



Innovative Applications of O.R.

A pairwise-frontier-based classification framework for two-group classification

Qianying Jin^a*, Kristiaan Kerstens^b, Ignace Van de Woestyne^c, Zhongbao Zhou^d

^a College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China

^b Univ. Lille, CNRS, IESEG School of Management, UMR 9221 - LEM - Lille Économie Management, F-59000, Lille, France

^c KU Leuven, Research Centre for Operations Research and Statistics (ORSTAT), Brussels Campus, Warmoesberg 26, B-1000 Brussels, Belgium

^d School of Business Administration, Hunan University, Changsha, 410082, China

ARTICLE INFO

Keywords:

Data Envelopment Analysis

Frontier

Nonconvex

Convex

Two-group classification

ABSTRACT

Mathematical programming-based methods are widely used to generate separating boundaries in two-group classification problems. While nonlinear boundaries may often yield better performance, they typically require pre-specifying a nonlinear functional form. This contribution proposes a novel pairwise-frontier-based classification (PFC) framework that approximates nonlinear separating boundaries without the need for such a specification. The proposed PFC framework clarifies the role of axioms in classification, providing a clear basis for axiom selection based on problem characteristics. Building on this, PFC methods with nonconvex (NC) frontiers are developed, ensuring that the overlap is not artificially enlarged. To further minimize overlap, certain training observations are deliberately controlled to misclassify, provided that the total misclassification cost does not increase. This enables the extension from hard frontiers to more stable soft frontiers. The applicability of the proposed PFC methods is validated through simulation studies and real-life datasets. The results show that the proposed PFC methods are competitive with some well-established classifiers. Specifically, the PFC-NCSOFT method demonstrates superior average performance and greater stability, making it particularly suitable for imbalanced classification tasks.

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 customer churn (e.g., De Caigny et al. (2020)), fraud and failure detection (e.g., De Bock et al. (2020)), credit scoring (e.g., Farbmacher et al. (2022)), Lessmann et al. (2015)), medical diagnosis (e.g., Merdan et al. (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 et al. (2007) and Silva (2017). For expository clarity, this contribution focuses on the two-group classification problem.¹

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.

* Corresponding author.

E-mail addresses: qianying.jin@nuaa.edu.cn (Q. Jin), k.kerstens@ieseg.fr (K. Kerstens), ignace.vandewoestyne@kuleuven.be (I. Van de Woestyne), Z.B.Zhou@hnu.edu.cn (Z. Zhou).

¹ The classification problem here should be distinguished from group performance evaluation (e.g., Walheer (2018)). While the latter assesses technology differences between predefined groups, classification aims to predict the unknown group membership. Another major supervised classification problem is multi-group classification, where a given dataset is divided into more than two distinct groups. The multi-group classification problem can be transformed into a series of two-group classification problems by adopting the one against one approach (Silva, 2017). Therefore, this contribution focuses on the most fundamental two-group classification problem.

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) model, which generates a C piecewise linear frontier, is of interest. The DEA model is a linear programming model evaluating the efficiency of observations by projecting them onto a C piecewise linear frontier. It is widely applied in production economics and finance (see recent surveys and historical developments in Ouenniche et al. (2017) and Emrouznejad et al. (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 et al. (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 & 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 (2002), Pendharkar et al. (2000, 1999)). 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 cannot 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 enveloping 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 choose to eliminate overlap entirely during the training process (Chang & Kuo, 2005, 2008; Kuo, 2013). With this treatment, the overlap is effectively 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 neighborhood approach (Pendharkar, 2011), membership functions (Pendharkar, 2012), interaction or Minimum Sum of Deviations (MSD) method (Pendharkar & 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.

Before addressing how to handle overlap, it is essential to recognize that the overlap is shaped by the mix of axioms used to construct the frontiers. Different mixes of axioms produce frontiers with distinct geometric properties, which in turn yield overlaps of varying extent. This foundational relationship has been largely overlooked in the PFC literature. The existing PFC methods follow the geometric intuition of DEA and directly adopt the C piecewise linear frontier for classification, without examining the meaning or justification of the underlying axioms. The only partial exception that we are aware of is Jin et al. (2024) who analyze the correspondence of axioms when applying frontier-based classification methods to anomaly detection. However, their study employs only a single frontier, leaving the issue of overlap between two groups unaddressed.

Therefore, the first purpose of this contribution is to establish an interpretable framework for PFC methods that clarifies the role of axioms in classification. Specifically, the free disposability axiom corresponds to the monotonicity relations among attributes, and the convexity axiom corresponds to attribute substitutability. This correspondence provides an interpretable basis for selecting axioms based on problem characteristics, thereby moving beyond purely geometric applications. This theoretical basis enables us, for the first time, to

formally demonstrate how different combinations of axioms shape the geometry of frontiers and, consequently, the extent of overlap.

Second, this is the first attempt to introduce pairwise nonconvex (NC) frontiers into a classification framework. Building on the theoretical basis, the convexity axiom should be relaxed if no attribute substitutability is specified. In this case, the Free Disposal Hull (FDH) model proposed by Deprins et al. (1984) is employed to derive the NC frontier. Compared to the C frontiers, the NC frontiers envelop the groups of training observations more tightly and generally produce a smaller overlap. Consequently, NC frontiers are expected to enhance classification performance when two groups are less separable.

Third, a new concept of soft frontiers is proposed to handle overlap. Although NC frontiers help reduce overlap, some overlap often persists where the groups intersect. To address this, an optimization algorithm is designed that iteratively minimizes overlap until stable soft frontiers are achieved. This algorithm reduces overlap by allowing a controlled set of training observations to be misclassified. Such controlled misclassification is only allowed when it strictly reduces ambiguity without increasing the total misclassification cost. This soft frontier approach offers a distinct alternative to existing methods of handling overlap. It neither introduces excessive misclassification nor relies on external classifiers at the expense of abandoning the frontier-based framework.

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

2. Geometric illustrations of PFC methods

Figs. 1–4 are provided to visually explain the intuitive idea of PFC methods. Every training observation is characterized by two attributes. Compared to Group 1, the training observations in Group 2 predominantly occupy the region with lower values on the horizontal axis and higher values on the vertical axis. This relative position suggests that the horizontal axis portrays an input-type attribute while the vertical axis portrays an output-type attribute.

Fig. 1 depicts a nonlinearly separable example. The two groups of training observations can be well separated by a pair of NC-hard frontiers as depicted in Fig. 1(a). However, when employing the C-hard frontiers, overlap is detected as the gray filled area in Fig. 1(b). This observed overlap is not inherent but introduced by imposing the convexity axiom.

Regardless of the modeling axioms, overlap caused by inherently ambiguous distributions or noise often exists in practice. In these cases, the training groups inevitably overlap, either using the C-hard or the NC-hard frontiers. To explicitly address overlaps arising beyond modeling axioms, we focus in the subsequent illustrations on the NC case.

Fig. 2 depicts two cases where groups of observations are inseparable. The overlap in Fig. 2(a) arises from the group distributions. By contrast, Fig. 2(b) maintains identical group distributions, but introduces a single mislabeled training observation. The former is described as the clean case, and the latter is described as the noisy case.

In the clean case depicted in Fig. 2(a), 25 training observations fall into the overlap when hard frontiers are employed. Training observations outside this overlap are classified with certainty, but those inside cannot be clearly classified based on their location relative to the frontiers. Although applying the distance-based classification rules could correctly assign labels to 13 training observations, 12 remain to be misclassified. If the classifier generalizes well, then this ambiguity in the training process would translate to the test process. That is, approximately 12.5% of the test samples will fall into the overlap

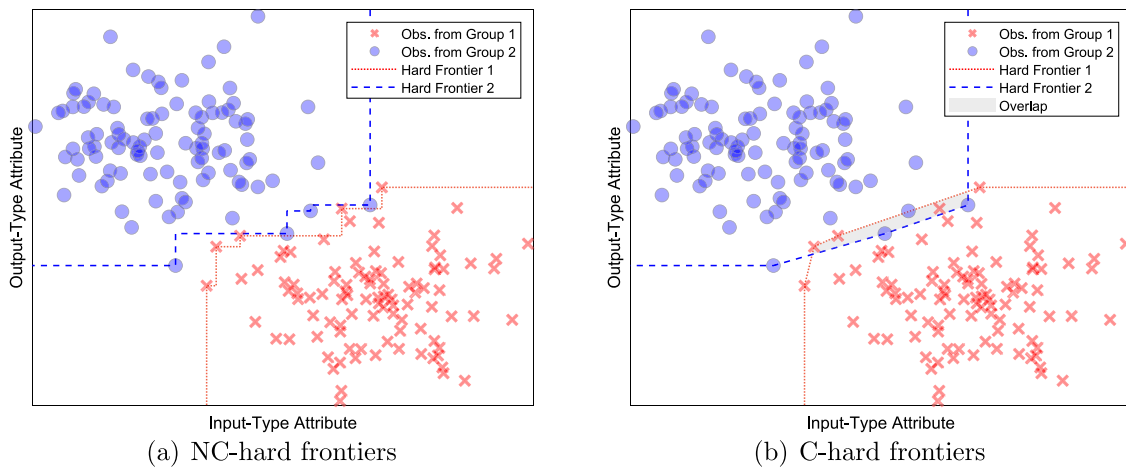


Fig. 1. An ideal example with nonlinearly separable groups.

