Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 346 (2019) 321–331

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

An Integer Programming Approach for the 2-class Single-group Classification Problem

Ricardo C. Corrêa[1]

Universidade Federal Rural do Rio de Janeiro
Departamento de Ciência da Computação
Av. Governador Roberto Silveira S/N, 26020-740 Nova Iguaçu - RJ, Brazil

Manuela Blaum, Javier Marenco [2]

Universidad Nacional de General Sarmiento, Instituto de Ciencias
J. M. Gutiérrez 1150, Malvinas Argentinas, (1613) Buenos Aires, Argentina

Ivo Koch [3]

Universidad Nacional de General Sarmiento, Instituto de Industria
J. M. Gutiérrez 1150, Malvinas Argentinas, (1613) Buenos Aires, Argentina

Marcelo Mydlarz [4]

Universidad Nacional de General Sarmiento, Instituto de Industria
J. M. Gutiérrez 1150, Malvinas Argentinas, (1613) Buenos Aires, Argentina
CONICET, Argentina

Abstract

Two sets $\mathcal{X}_B, \mathcal{X}_R \subseteq \mathbb{R}^d$ are *linearly separable* if their convex hulls are disjoint, implying that a hyperplane separating $\mathcal{X}_B$ from $\mathcal{X}_R$ exists. Such a hyperplane provides a method for classifying new points, according to the side of the hyperplane in which the new points lie. In this work we consider a particular case of the 2-class classification problem, which asks to select the maximum number of points from $\mathcal{X}_B$ and $\mathcal{X}_R$ in such a way that the selected points are linearly separable. We present an integer programming formulation for this problem, explore valid inequalities for the associated polytope, and develop a cutting plane approach coupled with a lazy-constraints scheme.

Keywords: classification, integer programming, polyhedral combinatorics

[1] Email: correa@ufrrj.br

[2] Email: mblaum@ungs.edu.ar, jmarenco@ungs.edu.ar

[3] Email: ikoch@ungs.edu.ar

[4] Email: mmydlarz@ungs.edu.ar

https://doi.org/10.1016/j.entcs.2019.08.029

1 Introduction

Classification problems with supervised learning involve separating training samples into categories, in such a way that future samples can be automatically categorized based on the categorization of the training samples. In this work, we are interested in a particular case of the 2-class classification problem, an optimization problem that arises within this context.

We are given a set $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subseteq \mathbb{R}^d$ of *samples* (also referred to as *points*) and a partition of the index set $[m] := \{1, \dots, m\}$ into two *classes* B and R, defining subsets $\mathcal{X}_B = \{\mathbf{x}_i : i \in B\}$ and $\mathcal{X}_R = \{\mathbf{x}_j : j \in R\}$. The sets of points $\mathcal{X}_B$ and $\mathcal{X}_R$ are said to be *linearly separable* if and only if $\mathrm{conv}(\mathcal{X}_B) \cap \mathrm{conv}(\mathcal{X}_R) = \emptyset$, where $\mathrm{conv}(X)$ denotes the convex hull of X. The linear separability of $\mathcal{X}_B$ and $\mathcal{X}_R$ is characterized by the fact that no vector $(\lambda_1, \dots, \lambda_m) \geq \mathbf{0}$ satisfies

$$\sum_{i \in B} \lambda_i \mathbf{x}_i = \sum_{j \in R} \lambda_j \mathbf{x}_j, \qquad \text{and} \qquad \sum_{i \in B} \lambda_i = \sum_{j \in R} \lambda_j = 1.$$

The objective of the *2-class single-group classification* problem is to determine subsets of $\mathcal{X}_B$ and $\mathcal{X}_R$ that are linearly separable, maximizing the sum of their cardinalities. Formally, we want to find $\mathcal{X}_S \subseteq \mathcal{X}_B$ and $\mathcal{X}_T \subseteq X_R$ maximizing $|\mathcal{X}_S| + |\mathcal{X}_T|$, and such that $\mathcal{X}_S$ and $\mathcal{X}_T$ are linearly separable. The points in $\mathcal{X}_O := \mathcal{X} \backslash (\mathcal{X}_S \cup \mathcal{X}_T)$ are called *outliers*, and the objective function of the problem implies that $|\mathcal{X}_O|$ is to be minimized.

