to identify the overlap. Importantly, this contribution proposes to construct frontiers based on background knowledge of classification, thus ensuring that their intersection (i.e., overlap) is not increased by blindly applying commonly used axioms. Depending on the axioms applied, pairwise frontiers can be either convex or nonconvex. The second step minimizes identified overlaps by allowing training observations to be misclassified, but all training observations that have been correctly classified must remain correctly classified. The PFC method with hard frontiers is then extended to the one with soft frontiers. The applicability of the proposed PFC methods is illustrated by simulation studies and real-life data sets. The results show that the proposed method is competitive with some well-established classifiers in the literature and even performs better with unbalanced data sets.

Keywords: Data Envelopment Analysis; Frontier; Nonconvex; Convex; Two-group Classification.

[*] Q. Jin appreciates financial support of the National Natural Science Foundation of China (No. 72201127).

[†] College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China. qianying.jin@nuaa.edu.cn, Corresponding author.

[‡] Univ. Lille, CNRS, IESEG School of Management, UMR 9221 - LEM - Lille Économie Management, F-59000, Lille, France. k.kerstens@ieseg.fr

[§] KU Leuven, Research Centre for Operations Research and Statistics (ORSTAT), Brussels Campus, Warmoesberg 26, B-1000 Brussels, Belgium, ignace.vandewoestyne@kuleuven.be

[¶] School of Business Administration, Hunan University, Changsha, 410082, China. Z.B.Zhou@hnu.edu.cn

1 Introduction

A classification aims to determine whether an observation belongs to a particular group by evaluating a set of attributes. As an important and widely studied topic, its applications include but are not limited to costumer churn (e.g., De Caigny, Coussement, De Bock, and Lessmann (2020)), fraud and failure detection (e.g., Zhao, Ouenniche, and De Smedt (2024); De Bock, Coussement, and Lessmann (2020)), credit scoring (e.g., Lessmann, Baesens, Seow, and Thomas (2015); Farbmacher, Löw, and Spindler (2022)), medical diagnosis (e.g., Merdan, Barnett, Denton, Montie, and Miller (2021)), etc. Numerous techniques and methods have been proposed, such as statistical methods, support vector machines, artificial neural networks, decision trees and ensemble classifiers. A comprehensive review of statistical and data mining techniques used for classification can be found in Kotsiantis, Zaharakis, and Pintelas (2007) and Silva (2017). For expository clarity, this contribution focuses on the two-group classification problem.

Mathematical programming-based (MP-based) classifiers have been given considerable attention since their introduction by Freed and Glover (1981). The basic idea of the MP-based classifier is to determine a separating boundary such that the two groups of training observations lie on opposite sides of the separating boundary. If the convex (C) hulls of the two groups do not intersect, then these two groups are linearly separable, i.e., the separating boundary is simply a hyperplane. Sometimes, even though the groups of training observations are distinct, their C hulls may intersect (i.e., there is an overlap). Alternatively, when the training groups themselves are not well distinguished, their C hulls naturally intersect. In both cases, nonlinear hypersurfaces are believed to provide better separation than hyperplanes. However, generating nonlinear hypersurfaces requires a pre-specification of a nonlinear function form. It is not impossible, but very difficult to predetermine a nonlinear functional form to fit for a real application.

Given the basic idea of MP-based classifiers, this contribution is also interested in determining a separating boundary that best separates the two groups. Moreover, the boundary is expected to be nonlinear, but without a predetermined nonlinear functional form. To meet this goal, the Data Envelopment Analysis (DEA) method, which generates a C piecewise linear frontier, is of interest. The DEA method is a linear programming model evaluating the efficiency of observations by projecting these onto a C piecewise linear frontier. It is widely applied in production economics and finance (see recent surveys and historical developments in Ouenniche, Carrales, Tone, and Fukuyama (2017) and Emrouznejad, Banker, and Neralic (2019), respectively). Despite its popularity in production and finance, the DEA method has not been very widely used as a classification tool up to now.

Troutt, Rai, and Zhang (1996) is the first application of DEA as a classification tool known to us. They propose to use the C piecewise linear frontier generated from a base group of training observations as a separating boundary. This idea of employing a C frontier as a separating boundary has been adapted by proposing alternative objective functions (Seiford and Zhu, 1998), incorporating various data types (e.g., Leon and Palacios (2009), Yan and Wei (2011)) and has been applied in different application areas (e.g., Pendharkar, Rodger, and Yaverbaum (1999); Pendharkar, Khosrowpour, and Rodger (2000); Pendharkar (2002)). All the above classification methods employ a single C frontier as the separating boundary. To some extent, they only utilize the information of the selected base group. For this reason, they can not recognize the overlap between the two groups.

To fully utilize the information from both groups for better classification, Chang and Kuo (2005, 2008) propose to train a pair of C frontiers each of which envelops a group of training observations. The trained pair of C frontiers jointly determines a nonlinear separating boundary. We refer to these classification methods as Pairwise-Frontier-based Classification (PFC) methods with C frontiers.

In PFC methods with C frontiers, the intersection of pairwise frontiers is identified as overlap. In the existing literature on PFC methods, some researchers have chosen to eliminate overlap during the training process (Chang and Kuo, 2005, 2008; Kuo, 2013). With this treatment, the overlap is completely removed, but at the cost of misclassifying some otherwise correctly classified training observations. Other researchers choose not to do anything with the overlap during training, but classify the overlap using other classification methods during testing, e.g., a cost-sensitive nearest neighbourhood approach (Pendharkar, 2011), membership functions (Pendharkar, 2012), interaction or Minimum Sum of Deviations (MSD) method (Pendharkar and Troutt, 2014), probabilistic DEA techniques (Pendharkar, 2018), among others. With this treatment, the DEA efficiency measurement is only used to identify the overlap, but it is not used to predict the group membership of a new observation.

Regardless of these various treatments on the overlap, all existing PFC methods are constructed and inspired by the geometrical advantages of the DEA model, i.e., approximating the nonlinear separating boundary without predetermining a nonlinear functional form. In general, there is a lack of reflection on the relation between the axioms implied by constructing a nonlinear frontier and the background knowledge of classification. The only exception that we are aware of is that Jin, Kerstens, and Van de Woestyne (2024) have explored the correspondence of axioms when applying frontier-based classification methods to anomaly detection. However, only a single frontier is employed in this work. Furthermore, to the best of our knowledge, no study has been conducted to examine how the different mixes of axioms affect the overlap.

Therefore, the first purpose of this contribution is to explore the correspondence of the axioms in the context of two-group classification and their influence on overlap. First, the axiom of free disposability corresponds to a monotonous relation between the attributes and the group membership. This monotonicity relation determines the relative positions of the pairwise frontiers and therefore greatly affects the magnitude of the overlap. In many applications, the monotonicity relation is often an implicit assumption rather than an explicit one. To address this situation, a MSD model is proposed to establish a monotonous relation with less overlap. Second, the convexity axiom corresponds to the substitution relation between attributes. All existing PFC methods adhere to the convexity axiom. However, if no substitution relation is given in advance, then the convexity axiom should be relaxed. This allows for the construction of a pair of nonconvex (NC) frontiers based on the Free Disposal Hull (FDH) model proposed by Deprins, Simar, and Tulkens (1984). Compared to the C frontiers, the NC frontiers envelop the groups of training observations more tightly. Therefore, the overlap resulting from the NC frontiers is normally smaller. Exploring the correspondence of the axioms aims to ensure that the identified overlap is not meaninglessly increased by the implementation of inappropriate axioms.

Second, this contribution aims to handle the identified overlap from two perspectives. On the one hand, the overlap is minimized during training, but to the extent that all training observations that are already correctly classified remain correctly classified. An algorithm is proposed to realize this minimization. After the overlap is minimized, a new PFC method with soft frontiers is constructed. On the other hand, a directional distance function (DDF) measure is introduced to determine the positions of an observation relative to the frontiers. Specifically, based on the comparison of the corresponding DDF measurements, a classification decision for the overlap can be designed. In this way, the group membership of the observations in the overlap is inferred from the DDF measurement itself without additional reliance on any other classification methods.

Third, this contribution also aims at offering the first empirical analysis on evaluating the classification performance of the PFC methods with unbalanced data sets. Fundamentally, the PFC method should be less vulnerable to unbalanced data sets. In the PFC method, the separating boundary is determined by pairs of hard or soft frontiers. The hard frontier of a group is independently determined by the training observations of that group, and more specifically, only the portion of the training observations that lie on the frontier contribute to its determination. The soft frontier of a group is slightly influenced by the training observations of the other group, but only if those training observations are located in the overlap. Therefore, when confronted with unbalanced data sets, the PFC method should be able to better balance the performance on the minority group and the majority group.

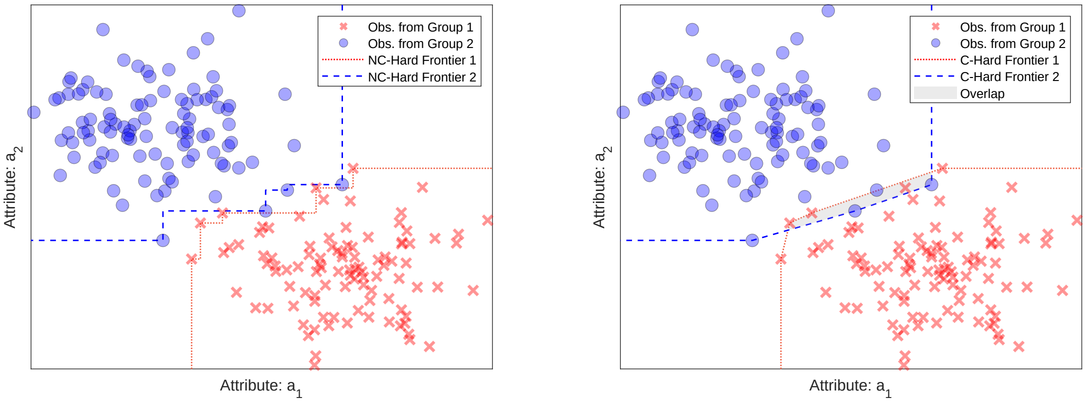
This contribution unfolds as follows. In Section 2, the intuitive idea of a PFC method is illustrated with a two-dimensional geometric example. Section 3 details the models and procedures for constructing a general PFC method. In Section 4 the proposed PFC methods are evaluated both with simulation data and with a real-life data set. Finally, Section 5 concludes with a summary of the contributions and a discussion of potential future research.

2 Geometric Illustration

Consider an illustrative example with two groups, each with 100 training observations. In Figures 1-3, the training observations from Group 1 and Group 2 are represented by red crosses and blue dots, respectively. Every training observation is characterized by two attributes, i.e., a_1 and a_2. The monotonicity relation of the attributes shows that a_1 is an input-type attribute and a_2 is an output-type attribute. This monotonicity relation is reflected in the relative values of the attributes in the different groups: in particular, the observations in Group 2 generally take smaller values of a_1 and larger values of a_2 than the observations in Group 1.

Figure 1 depicts an ideal example when two groups of training observations do not really intersect. They can be well separated by a pair of NC-hard frontiers as depicted in Figure 1(a). All the training observations from Group 1 are situated below the NC-hard frontier 1, indicated by the dotted polylines. By contrast, all the training observations from Group 2 are situated above the NC-hard frontier 2, indicated by the dashed polylines.

The NC-hard frontiers in Figure 1(a) are determined by the training observations and the axiom of free disposability. The axiom of free disposability is determined by the monotonicity relation of attributes, but is reflected differently in the two groups. For training observations from Group 1, the axiom of free disposability implies that neither an increase in a_1 nor a decrease in a_2 results in the corresponding observation being classified to Group 2. Thus, the NC-hard frontier 1 consists of the training observations with smallest a_1 and largest a_2 in Group 1. By contrast, for training observations from Group 2, the axiom of free disposability implies that neither a decrease in a_1 nor an increase in a_2 results in the corresponding observation being classified to Group 1. Thus, the NC-hard frontier 2 consists of the training observations with largest a_1 and smallest a_2 in Group 2. In this case, a pair of NC frontiers is produced. These frontiers are referred to as hard frontiers, because all training observations from each group are used to construct the corresponding frontier.



(a) Separable with NC-hard frontiers

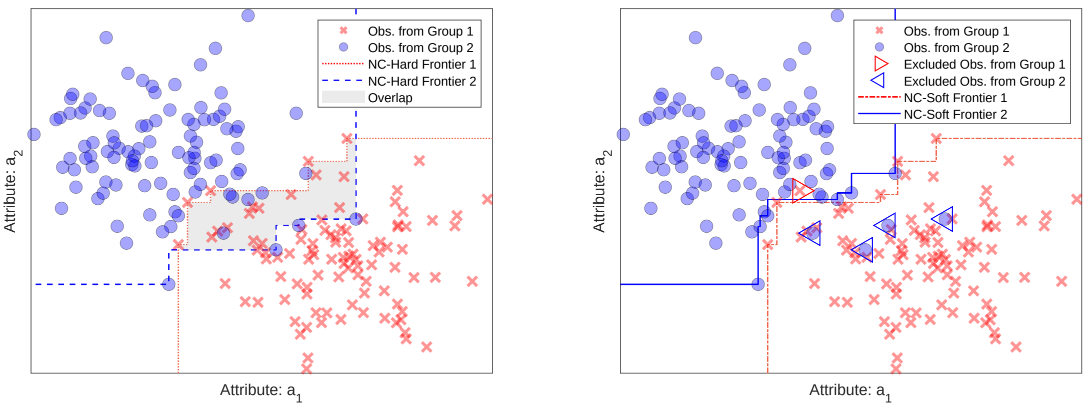
(b) Nonseparable with C-hard frontiers

Figure 1: An ideal example with two non-intersecting groups

In the literature on PFC methods, the convexity axiom is generally accepted without justifying its necessity. As shown in Figure 1(b), if also a convexity axiom is imposed, then a pair of C-hard frontiers is produced. The C-hard frontiers now have an intersection, i.e., there is overlap, while there is no overlap between the two training groups themselves. In this sense, the a priori imposition of an additional convexity axiom can lead to overlap, which is potentially detrimental to the classification performance.

