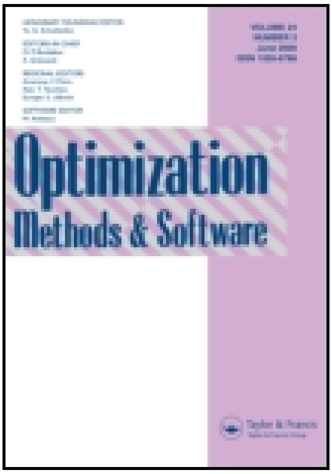

Optimization Methods and Software

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/goms20

Separation via polyhedral conic functions

Rafail N. Gasimov [a] & Gurkan Ozturk [a]

[a] Eskişehir Osmangazi University, Industrial Engineering Department , 26030, Bademlik, Eskişehir, Turkey

Published online: 16 Aug 2006.

Optimization Methods and Software
Vol. 21, No. 4, August 2006, 527–540

Separation via polyhedral conic functions

RAFAIL N. GASIMOV* and GURKAN OZTURK

Eskişehir Osmangazi University, Industrial Engineering Department, 26030, Bademlik, Eskişehir, Turkey

(Received 21 January 2006; in final form 24 March 2006)

We consider the problem of discriminating between two finite point sets A and B in the n-dimensional space by using a special type of polyhedral function. An effective finite algorithm for finding a separating function based on iterative solutions of linear programming subproblems is suggested. At each iteration a function whose graph is a polyhedral cone with vertex at a certain point is constructed and the resulting separating function is defined as a point-wise minimum of these functions. It has been shown that arbitrary two finite point disjoint sets can be separated by using this algorithm. An illustrative example is given and an application on classification problems with some real-world data sets has been implemented.

Keywords: Separability; Polyhedral functions; Polyhedral cones; Polyhedral conic functions; Classification

1. Introduction

We consider the problem of separation of two nonempty finite point sets A and B in $\mathbb{R}^n$. It is well-known that if the convex hulls of such sets are disjoint, that is $co(A) \cap co(B) = \emptyset$, a strictly separating plane can be constructed. If the convex hulls of A and B intersect then a linear programming (LP) technique can be applied to obtain a hyperplane which minimizes some misclassification measure. A technique for finding such a hyperplane is described by Bennett and Mangasarian [1]. Algorithms based on similar approach are developed also in [2–6]. Whenever the two sets A and B cannot be separated by a hyperplane, $co(A) \cap co(B) \neq \emptyset$ but the convex hull of A and the set B do not intersect, $co(A) \cap B = \emptyset$ they are h-polyhedrally separable [7]. That is, there exist h hyperplanes such that the points of A are contained in a convex polyhedron (intersection of h half-spaces) and the points of B are left outside the polyhedron. Astorino and Gaudioso [5] developed an algorithm on h-polyhedral separability. They introduced an error function which is piecewise linear, but not convex or concave and defined a procedure based on the iterative solution of the LP descent direction finding subproblems. Bagirov [6] introduced the notion of max–min separability which can be considered as a generalization of the h-polyhedral separability given in [5]. It was shown that if the sets

*Corresponding author. Email: gasimovr@ogu.edu.tr

Optimization Methods and Software
ISSN 1055-6788 print/ISSN 1029-4937 online
http://www.tandf.co.uk/journals
DOI: 10.1080/10556780600723252

A and B are disjoint then they are max–min separable. Bagirov described an error function for this case and developed an algorithm for its minimization.

Gasimov [8] considered a case of two nonempty sets A and B in real normed spaces, partially ordered by a pointed convex cone C, where $A - C$ is convex and $(A - C) \cap B = \emptyset$. He used a special class of polyhedral functions in order to separate $A - C$ and B and characterized properly minimal points of nonconvex vector optimization problems using such a separation.