A wide range of continuous optimization methods for this problem, including linear and quadratic programming, have been developed in the last years [4,6,8]. More recently, integer linear programming tools started to be used in conjunction with continuous methods [7,9,10,11]. The 2-class single-group classification problem is a particular case of the *2-class classification* problem introduced in [3]. In this more general setting, besides the set of points $\mathcal{X}$ partitioned into classes $\mathcal{X}_B$ and $\mathcal{X}_R$, we are given numbers $n_B, n_R \in \mathbb{Z}_+$, and the goal is to find a set $\mathcal{X}_O$ as small as possible of outliers so that $\mathcal{X}_B \setminus \mathcal{X}_O$ and $\mathcal{X}_R \setminus \mathcal{X}_O$ can be partitioned into n_B and n_R subsets, respectively, and each of the n_B subsets of $\mathcal{X}_B \setminus \mathcal{X}_O$ is linearly separable from each of the n_R subsets of $\mathcal{X}_R \setminus \mathcal{X}_O$.

Corrêa, Delle Donne, and Marenco [5] consider a mixed integer programming formulation for the 2-class classification problem, and explore facet-inducing inequalities for the associated polytope. Here we present a pure integer programming formulation (i.e., containing binary variables only) for the 2-class single-group classification problem, extending and strengthening the results presented in [5] to this particular setting. Our goal is to study the combinatorics associated with the problem, with the objective of designing efficient integer-programming based computational procedures for this problem. Besides the theoretical interest, such a study is of practical relevance since the obtained computational methods could be used within general classification tools. In order to give a first evaluation of this potential, we also present preliminary computational results for this formulation.

The remainder of this paper is organized as follows. Section 2 presents the integer programming formulation for the problem and a general family of valid inequalities for the associated polytope. Section 3 shows the lazy-constraints scheme used to tackle the exponential number of constraints in the model. Section 4 introduces a separation procedure for these inequalities, and Section 5 reports our computational experience. Finally, Section 6 provides final remarks.

2 Integer programming formulation and valid inequalities

The mixed integer programming formulation presented in [5] includes binary variables representing the assignment of points to groups, and continuous variables representing the hyperplanes that separate each pair of groups from different classes. In this work, we consider a formulation that only resorts to variables of the first kind, that is, binary variables representing the assignment of points to groups. Since we consider a single group for each class of points, these binary variables determine the selected points (i.e., points that are not declared to be outliers). The formulation can be extended to the multi-group case in a straightforward way.

For $i \in [m]$, we introduce the binary variable z_i representing whether $\mathbf{x}_i$ is chosen ($z_i = 1$) or not ($z_i = 0$). The 2-class single-group classification problem can be modeled as the problem of maximizing $\sum_{i\in B} z_i + \sum_{j\in R} z_j$ subject to the constraint that the sets $\{\mathbf{x}_i : z_i = 1, i \in B\}$ and $\{\mathbf{x}_j : z_j = 1, j \in R\}$ are linearly separable. Given an instance $\mathcal{I} = (\mathcal{X}, B, R)$ of the problem, we call $P_{\mathcal{I}}$ the convex hull of the points $z \in \{0,1\}^m$ satisfying the linear separability constraints. These constraints imply that any feasible solution z satisfies the *S,T-inequality*

$$\sum_{i\in S} z_i + \sum_{j\in T} z_j \;\leq\; |S| + |T| - 1 \tag{1}$$

for every $S \subseteq B$ and every $T \subseteq R$ such that $\mathcal{X}_S$ and $\mathcal{X}_T$ are not linearly separable. These inequalities restrict the selected points from each class to indeed correspond to a feasible solution, and –together with the integrality constraints– can be used as constraints defining $P_{\mathcal{I}}$. As we will show below, the inequality (1) is facet-inducing when $\mathcal{X}_S$ and $\mathcal{X}_T$ are minimal with respect to being linearly inseparable (i.e., when $\mathcal{X}_S \setminus \{\mathbf{x}_i\}$ and $\mathcal{X}_T \setminus \{\mathbf{x}_i\}$ are linearly separable for every $i \in S \cup T$).