While Figure 1 illustrates an ideal example where the training groups are well characterized and can be separated by pairwise NC-hard frontiers, in most applications the training groups themselves are not that well distinguished. In these cases, the training groups inevitably overlap, either with the NC-hard frontiers or with the C-hard frontiers, as shown in Figures 2(a) and 3(a), respectively. The identified overlap is the intersection of pairwise frontiers, indicated by the gray filled area in Figures 2 and 3. In this example, the overlap under C is larger than under NC. We return to this observation in Proposition 3.2.

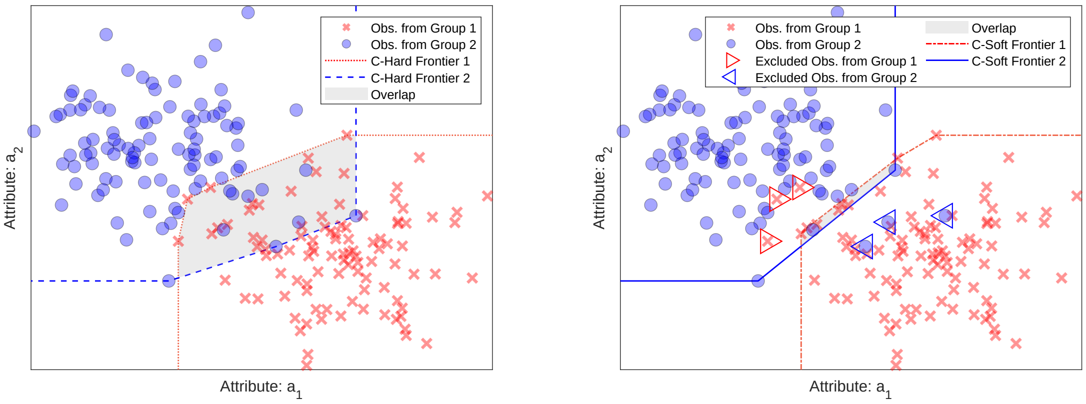
To minimize the identified overlap, we propose to exclude some of the training observations while constructing the frontiers, i.e., allow these to be misclassified during training. An algorithm is proposed to produce soft frontiers that realize a minimum overlap. Note that only training observations in the identified overlap are allowed to be misclassified. All other training observations outside the overlap are to remain correctly classified, i.e., these always remain within the corresponding frontier.



(a) NC-hard frontiers

(b) NC-soft frontiers

Figure 2: A realistic example with two intersecting groups: NC case



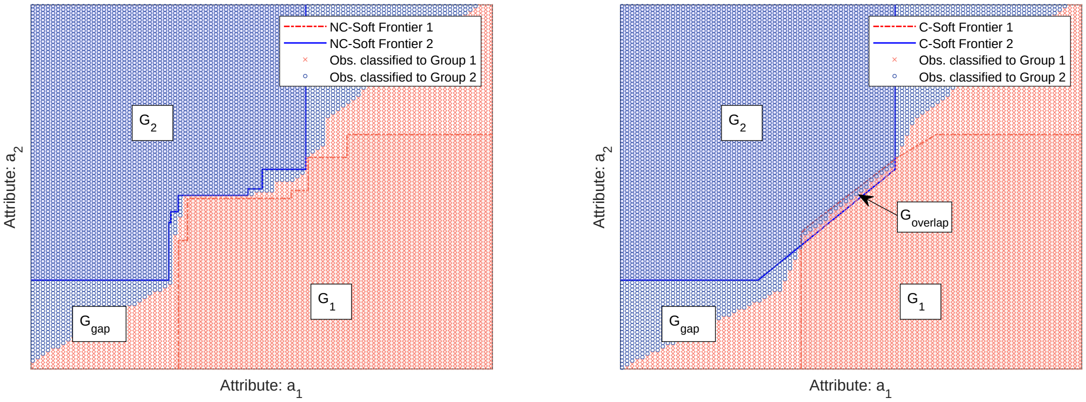
(a) C-hard frontiers

(b) C-soft frontiers

Figure 3: A realistic example with two intersecting groups: C case

The results for the NC and C cases are shown in Figures 2(b) and 3(b), respectively. For the NC case, 1 training observation from Group 1 and 4 training observations from Group 2 are identified as those that should be excluded. The overlap is then removed at the cost of misclassifying just 5 training observations. Similarly for the C case, the overlap is largely decreased at the cost of misclassifying 3 training observation from Group 1 and 3 training observations from Group 2. After excluding the training observations that should be misclassified, the soft frontiers are constructed from the remaining training observations.

With the constructed pairwise frontiers, the group membership of a new observation can now be determined by its relative location to these frontiers. There are four situations depending on the relative locations. Figure 4 visualizes the situations. First, if a new observation is located within frontier 1 and beyond frontier 2, then it is classified to Group 1. In Figure 4, this situation corresponds to the region below soft frontier 1 labeled with G_1. Second, if a new observation is located within frontier 2 and beyond frontier 1, then it is classified to Group 2. In Figure 4, this situation corresponds to the region above soft frontier 2 labeled with G_2. Third, if a new observation is located beyond both frontiers, then it is located in the gap. In Figure 4, this situation corresponds to the region between the soft frontier 1 and the soft frontier 2 labeled with G_{gap}. Its group membership needs to be determined based on which frontier it is closest to. Fourth, if a new observation is located within both frontiers, then it is located in the overlap. In Figure 4(b), it is the region where the soft frontier 1 and the soft frontier 2 intersect, labeled by G_{overlap}. Its group membership needs to be determined based on which frontier it is farthest from.



(a) Pairwise NC-soft frontiers

(b) Pairwise C-soft frontiers

Figure 4: Classification results with pairwise frontiers

The eventual separating boundary is jointly determined by the pairwise frontiers, as shown in Figure 4. The observations marked with red crosses are classified to Group 1, while the observations marked with blue circles are classified to Group 2.

3 Pairwise-Frontier-Based Classification: A Proposal

3.1 Problem Description

A two-group classification problem aims at classifying an observation to either Group 1 or Group 2. Such a classification problem is solved by learning from the historical classification information provided a priori, which consists of a set of training observations $Z = \{z_1, \ldots, z_n\}$. For every training observation z_j, its group label $I(z_j)$, where $I(z_j) \in \{1, 2\}$, refers to the group the training observation z_j belongs to. The training observations are exclusively classified into one of these two groups. Thus, $Z = Z_1 \cup Z_2$ and $Z_1 \cap Z_2 = \emptyset$, where Z_l ($l \in \{1, 2\}$) denotes the set of training observations that belong to group l, i.e.,

$Z_l = \{z_j \in Z \mid I(z_j) = l\}$.

The observations are evaluated by a number of attributes. The family of attributes is represented by $A = \{a_1, \ldots, a_K\}$. The evaluation of any observation z on attribute $a_k \in A$ is represented by $a_k(z)$.

To generate the frontiers, the monotonicity relation of attributes need to be made explicit. In some cases, the monotonicity relation of attributes is determined a priori by the decision maker. If a smaller evaluation on attribute a_k means that the corresponding observation is more likely to belong to Group 2, then attribute a_k is defined as an input-type attribute. By contrast, if a larger evaluation on attribute a_k means that the corresponding observation is more likely to belong to Group 2, then attribute a_k is defined as an output-type attribute.

In situations where the monotonicity relation for attributes is not explicitly given, we can use the MSD model (1) proposed by Freed and Glover (1986) to differentiate the attributes.

$$\begin{aligned}
\min_{\alpha_k, s^+_{1,j}, s^-_{2,j}} \quad & \sum_{z_j \in Z_1} s^+_{1,j} + \sum_{z_j \in Z_2} s^-_{2,j} & \\
s.t. \quad & \sum_{a_k \in A} \alpha_k a_k(z_j) - s^-_{1,j} \le d - \eta & \forall z_j \in Z_1 \\
& \sum_{a_k \in A} \alpha_k a_k(z_j) + s^+_{2,j} \ge d & \forall z_j \in Z_2 \\
& s^-_{1,j} \ge 0, s^+_{2,j} \ge 0, d \text{ and } \alpha_k \text{ are free} &
\end{aligned} \tag{1}$$

where d is a threshold value. To avoid a trivial solution (where $\alpha_k = 0$ and $d = 0$) and to have a clear separation between two groups, a small positive number η is introduced (see Glover (1990) for details).

Solving model (1) provides each attribute with a monotonicity relation as follows. If the optimal value α^*_k for α_k is negative, then the increase on attribute a_k reduces $\sum_{a_k \in A} \alpha^*_k a_k(z_j)$ which makes z_j a less favorable candidate of belonging to Group 2. This satisfies the behaviour of an input-type attribute. By contrast, if the optimal value α^*_k is positive, then the increase on attribute a_k makes z_j more favorable of belonging to Group 2. Therefore, an attribute a_k with a positive α^*_k behaves like an output-type attribute.

Based on different monotonicity relations, the family of attributes can be exclusively differentiated into two types, the input-type attributes reorganized into $X = \{x_1, \ldots, x_m\}$ and the output-type attributes reorganized into $Y = \{y_1, \ldots, y_s\}$. Note that the set of all attributes $A = X \cup Y$ and $K = m + s$. The evaluation of observation z on attribute x_i is represented by $x_i(z)$, and its evaluation on attribute y_r is represented by $y_r(z)$. Similarly, we introduce the notation $X(z) = (x_1(z), ..., x_m(z))$ and $Y(z) = (y_1(z), ..., y_s(z))$.

3.2 Acceptance Possibility Set and Its Estimates

In the context of classification, we introduce the Accepted Possibility Set (APS) to describe the attainable set of a certain group.[1] Specifically, it describes all possible combinations of attribute values whose corresponding observations are accepted as members of that group. The APS of Group l is expressed as

$$T_l = \{(x, y) \in \mathbb{R}^m \times \mathbb{R}^s \mid (x, y) \text{ is accepted as a member of Group } l\}. \tag{2}$$

Let S_l denote the set of training observations that are known to belong to Group l. Then, APS T_l can be estimated from S_l using the idea of minimal extrapolation. First, APS T_l is estimated as the smallest set containing data that satisfy the axiom of free disposability, denoted by $T^*_{NC,l}$. The axiom of strong or free disposability corresponds to the monotonicity relation of attributes. Specifically for Group 1, free disposability implies that the corresponding observations with larger attributes in X and smaller attributes in Y compared to the training observations from Group 1 are still acceptable as members of Group 1. By contrast, for Group 2, free disposability implies that the corresponding observations with smaller attributes in X and larger attributes in Y compared to the training observations from Group 2 remains acceptable as members of Group 2. Thus, based on the sets S_1 and S_2, respectively, the NC estimates of APSs T_1 and T_2 are:

$$T^*_{NC,1} = \left\{ (x, y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_1} \lambda_j X(z_j) \leq x, \sum_{z_j \in S_1} \lambda_j Y(z_j) \geq y, \sum_{z_j \in S_1} \lambda_j = 1, \lambda_j \in \{0, 1\} \right\}, \tag{3}$$

$$T^*_{NC,2} = \left\{ (x, y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_2} \lambda_j X(z_j) \geq x, \sum_{z_j \in S_2} \lambda_j Y(z_j) \leq y, \sum_{z_j \in S_2} \lambda_j = 1, \lambda_j \in \{0, 1\} \right\}. \tag{4}$$

Second, APS T_l can also be estimated as the smallest C set containing data that satisfy the axiom of free disposability, denoted by $T^*_{C,l}$. Comparing to the NC estimate $T^*_{NC,l}$, an additional convexity axiom is adopted. The convexity axiom in classification implies a substitution relation among the attributes. In empirical applications, there may exist this type of substitution relation. For example, in the admission to a college, the admission

[1] The concept of APS is derived from the Production Possibility Set (PPS) in production analysis, which describes the attainable set of a given technology. The PPS describes producibility, whereas the APS describes acceptability.

decision is made based on several attributes of a candidate, including the GMAT score and the SAT score. In evaluating a candidate, the disadvantage in the GMAT scores can be compensated to a certain degree by the advantages in SAT. In this case, there exists a substitution relation between the two attributes, namely the GMAT score and the SAT score. Thus, if prior information on such a substitution relation is provided, then based on the same observed sets S_1 and S_2, respectively, the C estimates of T_1 and T_2 are:

$$T^*_{C,1} = \left\{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_1} \lambda_j X(z_j) \leq x, \sum_{z_j \in S_1} \lambda_j Y(z_j) \geq y, \sum_{z_j \in S_1} \lambda_j = 1, \lambda_j \geq 0\right\}, \quad (5)$$

$$T^*_{C,2} = \left\{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_2} \lambda_j X(z_j) \geq x, \sum_{z_j \in S_2} \lambda_j Y(z_j) \leq y, \sum_{z_j \in S_2} \lambda_j = 1, \lambda_j \geq 0\right\}. \quad (6)$$

With an additional convexity axiom, the C estimates of T_1 and T_2 are no smaller than the corresponding NC estimates. This is formalised in the following Proposition 3.1.