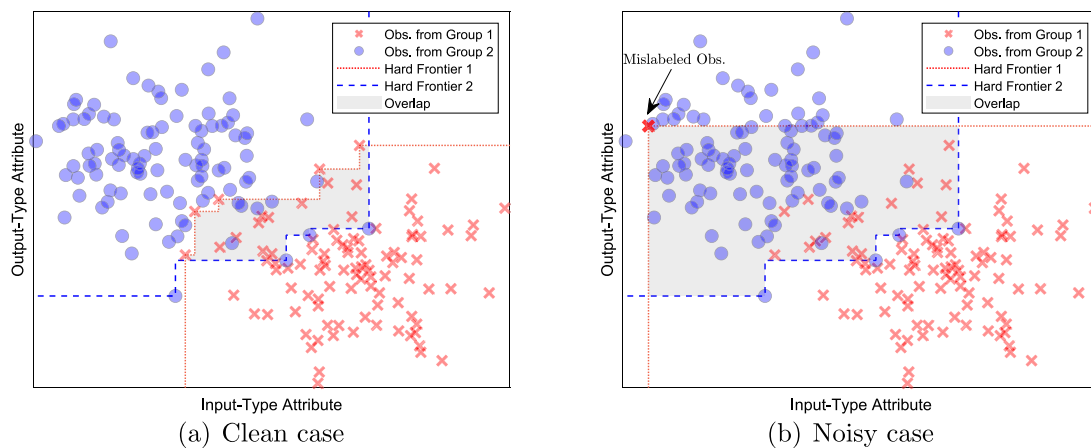


Fig. 2. A realistic example with inseparable groups: hard frontiers.

with labeling ambiguity, with 48% of these ambiguous cases being misclassified.

In the noisy case depicted in Fig. 2(b), the overlap expands from 25 to 89 training observations due to one mislabeled training observation. That is, 32% of the training observations, which would otherwise be clearly classified, now becomes subject to the choice of the direction vector. Even if the distance-based classification rules are applied, 24 training observations are still misclassified. In other words, a single mislabeled training observation amplifies the misclassification resulting from overlap by 200%.

Thus, a larger overlap introduces greater ambiguity, which in most cases leads to more misclassifications. To mitigate the ambiguity, overlap minimization must be prioritized during training. By intentionally excluding some of the training observations, the derived frontiers shift inwards and generate a smaller overlap. An algorithm is proposed in Section 3.4 to derive the soft frontiers that realize a minimum overlap.

Fig. 3 visualizes the derived soft frontiers. For the clean case in Fig. 3(a), no overlap is observed with the soft frontiers. Compared to Fig. 2(a), the original 12.5% overlap is now completely eliminated by intentionally misclassifying only 5 training observations. By sacrificing this 2.5% training accuracy, unambiguous classification is achieved for the remaining 97.5% of training samples.

For the noisy case in Fig. 3(b), the derived soft frontiers are the same as those for the clean case, demonstrating robustness against labeling noise. Compared to the 5 misclassifications in the clean case, the noisy case requires misclassifying 6 training observations. However, the additional misclassified single observation corresponds to the initially

mislabeled sample. While nominally counted as a misclassification, its relabeling through soft frontiers effectively corrects the original labeling error. Compared to Fig. 2(b), the original 44.5% overlap is eliminated and the training accuracy improves from 88% to 97%.

The final separating boundary is jointly determined by the pairwise frontiers. As visualized in Figs. 4(a) and 4(b), the separating boundary obtained from hard frontiers is highly sensitive to labeling noise, varying significantly between the noisy and clean cases. By contrast, the boundary derived from soft frontiers remains identical under both cases, as shown in Fig. 4(c). This demonstrates that, unlike their hard counterparts, soft frontiers yield a robust classifier that is stable against labeling noise.

3. Pairwise-frontier-based classification framework

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, which consists of a finite set of training observations.

For every training observation z_j from the finite set Z , its group label $L(z_j)$ is known. Specifically, the training observations in Z are exclusively classified into either Group 1 or Group 2. Thus, $Z = Z_1 \cup Z_2$ and $Z_1 \cap Z_2 = \emptyset$, where Z_l denotes the set of training observations that belong to Group l , i.e., $Z_l = \{z_j \in Z \mid L(z_j) = l\}$ ($l \in \{1, 2\}$).

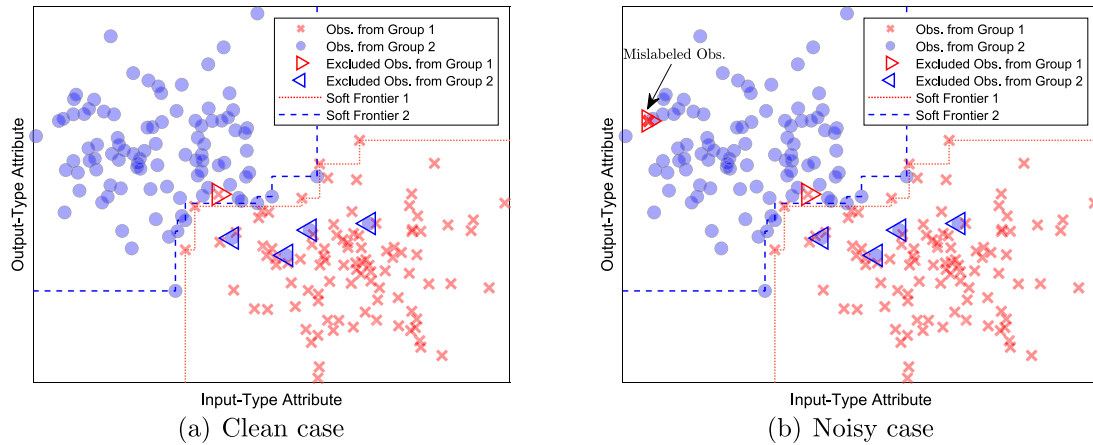


Fig. 3. A realistic example with inseparable groups: soft frontiers.

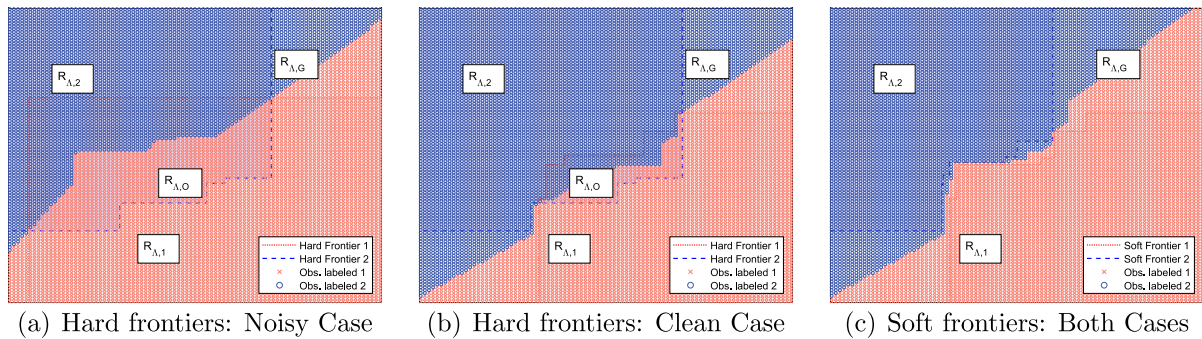


Fig. 4. Classification results with pairwise NC frontiers.

Typically in the classification process, observations are evaluated by a number of attributes, represented by $A = \{a_1, \dots, a_K\}$. The attribute vector of observation z can then be denoted by $A(z) = (a_1(z), \dots, a_K(z))$. At the core of classification process is some function f that, based on the attribute vector, returns either 1 representing Group 1 or 2 representing Group 2. This allows us to introduce the function I that maps an observation z to its predicted group label as follows: $I(z) = 1 + \mathbb{I}_{[0,\infty)}(f(A(z)))$, where $\mathbb{I}_{[0,\infty)}$ is the indicator function of $[0, \infty)$. The key to resolving the classification problem then boils down to finding an optimal function f such that for all training observations z_j , the predicted group labels $I(z_j)$ and the corresponding known group labels $L(z_j)$ are as identical as possible.

In MP-based classifiers, the functional form of f is pre-specified, which essentially constrains the form of the quantitative relation between attribute values and group labels. In this contribution, we do not impose any assumptions on the form of the quantitative relation, but we simply make an assumption on the direction of this relation.

The direction of the relation between attribute values and group labels is twofold.² If the value of $f(A(z))$ increases (decreases) with the increase (decrease) of a particular attribute $a_k(z)$, assuming all other attributes remain constant, then there is a monotonically increasing

relation between this attribute and group labels. Otherwise, there is a monotonically decreasing relation. The former relation is named the output-type, and the corresponding attributes are reorganized into $Y = \{y_1, \dots, y_s\}$. The latter is named the input-type, and the corresponding attributes are reorganized into $X = \{x_1, \dots, x_m\}$. That is, $A = X \cup Y$ and $K = m + s$. The function I that maps an observation to its predicted group label is now determined by $I(z) = 1 + \mathbb{I}_{[0,\infty)}(f(X(z), Y(z)))$, where $X(z)$ and $Y(z)$ represent the evaluation of observation z on the attributes X and Y , respectively.

3.2. Acceptance possibility set and its estimates

In the context of classification, we use the Acceptance Possibility Set (APS) to describe the attainable set of a certain group. The concept of APS is analogous to the Production Possibility Set (PPS) in production analysis. On the one hand, the PPS describes that certain resource-product correspondences are producible under a given technology. On the other hand, the APS describes that certain attribute-group label correspondences are acceptable by a given group. Specifically, the APS of Group l describes all possible combinations of attribute values that correspond to observations with predicted group label of $I(z) = l$.

Since the true attribute-group label correspondences are partially unknown, the basic idea is to construct an estimated APS from the observed training observations. The estimated APS is constructed according to the minimal extrapolation principle, i.e., it is the smallest subset that contains the observed training observations and satisfies some axioms.

² The direction of the relation between attribute values and group labels are normally prior-knowledge of a classification problem. In cases where this relation is not explicitly given, a linear programming (LP) model can be used to distinguish between attributes: see Appendix A for an example.