Given two non-empty sets $S \subseteq B$ and $T \subseteq R$ such that $\mathcal{X}_S$ and $\mathcal{X}_T$ are not lineary separable, any minimum cardinality set $N \subset S \cup T$ such that $\mathcal{X}_S \setminus \mathcal{X}_N$ and $\mathcal{X}_T \setminus \mathcal{X}_N$ are linearly separable, plays a key role. We call N an *$\mathcal{N}$-set of* S, T and define $\mathcal{N}(S,T) := \{N \subseteq S \cup T : N \text{ is an } \mathcal{N}\text{-set of } S,T\}$, and, for $i \in S \cup T$,

$$\nu_{S,T}(i) := \min\{|N| : i \in N \text{ and } N \in \mathcal{N}(S,T)\}, \text{ or } \infty \text{ if no } N \in \mathcal{N}(S,T) \text{ contains } i.$$

We denote $\nu_{S,T}(i)$ by ν_i when S and T are clear from the context. This setting gives rise to what we call the *$\mathcal{N}$-inequality* associated with S and T:

$$\sum_{i\in S} \frac{1 - z_i}{\nu_i} + \sum_{j\in T} \frac{1 - z_j}{\nu_j} \geq 1, \tag{2}$$

where the coefficient of $(1 - z_i)$ is null if $\nu_i = \infty$. This inequality generalizes facet-inducing inequalities explored in [5], corresponding to the cases where $\nu_i = 1$ for every $i \in S \cup T$. The following result, adapted from [5], illustrates this situation.

Theorem 2.1 ([5]) *The inequality* (2) *defines a facet of* $P_{\mathcal{I}}$ *if* ($|T| = 1$ *or* ($|T| = 2$ *and* $\mathbf{x}_T \cap \mathrm{conv}(\mathbf{x}_S) = \emptyset$)) *and* S *is minimal with respect to the property* $\mathrm{conv}(\mathbf{x}_T) \cap \mathrm{conv}(\mathbf{x}_S) \neq \emptyset$ *(i.e.,* $\mathrm{conv}(\mathbf{x}_T) \cap \mathrm{conv}(\mathbf{x}_{S'}) = \emptyset$ *for every* $S' \subsetneq S$*).*

A more general result is given by the following theorem, also settling a question raised in [5]. To state this result, let $G_{S,T} = (S \cup T, E)$ denote the *safe graph of* S, T in which an edge ij exists if there exist two $\mathcal{N}$-sets N_i and N_j of equal size ν such that $N_i \triangle N_j = \{i, j\}$ and $\nu_k = \nu$, for all $k \in N_i \cup N_j$. Observe that $\nu_i = \nu_j$ for each edge ij of $G_{S,T}$, and by transitivity, $\nu_{i'} = \nu_{j'}$ for each pair of vertices i' and j' connected in $G_{S,T}$. In other words, $\nu_i = \nu_j$ if i and j are two vertices in a same connected component of $G_{S,T}$. Let $G^{\nu}_{S,T} = (V^{\nu}_{S,T}, E^{\nu}_{S,T})$ denote the subgraph of $G_{S,T}$ induced by $V^{\nu}_{S,T} = \{i \in S \cup T \mid \nu_i = \nu\}$.

Theorem 2.2 *The inequality* (2) *is valid for* $P_{\mathcal{I}}$*. Moreover, if*

(i) *for every* $\nu \in \mathbb{N}$, $V^{\nu}_{S,T} = \emptyset$ *or* $\nu = 1$ *or* ($|V^{\nu}_{S,T}| > 1$ *and* $G^{\nu}_{S,T}$ *is connected), and*

(ii) *for every* $i \in B \setminus S$ *(resp.* $j \in R \setminus T$*), there exists* $t \in S \cup T$ *and an* $\mathcal{N}$*-set* N *with* $|N| = \nu_t$ *and* $\nu_k = \nu_t$ *for every* $k \in N$ *such that* $\mathcal{X}_{S \cup \{i\}} \setminus \mathcal{X}_N$ *and* $\mathcal{X}_T \setminus \mathcal{X}_N$ *(resp.* $\mathcal{X}_S \setminus \mathcal{X}_N$ *and* $\mathcal{X}_{T \cup \{j\}} \setminus \mathcal{X}_N$*) are linearly separable,*