Proposition 3.1. $T^*_{NC,1} \subseteq T^*_{C,1}$, $T^*_{NC,2} \subseteq T^*_{C,2}$

The proof of this proposition and all other theoretical results is found in the Appendix A.

If an observation satisfies both the descriptions of Group 1 and Group 2, then it is located in the overlap. The overlap is represented by the intersection of the estimates of T_1 and T_2. Specifically, the overlap under the NC case is $T^*_{NC,Overlap} = T^*_{NC,1} \cap T^*_{NC,2}$, and the overlap under the C case is $T^*_{C,Overlap} = T^*_{C,1} \cap T^*_{C,2}$. The overlap under the NC case is always included in the overlap under the C case, as shown in Proposition 3.2.

Proposition 3.2. $T^*_{NC,Overlap} \subseteq T^*_{C,Overlap}$

Corollary 3.1. *$T^*_{NC,Overlap} = T^*_{C,Overlap} \neq \emptyset$ if and only if $T^*_{C,Overlap} \neq \emptyset$, $T^*_{NC,Overlap} \neq \emptyset$, $T^*_{C,1} = T^*_{NC,1}$ and $T^*_{C,2} = T^*_{NC,2}$.*

Corollary 3.1 implies that the nonempty overlap under the NC and C cases is equal if and only if the estimates of the APSs under the two cases are identical.

If an observation fits neither the description of Group 1 nor Group 2, then it is considered to be located in the gap. The gap is represented by the complement of the union of the estimates of T_1 and T_2 with respect to the whole attribute space $\mathbb{R}^m \times \mathbb{R}^s$. Specifically, the gap under the NC case is $T^*_{NC,Gap} = \left(T^*_{NC,1} \cup T^*_{NC,2}\right)^{\complement}$, and the gap under the C case is $T^*_{C,Gap} = \left(T^*_{C,1} \cup T^*_{C,2}\right)^{\complement}$, where $(.)^{\complement}$ denotes the complement of a set with respect to $\mathbb{R}^m \times \mathbb{R}^s$.

Proposition 3.3. $T^*_{C,Gap} \subseteq T^*_{NC,Gap}$

Corollary 3.2. *$T^*_{NC,Gap} = T^*_{C,Gap}$ if and only if $T^*_{C,1} \backslash T^*_{NC,1} \subseteq T^*_{NC,2}$ and $T^*_{C,2} \backslash T^*_{NC,2} \subseteq T^*_{NC,1}$.*

Proposition 3.3 shows that the gap under the C case is included in the gap under the NC case. Corollary 3.2 implies that the gaps under the NC and C cases are equal if and only if the increment from the NC estimate of one group to its C estimate is captured by the NC estimate of the other group.

To unify expressions (3) to (6), we use the following notation to stand for the estimates of T_1 and T_2 under both the NC and C cases:

$$T^*_{\Lambda,1} = \left\{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_1} \lambda_j X(z_j) \leq x, \sum_{z_j \in S_1} \lambda_j Y(z_j) \geq y, \sum_{z_j \in S_1} \lambda_j = 1, \lambda_j \in \Lambda \right\}, \quad (7)$$

$$T^*_{\Lambda,2} = \left\{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_2} \lambda_j X(z_j) \geq x, \sum_{z_j \in S_2} \lambda_j Y(z_j) \leq y, \sum_{z_j \in S_2} \lambda_j = 1, \lambda_j \in \Lambda \right\}, \quad (8)$$

where

$$\text{(i) } \Lambda \equiv \Lambda^{\mathrm{C}} = \{\lambda_j \geq 0\} \text{ , or (ii) } \Lambda \equiv \Lambda^{\mathrm{NC}} = \{\lambda_j \in \{0,1\}\}.$$

Following Chambers, Chung, and Färe (1998), the directional distance function (DDF) can be a complete function representation of the APS, i.e., an observation is in the APS if and only if its DDF measurement is non-negative. Thus, $T^*_{\Lambda,1}$ and $T^*_{\Lambda,2}$ can be determined using the following DDF measures, respectively:

$$D_{T^*_{\Lambda,1}}(z_0, g_1) = \sup\{\delta_{\Lambda,1}(z_0) \in \mathbb{R} \mid (X(z_0), Y(z_0)) + \delta_{\Lambda,1}(z_0) g_1(z_0) \in T^*_{\Lambda,1}\}, \quad (9)$$

$$D_{T^*_{\Lambda,2}}(z_0, g_2) = \sup\{\delta_{\Lambda,2}(z_0) \in \mathbb{R} \mid (X(z_0), Y(z_0)) + \delta_{\Lambda,2}(z_0) g_2(z_0) \in T^*_{\Lambda,2}\}, \quad (10)$$

where $g_1 = (g_{X,1}, g_{Y,1}) \in \mathbb{R}^m_- \times \mathbb{R}^s_+$ and $g_2 = (g_{X,2}, g_{Y,2}) \in \mathbb{R}^m_+ \times \mathbb{R}^s_-$ represent the projection directions.

To make the DDF measure more interpretable, a proportional DDF measure is introduced in this contribution. Specifically, $g_1(z_0) = (-|X(z_0)|, |Y(z_0)|)$ and $g_2(z_0) = (|X(z_0)|, -|Y(z_0)|)$ are applied for the evaluated observation z_0. Note that in the classification context with potentially negative attribute values, the absolute value is used for preserving a proportional interpretation (see Kerstens and Van de Woestyne (2011) for details).

The APS estimate can be sufficiently represented using its boundary (also known as the frontier) without the need to use all the training observations. Specifically, only training observations with a DDF measure of 0 contribute to the construction of the corresponding frontier. The proportional DDF measures represented by (9) and (10) with the appropriate projection directions can be calculated simultaneously from model (11).

$$
\begin{aligned}
\max_{\lambda_{j,1},\delta_{\Lambda,1},\lambda_{j,2},\delta_{\Lambda,2}} \quad & \delta_{\Lambda,1}(z_0)+\delta_{\Lambda,2}(z_0) & & \\
s.t. \quad & \sum_{z_j\in S_1}\lambda_{j,1}x_i(z_j)\le x_i(z_0)-\delta_{\Lambda,1}(z_0)|x_i(z_0)| & & \forall x_i\in X \\
& \sum_{z_j\in S_1}\lambda_{j,1}y_r(z_j)\ge y_r(z_0)+\delta_{\Lambda,1}(z_0)|y_r(z_0)| & & \forall y_r\in Y \\
& \sum_{z_j\in S_1}\lambda_{j,1}=1 & & \\
& \lambda_{j,1}\in\Lambda & & \forall z_j\in S_1 \\
& \sum_{z_j\in S_2}\lambda_{j,2}x_i(z_j)\ge x_i(z_0)+\delta_{\Lambda,2}(z_0)|x_i(z_0)| & & \forall x_i\in X \\
& \sum_{z_j\in S_2}\lambda_{j,2}y_r(z_j)\le y_r(z_0)-\delta_{\Lambda,2}(z_0)|y_r(z_0)| & & \forall y_r\in Y \\
& \sum_{z_j\in S_2}\lambda_{j,2}=1 & & \\
& \lambda_{j,2}\in\Lambda & & \forall z_j\in S_2
\end{aligned}
\tag{11}
$$

where

$$\text{(i) } \Lambda\equiv\Lambda^{\mathrm{C}}=\{\lambda_j\ge 0\}\ \text{, or (ii) } \Lambda\equiv\Lambda^{\mathrm{NC}}=\{\lambda_j\in\{0,1\}\}\,.$$

In the C case, model (11) is solving a linear programming (LP) problem, while it involves solving a binary mixed integer program (BMIP) for the NC case. To speed up computations in the NC case, a fast implicit enumeration-based method is proposed by Cherchye, Kuosmanen, and Post (2001) requiring only to compute minima of lists of ratios. Thus, the following exact solutions are obtained for model (11) under the NC case:

$$\delta^*_{\Lambda^{\mathrm{NC}},1}(z_0)=\max_{z_j\in S_1}\left(\min_{x_i\in X}\left(\frac{x_i(z_0)-x_i(z_j)}{|x_i(z_0)|}\right),\min_{y_r\in Y}\left(\frac{y_r(z_j)-y_r(z_0)}{|y_r(z_0)|}\right)\right), \tag{12}$$

$$\delta^*_{\Lambda^{\mathrm{NC}},2}(z_0)=\max_{z_j\in S_2}\left(\min_{x_i\in X}\left(\frac{x_i(z_j)-x_i(z_0)}{|x_i(z_0)|}\right),\min_{y_r\in Y}\left(\frac{y_r(z_0)-y_r(z_j)}{|y_r(z_0)|}\right)\right). \tag{13}$$

3.3 Convex and Nonconvex Hard Frontiers

In this subsection, pairwise hard frontiers are constructed to represent the APS estimates for Group 1 and Group 2, respectively. The use of pairwise hard frontiers is sufficient to determine a separating boundary in two cases. The first case is that of well characterized training groups without overlap. The second case is where there is overlap but it is decided not to do anything about it during the training process. In both cases, the frontiers are constructed based on all the training observations of the corresponding groups and are, therefore, called hard frontiers.

To identify the training observations that determine the hard frontiers, model (11) needs to be solved for each observation $z_j \in Z_1 \cup Z_2$ with $S_1 = Z_1$ and $S_2 = Z_2$. If the DDF measurement of the evaluated observation z_j is 0, then it is collected into the corresponding frontier set, denoted by $F_{\Lambda,l}$ $(l \in \{1, 2\})$. Specifically,

$$F_{\Lambda,1} = \{z_j \in Z_1 \mid \delta^*_{\Lambda,1}(z_j) = 0\}, \quad F_{\Lambda,2} = \{z_j \in Z_2 \mid \delta^*_{\Lambda,2}(z_j) = 0\}, \tag{14}$$

where $\delta^*_{\Lambda,1}(z_j)$ and $\delta^*_{\Lambda,2}(z_j)$ are the optimized DDF measurements obtained from solving model (11).

The hard frontiers 1 and 2 are then represented by the boundaries of frontier sets $F_{\Lambda,1}$ and $F_{\Lambda,2}$, respectively. Note that both C or NC versions can be obtained, depending on whether the convexity axiom is adopted or not.

Instead of using all training observations in Z_1 and Z_2, the group membership of a new observation is now determined by its relative location compared to the hard frontiers. The classification rules are detailed in Subsection 3.5.

3.4 Convex and Nonconvex Soft Frontiers

In general, when the overlap gets larger the classification ability of a classifier tends to get worse. In this subsection, we propose to minimize the overlap that occurs during the training process to improve the classification ability.

The overlap is minimized by allowing some of the training observations to be misclassified with the restriction that all of the training observations that are correctly classified by the hard frontiers remain correctly classified. Thus, pairwise soft frontiers are constructed to represent the APS estimates and to jointly determine a separating boundary.[2] It takes two steps to construct the soft frontiers.

Step 1: Identify the Overlap

A training observation is located in the overlap if it is simultaneously situated on or below the frontier of Group 1 and on or below the frontier of Group 2. More specifically, it can be represented as follows:

$$R_{\Lambda,0} = \{z_j \in Z \mid \delta^*_{\Lambda,1}(z_j) \geq 0 \text{ and } \delta^*_{\Lambda,2}(z_j) \geq 0\}, \tag{15}$$

where $\delta^*_{\Lambda,1}(\cdot)$ and $\delta^*_{\Lambda,2}(\cdot)$ are the optimized DDF measurements solved from model (11) while constructing the hard frontiers.

Step 2: Minimize the Overlap

To minimize the overlap determined in Step 1, we propose to exclude some training observations while constructing the frontiers. Note that only the training observations that are located in the overlap $R_{\Lambda,0}$ can be considered for exclusion. Algorithm 1 (see infra) is designed to identify the training observations that should be excluded, and yields as outputs the soft frontier sets, namely, $\hat{F}_{\Lambda,1}$ and $\hat{F}_{\Lambda,2}$.

Model (16) is constructed to identify training observations in $R_{\Lambda,0}$ that potentially need to be excluded:

$$\begin{aligned}
\min_{s_{j,1}, s_{j,2}, p} \quad & c \cdot \sum_{z_j \in R_{\Lambda,0} \cap Z_1} s_1(z_j) + \sum_{z_j \in R_{\Lambda,0} \cap Z_2} s_2(z_j) \\
s.t. \quad & \delta^*_{\Lambda,1}(z_j) + s_1(z_j) \geq \delta^*_{\Lambda,2}(z_j) - s_1(z_j) + p \qquad \forall z_j \in R_{\Lambda,0} \cap Z_1 \\
& \delta^*_{\Lambda,2}(z_j) + s_2(z_j) \geq \delta^*_{\Lambda,1}(z_j) - s_2(z_j) - p \qquad \forall z_j \in R_{\Lambda,0} \cap Z_2 \\
& s_1(z_j), s_2(z_j) \geq 0 \\
& p \text{ unconstrained}
\end{aligned} \tag{16}$$

where $\delta^*_{\Lambda,1}(\cdot)$ and $\delta^*_{\Lambda,2}(\cdot)$ are the optimized DDF measurements solved from model (11) while constructing the hard frontiers.