The Monotonicity Axiom (also named free disposability axiom) ensures that the function f preserves the dominance relationship between observations. If observation z_1 dominates z_2 by exhibiting better attribute values, then the function must assign a value to z_1 that is no smaller than that of z_2 . Let $\mathcal{Z}$ be the domain of possible attribute combinations.

Monotonicity Axiom: For any $z_1, z_2 \in \mathcal{Z}$, if $(X(z_1), -Y(z_1)) \leq (X(z_2), -Y(z_2))$, then $f(X(z_1), Y(z_1)) \geq f(X(z_2), Y(z_2))$.

Under the monotonicity axiom, the classification depends on the dominance relationship relative to the training observations. If an observation z is dominated by a training observation from Group 1 characterized by larger X or smaller Y , then it receives a smaller $f(X(z), Y(z))$ and is assigned a predicted group label of $I(z) = 1$. By contrast, if z dominates a training observation from Group 2 characterized by smaller X or larger Y , then it achieves a higher $f(X(z), Y(z))$ resulting in a predicted group label of $I(z) = 2$.

The Convexity Axiom is expressed as a substitution relation between attributes. In some classification applications, there may exist this type of substitution relation. For example, the admission decision to a college is made based on several attributes of a candidate, including the GMAT and the SAT scores. In evaluating a candidate, the disadvantage in the GMAT score can be compensated to a certain degree by the advantage in the SAT score. In this case, there exists a substitution relation between GMAT and SAT scores. The following Convexity Axiom then needs to be introduced to characterize these two attributes.

Convexity Axiom: For any $z_1, z_2 \in \mathcal{Z}$ and any $\alpha \in [0, 1]$, the following condition holds: $f(\alpha(X(z_1), Y(z_1)) + (1 - \alpha)(X(z_2), Y(z_2))) \geq \alpha f(X(z_1), Y(z_1)) + (1 - \alpha)f(X(z_2), Y(z_2))$.

Based on the training observations and both above axioms, we can construct the estimated APSs. Let S_l denote the set of training observations known to belong to Group l . The estimated APS of Group l is the constructed as follows:

$$T_{A,1}^* = \left\{ (x, y) \in \mathbb{R}^m \times \mathbb{R}^s \mid \sum_{z_j \in S_1} \lambda_{j,1} X(z_j) \leq x, \sum_{z_j \in S_1} \lambda_{j,1} Y(z_j) \geq y, \sum_{z_j \in S_1} \lambda_{j,1} = 1, \lambda_{j,1} \in \Lambda \right\}, \quad (1)$$

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

where (i) $\Lambda \equiv \Lambda^{\text{NC}} = \{\lambda_{j,1}, \lambda_{j,2} \in \{0, 1\}\}$, or (ii) $\Lambda \equiv \Lambda^{\text{C}} = \{\lambda_{j,1}, \lambda_{j,2} \geq 0\}$.

Note that blindly adding the convexity axiom may increase overlap and thereby compromise classification ability. Based on the subset relationship between $T_{A,1}^*$ and $T_{A,2}^*$, the subset relationships for the overlap $T_{A,\text{Overlap}}^* = T_{A,1}^* \cap T_{A,2}^*$ and gap $T_{A,\text{Gap}}^* = (T_{A,1}^* \cup T_{A,2}^*)^{\text{C}}$ can be deduced, where $(\cdot)^{\text{C}}$ denotes the complement of a set with respect to $\mathbb{R}^m \times \mathbb{R}^s$. These subset relationships are formalized into the next Propositions 3.1 to 3.3 and proven in Appendix B.

Proposition 3.1. $T_{A,1}^{\text{NC}*} \subseteq T_{A,1}^*$, $T_{A,2}^{\text{NC}*} \subseteq T_{A,2}^*$

Proposition 3.2. $T_{A,\text{Overlap}}^{\text{NC}*} \subseteq T_{A,\text{Overlap}}^*$. In particular, $T_{A,\text{Overlap}}^{\text{NC}*} = T_{A,\text{Overlap}}^* \neq \emptyset$ if and only if $T_{A,\text{Overlap}}^* \neq \emptyset$, $T_{A,\text{Overlap}}^{\text{NC}*} \neq \emptyset$, $T_{A,1}^{\text{NC}*} = T_{A,1}^*$ and $T_{A,2}^{\text{NC}*} = T_{A,2}^*$.

Proposition 3.3. $T_{A,\text{Gap}}^{\text{NC}*} \subseteq T_{A,\text{Gap}}^*$. In particular, $T_{A,\text{Gap}}^{\text{NC}*} = T_{A,\text{Gap}}^*$ if and only if $T_{A,1}^* \setminus T_{A,1}^{\text{NC}*} \subseteq T_{A,2}^{\text{NC}*}$ and $T_{A,2}^* \setminus T_{A,2}^{\text{NC}*} \subseteq T_{A,1}^{\text{NC}*}$.

The NC overlap is a subset of the C overlap, while the C gap is a subset of the NC gap. During the training process, no training observation will be left out in the gap. Therefore, if there is no substitution relation, the inclusion of the convexity axiom would only lead to greater overlap. However, if there is substitution relation, then the convexity axiom should be included so that the training samples can properly represent the population.

For a functional representation of the estimated APSs, the directional distance function (DDF) is used (Chambers et al., 1998). Unlike conventional radial measures, which lack directional flexibility and force reliance on external methods for classifying overlap, the DDF enables a consistent classification rule across all regions, without requiring any other classification methods. Accordingly, $T_{A,1}^*$ and $T_{A,2}^*$ are determined using the following DDF measures, respectively:

$$D_{T_{A,l}^*}(z_0, g_l) = \sup \{ \delta_{A,l}(z_0; S_l, g_l) \in \mathbb{R} \mid (X(z_0), Y(z_0)) + \delta_{A,l}(z_0; S_l, g_l) g_l(z_0) \in T_{A,l}^* \}, \quad (3)$$

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 projected direction vectors. To ensure the feasibility of the DDF measure defined in expression (3), the direction vectors must be non-zero. For example, if an observation lies outside the APS and the projected direction vector assumes a zero value, then the DDF measure may become unbounded or fail to converge. Practically, a minimal positive perturbation term ϵ can be used to prevent this issue. That is, $g_1 = (-\max(|g_{X,1}|, \epsilon \cdot \mathbf{1}_m), \max(|g_{Y,1}|, \epsilon \cdot \mathbf{1}_s))$ and $g_2 = (\max(|g_{X,2}|, \epsilon \cdot \mathbf{1}_m), -\max(|g_{Y,2}|, \epsilon \cdot \mathbf{1}_s))$, where $\mathbf{1}_m$ and $\mathbf{1}_s$ denote all-ones vectors of dimensions m and s , respectively.

Throughout this work, $\delta_{A,l}(z_0; S_l, g_l)$ denotes the distance value from the observation z_0 along the projection direction g_l to the boundary of set S_l . When S_l and g_l are unchanged or have been specified in context, we abbreviate it to $\delta_{A,l}(z_0)$ for brevity. Specifically, $\delta_{A,l}(z_0; S_l, g_l)$ can be calculated from model (4).

$$\begin{aligned} \max_{\lambda_{j,l}, \delta_{A,l}} \quad & \delta_{A,1}(z_0; S_1, g_1) + \delta_{A,2}(z_0; S_2, g_2) \\ \text{s.t.} \quad & \sum_{z_j \in S_1} \lambda_{j,1} x_i(z_j) \leq x_i(z_0) + \delta_{A,1}(z_0; S_1, g_1) g_{x_i,1}(z_0) \quad \forall x_i \in X \\ & \sum_{z_j \in S_1} \lambda_{j,1} y_r(z_j) \geq y_r(z_0) + \delta_{A,1}(z_0; S_1, g_1) g_{y_r,1}(z_0) \quad \forall y_r \in Y \\ & \sum_{z_j \in S_1} \lambda_{j,1} = 1, \lambda_{j,1} \in \Lambda \\ & \sum_{z_j \in S_2} \lambda_{j,2} x_i(z_j) \geq x_i(z_0) + \delta_{A,2}(z_0; S_2, g_2) g_{x_i,2}(z_0) \quad \forall x_i \in X \\ & \sum_{z_j \in S_2} \lambda_{j,2} y_r(z_j) \leq y_r(z_0) + \delta_{A,2}(z_0; S_2, g_2) g_{y_r,2}(z_0) \quad \forall y_r \in Y \\ & \sum_{z_j \in S_2} \lambda_{j,2} = 1, \lambda_{j,2} \in \Lambda \end{aligned} \quad (4)$$

where (i) $\Lambda \equiv \Lambda^{\text{NC}} = \{\lambda_{j,1}, \lambda_{j,2} \in \{0, 1\}\}$, or (ii) $\Lambda \equiv \Lambda^{\text{C}} = \{\lambda_{j,1}, \lambda_{j,2} \geq 0\}$.

In the C case, model (4) is solving a LP problem, while it involves solving a binary mixed integer program for the NC case. To speed up computations in the NC case, a fast implicit enumeration-based method is proposed by Cherchye et al. (2001) requiring only to compute minima of lists of ratios. Thus, the following exact solutions are obtained for model (4) under the NC case inspired by Cherchye et al. (2001):

$$\delta_{A,1}^{\text{NC}}(z_0) = \max_{z_j \in S_1} \left\{ \min \left(\min_{x_i \in X} \left(\frac{x_i(z_0) - x_i(z_j)}{|g_{x_i,1}(z_0)|} \right), \min_{y_r \in Y} \left(\frac{y_r(z_j) - y_r(z_0)}{|g_{y_r,1}(z_0)|} \right) \right) \right\}, \quad (5)$$

$$\delta_{A,2}^{\text{NC}}(z_0) = \max_{z_j \in S_2} \left\{ \min \left(\min_{x_i \in X} \left(\frac{x_i(z_j) - x_i(z_0)}{|g_{x_i,2}(z_0)|} \right), \min_{y_r \in Y} \left(\frac{y_r(z_0) - y_r(z_j)}{|g_{y_r,2}(z_0)|} \right) \right) \right\}. \quad (6)$$

3.3. PFC with hard frontiers

In this subsection, pairwise hard frontiers are constructed to represent the APS estimates of Groups 1 and 2. The use of pairwise hard frontiers is sufficient to determine a separating boundary in two

situations. The first is that of well-characterized training groups without overlap; the second is where there is overlap but it is decided to do nothing about it during the training process. In both situations, the frontiers are constructed based on all the training observations and are therefore called hard frontiers.