In this paper, we extend the class of functions used by Gasimov [8], and define the so-called polyhedral conic functions which are used to construct a separation function for the given two arbitrary finite point disjoint sets A and B in $\mathbb{R}^n$. These functions are formed as an augmented l_1 norm – with a linear part added. A graph of such a function is a polyhedral cone with a sublevel set including at the utmost an intersection of 2^n half spaces. An iterative algorithm generating a nonlinear separating function by using polyhedral conic functions (PCF) and therefore called a PCF algorithm is developed. This algorithm is based on solutions of linear programming subproblems. A solution of these subproblems at each iteration results in the polyhedral conic function which separates a certain part of the set A from the whole set B. By excluding these points from A, the algorithm passes to the next iteration and so on. The resulting separation function is defined as a point-wise minimum of all the functions generated. We show that the algorithm terminates in a finite number of iterations and the maximum number of iterations required for separating two arbitrary finite point sets does not exceed the number of elements in one of these sets. An illustrative example has been constructed and an application on classification problems has been implemented. Numerical experiments using real-world data sets have been carried out.

The remainder of the paper is organized as follows. In the second section the class of polyhedral conic functions used here is introduced and an illustrative example is given. In this

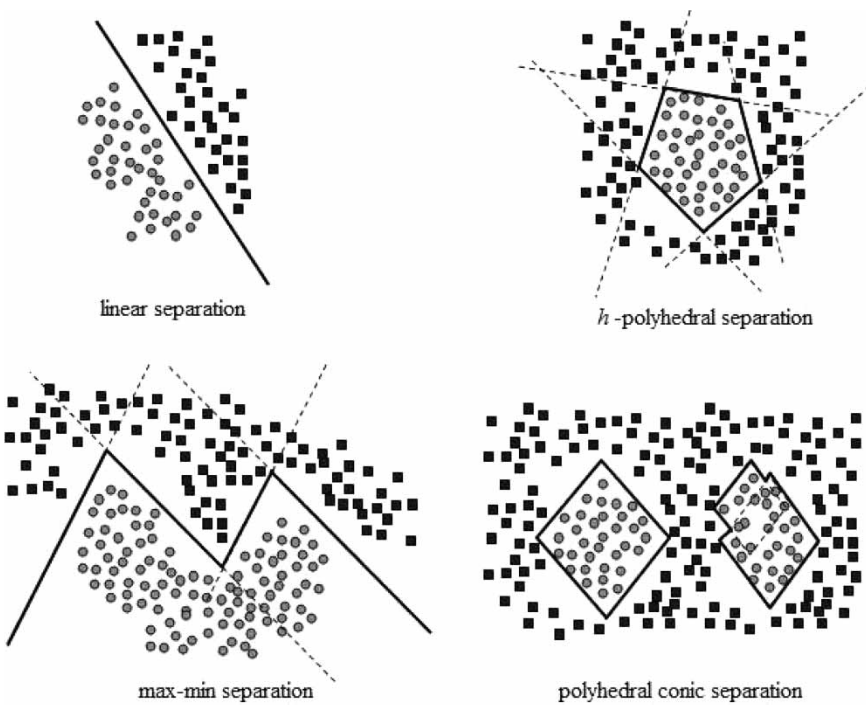

Figure 1. Geometric interpretation of four separation methods: linear, h-polyhedral, max–min and polyhedral conic.

section the algorithm for finding separating functions is also presented. Application on classification problems and results of numerical experiments are presented in section 3. Section 4 presents some conclusions from the study.

2. Polyhedral conic functions and the separating algorithm

In this section we introduce a special class of polyhedral functions used for separating two disjoint point sets in $\mathbb{R}^n$. This class consists of functions $g_{(w,\xi,\gamma,a)}: \mathbb{R}^n \to \mathbb{R}$ defined as:

$$g_{(w,\xi,\gamma,a)}(x) = w'(x-a) + \xi\|x-a\|_1 - \gamma, \tag{1}$$

where $w, a \in \mathbb{R}^n, \xi, \gamma \in \mathbb{R}$, $w'x = w_1x_1 + \cdots + w_nx_n$ is a scalar product of w and x, $\|x\|_1 = |x_1| + \cdots + |x_n|$ is a l_1-norm of the vector $x \in \mathbb{R}^n$.

LEMMA 2.1 *A graph of the function $g_{(w,\xi,\gamma,a)}$ defined in equation* (1) *is a polyhedral cone with vertex at* $(a, -\gamma) \in \mathbb{R}^n \times \mathbb{R}$

Proof To prove the lemma we show that:

(i) a graph of the function is a cone with vertex at $(a, -\gamma) \in \mathbb{R}^n \times \mathbb{R}$, and
(ii) each sublevel set of this function is a convex polyhedron.