In model (16), the weight c is used to mitigate the data imbalance. In this contribution, we use the ratio of the cardinality of $R_{\Lambda,0} \cap Z_2$ and the cardinality of $R_{\Lambda,0} \cap Z_1$. The optimized p^* reflects a preference for a certain group. If $p^* > 0$, then Group 2 is preferred. A positive p^* implies that the sum of deviations is minimized by having more training observations

[2] Similar to the terminologies of hard and soft margins in Support Vector Machine (SVM), the frontiers constructed by allowing for misclassification are referred to as soft frontiers.

from Group 2 correctly classified, while a non-positive p^* implies that the sum of deviations is minimized by having a preference for Group 1.

Algorithm 1 Generating the soft frontier sets

```
Inputs: $Z_1$, $Z_2$, $R_{\Lambda,0}$
Solve model (16), obtain the optimized values: $s_1^*(z_j)$, $s_2^*(z_j)$, $p^*$
Generate $E_{\Lambda,1}$ and $E_{\Lambda,2}$ based on equation (17)
Let $S_1 = Z_1 \setminus E_{\Lambda,1}$, $S_2 = Z_2 \setminus E_{\Lambda,2}$
Solve model (11) for $z_j \in E_{\Lambda,1} \cup E_{\Lambda,2}$, obtain $\delta^*_{\Lambda,1}(z_j)$, $\delta^*_{\Lambda,2}(z_j)$
Generate $Gap_{\Lambda,1}$ and $Gap_{\Lambda,2}$ based on equations (18) and (19)
if $Gap_{\Lambda,1} \cup Gap_{\Lambda,2} \neq \emptyset$, then
    if $p^* > 0$, then
        $E_{\Lambda,2} = E_{\Lambda,2} \setminus Gap_{\Lambda,2}$
        Let $S_1 = Z_1 \setminus E_{\Lambda,1}$ and $S_2 = Z_2 \setminus E_{\Lambda,2}$
        Solve model (11) for $z_j \in Gap_{\Lambda,1}$, obtain $\delta^*_{\Lambda,1}(z_j)$, $\delta^*_{\Lambda,2}(z_j)$
        $Gap_{\Lambda,1} = \{z_j \in Gap_{\Lambda,1} \mid \delta^*_{\Lambda,1}(z_j) < 0 \text{ and } \delta^*_{\Lambda,2}(z_j) < 0\}$
        $E_{\Lambda,1} = E_{\Lambda,1} \setminus Gap_{\Lambda,1}$
    else
        $E_{\Lambda,1} = E_{\Lambda,1} \setminus Gap_{\Lambda,1}$
        Let $S_1 = Z_1 \setminus E_{\Lambda,1}$ and $S_2 = Z_2 \setminus E_{\Lambda,2}$
        Solve model (11) for $z_j \in Gap_{\Lambda,2}$, obtain $\delta^*_{\Lambda,1}(z_j)$, $\delta^*_{\Lambda,2}(z_j)$
        $Gap_{\Lambda,2} = \{z_j \in Gap_{\Lambda,2} \mid \delta^*_{\Lambda,1}(z_j) < 0 \text{ and } \delta^*_{\Lambda,2}(z_j) < 0\}$
        $E_{\Lambda,2} = E_{\Lambda,2} \setminus Gap_{\Lambda,2}$
    end if
end if
Let $S_1 = Z_1 \setminus E_{\Lambda,1}$, $S_2 = Z_2 \setminus E_{\Lambda,2}$
Solve model (11) for $z_j \in (Z_1 \setminus E_{\Lambda,1}) \cup (Z_2 \setminus E_{\Lambda,2})$, obtain $\delta^*_{\Lambda,1}(z_j)$, $\delta^*_{\Lambda,2}(z_j)$
Generate $\hat{F}_{\Lambda,1}$ and $\hat{F}_{\Lambda,2}$ based on equation (20)
Outputs: $\hat{F}_{\Lambda,1}$, $\hat{F}_{\Lambda,2}$
```

A training observation $z_j \in R_{\Lambda,0} \cap Z_1$ can be excluded if it is more in the interior of $T^*_{\Lambda,2}$ rather than $T^*_{\Lambda,1}$. In model (16), this is reflected in a positive $s_1^*(z_j)$. Alternatively, a training observation $z_j \in R_{\Lambda,0} \cap Z_2$ can be excluded if it is more in the interior of $T^*_{\Lambda,1}$ rather than $T^*_{\Lambda,2}$. In model (16), this is reflected in a positive $s_2^*(z_j)$. The sets of the excluded training observations are then represented as follows:

$$E_{\Lambda,1} = \{z_j \in R_{\Lambda,0} \cap Z_1 | s_1^*(z_j) > 0\}, E_{\Lambda,2} = \{z_j \in R_{\Lambda,0} \cap Z_2 | s_2^*(z_j) > 0\}. \tag{17}$$

After excluding the training observations in the sets $E_{\Lambda,1} \cup E_{\Lambda,2}$, the overlap is minimized.

However, there may arise situations where the excluded training observations are located beyond both frontiers. In other words, the excluded training observations are located in the gap. To identify the excluded training observations that are located in the gap, model (11) is solved by letting $S_1 = Z_1 \setminus E_{\Lambda,1}$ and $S_2 = Z_2 \setminus E_{\Lambda,2}$. Solving model (11) for every training observation $z_j \in E_{\Lambda,1} \cup E_{\Lambda,2}$, the optimized DDF measurements are $\delta^*_{\Lambda,1}(z_j)$ and $\delta^*_{\Lambda,2}(z_j)$. The observations z_j are collected into the gap set, denoted by $Gap_{\Lambda,l}$, if it satisfies

$$Gap_{\Lambda,1} = \{z_j \in E_{\Lambda,1} \mid \delta^*_{\Lambda,1}(z_j) \leq 0 \text{ and } \delta^*_{\Lambda,2}(z_j) < 0\}, \tag{18}$$

$$Gap_{\Lambda,2} = \{z_j \in E_{\Lambda,2} \mid \delta^*_{\Lambda,1}(z_j) < 0 \text{ and } \delta^*_{\Lambda,2}(z_j) \leq 0\}. \tag{19}$$

If there are any excluded training observations located in the gap, then it means that the overlap has been over-minimized. Then, we need to add back these excluded training observations located in the gap provided that adding these observations back does not result in additional overlap. In particular, the order in which gap sets $Gap_{\Lambda,1}$ and $Gap_{\Lambda,2}$ are added is based on the group preference p^*. See steps 6-20 of Algorithm 1 for the specific procedures.

To identify the training observations that compose the soft frontiers, model (11) needs to be solved by taking $S_1 = Z_1 \setminus E_{\Lambda,1}$ and $S_2 = Z_2 \setminus E_{\Lambda,2}$. Solve model (11) for every training observation $z_j \in (Z_1 \setminus E_{\Lambda,1}) \cup (Z_2 \setminus E_{\Lambda,2})$. If the training observation z_j has a DDF measurement of 0, it is collected into the corresponding soft frontier sets:

$$\hat{F}_{\Lambda,1} = \{z_j \in Z_1 \setminus E_{\Lambda,1} \mid \delta^*_{\Lambda,1}(z_j) = 0\}, \hat{F}_{\Lambda,2} = \{z_j \in Z_2 \setminus E_{\Lambda,2} \mid \delta^*_{\Lambda,2}(z_j) = 0\}, \tag{20}$$

where $\delta^*_{\Lambda,1}(z_j)$ and $\delta^*_{\Lambda,2}(z_j)$ are the optimized DDF measurements obtained from solving model (11). All of the above is summarized in Algorithm 1.

3.5 Classification Rules

The constructed hard or soft frontiers can then be used to decide the group membership of a new observation z_0. Specifically, the group membership of a new observation z_0 is decided by its relative distances to the hard or soft frontiers. In particular, the relative distances of $(X(z_0), Y(z_0))$ to the frontiers are obtained by solving model (11) for z_0. Recall that if the PFC method with hard frontiers is used, then $S_1 = F_{\Lambda,1}$ and $S_2 = F_{\Lambda,2}$; if the PFC method with soft frontiers is used, then $S_1 = \hat{F}_{\Lambda,1}$ and $S_2 = \hat{F}_{\Lambda,2}$. The optimized DDF measurements are $\delta^*_{\Lambda,1}(z_0)$ and $\delta^*_{\Lambda,2}(z_0)$.

The classification rules do neither distinguish between C and NC cases, nor do they distinguish between hard and soft frontiers. The group membership is simply determined by the magnitude of the optimized DDF measurements. Based on the magnitude of the relative DDF measurements, the group membership of a new observation z_0 is decided as follows:

Rule A.1: If $\delta^*_{\Lambda,1}(z_0) \geq 0$ and $\delta^*_{\Lambda,2}(z_0) < 0$, then z_0 belongs to Group 1 and $I(z_0) = 1$;

Rule A.2: If $\delta^*_{\Lambda,1}(z_0) < 0$ and $\delta^*_{\Lambda,2}(z_0) \geq 0$, then z_0 belongs to Group 2 and $I(z_0) = 2$;

Rule A.3: If $\delta^*_{\Lambda,1}(z_0) < 0$ and $\delta^*_{\Lambda,2}(z_0) < 0$, then the observation z_0 is in the gap;

Rule A.4: If $\delta^*_{\Lambda,1}(z_0) \geq 0$ and $\delta^*_{\Lambda,2}(z_0) \geq 0$, then the observation z_0 is in the overlap.

The group membership of the observation z_0 is clear and interpretable under the first two rules. Rule A.1 illustrates that the group label of observation z_0 is 1 because there exists a training observation with smaller X and larger Y that still belongs to Group 1. Rule A.2 illustrates that the group label of observation z_0 is 2 because a training observation with larger X and smaller Y already belongs to Group 2.

If the observation z_0 satisfies the conditions in rules A.3 and A.4, then the current information is insufficient to specify whether it belongs to Group 1 or Group 2. However, by comparing the relative DDF measurements, we can give a reasonable suggestion as to which group the observation may belong to.

Specifically, when the observation z_0 is in the gap specified in Rule A.3, the closer it is located to a frontier, the more similar it is inferred to be to that group. Therefore, z_0 is determined to belong to the group whose frontier is closest. That is, if $0 > \delta^*_{\Lambda,1}(z_0) \geq \delta^*_{\Lambda,2}(z_0)$ holds, then $I(z_0) = 1$. By contrast, if $\delta^*_{\Lambda,1}(z_0) < \delta^*_{\Lambda,2}(z_0) < 0$ holds, then $I(z_0) = 2$.

When the observation z_0 is in the overlap specified in Rule A.4, the more interior it is located within a frontier, the more similar it is inferred to be to that group. Therefore, z_0 is determined to belong to the group with the farthest frontier. That is, if $\delta^*_{\Lambda,1}(z_0) \geq \delta^*_{\Lambda,2}(z_0) \geq 0$ holds, then $I(z_0) = 1$. By contrast, if $0 \leq \delta^*_{\Lambda,1}(z_0) < \delta^*_{\Lambda,2}(z_0)$ holds, then $I(z_0) = 2$.

Classification rules A.1-A.4 are designed in a conservative way. However, in this way the group membership of observations in gaps and overlaps (i.e., satisfying rules A.3 and A.4) can be ambiguous. To provide a uniquely determined group membership for all observations, the following classification rules are further designed based on some inferential information for the observations in the gap and overlap:

Rule B.1: If $\delta^*_{\Lambda,1}(z_0) \geq \delta^*_{\Lambda,2}(z_0)$, then z_0 belongs to Group 1 and $I(z_0) = 1$;

Rule B.2: If $\delta^*_{\Lambda,1}(z_0) < \delta^*_{\Lambda,2}(z_0)$, then z_0 belongs to Group 2 and $I(z_0) = 2$.

4 Experimental Analysis

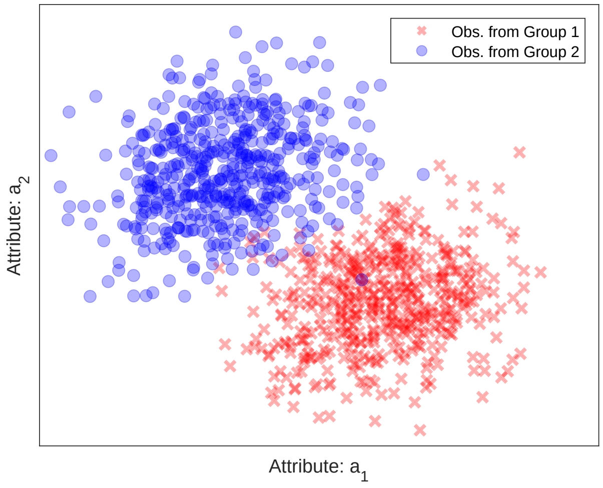
4.1 Simulation Studies

In this subsection, an artificial data set is used to test the four proposed PFC methods, namely the PFC methods with C-hard frontiers (PFC-CHard), the PFC methods with NC-hard frontiers (PFC-NCHard), the PFC methods with C-soft frontiers (PFC-CSoft), and the

PFC methods with NC-soft frontiers (PFC-NCSoft).