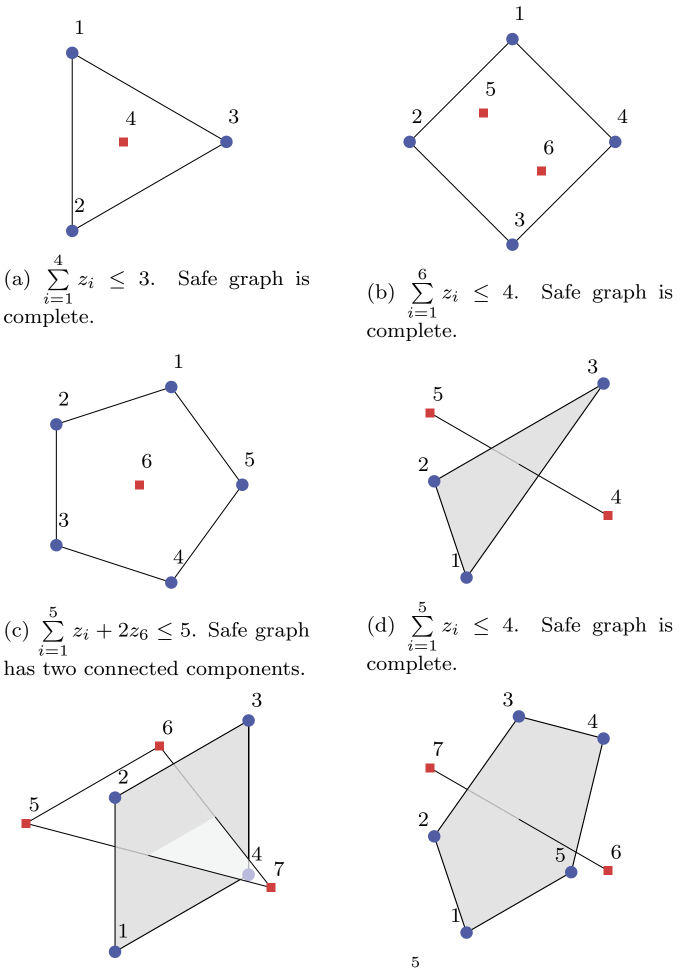
then (2) *defines a facet of* $P_{\mathcal{I}}$*.*

It is interesting to note that the minimality hypothesis in Theorem 2.1 implies the hypothesis (ii) of Theorem 2.2. It is worth remarking that the facetness conditions specified by Theorem 2.2 hold for many simple structures, including those depicted in Fig. 1 for $d = 2$ and $d = 3$.

3 Checking for feasibility

Since the number of constraints in the model is exponential, we resort to the following *lazy constraint* scheme in order to quickly detect infeasible integral points. According to [5], a fractional solution $\bar{z}$ belongs to the linear relaxation of the integer programming formulation if and only the following linear programming formulation

$$\begin{aligned}
\max \quad & 2M - (M+1)\Big(\sum_{i \in B} \upsilon_i \bar{z}_i + \sum_{j \in R} \upsilon_j \bar{z}_j\Big) \\
\text{s.t.} \quad & \sum_{i \in B} \upsilon_i \mathbf{x}_i = \sum_{j \in R} \upsilon_j \mathbf{x}_j \\
& \sum_{i \in B} \upsilon_i = \sum_{j \in R} \upsilon_j \\
& \upsilon_i \leq \bar{z}_i, \qquad i \in B \\
& \upsilon_j \leq \bar{z}_j, \qquad j \in R \\
& (\upsilon_B, \upsilon_R) \geq \mathbf{0}
\end{aligned}$$



(a) $\sum_{i=1}^{4} z_i \leq 3$. Safe graph is complete.

(b) $\sum_{i=1}^{6} z_i \leq 4$. Safe graph is complete.

(c) $\sum_{i=1}^{5} z_i + 2z_6 \leq 5$. Safe graph has two connected components.

(d) $\sum_{i=1}^{5} z_i \leq 4$. Safe graph is complete.