To prove the first part, consider a set $graph(g_{(w,\xi,\gamma,a)}) - (a, -\gamma)$:

$$graph(g_{(w,\xi,\gamma,a)}) - (a, -\gamma) = \{(x-a, \alpha+\gamma) : w'(x-a) + \xi\|x-a\|_1 - \gamma = \alpha\}.$$

By letting $x - a = y, \alpha + \gamma = \beta$, this set can be written also as

$$graph(g_{(w,\xi,\gamma,a)}) - (a, -\gamma) = \{(y, \beta) : w'y + \xi\|y\|_1 = \beta\}. \tag{2}$$

It is obvious that this set is a cone with vertex at the origin. Indeed if $(y, \beta) \in graph(g_{(w,\xi,\gamma,a)}) - (a, -\gamma)$ then

$$w'y + \xi\|y\|_1 = \beta.$$

Hence, for any $\lambda > 0$ we have:

$$\lambda w'y + \lambda\xi\|y\|_1 = \lambda\beta,$$

or

$$w'(\lambda y) + \xi\|\lambda y\|_1 = \lambda\beta,$$

which implies that $(\lambda y, \lambda\beta)$ also belongs to $graph(g_{(w,\xi,\gamma,a)}) - (a, -\gamma)$ and therefore this set is a cone with vertex at the origin.

Now show the second part of the proof. Let α be a real number. Then the sublevel set of the function $g_{(w,\xi,\gamma,a)}$ given by (1) is:

$$S_\alpha = \{x \in \mathbb{R}^n : g_{(w,\xi,\gamma,a)}(x) = w'(x-a) + \xi \|x-a\|_1 - \gamma \leq \alpha\}.$$

By using a definition of l_1-norm, this set can equivalently be written as

$$S_\alpha = \{x \in \mathbb{R}^n : \tilde{w}'(x-a) - \gamma \leq \alpha\},$$

where

$$\tilde{w}'(x-a) = \sum_{i=1}^{n}(w_i + \xi sgn(x_i - a_i))(x_i - a_i).$$

This means that the sublevel set S_α is an intersection of utmost 2^n half-spaces and therefore is a convex polyhedron. The proof is completed. ■

Definition 2.2 *A function* $g\colon \mathbb{R}^n \to \mathbb{R}$ *is called polyhedral conic if its graph is a cone and all its sublevel sets*

$$S_\alpha = \{x \in \mathbb{R}^n : g(x) \leq \alpha\},$$

for $\alpha \in \mathbb{R}$, *are polyhedrons.*

It follows from Lemma 2.1 that each function of the form (1) is a polyhedral conic function (PCF).

We state now our algorithm for solving the separation problem. Since the algorithm is based on polyhedral conic functions we will call it PCF Algorithm.

2.1 *PCF algorithm*

Let A and B be two given sets in $\mathbb{R}^n$:

$$A = \{a^i \in \mathbb{R}^n : i \in I\},\ B = \{b^j \in \mathbb{R}^n : j \in J\}$$

where $I = \{1, \ldots, m\}$, $J = \{1, \ldots, p\}$. The algorithm presented below generates at each iteration a function of the form (1) by calculating the parameters w, ξ and γ as a solution of a certain linear programming (LP) subproblem. These parameters are used to define a polyhedral conic function whose sublevel set divides the whole space into two parts such that all the points of B remain 'outside', and as many points of A as possible remain 'inside' this sublevel set. By excluding these latter points from A, the algorithm passes to the next iteration and generates a new separating function for this modified set. The process continues until an empty set is obtained. The resulting separating function is defined as a point-wise minimum of all functions so generated. We will prove that the algorithm terminates in finite steps.

Initialization Step: Let $l = 1$, $I_l = I$, $A_l = A$ and go to Step 1.