In the simulation, the observations from Groups 1 and 2 are generated from two bivariate Normal density distributions $\mathcal{N}_1(\mu_1, \Sigma_1)$ and $\mathcal{N}_2(\mu_2, \Sigma_2)$, respectively. These two bivariate Normal density distributions are characterized with the following parameters: $\mu_1 = (7, 2)$, $\Sigma_1 = \left(\begin{smallmatrix} 4 & 1 \\ 1 & 4 \end{smallmatrix}\right)$, $\mu_2 = (2, 7)$, $\Sigma_2 = \left(\begin{smallmatrix} 4 & 1 \\ 1 & 4 \end{smallmatrix}\right)$. A graphical representation of the artificial data set is depicted in Figure 5. The red diagonal crosses represent the observations belonging to Group 1, while the blue dots represent the observations belonging to Group 2.

Figure 5: Illustration of the Simulated Data set



Each time, the simulation generates 500 observations following $\mathcal{N}_1(\mu_1, \Sigma_1)$, and 500 observations following $\mathcal{N}_2(\mu_2, \Sigma_2)$. From the 500 observations in each group, 400 are used for training and the remaining 100 for testing. This simulation is repeated 100 times and the average performance is reported.

To implement the PFC methods, the monotonicity relation of attributes need to be made explicit. However, in this simulation study, the monotonicity relation of attributes is not explicitly given. As proposed in Section 3, the MSD model (1) can be used to differentiate the attributes. The attribute a_1 has a negative optimal weight value, while the attribute a_2 has a positive one. Thus, attribute a_1 should be considered as an input type and attribute a_2 as an output type.

The performance results of the four proposed PFC methods are summarized in Table 1. Columns 2-3 report the validation accuracy and the percentage of training observations located in the overlap. Columns 4-5 report the prediction accuracy and the percentage of testing observations located in the overlap and testing observations located in the gap. The average CPU time for executing different PFC methods are reported in the last column.

Horizontally, each row reports the average classification performances of a PFC method.

Table 1: Performance Results of the Simulation Study

PFC	Training		Testing			CPU Time (s)
	Accuracy	Overlap	Accuracy	Overlap	Gap	
CHard	0.9766	0.0920	0.9740	0.0810	0.0035	15.3774
NCHard	0.9768	0.0530	0.9776	0.0389	0.0076	0.7552
CSoft	0.9818	0.0023	0.9773	0.0009	0.0105	18.6525
NCSoft	0.9887	0	0.9796	0	0.0252	1.4815

Four observations can be made from Table 1. First, a blind imposition of the convexity axiom leads to additional overlap, but reduces the gap. From the PFC-NCHard method to the PFC-CHard method the percentage of overlap during training increases from 5.30% to 9.20%. This observation on overlap substantiates Proposition 3.2. As for the gap, its percentage change is not revealed in the training process. However, the anticipated increase in gap is reflected in the test sample, specifically from 0.35% to 0.76%. This substantiates Proposition 3.3 to some extent.

Second, the use of soft frontiers can dramatically reduce the overlap without compromising the validation accuracy. In fact, it even improves the validation accuracy in this simulation. From the PFC-CHard method to the PFC-CSoft method, the overlap decreases by 8.97% and the validation accuracy improves by 0.52%. For the NC case, the overlap decreases from 5.30% to 0% and the validation accuracy improves by 1.19%.

Third, more overlap during training is indeed detrimental to the classification performance in this simulation. Put it differently, a smaller overlap during training normally translates into better performance during testing. From the PFC-CHard method to the PFC-NCSoft method, the overlap during training keeps decreasing (from 9.20% to 0%), while the prediction accuracy keeps improving (from 97.66% to 98.87%).

Fourth, while the PFC-NC methods would have required solving the complex BMIP problem, the enumeration-based approach makes it much more computationally efficient. Therefore, the PFC-C methods involve solving the LP problems with an average computational time of 17.0150 seconds. By contrast, the PFC-NC methods take an average of just 1.1184 seconds.

To sum up, the simulation results demonstrate that relaxing the unnecessary convexity axiom can rather substantially reduce the overlap and thus improve the classification performance. In addition, the use of soft frontiers can further reduce the overlap and improve the classification performance. Among the four proposed methods, the PFC-NCSoft method achieves the best classification performance coupled with a good performance in terms of computational time.

4.2 Experiments on a Real-life Data set

In this subsection, an experimental study on a real life data set is conducted to validate the practical classification performance of the proposed PFC methods. It is worth noting that this experimental study intends to show that the proposed PFC method can be a good candidate for two-group classification, rather than defeating some of the well-established classification methods already available.

4.2.1 Data Description and Experimental Design

A credit scoring data set provided by Yeh and Lien (2009) is used. This data set collects data of 30000 credit card holders of a bank in Taiwan. All the observations are characterized by 23 variables, including demographic characteristics, given credit limit, repayment status, bill statement, and history of past payments.[3]

The data set is screened for observations that are not providing effective information. First, if bills and payments for the last 6 months are zero, then the corresponding observations are considered as not providing effective information. Therefore, these observations are removed from the sample. Second, if all payments in the last 5 months are delayed, but the corresponding observations are registered as non-default, then these contradictory observations are removed from the sample. Third, if all payments in the last 5 months are made on time, but the corresponding observations are found to be default, then these contradictory observations are also removed from the sample. After removing 13928 observations according to our selection criteria, the cleaned data set contains 16072 observations. The cleaned data set contains 5492 defaulters and 10580 non-defaulters, which constitutes an unbalanced data set. To respect the unbalanced nature of the problem, a stratified 10-fold cross-validation is conducted.

The proposed PFC methods are designed for problems with monotonicity relations. Thus, four attributes with monotonicity relations are extracted and constructed from the original 23

[3] The data set and its detailed descriptions are available at the UCI Machine Learning Repository database: `https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients`.

variables. A detailed description of these four attributes is displayed in Table 2. Specifically, the first attribute (y_1) is an output-type attribute. A larger value of y_1 represents a better credit history and therefore implies a lower probability of default. The other three attributes, namely x_1, x_2 and x_3, are input-type attributes. Smaller values of x_1, x_2, and x_3 correspond to shorter average delay in repayment, smaller overdue ratio, and smaller credit utilization ratio, respectively. All these input-type attributes imply a lower probability of default.

Table 2: Description of the Constructed Attributes

Attributes	Description
y_1	Given credit limit: LIMIT_BAL
x_1	Average delay in repayment: Based on PAY_1 to PAY_6. Only delayed repayment time is counted, due time repayment (i.e., values of -1 and -2) is not counted, and 0 is replaced by 0.5
x_2	Total overdue amounts/LIMIT_BAL: Based on BILL_AMT1 to BILL_AMT6 and PAY_AMT1 to PAY_AMT6. The overdue amount for month t is calculated by subtracting PAY_AMT(t+1) from BILL_AMT(t). If the overdue amount is negative, replace it with 0.
x_3	September credit utilization ratio: BILL_AMT1/LIMIT_BAL

In addition to the experiment on the whole cleaned data set, we are also interested in investigating the effect of the unbalanced nature of the problem. Thus, three complementary experiments with increased unbalanced ratios are also conducted. Specifically, the unbalanced ratios are 1:5, 1:10 and 1:20. Each time, the minority group randomly selects 100 defaulters from a total of 5492 defaulters, while the majority group randomly selects 500, 1000, and 2000 non-defaulters, respectively, from a total of 10580 non-defaulters. Each time, a stratified 10-fold cross-validation is performed for each different unbalanced ratio. This process is repeated 10 times, and the average performance is reported for different performance measures.

The proposed four PFC methods are compared to in total eleven existing classification methods. These eleven classification methods can be classified into two categories. One category consists of three Data Envelopment Analysis-Discriminant Analysis (DEA-DA) methods, namely, the basic DEA-DA (Sueyoshi, 1999), the extended DEA-DA (Sueyoshi, 2001), and the MIP DEA-DA (Sueyoshi, 2004). The DEA-DA method is essentially based on the use of goal programming. Banker, Chang, and Cooper (2002) have argued that researchers should avoid referring to goal programming models that parameterize prescribed functional forms as DEA models. This justifies why we ignore these basic DEA-DA methods and their further developments in the methodological review supra.[4] Instead, we only focus on the three basic DEA-DA proposals of Sueyoshi (1999, 2001, 2004) in this empirical section. The other category consists of 8 non-DEA methods, namely, MSD-DA, Two-stage MSD-DA, linear DA, quadratic DA, logistic regression, decision tree, Gaussian SVM, and K-Nearest Neighbour (KNN).[5]

The classification performance of these various classification methods is evaluated by 5 measures, namely, precision, recall, specificity, F_2 score and G-mean. Among these, the measures of precision, recall and specificity are calculated from the confusion matrix. The F_2 score and G-mean are overall performance measures. The F_2 score considers recall to be twice as important as precision in calculating overall performance (Christen, Hand, and Kirielle (2023)). This is because in the case of credit card default prediction, failing to identify customers who will likely default is much worse than giving a false alarm with regard to customers who will not default. G-mean is the geometric mean of recall and specificity.

4.2.2 Performance Results

The performance results of the whole cleaned data set are presented in Table 3. The first block reports the performance results of four PFC methods. The second block report the performance results of three DEA-DA methods. The third block report the performance results of eight non-DEA classification methods. Columns 3-7 correspond to five performance measures. In each column, the best result within a specific block is highlighted in bold.

Several observations can be made from the performance results reported in Table 3. First, a comparison of the overall performance of the four proposed PFC methods reveals that the findings of the simulation study are still valid in this experimental analysis. Specifically, the relaxation of the convexity axiom improves the overall performance of the F_2 score and G-mean by between 7.06% and 16.73%. The adoption of soft frontiers improves the overall performance by a more pronounced increase situated between 13.25% and 26.78%. Among the four proposed methods, the PFC-NCSoft method achieves the best performance in terms of both overall measures and it is well ahead of the second best PFC method.

[4] In the literature, DEA-DA has been methodologically refined in a variety of ways (see Lotfi and Mansouri (2008); Boudaghi and Saen (2018) for details on successive developments). It has also been used quite widely in empirical analysis (see Tsai, Lin, Cheng, and Lin (2009); Toloo, Farzipoor Saen, and Azadi (2015)).

[5] Of the 8 non-DEA methods, the code for the last six are directly exported from the Matlab Classification Learner app. All other methods are coded in Matlab by the authors: these codes are available upon simple request.

Second, in terms of the performance of the different groups, from the PFC-CHard method to the PFC-NCSoft method, the performance of identifying the defaulters continues to improve dramatically, while the performance of predicting the non-defaulters deteriorates only slightly. Specifically, from C to NC, recall increases by an average of 18.34%, while specificity decreases by an average of only 4.72%. From the hard frontiers to the soft frontiers, recall increases by an average of 29.97%, while specificity decreases by an average of only 7.29%. Moreover, the difference between recall and specificity drops from 59.18% to 1.14%. It implies that, from C to NC and from hard to soft frontiers, the PFC methods show a more and more balanced performance in predicting different groups.

Table 3: The 10-fold cross validation performance on the whole data set

		Precision	Recall	Specificity	F_2	G-Mean
PFC	CHard	**0.8296**	0.3689	**0.9607**	0.4150	0.5953
	NCHard	0.8201	0.5430	0.9382	0.5823	0.7137
	Csoft	0.7963	0.6593	0.9125	0.6828	0.7756
	NCSoft	0.7350	**0.8520**	0.8405	**0.8257**	**0.8462**
DEA-DA	Basic	**0.7621**	0.7500	**0.8784**	0.7524	0.8117
	Extended	0.7201	**0.8598**	0.8266	**0.8277**	**0.8430**
	MIP	0.6568	0.7591	0.7940	0.7362	0.7764
Non-DEA	MSD DA	0.6606	**0.9039**	0.7590	**0.8419**	0.8283
	Two-stage MSD DA	0.7214	0.8587	0.8279	0.8272	0.8432
	Linear DA	0.8802	0.6890	0.9513	0.7203	0.8096
	Quadratic DA	0.7983	0.7562	0.9009	0.7643	0.8254
	Logitic Regression	0.8218	0.7522	0.9153	0.7651	0.8298
	Decision Tree	0.8855	0.7536	0.9494	0.7768	0.8459
	Gaussian SVM	**0.8937**	0.7566	**0.9533**	0.7805	**0.8493**
	KNN	0.7730	0.7746	0.8819	0.7743	0.8265

Third, in comparison with the best performing DEA-DA method, i.e., the extended DEA-DA method, the PFC-NCSoft method offers a more balanced performance, while being competitive on overall performance. Specifically, the extended DEA-DA method performs a little better in identifying defaulters, with a 0.78% higher recall. However, it deteriorates in predicting non-defaulters, with a 1.40% lower specificity. Thus, the extended DEA-DA method shows more difference in performance across groups.

Fourth, in comparison with the non-DEA classification methods, the PFC-NCSoft method continues to offer a more balanced performance, while remaining competitive on overall performance. Specifically, the MSD DA method has the highest F_2 score (1.62% higher than the PFC-NCSoft method), while the Gaussian SVM has the highest value of G-mean (0.3% higher than the PFC-NCSoft method). However, the MSD DA method shows a difference of up to 14.49% in correctly predicting defaulters and non-defaulters (i.e., the difference between recall and specificity). This difference for the Gaussian SVM method is even larger and can be up to 19.68%.