(e) $\sum_{i=1}^{6} z_i + 2z_7 \geq 2$. Safe graph has two connected components.

(f) $\sum_{i=1}^{5} z_i + 2z_6 + 2z_7 \leq 7$. Safe graph has two connected components.

Fig. 1. Facet defining structures according to Theorem 2.2. Solid segments represent the convex hull of their endpoints.

has a nonnegative optimal value, where M is a big positive number. This follows from the application of a classical result by Balas [2] to the mixed integer programming formulation for the 2-class classification problem considered in [5]. Since this model is a linear programming formulation, then it can be solved efficiently, hence such a feasibility check is of practical use.

Note that the feasibility of this linear programming model does not depend on the actual value of M, namely if M is small then some solutions are lost but the problem remains feasible. However, small values of M will lead to meaningless mathematical solutions. As observed in [1], setting M to a value a few orders of magnitude larger than the size of the box encapsulating the data points typically suffices to produce good quality solutions.

4 Separation procedures

The family of $\mathcal{N}$-inequalities to consider has exponential size, hence separation procedures are needed for a practical implementation. We describe in this section an integer programming approach for performing such a separation. Although not guaranteed to run in polynomial time, the proposed approach turned out to be quite effective in practice.

4.1 Convex inclusion inequalities

Consider first the inequalities given by Theorem 2.1 when $|T| = 1$, called *convex inclusion inequalities* in [5] (see figures 1a and 1c for illustrations). Given a solution $\bar{z}$ of the relaxed model, the separation problem for these inequalities consists in finding a point $\mathbf{x}_j$, $j \in R$, and a set $S \subseteq B$, such that $\mathbf{x}_j \in \mathrm{conv}(\mathcal{X}_S)$, and such that $\bar{z}_j + \sum_{i\in S} \bar{z}_i > |S|$. We call $\mathbf{x}_j$ to be the *center* of the inequality. We restrict ourselves to a subset of the variables with fractional values, by considering the subsets $B_F = \{i \in B \mid LB \le \bar{z}_i \le UB\}$ and $R_F = \{j \in R \mid LB \le \bar{z}_i \le UB\}$, where LB and UB are two parameters such that $0 \le LB < UB \le 1$.

The separation can be accomplished with the following integer programming model. For each $j \in R$, we introduce the binary variable b_j specifying whether $\mathbf{x}_j$ is the center of the inequality or not. For each $i \in B$, we introduce the binary variable b_i specifying whether $i \in S$. We also have a continuous variable $\lambda_i \in [0, 1]$ for each $i \in B$, in such a way that $\{\lambda_i\}_{i\in B}$ represent the multipliers associated with the points in $\mathcal{X}_B$ showing that $\mathbf{x}_j$ is a convex combination of the points selected to be included in the set S. In this setting, the separation problem can be formulated as follows.

$$
\begin{aligned}
\max\quad & \sum_{i\in B_F} (\bar{z}_i - 1) b_i + \sum_{i\in R_F} \bar{z}_i b_i \\
\text{s.t.}\quad & \sum_{i\in B_F} \lambda_i \mathbf{x}_i = \sum_{j\in R_F} b_j \mathbf{x}_j && (3)\\
& \sum_{i\in B_F} \lambda_i = \sum_{j\in R_F} b_j = 1 && (4)\\
& \sum_{i\in B_F} b_i = d + 1 && (5)\\
& 0 \le \lambda_i \le b_i, \qquad i \in B_F && (6)\\
& b_i \in \{0, 1\}, \qquad i \in B_F \cup R_F
\end{aligned}
$$

The objective function asks to maximize the difference between the left-hand side and the right-hand-side of the inequality $\bar{z}_j + \sum_{i\in S} \bar{z}_i \le |S|$. Constraint (3) ensures that the selected point $\mathbf{x}_j$ is indeed a convex combination of the points selected to form S. Constraint (4) asserts that exactly one point in R is selected (and this point will be $\mathbf{x}_j$ in the inequality), and that the variables $\{\lambda_i\}_{i\in B}$ indeed represent a convex combination of the selected points from $\mathcal{X}_B$. Constraint (5) asks to select exactly $d+1$ points, which corresponds to the maximum number of points generating a minimal set S such that $\mathbf{x}_j \in \text{conv}(\mathcal{X}_S)$. Finally, constraints (6) ensure that the convex combination in constraint (3) is taken among the selected points in S.