Step 1: Let a^l be an arbitrary point of A_l. Solve subproblem P_l:

$$(P_l) \quad \min\left(\frac{y' e_m}{m}\right) \tag{3}$$

subject to

$$w'\left(a^i - a^l\right) + \xi\|a^i - a^l\|_1 - \gamma + 1 \leq y_i, \quad \forall i \in I_l, \tag{4}$$

$$-w'\left(b^j - a^l\right) - \xi\|b^j - a^l\|_1 + \gamma + 1 \leq 0, \quad \forall j \in J, \tag{5}$$

$$y = (y_1, \ldots, y_m) \in \mathbb{R}^m_+, w \in \mathbb{R}^n, \xi \in \mathbb{R}, \gamma \geq 1 \tag{6}$$

Let $w^l, \xi^l, \gamma^l, y^l$ be a solution of (P_l). Let

$$g_l(x) = g_{(w^l,\xi^l,\gamma^l,a^l)}(x) \tag{7}$$

and go to Step 2.

Step 2: Let $I_{l+1} = \{i \in I_l : g_l(a^i) + 1 > 0\}$, $A_{l+1} = \{a^i \in A_l : i \in I_{l+1}\}$, $l = l + 1$. If $A_l \neq \emptyset$ go to Step 1.

Step 3: Define the function $g(x)$ (separating the sets A and B) as

$$g(x) = \min_l g_l(x) \tag{8}$$

and stop.

At each iteration l, the algorithm arbitrarily chooses some element a^l from the set A_l and calculates parameters (w^l, ξ^l, γ^l) by solving a linear subproblem (P_l). All these parameters are then used in equation (7) for defining the function g_l. It follows from Lemma 2.1 that the graph of the function g_l consisting of points $(x, z) \in \mathbb{R}^n \times \mathbb{R}$ with $z = g_l(x)$ is a cone with vertex at $(a^l, -\gamma^l)$. A constraint $\gamma \geq 1$ stated in constraint set (6) ensures that the vertex of this cone has to be placed 'under' the hyperplane $z = 0$, that is in the half-space $\mathbb{R}^n \times (0, -\infty)$. The constraint set (4) ensures that the point a^l and all the points of the set A^l which are 'close' to a^l have to be in the polyhedron corresponding to the sublevel set $\{x : g_l(x) \leq -1\}$. The closeness of these points of A_l to a^l is defined by the optimal value of the objective function (3) in (P_l). That is, the sublevel set $\{x : g_l(x) \leq -1\}$ will include as many elements of A_l (besides a^l) as the value of this objective function is close to zero. Thus the objective function (3) and the constraint sets (4) and (6) ensure that all the elements of A_l will be enclosed to the sublevel set $\{x : g_l(x) \leq -1\}$ of g_l if this minimum is zero. On the other hand, the constraint set (5) ensures that all the elements of the set B have to be remained outside of the sublevel set $\{x : g_l(x) < 1\}$ at each iteration. Note that such a 'separability' at each iteration becomes possible due to the characteristics of polyhedral conic functions described in Lemma 2.1. We will call the method of separation described in the algorithm the PCF separation.

The following theorem proves that the presented algorithm terminates in a finite number of iterations and the resulting function g defined by equation (8) separates arbitrary disjoint sets A and B consisting of finite elements in R^n.

THEOREM 2.3 *PCF Algorithm terminates in a finite number of iterations and the function* $g : \mathbb{R}^n \to \mathbb{R}$ *defined by equation* (8) *strictly separates the sets A and B in the sense that*

$$g(a) < 0, \quad \forall a \in A, \tag{9}$$

$$g(b) > 0, \quad \forall b \in B. \tag{10}$$

Proof First show that the problem (P_l) has a solution $(w_l, \xi_l, \gamma_l) \in \mathbb{R}^n \times \mathbb{R}_+ \times [1, \infty)$ such that the corresponding function g_l separates at least one element, (say a_l) of A_l and the whole set B. By taking $w_l = 0, \gamma_l = 1$ we obtain a function $g_l(x) = \xi \|x - a^l\|_1 - 1$, for which we have $g_l(a^l) = -1 < 0$ and $g_l(b^j) = \xi \|b^j - a^l\|_1 - 1, \forall j \in J$. Since $b^j \in B$ we have $\|b^j - a^l\|_1 > 0, \forall j \in J$. Therefore when ξ is sufficiently large, the term $\xi \|b^j - a^l\|_1$ can be made large enough. Then for ξ sufficiently large, we have $g_l(b^j) > 0, \forall j \in J$, which means that the function $g_l(x) = \xi_l \|x - a^l\|_1 - 1$ separates a^l and the set B in the sense that $g_l(a_l) < 0$ and $g_l(b) > 0, \quad \forall b \in B$.