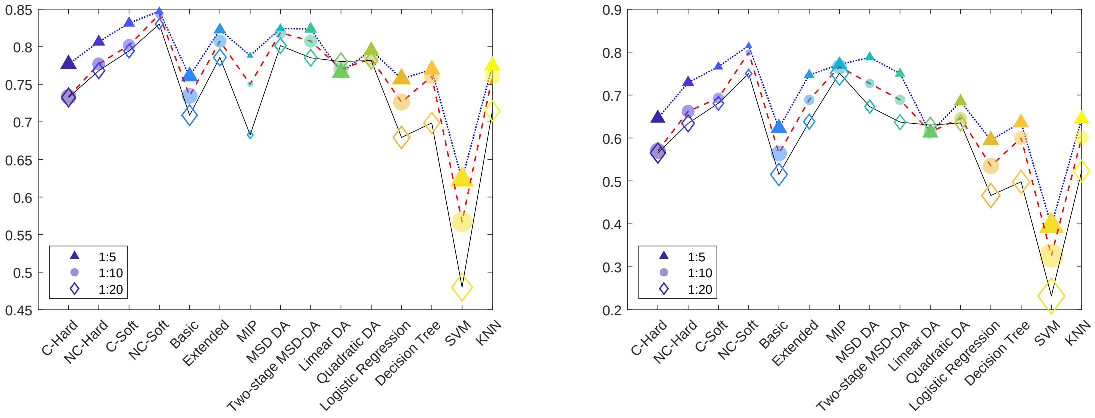
Table 4: Performance results under different unbalanced ratios

	Methods	G-Mean			Recall			Specificity		
		1:5	1:10	1:20	1:5	1:10	1:20	1:5	1:10	1:20
PFC	CHard	0.7768	0.7335	0.7325	0.6460	0.5700	0.5650	**0.9352**	**0.9474**	**0.9548**
	NCHard	0.8065	0.7771	0.7681	0.7290	0.6620	0.6340	0.8934	0.9177	0.9330
	CSoft	0.8314	0.8020	0.7945	0.7660	0.6930	0.6810	0.9038	0.9316	0.9278
	NCSoft	**0.8472**	**0.8429**	**0.8302**	**0.8150**	**0.7990**	**0.7500**	0.8810	0.8895	0.9196
DEA-DA	Basic	0.7606	0.7346	0.7088	0.6230	0.5640	0.5150	**0.9288**	**0.9579**	**0.9759**
	Extended	**0.8226**	**0.8074**	**0.7858**	0.7470	0.6890	0.6380	0.9060	0.9465	0.9687
	MIP	0.7882	0.7502	0.6828	**0.7710**	**0.7660**	**0.7510**	0.8070	0.7402	0.6355
Non-DEA	MSD DA	**0.8243**	**0.8188**	**0.8014**	**0.7880**	**0.7270**	**0.6730**	0.8624	0.9226	0.9550
	Two-stage MSD DA	0.8236	0.8075	0.7853	0.7500	0.6890	0.6370	0.9046	0.9468	0.9690
	Linear DA	0.7657	0.7704	0.7803	0.6120	0.6140	0.6300	0.9588	0.9675	0.9681
	Quadratic DA	0.7954	0.7839	0.7817	0.6850	0.6460	0.6350	0.9240	0.9520	0.9639
	Logiatic Regression	0.7572	0.7263	0.6793	0.5950	0.5350	0.4660	0.9646	0.9871	0.9940
	Decision Tree	0.7698	0.7607	0.6986	0.6360	0.6000	0.4980	0.9320	0.9657	0.9835
	Gaussian SVM	0.6233	0.5669	0.4799	0.3960	0.3260	0.2320	**0.9842**	**0.9970**	**0.9987**
	KNN	0.7747	0.7604	0.7146	0.6450	0.6010	0.5230	0.9312	0.9631	0.9785

The performance results under different unbalanced ratios are summarized in Table 4. When confronted with unbalanced data sets, some classification methods tend to trivialize the minority group and pursue a good overall performance by predicting the majority group as correctly as possible. Therefore, we pay special attention to the classification performance of the default (minority) and non-default (majority) groups, corresponding to recall and specificity, respectively. When characterizing overall performance, we report G-mean calculated based on recall and specificity, rather than F_2 scores calculated based on recall and precision.

The first two findings related to the PFC methods derived from Table 3 are still observed from Table 4. The last two findings have changed slightly. Specifically, in terms of the overall performance, the PFC-NCSoft method becomes the one that consistently performs best compared to all the other methods. Moreover, the performance superiority of the PFC-NCSoft method increases as the unbalanced ratio increases. When focusing on recall, the PFC-NCSoft method is also consistently the best performing model. The only minor exception is that the recall of the MIP DEA-DA method is 0.1% higher when the unbalanced ratio is 1:20. The specificity of the PFC-NCSoft method is not as high as the other methods, but its difference with recall is smaller, which implies a more balanced performance.

The performance results in Table 4 are further visualized in Figure 6. The vertical axes of Figures 6(a) and 6(b) report the G-mean and recall, respectively. The marker size in Figure 6(a) represents the differences between recall and specificity. When the marker is smaller, then the difference is smaller, which implies a more balanced performance. The marker size in Figure 6(b) shows the performance difference from the best performing method. When the marker is smaller, then the difference is smaller, indicating a better overall performance.



(a) G-mean

(b) Recall

Figure 6: Performance comparison under different unbalanced ratios

Three key messages can be deduced from Figure 6. First, the overall performance of the PFC method with less overlap is less negatively affected by the increase in the unbalanced ratio. The overlap decreases from the PFC-CHard method to the PFC-NCSoft method. Thus, from left to right in Figure 6, the overall performance polyline for an unbalanced ratio of 1:20 is converging with the overall performance polyline for an unbalanced ratio of 1:5. Second, under different unbalanced ratios, the PFC-NCSoft method always achieves the best overall performance and achieves a fairly balanced performance. Third, under different unbalanced ratios, the PFC-NCSoft method always has the best recall coupled with the best overall performance. Note that the MIP DEA-DA method also performs well in terms of recall, but rather poorly in terms of overall performance.

In general, the proposed PFC-NCSoft method shows a very competitive overall performance. In addition, it shows a unique advantage in the case of unbalanced data sets, i.e., it achieves a more balanced performance in predicting two unbalanced groups.

5 Conclusions

Classification is one of the most important and challenging problems in machine learning. Despite recent advances in the MP-based classification methods, there remains the need for development of a simpler, automated, and interpretable method. The PFC method generates a nonlinear separating boundary based on part of the training observations. Moreover, the shape of the generated separating boundary depends largely on the data and does not need to be pre-specified. In this sense, the PFC method should be a good candidate in estimating the separating boundary whose shape is mostly unspecified in applications. However, the PFC method has not yet been able to prevail as a standard classification tool.

In this contribution, three methodological contributions result in the construction of a general PFC method with competitive classification performance. First, for situations where the monotonicity relation is not available, a MSD model is proposed to decide on the monotonicity relation prevailing for the attributes. Second, the axiom of convexity that has previously been blindly retained can be relaxed. By relaxing the convexity axiom, we propose a PFC method that generates a pair of NC frontiers with, among others, substantially less overlap. We derive some results on the relations between overlap and gap under convexity and nonconvexity. Third, for cases where overlap exists, an algorithm is designed to minimize the overlap so that the training observations in the overlap are classified as correctly as possible, while all training observations that have been correctly classified remain correctly classified. In this sense, the PFC method with hard frontiers is extended to the case of soft frontiers.

Both simulation studies and experimental analysis show that the overall performance of the proposed PFC methods is improving from C to NC and from hard to soft frontiers due to the effective reduction of overlap, as somehow anticipated in our theoretical results. Experimental analysis of credit card data shows that the PFC-NCSoft method is quite competitive in terms of overall performance compared to some existing classification methods. In addition, the experimental results also demonstrate that the PFC-NCSoft method has a unique advantage on unbalanced data sets.

There remain some directions for future investigation. Just as relaxing the convexity axiom inherited from production theory delivers promising results, one may wonder whether it is also possible to relax the free disposability axiom that is also inherited from production theory, and to which extent this could further improve the classification performance of the proposed PFC method. In production analysis, one recent attempt to relax the free disposability axiom is developed in Briec, Kerstens, and Van de Woestyne (2016) and empirically implemented in Briec, Kerstens, and Van de Woestyne (2018). Another direction worth exploring is to test the proposed PFC methods with even more unbalanced data sets to check the consistency of its advantage on unbalanced data. Finally, one may wonder to which extent more robust versions of the PFC methods (e.g., order-m or order-α) would improve the classification performance.

References

BANKER, R., H. CHANG, AND W. COOPER (2002): ""Small Sample Properties of ML, COLS and DEA Estimators of Frontier Models in the Presence of Heteroscedasticity" by AN Bojanic, SB Caudill and JM Ford, European Journal of Operational Research 108, 1998, 140–148: A Comment," *European Journal of Operational Research*, 136(2), 466–467.

BOUDAGHI, E., AND R. F. SAEN (2018): "Developing a Novel Model of Data Envelopment Analysis–Discriminant Analysis for Predicting Group Membership of Suppliers in Sustainable Supply Chain," *Computers & Operations Research*, 89, 348–359.

BRIEC, W., K. KERSTENS, AND I. VAN DE WOESTYNE (2016): "Congestion in Production Correspondences," *Journal of Economics*, 119(1), 65–90.

——— (2018): "Hypercongestion in Production Correspondences: An Empirical Exploration," *Applied Economics*, 50(27), 2938–2956.

CHAMBERS, R., Y. CHUNG, AND R. FÄRE (1998): "Profit, Directional Distance Functions, and Nerlovian Efficiency," *Journal of Optimization Theory and Applications*, 98(2), 351–364.

CHANG, D. S., AND Y. C. KUO (2005): "A Novel Procedure to Identify the Minimized Overlap Boundary of Two Groups by DEA Model," in *Lecture Notes in Computer Science*, ed. by O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganà, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, vol. 3483, pp. 577–586. Springer, Berlin, Heidelberg.

——— (2008): "An Approach for the Two-group Discriminant Analysis: An Application of DEA," *Mathematical and Computer Modelling*, 47(9-10), 970–981.

CHERCHYE, L., T. KUOSMANEN, AND T. POST (2001): "FDH Directional Distance Functions with An Application to European Commercial Banks," *Journal of Productivity Analysis*, 15(3), 201–215.

CHRISTEN, P., D. J. HAND, AND N. KIRIELLE (2023): "A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives," *ACM Computing Surveys*, 56(3), 1–24.

DE BOCK, K. W., K. COUSSEMENT, AND S. LESSMANN (2020): "Cost-sensitive Business Failure Prediction when Misclassification Costs are Uncertain: A Heterogeneous Ensemble Selection Approach," *European Journal of Operational Research*, 285(2), 612–630.

De Caigny, A., K. Coussement, K. W. De Bock, and S. Lessmann (2020): "Incorporating Textual Information in Customer Churn Prediction Models based on a Convolutional Neural Network," *International Journal of Forecasting*, 36(4), 1563–1578.

Deprins, D., L. Simar, and H. Tulkens (1984): "Measuring Labor Efficiency in Post Offices," in *The Performance of Public Enterprises: Concepts and Measurements*, ed. by M. Marchand, P. Pestieau, and H. Tulkens, pp. 243–268. North Holland, Amsterdam.

Emrouznejad, A., R. D. Banker, and L. Neralic (2019): "Advances in Data Envelopment Analysis: Celebrating the 40th Anniversary of DEA and the 100th Anniversary of Professor Abraham Charnes' Birthday," *European Journal of Operational Research*, 278(2), 365–367.

Farbmacher, H., L. Löw, and M. Spindler (2022): "An Explainable Attention Network for Fraud Detection in Claims Management," *Journal of Econometrics*, 228(2), 244–258.

Freed, N., and F. Glover (1981): "Simple but Powerful Goal Programming Models for Discriminant Problems," *European Journal of Operational Research*, 7(1), 44–60.

Freed, N., and F. Glover (1986): "Evaluating Alternative Linear Programming Models to Solve the Two-Group Discriminant Problem," *Decision Sciences*, 17(2), 151–162.

Glover, F. (1990): "Improved Linear Programming Models for Discriminant Analysis," *Decision Sciences*, 21(4), 771–785.

Jin, Q., K. Kerstens, and I. Van de Woestyne (2024): "Convex and Nonconvex Nonparametric Frontier-based Classification Methods for Anomaly Detection," *OR Spectrum*, forthcoming.

Kerstens, K., and I. Van de Woestyne (2011): "Negative Data in DEA: A Simple Proportional Distance Function Approach," *Journal of the Operational Research Society*, 62(7), 1413–1419.

Kotsiantis, S. B., I. Zaharakis, and P. Pintelas (2007): "Supervised Machine Learning: A Review of Classification Techniques," *Emerging Artificial Intelligence Applications in Computer Engineering*, 160, 3–24.

Kuo, Y. C. (2013): "Consideration of Uneven Misclassification Cost and Group Size for Bankruptcy Prediction," *American Journal of Industrial and Business Management*, 3(08), 708.

Leon, C. F., and F. Palacios (2009): “Evaluation of Rejected Cases in an Acceptance System with Data Envelopment Analysis and Goal Programming,” *Journal of the Operational Research Society*, 60(10), 1411–1420.