To identify the training observations that determine the hard frontier of Group l ($l \in \{1, 2\}$), model (4) is solved by letting $S_l = Z_l$. For every $z_0 \in Z_l$, if its optimized DDF value is 0, then it is collected into the hard frontier set, denoted by $F_{A,l}$, as follows:

$$F_{A,l} = \{z_0 \in Z_l \mid \delta_{A,l}^*(z_0; Z_l, g_l) = 0\} \quad l \in \{1, 2\}. \quad (7)$$

Note that different choices of the direction vectors g_l lead to varying DDF values. However, this choice does not affect the sign of the DDF measure, and thus does not influence the construction of the frontiers.

The two hard frontiers, represented by $F_{A,1}$ and $F_{A,2}$, partition the attribute space into four disjoint subsets:

$$R_{A,1}(\cdot) = \{z_0 \in \cdot \mid \delta_{A,1}^*(z_0) \geq 0 \text{ and } \delta_{A,2}^*(z_0) < 0\}, \quad (8)$$

$$R_{A,2}(\cdot) = \{z_0 \in \cdot \mid \delta_{A,1}^*(z_0) < 0 \text{ and } \delta_{A,2}^*(z_0) \geq 0\}, \quad (9)$$

$$R_{A,O}(\cdot) = \{z_0 \in \cdot \mid \delta_{A,1}^*(z_0) \geq 0 \text{ and } \delta_{A,2}^*(z_0) \geq 0\}, \quad (10)$$

$$R_{A,G}(\cdot) = \{z_0 \in \cdot \mid \delta_{A,1}^*(z_0) < 0 \text{ and } \delta_{A,2}^*(z_0) < 0\}. \quad (11)$$

In these definitions, $\cdot$ is a placeholder for a set of input observations, such as the set of training observations. The optimized DDF values $\delta_{A,1}^*(z_0)$ and $\delta_{A,2}^*(z_0)$ for an observation z_0 are computed by solving model (4) with S_l configured as $F_{A,l}$.

The subsets $R_{A,1}$ and $R_{A,2}$ represent observations located in disjoint regions, with each subset lying on opposite sides of the frontiers. Specifically, $R_{A,1}$ includes observations that lie within frontier 1 but outside frontier 2, while $R_{A,2}$ includes those within frontier 2 but outside frontier 1. Training observations that satisfy the inequality conditions defining $R_{A,l}$ all belong to Group l . This empirical pattern suggests that any new observation falling in $R_{A,l}$ should be assigned to Group l .

The subsets $R_{A,O}$ and $R_{A,G}$ constitute the ambiguous region. The overlap region, defined by subset $R_{A,O}$, contains observations located in the intersection of the two frontiers. Training observations in $R_{A,O}$ originate from both groups. Therefore, the group membership of a new observation falling into this region is ambiguous, as evidence supports both groups.

By contrast, the gap region, defined by subset $R_{A,G}$, consists of observations located outside both frontiers. This subset is typically empty in the training data, as no training observation would satisfy the corresponding inequality conditions. Consequently, there is no evidence available to support any group for new observations that fall into the gap region.

Although the mathematical expressions of $R_{A,O}$ and $R_{A,G}$ differ, a unified classification rule based on the relative magnitudes of the DDF values can be applied to both regions. In both regions, the inequality $\delta_{A,1}^*(z_0) \geq \delta_{A,2}^*(z_0)$ identifies the group to which z_0 is more similar. Specifically, in $R_{A,O}$, a larger DDF value (more positive) indicates that the observation lies more interior to the frontier, suggesting greater similarity to the corresponding group. In $R_{A,G}$, a larger DDF value (less negative) indicates proximity to the frontier from the outside, also suggesting more similarity to the corresponding group.

To sum up, a unique group label can be assigned to any observation z_0 with the frontier sets $F_{A,1}$ and $F_{A,2}$. The unified classification rule is defined by the following mapping function:

$$I(z_0) = 1 + \mathbb{I}_{(0,\infty)}(\delta_{A,2}^*(z_0; F_{A,1}, g_1) - \delta_{A,1}^*(z_0; F_{A,2}, g_2)). \quad (12)$$

Note that group labels for observations in $R_{A,1}$ and $R_{A,2}$ are determined by the sign of DDF measures, which remains invariant to variations

in the direction vector. However, assigning observations in overlap requires comparisons of DDF values, which are sensitive to the choice of the direction vector. Consequently, greater overlap amplifies the classifier's dependency on the direction vector, introducing potential instability.

3.4. PFC with soft frontiers

To mitigate the directional dependency and strengthen robustness, we propose to minimize overlap during training. After minimizing the overlap, pairwise soft frontiers are constructed to determine a unique group label for every observation.³

The overlap is minimized by implementing Algorithm 1 (see *infra*). It is designed to minimize the ambiguous region during training, without increasing the total misclassification cost. Specifically, the total misclassification cost is defined as the sum of controlled misclassified observations (n_{Miscla}) and ambiguous observations (n_{Ambig}).

Beginning with the initial n_{Miscla} and n_{Ambig} (Steps 1–3), the procedure enters a main optimization loop (Steps 5–25). The stopping condition for this loop in Step 5 states that an iteration is accepted if and only if the controlled misclassification reduces the ambiguous region without increasing the total misclassification cost.

Each iteration employs Model (13) to identify two sets: $E_l(\cdot)$ (observations controlled to be misclassified) and $C_l(\cdot)$ (observations to be contained in frontier construction), then updating the working set to S'_l (Steps 6–9).

$$\begin{aligned} \min_{s_j, l, p} \quad & c \cdot \sum_{z_j \in R_{A,F}(S_1)} s_1(z_j) + \sum_{z_j \in R_{A,F}(S_2)} s_2(z_j) \\ \text{s.t.} \quad & \delta_{A,1}^*(z_j; S_1, g_1) + s_1(z_j) \geq \delta_{A,2}^*(z_j; S_2, g_2) - s_1(z_j) + p \quad \forall z_j \in R_{A,F}(Z_1) \\ & \delta_{A,2}^*(z_j; S_2, g_2) + s_2(z_j) \geq \delta_{A,1}^*(z_j; S_1, g_1) - s_2(z_j) - p \quad \forall z_j \in R_{A,F}(Z_2) \\ & s_1(z_j), s_2(z_j) \geq 0, p \in P \end{aligned} \quad (13)$$

where P is defined as follows: (i) $P = [0, +\infty)$ if $R_{A,F}(Z_1) = \emptyset$; (ii) $P = (-\infty, 0]$ if $R_{A,F}(Z_2) = \emptyset$; (iii) $P = (-\infty, +\infty)$ otherwise. The weight c is used to mitigate the data imbalance by taking the ratio of group sizes ($|Z_2|/|Z_1|$).

Based on the optimized $s_l^*(z_j)$, $E_l(\cdot)$ and $C_l(\cdot)$ are obtained as follows:

$$E(R_{A,F}(Z_l)) = \{z_j \in R_{A,F}(Z_l) \mid s_l^*(z_j) > 0\}, \quad (14)$$

$$C(R_{A,F}(Z_l)) = \{z_j \in R_{A,F}(Z_l) \mid s_l^*(z_j) \leq 0\}. \quad (15)$$

Note that in the C case, if a training observation needs to be excluded, but its associated benchmark lies outside the overlap, then excluding that training observation alone will not achieve an inward shift of the frontier. To achieve this inward shift, both the training observations and their corresponding benchmarks must be excluded (Step 8).

Step 10 reconstructs the sets of ambiguous observations and misclassified observations using the updated set S'_l . If the gap region is non-empty (Step 11), then the algorithm proceeds to test whether including certain gap observation will not introduce additional overlap (Steps 12–19). For any gap observation meeting this condition, the set S'_l is subsequently updated to include it (Steps 20–22).

Step 24 calculates the n'_{Miscla} and n'_{Ambig} with the latest S'_l . Based on the new numbers, Step 5 checks if further iteration is necessary. The iteration stops if additional misclassifications no longer yield larger decrease of ambiguous observations (Step 25). This ensures that the algorithm converges to stable soft frontiers.

³ 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.