4.2 S, T-Inequalities

We can extend the idea discussed above in order to separate the S, T-inequalities (1). To this end, we again introduce a binary variable b_i for each $i \in B$ and a binary variable b_j for each $j \in R$, representing whether the associated point is selected to belong to S and T, respectively. In this setting, we need $\text{conv}(\mathcal{X}_S) \cap \text{conv}(\mathcal{X}_B) \neq \emptyset$, and this is enforced with the introduction of a variable $\lambda_i \in [0,1]$ for each $i \in B$ and a variable $\lambda_j \in [0,1]$ for each $j \in R$. The resulting model is as follows.

$$
\begin{aligned}
\max \quad & 1 + \sum_{i \in B_F \cup R_F} (\bar{z}_i - 1) b_i & \\
\text{s.t.} \quad & \sum_{i \in B_F} \lambda_i \mathbf{x}_i = \sum_{j \in R_F} \lambda_j \mathbf{x}_j & (7) \\
& \sum_{i \in B_F} \lambda_i = \sum_{j \in R_F} \lambda_j = 1 & (8) \\
& \sum_{i \in B_F} b_j \geq 2 & (9) \\
& \sum_{j \in R_F} b_j \geq 2 & (10) \\
& \sum_{i \in B_F} b_i + \sum_{j \in R_F} b_j \leq d + 2 & (11) \\
& 0 \leq \lambda_t \leq b_t, \qquad t \in B_F \cup R_F & (12) \\
& b_t \in \{0,1\}, \qquad t \in B_F \cup R_F &
\end{aligned}
$$

Again, the objective function asks to maximize the difference between the left-hand-size and the right-hand-side of constraint (1) at the solution $\bar{z}$. Constraint (7) asks for a nonempty intersection of $\text{conv}(\mathcal{X}_S)$ and $\text{conv}(\mathcal{X}_T)$, by ensuring the existence of coefficients $\{\lambda_i\}_{i \in S}$ and $\{\lambda_j\}_{j \in T}$ representing coincident convex combinations from each set. Constraint (8) asks these coefficients to indeed represent a convex combination. Constraints (9) and (10) ensure that $|S| \geq 2$ and $|T| \geq 2$, respectively, in order to avoid generating convex inclusion cuts. Constraint (11) asks to select at most $d+2$ points, since this is the maximum value of $|S| + |T|$ generating a minimal set of linearly inseparable points. Finally, constraints (12) ensure that the convex combination in constraint (3) is taken among the selected points in S and T.

4.3 Enforcing fractional variables

Contrary to the aim of the lazy constraints discussed in Section 3 (ensuring the feasibility of integral solutions), the purpose of the separation procedures described above is to cut fractional solutions. Since not all the constraints (1) are included in the model, a fractional solution can violate some of them –defined by subsets S and T corresponding to integral variables. Hence, in order to force fractional variables in the generated cut, we add the constraint $\sum_{i \in B_F} f_{\bar{z}_i} b_i \geq F$, where $f_{\bar{z}_i} = 1$ if $D\bar{z}_i < LB + UB$ and $f_{\bar{z}_i} = 0$ otherwise, for suitable values $F, D \in \mathbb{R}$. The parameter D is set to a value such that $f_{\bar{z}_i} = 1$ only if $\bar{z}_i$ lies in the interval (LB, UB). The parameter F corresponds to a lower bound on the number of fractional variables in the generated cut.

5 Experiments

We have implemented a branch and cut procedure for the 2-class single-group classification problem, based on the results and algorithms mentioned in the previous sections. In this section we provide some preliminary computational experiments with separation heuristics for some special cases of $\mathcal{N}$-inequalities in order to explore such combinatorial methods as effective tools for solving classification problems. Our main goal with these experiments is not to provide a competitive algorithm for the 2-class single-group classification problem, since continuous optimization methods are much more effective than cutting-plane algorithms for this problem. Instead, we intend to assess whether combinatorial tools can potentially improve the overall efficiency when coupled to existing solution methods. To this end, we implemented a branch and bound algorithm to solve the basic formulation.