Let $\tilde{A}_l$ be the subset of A_l consisting of elements which are separated from B by the function g_l formed using the solution of the problem (P_l) at lth iteration, and let A_{l+1} be the subset of A_l consisting of the elements which could not be separated from B by g_l. If this set is not empty the algorithm will be continued. Since the set A has a finite number of elements, the process will be terminated after the finite number of iterations. Thus we will have a partition $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_L$ of the set A and functions $g_1, g_2, \ldots, g_L$ with properties:

$$A = \bigcup \tilde{A}_l$$

$$g_l(a) < 0, \forall a \in \tilde{A}_l,$$

$$g_l(b) > 0, \forall b \in B, l = 1, \ldots, L.$$

Then the function $g(x) = \min_l\{g_l(x)\}$ will obviously have properties (9) and (10). Indeed, since for every $a \in A$ there exists $l \in \{1, 2, \ldots, L\}$ such that $a \in \tilde{A}_l$, we have $g_l(a) < 0$ and therefore $g(a) = \min_l\{g_l(a) < 0\}$. On the other hand, since $g_l(b) > 0$ for all $l \in \{1, 2, \ldots, L\}$, $b \in B$, we have $g(b) > 0$. ■

Corollary 2.4 *Let A and B be two arbitrary sets consisting of finite number of points in $\mathbb{R}^n$. Then*

(i) *there exists a partition of $A : A = \bigcup \tilde{A}_l$ such that $co(\tilde{A}_l) \cap B = \emptyset$ and functions $g_l(x) = w_l'x + \xi_l\|x\|_1 - \gamma_l$, for $l = 1, 2, \ldots,$ L with $g_l(a) < 0, \forall a \in co(\tilde{A}_l)$, $g_l(b) > 0, \forall b \in B$, and*

(ii) *the function $g(x) = \min_l g_l(x)$ separates A and B in the sense of* (9) *and* (10).

Proof

(i) The existence of a partition $A : A = \bigcup \tilde{A}_l$ and functions $g_l(x)$ with a property

$$g_l(a) < 0, \forall a \in \tilde{A}_l, g_l(b) > 0, \forall b \in B,$$

follows from the proof of Theorem 2.3. Let $C_l = \{x \in \mathbb{R}^n : g_l(x) \leq 0\}$. Then $\tilde{A}_l \subset C$, $B \subset \{x \in \mathbb{R}^n : g_l(x) > 0\}$, and $C \cap B = \emptyset$ by construction. By Lemma 2.1, C_l is a convex polyhedron. Since it contains $\tilde{A}_l$, it contains also $co(\tilde{A}_l)$ – the smallest convex set containing $\tilde{A}_l$. Thus, $co(\tilde{A}_l) \cap B = \emptyset$.

(ii) Is obvious. ■

Example 2.5 Consider two finite point sets A and B in $\mathbb{R}^2$ shown in figure 2. Note that the set A is taken to consist of two isolated parts. The coordinates x_1 and x_2 of the points described in this figure are given in table 1.

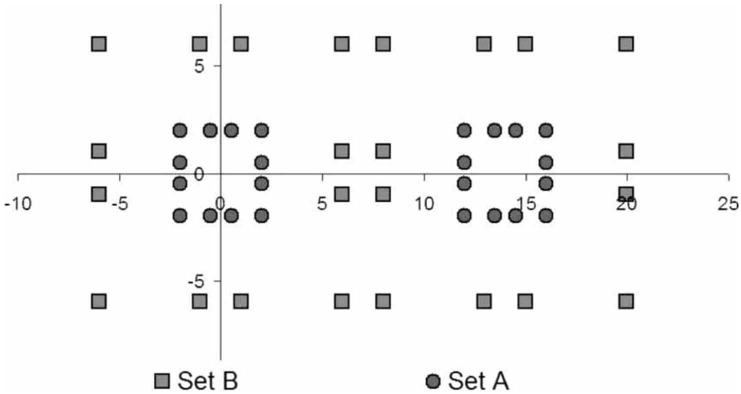

Figure 2. Two finite point sets A and B in $\mathbb{R}^2$.