Algorithm 1 Generating the frontier sets of soft frontiers

Require: Z_1, Z_2, g_1, g_2
Ensure: $\bar{F}_{A,1}$ and $\bar{F}_{A,2}$

- 1: Initialize $S_1 = Z_1$ and $S_2 = Z_2$
- 2: Solve model (4) to obtain $\delta_{A,1}^*(z_0; S_1, g_1)$ and $\delta_{A,2}^*(z_0; S_2, g_2)$ for every $z_0 \in Z$, and construct the sets $R_{A,1}(Z_2)$, $R_{A,2}(Z_1)$, $R_{A,O}(Z)$ and $R_{A,G}(Z)$ using expressions (8)–(11)
- 3: Calculate $n_{Ambig} = |R_{A,O}(Z) \cup R_{A,G}(Z)|$, $n_{Miscla} = |R_{A,1}(Z_2) \cup R_{A,2}(Z_1)|$
- 4: Initialize $S'_1 = S_1$, $n'_{Ambig} = n_{Ambig} - 1$, $n'_{Miscla} = n_{Miscla}$
- 5: **while** $n'_{Miscla} + n'_{Ambig} \leq n_{Miscla} + n_{Ambig}$ and $n'_{Ambig} < n_{Ambig}$ **do**
- 6: Let $S'_1 = S'_1$, $n_{Ambig} = |R_{A,O}(Z) \cup R_{A,G}(Z)|$, $n_{Miscla} = |R_{A,1}(Z_2) \cup R_{A,2}(Z_1)|$
- 7: Solve model (13) for all $z_j \in R_{A,F}(Z) = R_{A,O}(Z) \cup R_{A,G}(Z)$ and construct the sets $E(R_{A,F}(Z_1))$ and $C(R_{A,F}(Z_1))$ using expressions (14)–(15)
- 8: Solve model (4) for every $z_0 \in E(R_{A,F}(Z_1)) \cap R_{A,O}(Z)$, and generate the benchmark set $B_l(z_0) = \{z_j \in S_l \mid \lambda_{j,l} > 0\}$, thus the aggregated benchmark set $B_l = \bigcup_{z_0 \in E(R_{A,F}(Z_1)) \cap R_{A,O}(Z)} B_l(z_0)$
- 9: Let $S'_1 = S_l \cup C(R_{A,F}(Z_1)) \setminus (E(R_{A,F}(Z_1)) \cup B_l)$ ($\forall l$)
- 10: Solve model (4) to obtain $\delta_{A,1}^*(z_0; S'_1, g_1)$ and $\delta_{A,2}^*(z_0; S'_2, g_2)$ for every $z_0 \in Z$, and construct the sets $R_{A,1}(Z_2)$, $R_{A,2}(Z_1)$, $R_{A,O}(Z)$ and $R_{A,G}(Z)$
- 11: **if** $R_{A,G}(Z) = \emptyset$ **then**
- 12: Initialize $H_1 = H_2 = \emptyset$
- 13: **for each** $l \in \{1, 2\}$ **do**
- 14: Let $l_i = l$, $l_o = \{1, 2\} \setminus \{l\}$
- 15: **for each** $z_j \in R_{A,G}(Z_{l_i})$ **do**
- 16: Solve model (4) to obtain $\delta_{A,l_i}^*(z_0; S'_{l_i} \cup \{z_j\}, g_{l_i})$ for every $z_0 \in R_{A,G}(Z_{l_i}) \cup S'_{l_i}$, and construct the set $R_{A,O}(R_{A,G}(Z_{l_i}) \cup S'_{l_i})$ using expression (10)
- 17: **if** $R_{A,O}(R_{A,G}(Z_{l_i}) \cup S'_{l_i}) = \emptyset$ **then** $H_{l_i} = H_{l_i} \cup \{z_j\}$ **end if**
- 18: **end for**
- 19: **end for**
- 20: **if** $H_1 \cup H_2 \neq \emptyset$ **then**
- 21: Update $S'_1 = S'_1 \cup H_1$ and $S'_2 = S'_2 \cup H_2$, $R_{A,G}(Z) = R_{A,G}(Z) \setminus (H_1 \cup H_2)$
- 22: **end if**
- 23: **end if**
- 24: Calculate $n'_{Ambig} = n(R_{A,O}(Z) \cup R_{A,G}(Z))$, $n'_{Miscla} = n(R_{A,1}(Z_2) \cup R_{A,2}(Z_1))$
- 25: **end while**
- 26: Obtain $\bar{F}_{A,l} = \{z_0 \in Z_l \mid \delta_{A,l}^*(z_0; S'_l, g_l) = 0\}$ ($\forall l$) following expression (7)

The frontier sets $\bar{F}_{A,l}$ ($\forall l$), describing the soft frontiers, are then obtained following expression (7). Note that the set S_l here in Step 26 is no longer the complete training dataset as used for constructing the hard frontiers.

The classification rule for PFC with soft frontiers maintains the same functional form as for hard frontiers, but with different frontier sets $\bar{F}_{A,1}$ and $\bar{F}_{A,2}$:

$$I(z_0) = 1 + \mathbb{I}_{(0,\infty)}(\delta_{A,1}^*(z_0; \bar{F}_{A,1}, g_1) - \delta_{A,2}^*(z_0; \bar{F}_{A,2}, g_2)), \quad (16)$$

where $\delta_{A,1}^*(z_0; \bar{F}_{A,1}, g_1)$ and $\delta_{A,2}^*(z_0; \bar{F}_{A,2}, g_2)$ are computed for an observation z_0 by solving model (4) with S_l configured as $\bar{F}_{A,l}$.

Compared to the hard frontiers, the clearly classified regions $R_{A,1}$ and $R_{A,2}$ expand while the overlap contracts, so more observations receive an unambiguous group label independent of the direction vector. Although the soft frontiers may leave some observations in the gap region, Algorithm 1 constrains this to a minimal level. Overall, the soft frontier approach yields a clearer classification for the majority of observations and enhances the PFC method by reducing the ambiguity associated with the direction vector.

4. Experimental analysis

4.1. Simulation studies

In this subsection, two types of simulated datasets are generated to test the proposed PFC methods. One type is described as C since its boundary is generally in a C shape. The other type is described as NC since its boundary follows a NC shape. For the simulated C datasets,

observations in Group 1 and Group 2 are generated from two distinct bivariate Normal distributions: (i) Group 1 follows $\mathcal{N}_{1,1}(\mu_{1,1}, \Sigma_{1,1})$ with $\mu_{1,1} = (2, -4)$ and $\Sigma_{1,1} = \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix}$; (ii) Group 2 follows $\mathcal{N}_{2,1}(\mu_{2,1}, \Sigma_{2,1})$ with $\mu_{2,1} = (-4, 2)$ and $\Sigma_{2,1} = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}$. For the simulated NC datasets, each group is generated from a mixture of two bivariate Normal distributions: (i) Group 1 is a mixture of $\mathcal{N}_{1',1}(\mu_{1',1}, \Sigma_{1',1})$ and $\mathcal{N}_{1',2}(\mu_{1',2}, \Sigma_{1',2})$, where $\mu_{1',1} = (1, 0)$, $\Sigma_{1',1} = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix}$, $\mu_{1',2} = (5, 4)$, and $\Sigma_{1',2} = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}$; (ii) Group 2 is a mixture of $\mathcal{N}_{2',1}(\mu_{2',1}, \Sigma_{2',1})$ and $\mathcal{N}_{2',2}(\mu_{2',2}, \Sigma_{2',2})$, where $\mu_{2',1} = (2, 6)$, $\Sigma_{2',1} = \begin{pmatrix} 2 & 1.5 \\ 1.5 & 5 \end{pmatrix}$, $\mu_{2',2} = (6, 10)$, and $\Sigma_{2',2} = \begin{pmatrix} 5 & 1.5 \\ 1.5 & 2 \end{pmatrix}$.

We considered four data scenarios: C-Clean, C-Noisy, NC-Clean, and NC-Noisy. For the noisy scenarios, labeling noise is introduced by randomly flipping 1% of the training labels. A graphical representation of the simulated C and NC datasets is depicted in Fig. 5.

For each scenario, the simulation is repeated 100 times. In each replication, a balanced training set of 2000 observations is generated, with observations evenly sampled from the underlying distributions. An independent test set of 2000 observations is sampled from the same distributions. Model performance is assessed by averaging results over all repetitions.

All four PFC methods are implemented using the proportional DDF.⁴ The monotonicity relation of attributes is derived by solving the MSD model in Appendix A. In this setup, the x-axis attribute is considered input-type, while the y-axis attribute is considered output-type.

The performance results under different data scenarios are summarized in Table 1. The table is organized as follows: the three columns under “Accuracy” report classification accuracy over the entire test set and within each group, respectively; columns 6–7 present the percentage of overlap and gap; the last two columns provide the p -values from the Wilcoxon signed-rank test, which verifies whether the performance differences between the corresponding comparisons are statistically significant (Demšar, 2006). As the results from the training and test sets differ very little, only the test set results are presented here. Comprehensive results under different scenarios are provided in Appendix C.

Four observations can be made from Table 1. First, consistent with Propositions 3.2 and 3.3, C models exhibit larger overlaps and slightly smaller gaps than the NC models across all scenarios. Specifically, on C datasets, C models have on average a 4.58% larger overlap and a 0.03% smaller gap. On NC datasets, the difference is much more pronounced, with a 26.75% larger overlap and a 0.12% smaller gap. In every scenario, the overlap is substantially larger than the gap. Therefore, C models produce a larger ambiguous region than the NC models, indicating a higher level of inherent predictive uncertainty.

Second, models with soft frontiers successfully reduce overlap compared to those with hard frontiers. Although this reduction comes with a slight increase in the gap, the overall ambiguity is still reduced, with reduction rates ranging from 85.00% (for the NC model under the NC-Clean scenario) to 99.00% (for the C model under the C-Noisy scenario). Moreover, this ambiguity reduction is attained at a relatively low cost, manifested through controlled misclassifications during training. Specifically, under the C-Clean (NC-Clean) scenario, an average ambiguity reduction of 14.21% (29.48%) is achieved by misclassifying only 1.58% (3.79%) of the training observations. Under the C-Noisy (NC-Noisy) scenario, an average ambiguity reduction of 76.67% (64.95%) is achieved by misclassifying just 1.67% (3.86%) of the clean training observations.

Third, on C datasets, the CHard model has a modest performance advantage of 0.31% on average over the NCHard model. However, this advantage diminishes to just 0.06% and becomes less significant

⁴ While radial measures are more common in the DEA literature, they implicitly require that the unmeasured attributes remain fixed and invariant. However, most attributes in classification are variable. Therefore, we argue that proportional DDF is a more reasonable default choice in the absence of prior information about the attributes.