Lessmann, S., B. Baesens, H.-V. Seow, and L. C. Thomas (2015): “Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring: An Update of Research,” *European Journal of Operational Research*, 247(1), 124–136.

Lotfi, F. H., and B. Mansouri (2008): “The Extended Data Envelopment Analysis/Discriminant Analysis Approach of Fuzzy Models,” *Applied Mathematical Sciences*, 2(30), 1465–1477.

Merdan, S., C. L. Barnett, B. T. Denton, J. E. Montie, and D. C. Miller (2021): “Data Analytics for Optimal Detection of Metastatic Prostate Cancer,” *Operations Research*, 69(3), 774–794.

Ouenniche, J., S. Carrales, K. Tone, and H. Fukuyama (2017): “An Account of DEA-Based Contributions in the Banking Sector,” in *Advances in DEA theory and applications: With extensions to forecasting models*, ed. by K. Tone, chap. 14, pp. 141–171. John Wiley & Sons, Ltd.

Pendharkar, P. (2012): “Fuzzy Classification Using the Data Envelopment Analysis,” *Knowledge-Based Systems*, 31, 183–192.

Pendharkar, P. (2018): “Data Envelopment Analysis Models for Probabilistic Classification,” *Computers & Industrial Engineering*, 119, 181–192.

Pendharkar, P., M. Khosrowpour, and J. Rodger (2000): “Application of Bayesian Network Classifiers and Data Envelopment Analysis for Mining Breast Cancer Patterns,” *Journal of Computer Information Systems*, 40(4), 127–132.

Pendharkar, P., J. Rodger, and G. Yaverbaum (1999): “Association, Statistical, Mathematical and Neural Approaches for Mining Breast Cancer Patterns,” *Expert Systems with Applications*, 17(3), 223–232.

Pendharkar, P. C. (2002): “A Potential Use of Data Envelopment Analysis for the Inverse Classification Problem,” *Omega*, 30(3), 243–248.

Pendharkar, P. C. (2011): “A Hybrid Radial Basis Function and Data Envelopment Analysis Neural Network for Classification,” *Computers & Operations Research*, 38(1), 256–266.

PENDHARKAR, P. C., AND M. D. TROUTT (2014): "Interactive Classification Using Data Envelopment Analysis," *Annals of Operations Research*, 214(1), 125–141.

SEIFORD, L., AND J. ZHU (1998): "An Acceptance System Decision Rule with Data Envelopment Analysis," *Computers & Operations Research*, 25(4), 329–332.

SILVA, A. P. D. (2017): "Optimization Approaches to Supervised Classification," *European Journal of Operational Research*, 261(2), 772–788.

SUEYOSHI, T. (1999): "DEA-Discriminant Analysis in the View of Goal Programming," *European Journal of Operational Research*, 115(3), 564–582.

——— (2001): "Extended DEA-Discriminant Analysis," *European Journal of Operational Research*, 131(2), 324–351.

——— (2004): "Mixed Integer Programming Approach of Extended DEA–Discriminant Analysis," *European Journal of Operational Research*, 152(1), 45–55.

TOLOO, M., R. FARZIPOOR SAEN, AND M. AZADI (2015): "Obviating Some of the Theoretical Barriers of Data Envelopment Analysis-Discriminant Analysis: An Application in Predicting Cluster Membership of Customers," *Journal of the Operational Research Society*, 66, 674–683.

TROUTT, M., A. RAI, AND A. ZHANG (1996): "The Potential Use of DEA for Credit Applicant Acceptance Systems," *Computers & Operations Research*, 23(4), 405–408.

TSAI, M.-C., S.-P. LIN, C.-C. CHENG, AND Y.-P. LIN (2009): "The Consumer Loan Default Predicting Model–An Application of DEA–DA and Neural Network," *Expert Systems with Applications*, 36(9), 11682–11690.

YAN, H., AND Q. WEI (2011): "Data Envelopment Analysis Classification Machine," *Information Sciences*, 181(22), 5029–5041.

YEH, I., AND C. LIEN (2009): "The Comparisons of Data Mining Techniques for the Predictive Accuracy of Probability of Default of Credit Card Clients," *Expert Systems with Applications*, 36(2), 2473–2480.

ZHAO, J., J. OUENNICHE, AND J. DE SMEDT (2024): "Survey, Classification and Critical Analysis of the Literature on Corporate Bankruptcy and Financial Distress Prediction," *Machine Learning with Applications*, 15, 100527.

Appendices: Supplementary Material

A Appendix: Proofs

Proof of Proposition 3.1:

Proof. Let $(x_0, y_0) \in T^*_{NC,1}$, then there are weights $\lambda_{j_0} = 1$ and $\lambda_j = 0$ $(z_j \in S_1, j \neq j_0)$ such that $\sum_{z_j \in S_1} \lambda_j X(z_j) \leq x_0$, $\sum_{z_j \in S_1} \lambda_j Y(z_j) \geq y_0$ and $\sum_{z_j \in S_1} \lambda_j = 1$. Since $\lambda_{j_0} = 1$ and $\lambda_j = 0$ $(z_j \in S_1, j \neq j_0)$ also satisfy the constraint $\lambda_j \geq 0$, $(x_0, y_0) \in T^*_{C,1}$, thus $T^*_{NC,1} \subseteq T^*_{C,1}$.

Using arguments paralleling the above, one can prove that $T^*_{NC,2} \subseteq T^*_{C,2}$.

Proof of Proposition 3.2:

Proof. Using Proposition 3.1, let $\Delta_1 = T^*_{C,1} \setminus T^*_{NC,1}$ and $\Delta_2 = T^*_{C,2} \setminus T^*_{NC,2}$. Thus, $T^*_{C,1} = T^*_{NC,1} \cup \Delta_1$, $T^*_{C,2} = T^*_{NC,2} \cup \Delta_2$, while $T^*_{NC,1} \cap \Delta_1 = \emptyset$ and $T^*_{NC,2} \cap \Delta_2 = \emptyset$. The overlap under the C case, $T^*_{C,Overlap}$, can be expressed as a union of several parts as follows:

$$\begin{aligned}
T^*_{C,Overlap} &= T^*_{C,1} \cap T^*_{C,2} = T^*_{C,1} \cap \left(T^*_{NC,2} \cup \Delta_2\right) = \left(T^*_{C,1} \cap T^*_{NC,2}\right) \cup \left(T^*_{C,1} \cap \Delta_2\right) \\
&= \left[\left(T^*_{NC,1} \cup \Delta_1\right) \cap T^*_{NC,2}\right] \cup \left[\left(T^*_{NC,1} \cup \Delta_1\right) \cap \Delta_2\right] \\
&= \left[\left(T^*_{NC,1} \cap T^*_{NC,2}\right) \cup \left(\Delta_1 \cap T^*_{NC,2}\right)\right] \cup \left[\left(T^*_{NC,1} \cap \Delta_2\right) \cup \left(\Delta_1 \cap \Delta_2\right)\right] \\
&= T^*_{NC,Overlap} \cup \left(\Delta_1 \cap T^*_{NC,2}\right) \cup \left(T^*_{NC,1} \cap \Delta_2\right) \cup \left(\Delta_1 \cap \Delta_2\right).
\end{aligned} \tag{A1}$$

Thus, $T^*_{NC,Overlap} \subseteq T^*_{C,Overlap}$.

Proof of Corollary 3.1:

Proof. We first prove that if $T^*_{C,Overlap} \neq \emptyset$, $T^*_{NC,Overlap} \neq \emptyset$, $T^*_{C,1} = T^*_{NC,1}$ and $T^*_{C,2} = T^*_{NC,2}$ hold, then $T^*_{C,Overlap} = T^*_{NC,Overlap} \neq \emptyset$ holds.

Using Proposition 3.1, let $\Delta_1 = T^*_{C,1} \backslash T^*_{NC,1}$ and $\Delta_2 = T^*_{C,2} \backslash T^*_{NC,2}$. Since $T^*_{C,1} = T^*_{NC,1}$ and $T^*_{C,2} = T^*_{NC,2}$, we have $\Delta_1 = \Delta_2 = \emptyset$. Correspondingly, $\Delta_1 \cap T^*_{NC,2}$, $T^*_{NC,1} \cap \Delta_2$ and $\Delta_1 \cap \Delta_2$ are all empty. With expression (A1), $T^*_{C,Overlap} = T^*_{NC,Overlap}$ holds. Since $T^*_{C,Overlap} \neq \emptyset$ and $T^*_{NC,Overlap} \neq \emptyset$, naturally $T^*_{C,Overlap} = T^*_{NC,Overlap} \neq \emptyset$ holds, as desired.

We then prove that if $T^*_{C,Overlap} = T^*_{NC,Overlap} \neq \emptyset$ holds, then $T^*_{C,Overlap} \neq \emptyset$, $T^*_{NC,Overlap} \neq \emptyset$, $T^*_{C,1} = T^*_{NC,1}$ and $T^*_{C,2} = T^*_{NC,2}$ hold.

Since $T^*_{C,Overlap} = T^*_{NC,Overlap} \neq \emptyset$, $T^*_{C,Overlap} \neq \emptyset$ and $T^*_{NC,Overlap} \neq \emptyset$ naturally hold. The nonempty overlaps under the NC and the C cases can be represented as follows:

$$\begin{aligned} T^*_{NC,Overlap} &= T^*_{NC,1} \cap T^*_{NC,2} = \Bigg\{ (x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_1} \lambda_{j,1} X(z_j) \leq x \leq \sum_{z_j \in S_2} \lambda_{j,2} X(z_j), \\ &\sum_{z_j \in S_1} \lambda_{j,1} Y(z_j) \geq y \geq \sum_{z_j \in S_2} \lambda_{j,2} Y(z_j), \sum_{z_j \in S_1} \lambda_{j,1} = \sum_{z_j \in S_2} \lambda_{j,2} = 1, \lambda_{j,1}, \lambda_{j,2} \in \{0,1\} \Bigg\}, \end{aligned} \tag{A2}$$

$$\begin{aligned} T^*_{C,Overlap} &= T^*_{C,1} \cap T^*_{C,2} = \Bigg\{ (x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_1} \lambda_{j,1} X(z_j) \leq x \leq \sum_{z_j \in S_2} \lambda_{j,2} X(z_j), \\ &\sum_{z_j \in S_1} \lambda_{j,1} Y(z_j) \geq y \geq \sum_{z_j \in S_2} \lambda_{j,2} Y(z_j), \sum_{z_j \in S_1} \lambda_{j,1} = \sum_{z_j \in S_2} \lambda_{j,2} = 1, \lambda_{j,1}, \lambda_{j,2} \geq 0 \Bigg\}. \end{aligned} \tag{A3}$$

Let the sets of all strongly efficient observations under the NC case in S_1 and S_2 be:

$$S'_{NC,1} = \left\{ z_j \in S_1 \mid x \leq X(z_j), y \geq Y(z_j) \text{ and } x \neq X(z_j), y \neq Y(z_j) \Rightarrow (x,y) \notin T^*_{NC,1} \right\}, \tag{A4}$$

$$S'_{NC,2} = \left\{ z_j \in S_2 \mid x \geq X(z_j), y \leq Y(z_j) \text{ and } x \neq X(z_j), y \neq Y(z_j) \Rightarrow (x,y) \notin T^*_{NC,2} \right\}. \tag{A5}$$

Moreover, $T^*_{NC,Overlap}$ is only determined by the strongly efficient observations that are located in the overlap. Let $S''_{NC,1}$ and $S''_{NC,2}$ represent the subsets containing all strongly efficient observations that are located in $T^*_{C,Overlap}$. Specifically, $S''_{NC,1} = \{z_j \in S'_{NC,1} \mid (X(z_j), Y(z_j)) \in T^*_{NC,Overlap}\}$, and $S''_{NC,2} = \{z_j \in S'_{NC,2} \mid (X(z_j), Y(z_j)) \in T^*_{NC,Overlap}\}$.