The initial model is composed by the z-variables with no constraints, and lazy constraints are dynamically added with the procedure mentioned in Section 3. This implementation was compared with the same algorithm when cuts are incorporated with the integer-programming-based separation procedures described in Section 4. The algorithm was coded in the Java programming language, using Cplex 12.8 as the linear programming solver (for the linear relaxations and checking for feasibility) and mixed integer programming solver (for the separation problems).

Table 1 summarizes the preliminary experiments with synthetic 2-class instances generated with the following random procedure. Initially, two points are defined in opposite sides and at distance 1 of a given hyperplane of dimension $d - 1$ to act as the centers of the two classes. Then, for each one of the centers, a cluster of points is created normally distributed about vertices of a d-dimensional hypercube with sides of length 2. Finally, a number of noise points are introduced in each class. For the results reported, the instances have 2500 random and 16 noise points and dimensions ranging from $d = 2$ to $d = 6$. An illustration of the instance for $d = 2$ is shown in Fig. 2.

For every instance, we report the number of nodes in the enumeration tree, the number of generated lazy constraints, the total running time in seconds, and the optimality gap (in %). The parameter M was set to 200 in all cases, which is two

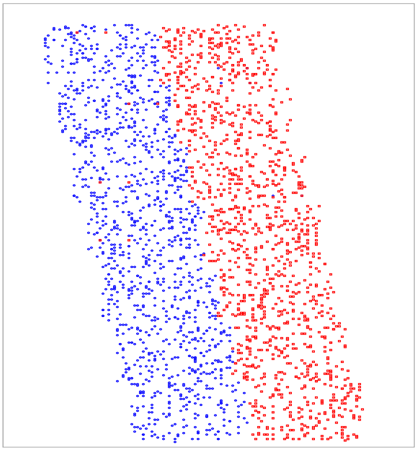


Fig. 2. Instance for $d = 2$.

orders of magnitude larger than the coordinates of the instance points. The columns labeled "No cuts" correspond to the execution of the procedure with no additional cuts, besides the mandatory lazy constraints. The columns labeled "Inclusion cuts" correspond to the dynamical addition of the convex inclusion cuts, with the separation model presented in Section 4.1. The columns labeled "S, T-cuts" correspond to the dynamical addition of the S, T-cuts, with the separation model presented in Section 4.2. In both cases, the inequality found is added only if the objective value of the corresponding separation model is larger than a parameter $VIOL$. Finally, the results corresponding to the dynamical addition of S, T-cuts and, if it fails, followed by the dynamical addition of convex inclusion ones, appear in the columns labeled "Mixed". Note that the parameters LB and D used in the separation procedures ensure that only fractional solutions are separated (such parameters for the convex inclusion and S, T-inequalities are identified with indices I and S, T, respectively). We conducted experiments with several combinations of values for LB and D. The results in Table 1 correspond to the best configuration for each case.

As Table 1 suggests, the separation procedures are able to generate a large number of cuts and provide upper bounds that are competitive with those of the pure branch and bound. In almost all cases (there is only one exception, with $d = 2$), the addition of cuts drastically reduces the addition of lazy constraints within the time limit of 1 hour. Another characteristic that can be observed is that the S, T-cuts are more effective than convex inclusion cuts.

6 Concluding remarks

We have presented in this work a first computational study of the effectiveness of the valid inequalities introduced in [5] for the 2-class classification problem, by