This example has been solved in two ways. In the first way the PCF algorithm is applied directly. Then we slightly modify Step 1 of the algorithm and apply it to obtain another separation function. The geometrical interpretations for separation functions obtained in these ways are presented in figures 3 and 4, respectively.

2.1.1 First way for the solution of Example 2.5. The PCF algorithm is applied for constructing a separation function. GAMS/CPLEX solver is used for solving the LP subproblems. The algorithm has been terminated in seven iterations. The subsets $\tilde{A}_l, l = 1, \ldots, 7$ partitioning the set A and the corresponding polyhedral conic functions g_l separating these sets from B at each iteration are presented below:

$$g_1(x) = 0.06(x_1 - 14.5) + 0.11(x_2 - 2) + 0.34(|x_1 - 14.5| + |x_2 - 2|) - 1$$

$$g_2(x) = -0.06(x_1 + 0.5) + 0.11(x_2 - 2) + 0.34(|x_1 + 0.5| + |x_2 - 2|) - 1$$

$$g_3(x) = -0.05(x_1 - 2) + 0.23(x_2 - 0.5) + 0.46(|x_1 - 2| + |x_2 - 0.5|) - 1$$

$$g_4(x) = 0.11(x_1 - 16) + 0.02(x_2 + 0.5) + 0.34(|x_1 - 16| + |x_2 + 0.5|) - 1$$

$$g_5(x) = -0.02(x_1 - 0.5) - 0.11(x_2 + 2) + 0.34(|x_1 - 0.5| + |x_2 + 2|) - 1$$

$$g_6(x) = -0.11(x_1 + 2) - 0.06(x_2 + 0.5) + 0.34(|x_1 + 2| + |x_2 + 0.5|) - 1$$

$$g_7(x) = -0.07(x_1 - 12) - 0.26(x_2 + 2) + 0.53(|x_1 - 12| + |x_2 + 2|) - 1.7$$

$$\tilde{A}_1(x) = \{(16, 2), (14.5, 2), (16, 0.5), (12, 2), (13.5, 2), (14.5, -2)\}$$

$$\tilde{A}_2(x) = \{(2, 2), (0.5, 2), (-2, 2), (-0.5, 2), (-2, 0.5), (-0.5, -2)\}$$

$$\tilde{A}_3(x) = \{(2, 0.5), (2, -2), (2, -0.5)\}$$

$$\tilde{A}_4(x) = \{(16, -2), (16, -0.5), (12, -0.5)\}$$

$$\tilde{A}_5(x) = \{(0.5, -2), (-2, -2)\}$$

$$\tilde{A}_6(x) = \{(-2, -0.5)\}$$

$$\tilde{A}_7(x) = \{(12, 0.5), (12, -2), (13.5, -2)\}$$

Table 1. The data points for example 2.5.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A																								
x_1	−2	−2	−2	−2	−0.5	−0.5	0.5	0.5	2	2	2	2	12	12	12	12	13.5	13.5	14.5	14.5	16	16	16	16
x_2	0.5	−0.5	2	−2	2	−2	2	−2	0.5	−2	−0.5	2	2	−2	0.5	−0.5	2	−2	2	−2	−0.5	0.5	2	−2
B																								
x_1	8	6	20	6	15	20	1	−1	−6	20	−6	−6	8	13	20	6	15	13	8	−1	−6	6	8	1
x_2	−6	−1	6	−6	6	1	6	6	1	−6	−1	−6	−1	6	−1	1	−6	−6	1	−6	6	6	6	−6