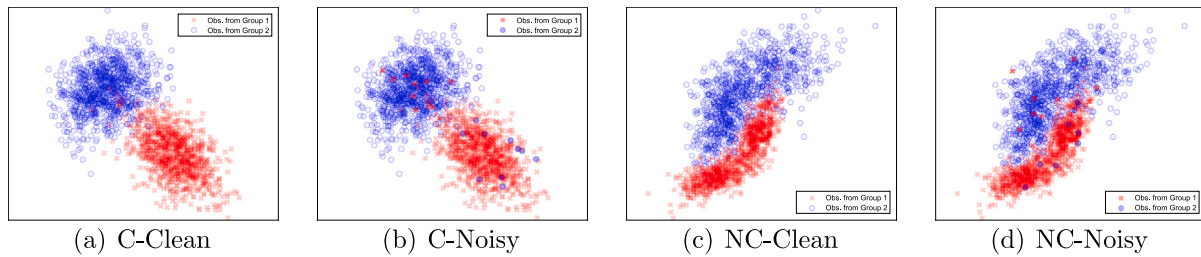


Fig. 5. Illustration of the simulated datasets.

Table 1
Performance results under different data scenarios.

Datasets	PFC-	Accuracy			Ambiguity		Wilcoxon signed-rank test	
		Overall	Group 1	Group 2	Overlap	Gap	CHard	NCSOft
C-Clean	CHard	0.9763	0.9802	0.9724	0.1594	0.0008		
	NCHard	0.9727	0.9836	0.9617	0.1213	0.0014	0.0000***	0.0000***
	CSOft	0.9789	0.9732	0.9845	0.0014	0.0066	0.0000***	0.0011**
	NCSOft	0.9782	0.9779	0.9786	0	0.0116		
C-Noisy	CHard	0.9653	0.9417	0.9890	0.7924	0.0000		
	NCHard	0.9628	0.9354	0.9903	0.7389	0.0000	0.0000***	0.0000***
	CSOft	0.9783	0.9710	0.9856	0.0011	0.0068	0.0000***	0.0300*
	NCSOft	0.9778	0.9751	0.9805	0.0000	0.0110		
NC-Clean	CHard	0.9328	0.8980	0.9675	0.4300	0.0013		
	NCHard	0.9375	0.9090	0.9659	0.1599	0.0033	0.0037**	0.0000***
	CSOft	0.9434	0.9385	0.9482	0.0174	0.0068	0.0002***	0.0000***
	NCSOft	0.9586	0.9678	0.9495	0	0.0245		
NC-Noisy	CHard	0.9221	0.9481	0.8962	0.7929	0.0006		
	NCHard	0.8893	0.9064	0.8721	0.5283	0.0011	0.0000***	0.0000***
	CSOft	0.9444	0.9391	0.9497	0.0236	0.0086	0.0000***	0.0000***
	NCSOft	0.9520	0.9555	0.9485	0	0.0258		

* Denotes significance at 0.05.

** Denotes significance at 0.01.

*** Denotes significance at 0.001.

when comparing to their soft frontier variants. On NC datasets, the NC models perform significantly better than the C models, except with the NC-Noisy dataset. In that scenario, the PFC-CHard outperforms the PFC-NCHard model. This can be attributed to its larger ambiguous region, which enable more flexible handling of ambiguous observations, thereby diminishing the influence of noisy labels.

Fourth, models with soft frontiers achieve consistently higher overall accuracy than those with hard frontiers. With an average improvement of 0.98% on clean datasets and 2.86% on noisy datasets, the models with soft frontiers demonstrates particularly strong performance in the presence of labeling noise. Additionally, while models with hard frontiers exhibit a substantial performance gap between Group 1 and Group 2 under noisy conditions, those with soft frontiers deliver more balanced predictive accuracy across groups in such scenarios.

In summary, the simulation results demonstrate that the relative performance superiority of C and NC models depends on the underlying datasets (C or NC). However, PFC models with soft frontiers consistently outperform those with hard frontiers, particularly in the presence of labeling noise. By significantly reducing the ambiguous region, PFC models with soft frontiers achieve higher overall accuracy, more balanced performance across groups, and greater resilience to mislabeled samples.

4.2. Experiments on a real-life dataset

Compared to simulated datasets, real-world datasets tend to be more complex and noisy. In this subsection, an experimental study on a real life dataset 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

A credit scoring dataset provided by Yeh and Lien (2009) is used. This dataset 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.⁵

The dataset 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 dataset contains 16072 observations. The cleaned dataset contains 5492 defaulters and 10580 non-defaulters.

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 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

⁵ The dataset 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>.

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 $BILL_AMT(t)$. A negative overdue amount is replaced by 0.
x_3	September credit utilization ratio: $BILL_AMT1/LIMIT_BAL$

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.

All four PFC methods are implemented using the proportional DDF. The PFC methods are compared to 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, basic DEA-DA (Sueyoshi, 1999), extended DEA-DA (Sueyoshi, 2001), and MIP DEA-DA (Sueyoshi, 2004). The DEA-DA method is essentially based on the use of goal programming. Banker et al. (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 DEA-DA methods and their further developments in the methodological review *supra*.⁶ The other category consists of 8 non-DEA methods, namely, MSD, two-stage MSD, linear DA, quadratic DA, logistic regression, decision tree, Gaussian SVM, and K-Nearest Neighbor (KNN).⁷ Since the proposed PFC framework acts as a base classifier, the main text focuses exclusively on the above base classifiers to ensure a fair baseline comparison. For completeness sake, supplementary comparisons with several widely known ensemble methods (including Random Forest, XGBoost and LightGBM) are provided in Appendix D.

The classification performance of the various methods is evaluated by the F_2 score, G-mean, recall, specificity, and their differences. Among these, recall and specificity reflect the performance on defaulters and non-defaulters, respectively. 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 et al., 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. Note that for imbalanced datasets, accuracy is no longer appropriate to quantify overall performance as it is biased towards the majority group (He & Garcia, 2009).

⁶ In the literature, DEA-DA has been methodologically refined in a variety of ways (see Boudaghi and Farzipoor Saen (2018), Lotfi and Mansouri (2008) for details on successive developments). It has also been used quite widely in empirical analysis (see Toloo et al. (2015), Tsai et al. (2009)).

⁷ Of the 8 non-DEA methods, the code for the last six are exported from the Matlab Classification Learner app. All other methods are coded in Matlab by the authors: these codes are available upon simple request. For the DEA-DA methods and two MSD models, a small positive constant is required to prevent trivial zero coefficients and ensure a clear separation. This parameter is uniformly set to 0.01 in our experiments.

4.2.2. Performance results on the full dataset

To respect the imbalanced nature of the problem, a stratified 10-fold cross-validation (CV) is conducted. The performance results are presented in Table 3. Columns 3–7 correspond to five performance measures. In each column, the best result within a specific block is highlighted in bold.

Table 3 shows that the PFC-NCSOFT method outperforms the other PFC variants in both overall performance and balance across groups. Specifically, switching from C to NC models yields an overall performance improvement ranging from 3.43% to 16.89%. This improvement is much more substantial when shifting from hard to soft frontiers, with performance gains between 13.20% and 33.15%. Moreover, the between-group performance difference decreases sharply from 58.84% to merely 0.63%, highlighting the exceptional predictive balance achieved by the PFC-NCSOFT model.

Compared with non-PFC alternatives, PFC-NCSOFT remains highly competitive. Several alternatives achieve comparable or superior results on individual measures, but they do so by trading performance between the two groups. Specifically, the MSD method shows a better performance on identifying defaulters, resulting in higher F_2 scores. However, this comes with poorer performance in predicting non-defaulters, and is eventually reflected in its lower G-mean values. By contrast, the SVM Gaussian demonstrates the best performance in recognizing non-defaulters and attains a superior G-mean. Yet, this strength is offset by its relatively low recall of 75.47%, thereby yielding a lower F_2 score. Thus, none of the listed classifiers outperform PFC-NCSOFT on both overall performance measures.

To verify whether the observed performance differences are statistically significant, a Friedman test followed by Nemenyi post-hoc comparisons is conducted (Demšar, 2006). In addition to the experiments summarized in Table 3, the 10-fold CV is repeated four more times to enhance the robustness of the evaluation. Note that the DEA-DA MIP method is excluded from these additional runs because it shows no superiority on any measures but incurring substantial computation time.

The Friedman test reveals significant differences among the 14 classifiers for both the F_2 score and G-mean (see Appendix E for detailed Friedman statistics). Then, the Nemenyi post-hoc comparisons are performed with a critical difference (CD) of 3.0478 ($\alpha = 0.01$). The resulting CD diagrams are shown in Fig. 6. Each CD diagram shows classifiers ordered by average rank on the vertical axis and average rank values on the horizontal axis (lower = better). A vertical dashed line marks the right-hand end of the PFC-NCSOFT CD bar to indicate the classification models over which it significantly dominates.

The CD diagrams in Fig. 6 show that the PFC-NCSOFT method significantly outperforms all other PFC variants across both overall performance measures. Compared with non-PFC alternatives, the PFC-NCSOFT method consistently appears in the leading group across both overall performance measures. Regarding the F_2 score, the MSD model has better average rank than PFC-NCSOFT. However, this difference is not statistically significant. Similarly, for the G-mean value, PFC-NCSOFT

Table 3
Stratified 10-fold CV performance on the full cleaned dataset.