Instead of using all the training observations in S_1 and S_2, the estimates $T^*_{NC,Overlap}$ and $T^*_{C,Overlap}$ can now be represented as follows:

$$\begin{aligned} T^*_{NC,Overlap} = &\Bigg\{ (x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S''_{NC,1}} \lambda_{j,1} X(z_j) \leq x \leq \sum_{z_j \in S''_{NC,2}} \lambda_{j,2} X(z_j), \\ &\sum_{z_j \in S''_{NC,1}} \lambda_{j,1} Y(z_j) \geq y \geq \sum_{z_j \in S''_{NC,2}} \lambda_{j,2} Y(z_j), \sum_{z_j \in S''_{NC,1}} \lambda_{j,1} = \sum_{z_j \in S''_{NC,2}} \lambda_{j,2} = 1, \lambda_{j,1}, \lambda_{j,2} \in \{0,1\} \Bigg\}, \end{aligned} \tag{A6}$$

$$\begin{aligned} T^*_{C,Overlap} = &\Bigg\{ (x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \\ &\sum_{z_j \in S''_{NC,1}} \lambda_{j,1} X(z_j) + \sum_{z_j \in S'_{NC,1} \backslash S''_{NC,1}} \lambda_{j,1} X(z_j) \leq x \leq \sum_{z_j \in S''_{NC,2}} \lambda_{j,2} X(z_j) + \sum_{z_j \in S'_{NC,2} \backslash S''_{NC,2}} \lambda_{j,2} X(z_j), \\ &\sum_{z_j \in S''_{NC,1}} \lambda_{j,1} Y(z_j) + \sum_{z_j \in S'_{NC,1} \backslash S''_{NC,1}} \lambda_{j,1} Y(z_j) \geq y \geq \sum_{z_j \in S''_{NC,2}} \lambda_{j,2} Y(z_j) + \sum_{z_j \in S'_{NC,2} \backslash S''_{NC,2}} \lambda_{j,2} Y(z_j), \\ &\sum_{z_j \in S''_{NC,1}} \lambda_{j,1} + \sum_{z_j \in S'_{NC,1} \backslash S''_{NC,1}} \lambda_{j,1} = \sum_{z_j \in S''_{NC,2}} \lambda_{j,2} + \sum_{z_j \in S'_{NC,2} \backslash S''_{NC,2}} \lambda_{j,2} = 1, \lambda_{j,1}, \lambda_{j,2} \geq 0 \Bigg\}. \end{aligned} \tag{A7}$$

Since $T^*_{NC,Overlap} \neq \emptyset$, there exist at least one strongly efficient observation $z_{j_0,1} \in S''_{NC,1}$ and at least one strongly efficient observation $z_{j_0,2} \in S''_{NC,2}$ such that the constraints $X(z_{j_0,1}) \leq X(z_{j_0,2})$ and $Y(z_{j_0,1}) \geq Y(z_{j_0,2})$ are satisfied. That is, part of $T^*_{NC,Overlap}$ represented by $z_{j_0,1}$ and $z_{j_0,2}$ can be described as: $T^*_{NC,Overlap}(z_{j_0,1}, z_{j_0,2}) = \{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid X(z_{j_0,1}) \leq x \leq X(z_{j_0,2}), Y(z_{j_0,1}) \geq y \geq Y(z_{j_0,2})\}$.

Assuming that there is another strongly efficient observation $z_{j_1,1} \in S''_{NC,1}$, then the partial overlap under the NC and the C cases are described as follows:

$$\begin{aligned} T^*_{NC,Overlap} \quad & (z_{j_0,1}, z_{j_1,1}, z_{j_0,2}) = \{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid X(z_{j_0,1}) \leq x \leq X(z_{j_0,2}), Y(z_{j_0,1}) \geq y \geq Y(z_{j_0,2})\} \\ & \cup \{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid X(z_{j_1,1}) \leq x \leq X(z_{j_0,2}), Y(z_{j_1,1}) \geq y \geq Y(z_{j_0,2})\}, \end{aligned} \tag{A8}$$

$$\begin{aligned} T^*_{C,Overlap} \quad & (z_{j_0,1}, z_{j_1,1}, z_{j_0,2}) = \bigcup_{\lambda \in [0,1]} T^*_{NC,Overlap}(\lambda z_{j_0,1} + (1-\lambda) z_{j_1,1}, z_{j_0,2}) \\ & = \bigcup_{\lambda \in [0,1]} \{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \lambda X(z_{j_0,1}) + (1-\lambda) X(z_{j_1,1}) \leq x \leq X(z_{j_0,2}), \lambda Y(z_{j_0,1}) + (1-\lambda) Y(z_{j_1,1}) \geq y \geq Y(z_{j_0,2})\}. \end{aligned} \tag{A9}$$

Apparently, the additional strongly efficient observation $z_{j_1,1} \in S''_{NC,1}$ makes expression (A8) a proper subset of expression (A9). Using similar arguments, one can prove that any additional strongly efficient observations in $S''_{NC,1}$ and $S''_{NC,2}$ leads to $T^*_{NC,Overlap} \subset T^*_{C,Overlap}$, which contradicts with $T^*_{NC,Overlap} = T^*_{C,Overlap}$. Thus, there can be only one strongly efficient observation from each group located in the overlap so that $T^*_{NC,Overlap} = T^*_{C,Overlap}$ holds. The estimates $T^*_{NC,Overlap}$ and $T^*_{C,Overlap}$ can be now expressed as follows:

$$T^*_{NC,Overlap} = \{(x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid X(z_{j_0,1}) \leq x \leq X(z_{j_0,2}), Y(z_{j_0,1}) \geq y \geq Y(z_{j_0,2})\} \tag{A10}$$

$$\begin{aligned} T^*_{C,Overlap} = \quad & \Bigg\{ (x,y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \\ & \lambda_{j_0,1} X(z_{j_0,1}) + \sum_{z_j \in S'_{NC,1} \setminus \{z_{j_0,1}\}} \lambda_{j,1} X(z_j) \leq x \leq \lambda_{j_0,2} X(z_{j_0,2}) + \sum_{z_j \in S'_{NC,2} \setminus \{z_{j_0,2}\}} \lambda_{j,2} X(z_j), \\ & \lambda_{j_0,1} Y(z_{j_0,1}) + \sum_{z_j \in S'_{NC,1} \setminus \{z_{j_0,1}\}} \lambda_{j_0,1} Y(z_j) \geq y \geq \lambda_{j_0,2} Y(z_{j_0,2}) + \sum_{z_j \in S'_{NC,2} \setminus \{z_{j_0,2}\}} \lambda_{j,2} Y(z_j), \\ & \lambda_{j_0,1} + \sum_{z_j \in S'_{NC,1} \setminus \{z_{j_0,1}\}} \lambda_{j,1} = \lambda_{j_0,2} + \sum_{z_j \in S'_{NC,2} \setminus \{z_{j_0,2}\}} \lambda_{j,2} = 1, \lambda_{j_0,1}, \lambda_{j_0,2}, \lambda_{j,1}, \lambda_{j,2} \geq 0 \Bigg\}. \end{aligned} \tag{A11}$$

If $S'_{NC,1} \setminus \{z_{j_0,1}\} \neq \emptyset$ or $S'_{NC,2} \setminus \{z_{j_0,2}\} \neq \emptyset$, then expression (A10) will be a proper subset of expression (A11), which contradicts with $T^*_{NC,Overlap} = T^*_{C,Overlap}$. Thus, $S'_{NC,1} \setminus \{z_{j_0,1}\} = S'_{NC,2} \setminus \{z_{j_0,2}\} = \emptyset$ must be met to ensure that $T^*_{NC,Overlap} = T^*_{C,Overlap}$ holds. In other words, there should be no strongly efficient observation located outside the overlap.

Summarizing the arguments above, $T^*_{NC,Overlap} = T^*_{C,Overlap}$ implies that there is only one strongly efficient observation for each group and they are located in the overlap. When there is only one strongly efficient observation for each group, $T^*_{C,1} = T^*_{NC,1}$ and $T^*_{C,2} = T^*_{NC,2}$ hold.

Thus, $T^*_{C,Overlap} = T^*_{NC,Overlap} \neq \emptyset$ if and only if $T^*_{C,Overlap} \neq \emptyset$, $T^*_{NC,Overlap} \neq \emptyset$, $T^*_{C,1} = T^*_{NC,1}$ and $T^*_{C,2} = T^*_{NC,2}$ hold.

Proof of Proposition 3.3:

Proof. Using Proposition 3.1, let $\Delta_1 = T^*_{C,1} \setminus T^*_{NC,1}$ and $\Delta_2 = T^*_{C,2} \setminus T^*_{NC,2}$. That is, $T^*_{C,1} = T^*_{NC,1} \cup \Delta_1$, $T^*_{C,2} = T^*_{NC,2} \cup \Delta_2$, while $T^*_{NC,1} \cap \Delta_1 = \emptyset$ and $T^*_{NC,2} \cap \Delta_2 = \emptyset$. With Δ_1 and Δ_2, we have $T^*_{C,1} \cup T^*_{C,2} = \big(T^*_{NC,1} \cup \Delta_1\big) \cup \big(T^*_{NC,2} \cup \Delta_2\big) = T^*_{NC,1} \cup T^*_{NC,2} \cup \Delta_1 \cup \Delta_2$, therefore $\big(T^*_{C,1} \cup T^*_{C,2}\big) \supseteq \big(T^*_{NC,1} \cup T^*_{NC,2}\big)$. Since $\big(T^*_{C,1} \cup T^*_{C,2}\big) \supseteq \big(T^*_{NC,1} \cup T^*_{NC,2}\big)$, define $\Delta = \big(T^*_{C,1} \cup T^*_{C,2}\big) \setminus \big(T^*_{NC,1} \cup T^*_{NC,2}\big)$. Correspondingly, $\big(T^*_{NC,1} \cup T^*_{NC,2}\big)^{\complement} = \big[\big(T^*_{C,1} \cup T^*_{C,2}\big) \setminus \Delta\big]^{\complement} = \big[\big(T^*_{C,1} \cup T^*_{C,2}\big) \cap \Delta^{\complement}\big]^{\complement} = \big(T^*_{C,1} \cup T^*_{C,2}\big)^{\complement} \cup \Delta$. Thus, $T^*_{C,Gap} \subseteq T^*_{NC,Gap}$.

Proof of Corollary 3.2:

Proof. Using Proposition 3.1, define $\Delta_1 = T^*_{C,1} \setminus T^*_{NC,1}$ and $\Delta_2 = T^*_{C,2} \setminus T^*_{NC,2}$. Then, $T^*_{C,1} \setminus T^*_{NC,1} \subseteq T^*_{NC,2}$ can be expressed as $\Delta_1 \subseteq T^*_{NC,2}$. Similarly, $T^*_{C,2} \setminus T^*_{NC,2} \subseteq T^*_{NC,1}$ can be expressed as $\Delta_2 \subseteq T^*_{NC,1}$.

We first prove that if $\Delta_1 \subseteq T^*_{NC,2}$ and $\Delta_2 \subseteq T^*_{NC,1}$ hold, then $T^*_{NC,Gap} = T^*_{C,Gap}$ holds. Since $\Delta_1 \subseteq T^*_{NC,2}$, we have $\Delta_1 \cup T^*_{NC,2} = T^*_{NC,2}$. Correspondingly with $\Delta_2 \subseteq T^*_{NC,1}$, we have $\Delta_2 \cup T^*_{NC,1} = T^*_{NC,1}$. With Δ_1 and Δ_2, we also have $T^*_{C,1} \cup T^*_{C,2} = \big(T^*_{NC,1} \cup \Delta_1\big) \cup \big(T^*_{NC,2} \cup \Delta_2\big) = T^*_{NC,1} \cup T^*_{NC,2} \cup \Delta_1 \cup \Delta_2 = \big(T^*_{NC,1} \cup \Delta_2\big) \cup \big(T^*_{NC,2} \cup \Delta_1\big)$. Since $\Delta_1 \cup T^*_{NC,2} = T^*_{NC,2}$ and $\Delta_2 \cup T^*_{NC,1} = T^*_{NC,1}$, therefore $T^*_{C,1} \cup T^*_{C,2} = T^*_{NC,1} \cup T^*_{NC,2}$. Correspondingly, $\big(T^*_{C,1} \cup T^*_{C,2}\big)^{\complement} = \big(T^*_{NC,1} \cup T^*_{NC,2}\big)^{\complement}$. Thus, $T^*_{C,Gap} = T^*_{NC,Gap}$ holds, as desired.

We then prove that if $T^*_{C,Gap} = T^*_{NC,Gap}$ holds, then $\Delta_1 \subseteq T^*_{NC,2}$ and $\Delta_2 \subseteq T^*_{NC,1}$ hold. $T^*_{C,Gap} = T^*_{NC,Gap}$ can be expressed as $\big(T^*_{C,1} \cup T^*_{C,2}\big)^{\complement} = \big(T^*_{NC,1} \cup T^*_{NC,2}\big)^{\complement}$. Therefore, $T^*_{C,1} \cup T^*_{C,2} = T^*_{NC,1} \cup T^*_{NC,2}$ holds. Since $T^*_{C,1} \cup T^*_{C,2}$ can be expressed as $\big(T^*_{NC,1} \cup \Delta_1\big) \cup \big(T^*_{NC,2} \cup \Delta_2\big) = \big(T^*_{NC,1} \cup T^*_{NC,2}\big) \cup (\Delta_1 \cup \Delta_2)$, it follows that $\big(T^*_{NC,1} \cup T^*_{NC,2}\big) \cup (\Delta_1 \cup \Delta_2) = T^*_{NC,1} \cup T^*_{NC,2}$. That is, $\Delta_1 \cup \Delta_2 \subseteq T^*_{NC,1} \cup T^*_{NC,2}$.

According to the transitive relation, since $\Delta_1 \subseteq (\Delta_1 \cup \Delta_2)$ and $(\Delta_1 \cup \Delta_2) \subseteq (T^*_{NC,1} \cup T^*_{NC,2})$, therefore $\Delta_1 \subseteq (T^*_{NC,1} \cup T^*_{NC,2})$. Since $\Delta_1 \cap T^*_{NC,1} = \emptyset$, therefore $\Delta_1 \subseteq T^*_{NC,2}$, as desired.

Using arguments paralleling the above, one can prove that $\Delta_2 \subseteq T^*_{NC,1}$.

Thus, $T^*_{NC,Gap} = T^*_{C,Gap}$ if and only if $T^*_{C,1} \setminus T^*_{NC,1} \subseteq T^*_{NC,2}$ and $T^*_{C,2} \setminus T^*_{NC,2} \subseteq T^*_{NC,1}$.