d	m	$BEST$	**Nodes**	**Lazy**	**Time**	**Cuts**	**Gap**
			No cuts				
2	2516	2479	1725	5880	**60.46**	–	–
3	2516	2474	65493	4746	738.50	–	–
4	2516	2469	80835	11921	3600	–	0.32
5	2516	2484	324859	4102	3587.73	–	–
6	2516	2480	105174	8215	3600	–	0.28
d	m	$BEST$	**Nodes**	**Lazy**	**Time**	**Cuts**	**Gap**
			Mixed $(LB_{S,T} = 0.7, D_{S,T} = 1.75)/(LB_I = 0.5, D_I = 1.61)$				
2	2516	2479	812	4771	81.51	54/**82**	–
3	2516	2474	**758**	**1602**	662.27	339/357	–
4	2516	2469	**656**	**4902**	3600	1734/**4020**	10.4
5	2516	2484	**380**	**877**	3600	**2011/4338**	98.1
6	2516	2480	**2784**	3928	3600	1128/**1174**	4.03
			S, T-**cuts** $(LB_{S,T} = 0.7, D_{S,T} = 1.75)$				
2	2516	2479	1276	8715	84.46	69	–
3	2516	2474	8094	2242	**490.15**	**1336**	–
4	2516	2469	34287	**6987**	3600	2055	**0.27**
5	2516	2484	39722	2904	**2022.35**	**1189**	–
6	2516	2480	37905	3841	3600	**2410**	**0.25**
			Inclusion cuts $(LB_I = 0.6, D_I = 1.61)$				
2	2516	2479	**805**	**4440**	73.79	**82**	–
3	2516	2474	946	**1745**	783.51	862	–
4	2516	2469	**3166**	7437	3600	**3457**	0.87
5	2516	2484	**4455**	**2532**	3600	393	0.37
6	2516	2480	**3638**	**3617**	3600	582	1.09

Table 1
Computational results for the procedure introduced in Section 5. We have used $M = 200$, $F = 1$, $VIOL = 0.2$, and $UB = 1.0$ in these experiments. Column $BEST$ indicates the value of the best solution found in all configurations.

resorting to the particular case of single groups for each class. The number of constraints in the initial model makes it necessary to resort to lazy constraints, which can be readily separated by a linear programming model. The separation of families of valid inequalities appears to be a tougher issue, and we resorted to mixed integer programming models for accomplishing this task. Our computational experience suggests that this strategy may be effective. As a future work, we intend to perform extensive computational experiments with the proposed cut generating procedure appended to other optimization methods for classification problems.

Acknowledgment

We are very grateful to the anonymous reviewers for their helpful remarks.

References

[1] Amaldi, E., S. Coniglio and L. Taccari, *Discrete optimization methods to fit piecewise affine models to data points*, Computers & Operations Research **75** (2016), pp. 214–230.
URL http://www.sciencedirect.com/science/article/pii/S0305054816301022

[2] Balas, E., *Projection, lifting and extended formulation in integer and combinatorial optimization*, Annals of Operations Research **140** (2005), pp. 125–161.

[3] Bertsimas, D. and R. Shioda, *Classification and regression via integer optimization*, Operations Research **55** (2007), pp. 252–271.

[4] Carrizosa, E. and D. R. Morales, *A mixed integer optimisation model for data classification*, Computers & Operations Research **40** (2013), pp. 150–165.

[5] Corrêa, R. C., D. Delle Donne and J. Marenco, *On the combinatorics of the 2-class classification problem*, Discrete Optimization (2018), pp. 1572–5286.
URL http://www.sciencedirect.com/science/article/pii/S1572528617302748

[6] Freed, N. and F. Glover, *Evaluating alternative linear programming models to solve the two group discriminant problem*, Decision Sciences **17** (2007), pp. 151–162.

[7] Maskooki, A., *Improving the efficiency of a mixed integer linear programming based approach for multi-class classification problem*, Comput. Ind. Eng. **66** (2013), pp. 383–388.
URL http://dx.doi.org/10.1016/j.cie.2013.07.005

[8] Pardalos, P. M. and P. Hansen, **45**, American Mathematical Society, Providence, RI, 2008.

[9] Sun, M., *A mixed integer programming model for multiple-class discriminant analysis*, International Journal of Information Technology and Decision Making **10** (2011), pp. 589–612.

[10] Uney, F. and M. Turkay, *A mixed-integer programming approach to multi-class data classification problem*, European Journal of Operational Research **173** (2006), pp. 910–920.
URL https://EconPapers.repec.orgRePEc:eee:ejores:v:173:y:2006:i:3:p:910-920

[11] Xu, G. and L. G. Papageorgiou, *A mixed integer optimisation model for data classification*, Comput. Ind. Eng. **56** (2009), pp. 1205–1215.
URL http://dx.doi.org/10.1016/j.cie.2008.07.012