Classification methods		Overall		Group-wise		
		F_2 Score	G-Mean	Recall	Specificity	Difference
PFC-	CHard	0.4167	0.5963	0.3709	0.9593	0.5884
	NCHard	0.5856	0.7157	0.5468	0.9372	0.3904
	CSoft	0.7482	0.8134	0.7396	0.8950	0.1554
	NCSoft	0.8260	0.8477	0.8509	0.8446	0.0063
	Basic ^a	0.2306	0.3707	0.2311	0.5955	0.3644
DEA-DA	Extend	0.7909	0.8388	0.7921	0.8886	0.0965
	MIP	0.7462	0.7745	0.7818	0.7765	0.0054
	MSD	0.8417	0.8280	0.9037	0.7589	0.1448
	Two-Stage MSD	0.7913	0.8394	0.7920	0.8899	0.0978
	Linear DA	0.7190	0.8087	0.6877	0.9512	0.2635
Others	Quadratic DA	0.7643	0.8252	0.7565	0.9005	0.1439
	Logistic Regression	0.7653	0.8298	0.7524	0.9155	0.1631
	Decision Tree	0.7776	0.8466	0.7544	0.9503	0.1959
	SVM Gaussian	0.7789	0.8482	0.7547	0.9535	0.1988
	KNN	0.7759	0.8280	0.7760	0.8836	0.1076

^a The weak performance of the DEA-DA Basic model arises from its constraint of non-negative weights, which inherently requires a positive correlation between attributes and the discriminant score. The current dataset does not satisfy this requirement. Preprocessing the data would improve performance.

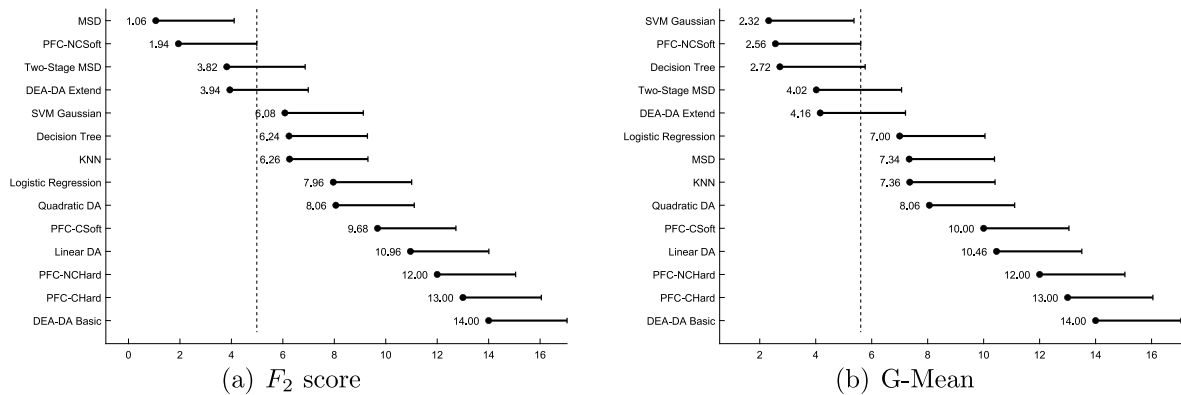


Fig. 6. Critical difference diagrams.

is slightly outperformed in average rank only by SVM Gaussian, but without statistical significance.

Taken together, these results show that PFC-NCSoft does not necessarily achieve the best performance on every individual measure, but it remains highly competitive across all alternatives without significant weaknesses. More importantly, it achieves a balanced performance in predicting defaulters and non-defaulters.

4.2.3. Performance results for varying default rates

In addition to the primary experiment on the full cleaned dataset, we conduct a series of further tests to investigate the robustness of the models against data imbalance. These experiments are conducted using datasets of the same sample size, but with varying default rates (inspired by Chen et al. (2024)).

Each time, a test set consisting of 492 defaulters and 492 non-defaulters, is randomly generated. To isolate the effect of class imbalance from that of sample size, we fix the training set size at 10000. This is constructed by including all remaining 5000 defaulters and randomly selecting 5000 non-defaulters, resulting in a balanced baseline. We then create 12 training sets of size 10000 with different default rates u ($u = 1\%, 2.5\%, 5\%, 10\%, \dots, 45\%, 50\%$). It is repeated 20 times with different random seeds to obtain robust performance estimates.

The performance results of G-mean across varying default rates are summarized in the heat map in Fig. 7. Each cell reports the G-mean of a classifier at a given default rate. Rows are ordered by average G-mean, with the colors in each cell representing performance relative to the PFC-NCSoft model. White cells correspond to the baseline model, the PFC-NCSoft model. Blue cells indicate poorer performance

relative to the PFC-NCSoft model, while orange cells indicate superior performance. Due to similar trends observed under the F_2 score, the corresponding heat map is provided in Appendix F.

The PFC-NCSoft model, serving as the baseline model, achieves the highest average G-mean of 0.8417 and is therefore placed in the top row. The predominance of blue cells throughout the heat map indicates that most classifiers underperform relative to PFC-NCSoft. Only Decision Tree and SVM exhibit orange cells when the default rate exceeds 40%. Note that their marginally higher G-means does not exceed 1.11%.

In particular, the PFC-NCSoft model significantly outperforms all other classifiers under severe imbalance. As the default rate decreases towards extreme imbalance, most models experience a sharp decline in performance. At the most extreme case of 1%, DEA-DA Extend, MSD and Two-stage MSD fail completely to identify any defaulters. Even the best-performing alternative, Quadratic DA, attains a G-mean of only 0.7351. However, PFC-NCSoft model achieves a substantially higher G-mean of 0.8141.

To further illustrate the stability of each model, the box plot in Fig. 8 shows the distribution of G-mean values across different default rates. Models are ordered by average G-mean. A compact box reflects consistent performance, whereas a wide box or long whiskers indicate high sensitivity to the default rate.

In Fig. 8, the PFC-NCSoft is located at the top-right corner with the smallest interquartile range and shortest whiskers. That confirms that it leads in central tendency, with both high median and average G-Mean values. Moreover, its performance across different default rates is the



Fig. 7. Heat map of G-mean under varying default rates.

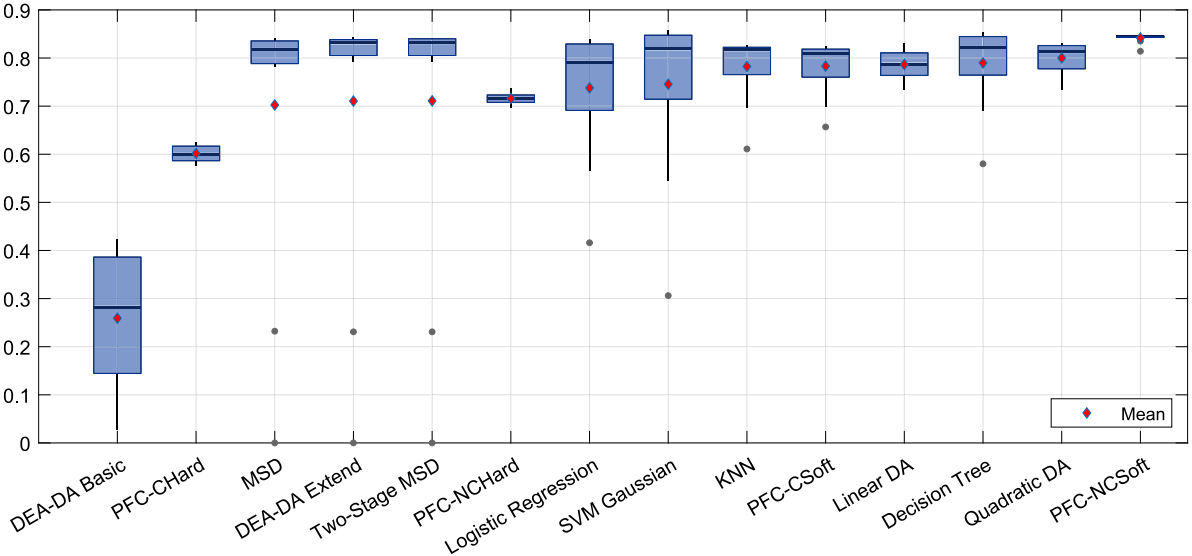


Fig. 8. Box-plots of G-mean across the 12 default rates.

most stable. By contrast, although SVM and Decision Tree occasionally outperform PFC-NCSOft at higher default rates, they show much greater fluctuation, indicating less reliability in real-world applications.

Building on the insights from the heat map and box plot, Fig. 9 further investigates the performance of the classifiers, specifically focusing on recall and specificity. These line plots reveal performance trends for only a subset of classifiers that outperform PFC-NCSOft in at least one overall performance measure.

From the line plots, it is evident that due to the limited number of defaulters, all models exhibit a classification bias towards correctly identifying non-defaulters, resulting in high specificity but low recall. As the imbalance decreases, recall generally increases, while specificity decreases. Specifically, the recall of other models fluctuates between 0.4823 and 0.9366, while their specificity decreases between 0.1095 to 0.3484. In contrast, PFC-NCSOft shows much smaller fluctuations, with recall varying by only 0.1314 and specificity by merely 0.0818.

When further analyzing the performance difference between recall and specificity, it is clear that competing models demonstrate a considerable gap, with the average gap ranging from 0.2543 to 0.3953. However, PFC-NCSOft maintains a consistently smaller gap across all default

rates, with an average gap of just 0.0479. This relatively small gap of PFC-NCSOft accounts for its robust and well-balanced performance under varying default rates.

Overall, the PFC-NCSOft model demonstrates superior average performance and greater stability across the tested default rates. Under severe data imbalance, the PFC-NCSOft model greatly outperforms the compared classifiers, exhibiting smaller fluctuations in recall and specificity. This makes it a reliable choice for such imbalanced classification tasks.

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

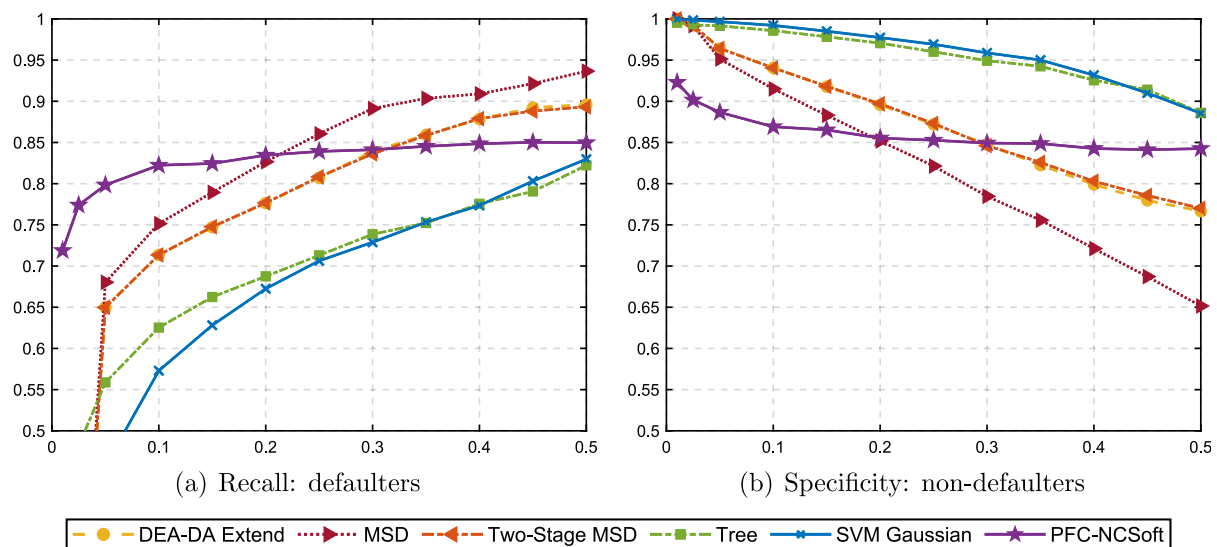


Fig. 9. Performance comparison under different default rates.

this sense, the PFC method should be a good candidate in estimating the separating boundary whose shape is mostly unspecified in applications.

In this contribution, an interpretable and flexible framework for PFC methods is established. The role of axioms in classification is clarified by linking the free disposability and convexity axioms to monotonicity and attribute substitutability, respectively. Building on this, PFC methods with NC frontiers are constructed when no substitution relation is specified. The NC frontiers offer tighter group envelopment and reduce overlap compared to C frontiers. In cases where overlap persists, PFC methods with soft frontiers are proposed. The soft frontiers are obtained from an optimization algorithm that minimize overlap through controlled misclassification. It offers a novel solution to handle overlap which avoids excessive misclassification while maintaining the frontier-based framework.

Both simulation studies and experimental analysis show that the overall performance of the PFC method improves from C to NC and from hard to soft frontiers. Experimental analysis of credit card data shows that PFC-NCSOft is quite competitive in terms of overall performance compared to some existing classification methods. Specifically, it demonstrates superior average performance and greater stability when the level of class imbalance incrementally increases. This makes it a reliable choice for imbalanced classification tasks.

There remain some directions for future investigation. First, it is worth exploring whether other axioms common in production analysis (e.g., the axiom of returns-to-scale) can correspond similarly to some relation in the classification problem (as convexity and monotonicity do). Second, although the monotonicity relation contributes to explain the classification results, it is not always satisfied in applications. For situations where the monotonicity relation does not hold, one may wonder whether it is possible to weaken the axiom of disposability. In production analysis, one recent attempt to relax the free disposability axiom is developed in Brieu et al. (2016) and empirically implemented in Brieu et al. (2018). Finally, one may wonder to which extent more robust versions of the PFC methods (e.g., order- m or order- α partial instead of full frontiers) would improve the classification performance.

CRedit authorship contribution statement

Qianying Jin: Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Conceptualization. **Kristiaan Kerstens:** Writing – review & editing, Validation, Supervision, Conceptualization. **Ignace Van de Woestyne:** Writing – review &

editing, Validation, Methodology. **Zhongbao Zhou:** Writing – review & editing, Validation, Supervision, Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Qianying Jin reports financial support was provided by National Natural Science Foundation of China. The other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

Acknowledgments

We sincerely appreciate the constructive comments of three anonymous referees who helped to improve this contribution. The usual disclaimer applies. Q. Jin appreciates financial support of the National Natural Science Foundation of China (No. 72201127).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.ejor.2026.05.009>.

References

- Banker, R., Chang, H., & Cooper, W. (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., & Farzipoor Saen, R. (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.
- Brieu, W., Kerstens, K., & Van de Woestyne, I. (2016). Congestion in production correspondences. *Journal of Economics*, 119(1), 65–90.
- Brieu, W., Kerstens, K., & Van de Woestyne, I. (2018). Hypercongestion in production correspondences: An empirical exploration. *Applied Economics*, 50(27), 2938–2956.
- Chambers, R., Chung, Y., & Färe, R. (1998). Profit, directional distance functions, and Nerlovian efficiency. *Journal of Optimization Theory and Applications*, 98(2), 351–364.
- Chang, D. S., & Kuo, Y. C. (2005). A novel procedure to identify the minimized overlap boundary of two groups by DEA model. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganà, H. P. Lee, Y. Mun, D. Taniar, & C. J. K. Tan (Eds.), *Lecture notes in computer science*: vol. 3483, (pp. 577–586). Heidelberg: Springer, ISBN: 978-3-540-25863-6.

- Chang, D. S., & Kuo, Y. C. (2008). An approach for the two-group discriminant analysis: An application of DEA. *Mathematical and Computer Modelling*, 47(9–10), 970–981.
- Chen, Y., Calabrese, R., & Martin-Barragan, B. (2024). Interpretable machine learning for imbalanced credit scoring datasets. *European Journal of Operational Research*, 312(1), 357–372.
- Cherchye, L., Kuosmanen, T., & Post, T. (2001). FDH directional distance functions with an application to European commercial banks. *Journal of Productivity Analysis*, 15(3), 201–215.
- Christen, P., Hand, D. J., & Kirielle, N. (2023). A review of the F-measure: Its history, properties, criticism, and alternatives. *ACM Computing Surveys*, 56(3), 1–24.
- De Bock, K. W., Coussement, K., & Lessmann, S. (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., Coussement, K., De Bock, K. W., & Lessmann, S. (2020). Incorporating textual information in customer churn prediction models based on a convolutional neural network. *International Journal of Forecasting*, 36(4), 1563–1578.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan), 1–30.
- Depriens, D., Simar, L., & Tulkens, H. (1984). Measuring labor efficiency in post offices. In M. Marchand, P. Pestieau, & H. Tulkens (Eds.), *The performance of public enterprises: concepts and measurements* (pp. 243–268). Amsterdam: North Holland.
- Emrouznejad, A., Banker, R. D., & Neralić, L. (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., Löw, L., & Spindler, M. (2022). An explainable attention network for fraud detection in claims management. *Journal of Econometrics*, 228(2), 244–258.
- Freed, N., & Glover, F. (1981). Simple but powerful goal programming models for discriminant problems. *European Journal of Operational Research*, 7(1), 44–60.
- He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.
- Jin, Q., Kerstens, K., & Van de Woestyne, I. (2024). Convex and nonconvex nonparametric frontier-based classification methods for anomaly detection. *OR Spectrum*, 46(4), 1213–1239.
- Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (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., & Palacios, F. (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., Baesens, B., Seow, H.-V., & Thomas, L. C. (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., & Mansouri, B. (2008). The extended data envelopment analysis/discriminant analysis approach of fuzzy models. *Applied Mathematical Sciences*, 2(30), 1465–1477.
- Merdan, S., Barnett, C. L., Denton, B. T., Montie, J. E., & Miller, D. C. (2021). Data analytics for optimal detection of metastatic prostate cancer. *Operations Research*, 69(3), 774–794.
- Ouenniche, J., Carrales, S., Tone, K., & Fukuyama, H. (2017). An account of DEA-based contributions in the banking sector. In K. Tone (Ed.), *Advances in DEA theory and applications: with extensions to forecasting models* (pp. 141–171). John Wiley & Sons, Ltd.
- 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. (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., Khosrowpour, M., & Rodger, J. (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., Rodger, J., & Yaverbaum, G. (1999). Association, statistical, mathematical and neural approaches for mining breast cancer patterns. *Expert Systems with Applications*, 17(3), 223–232.
- Pendharkar, P. C., & Troutt, M. D. (2014). Interactive classification using data envelopment analysis. *Annals of Operations Research*, 214(1), 125–141.
- Seiford, L., & Zhu, J. (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.
- Sueyoshi, T. (2001). Extended DEA-discriminant analysis. *European Journal of Operational Research*, 131(2), 324–351.
- Sueyoshi, T. (2004). Mixed integer programming approach of extended DEA-discriminant analysis. *European Journal of Operational Research*, 152(1), 45–55.
- Toloo, M., Farzipoor Saen, R., & Azadi, M. (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., Rai, A., & Zhang, A. (1996). The potential use of DEA for credit applicant acceptance systems. *Computers & Operations Research*, 23(4), 405–408.
- Tsai, M.-C., Lin, S.-P., Cheng, C.-C., & Lin, Y.-P. (2009). The consumer loan default predicting model—an application of DEA-DA and neural network. *Expert Systems with Applications*, 36(9), 11682–11690.
- Walheer, B. (2018). Cost malmquist productivity index: An output-specific approach for group comparison. *Journal of Productivity Analysis*, 49(1), 79–94.
- Yan, H., & Wei, Q. (2011). Data envelopment analysis classification machine. *Information Sciences*, 181(22), 5029–5041.
- Yeh, I., & Lien, C. (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.