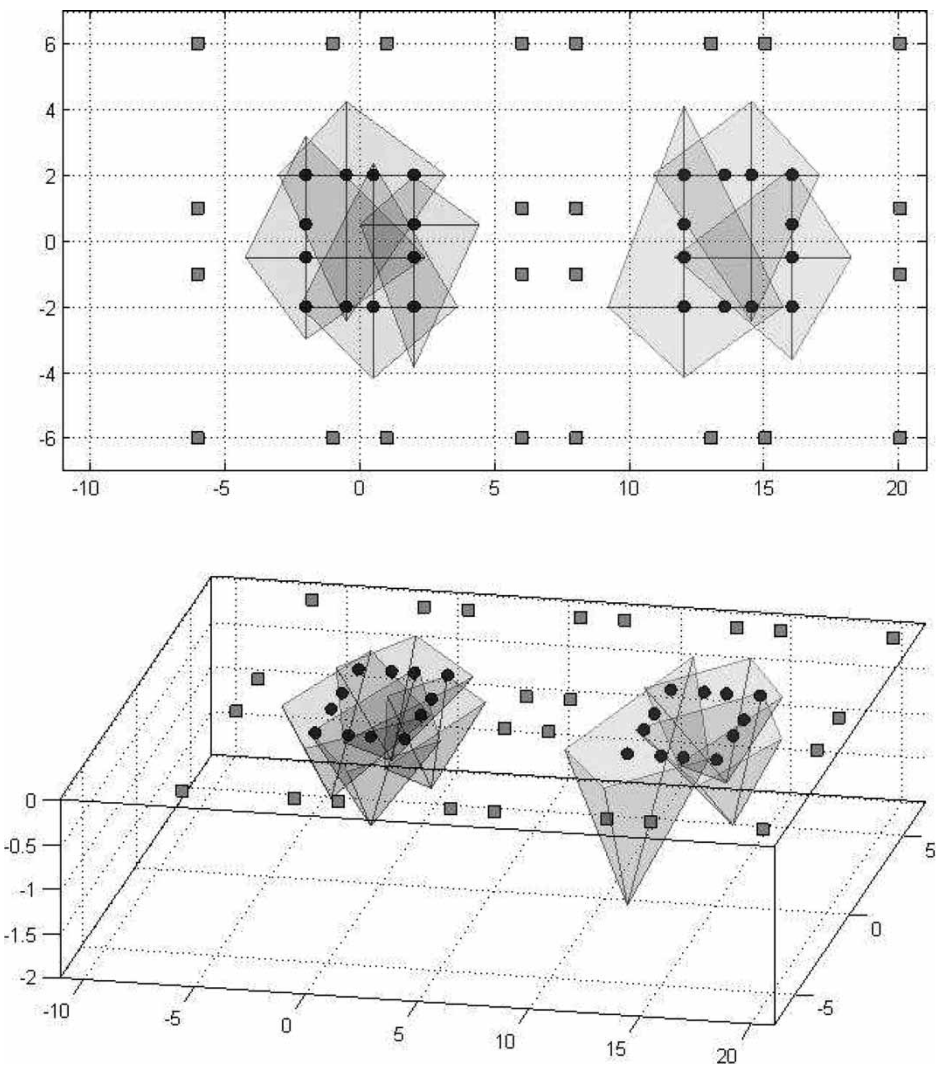

Figure 3. Two- and 3-dimensional views of polyhedral functions obtained by the first way.

2.1.2 The Modified PCF algorithm. It is clear from Lemma 2.1 and the proof of Theorem 2.3 that the number of iterations implemented by the PCF algorithm for a complete separation of the given two sets strongly depends on the place $(a, -\gamma) \in \mathbb{R}^n \times \mathbb{R}$ of vertex of the polyhedral cone representing the graph of the polyhedral conic function generated at each iteration. The γ^l coordinate of this vertex identified as a decision variable is optimally calculated by the LP subproblem, while the coordinates a^l are chosen randomly from the set A_l at each iteration l. It is obvious that the number of iterations and the separation quality may be different, if these points could be chosen in a more efficient way. A natural way to determine an optimal a^l is to identify it also as a decision variable in the subproblem (P_l) defined by (3)–(6) at each iteration. By identifying these points as decision variables the subproblem (P_l) becomes nonlinear. On the other hand, when the set A under consideration is not too large, Step 1 of the PCF algorithm can be modified by the following way. At each iteration l, solve the problem (P_l) for each point $a_i^l \in A_l$ and find the number of elements l_i of A_l separated from B. Then define a^l as $a^l = a^{l_0}$, where l_0=max$\{l_i : i \in I_l\}$.

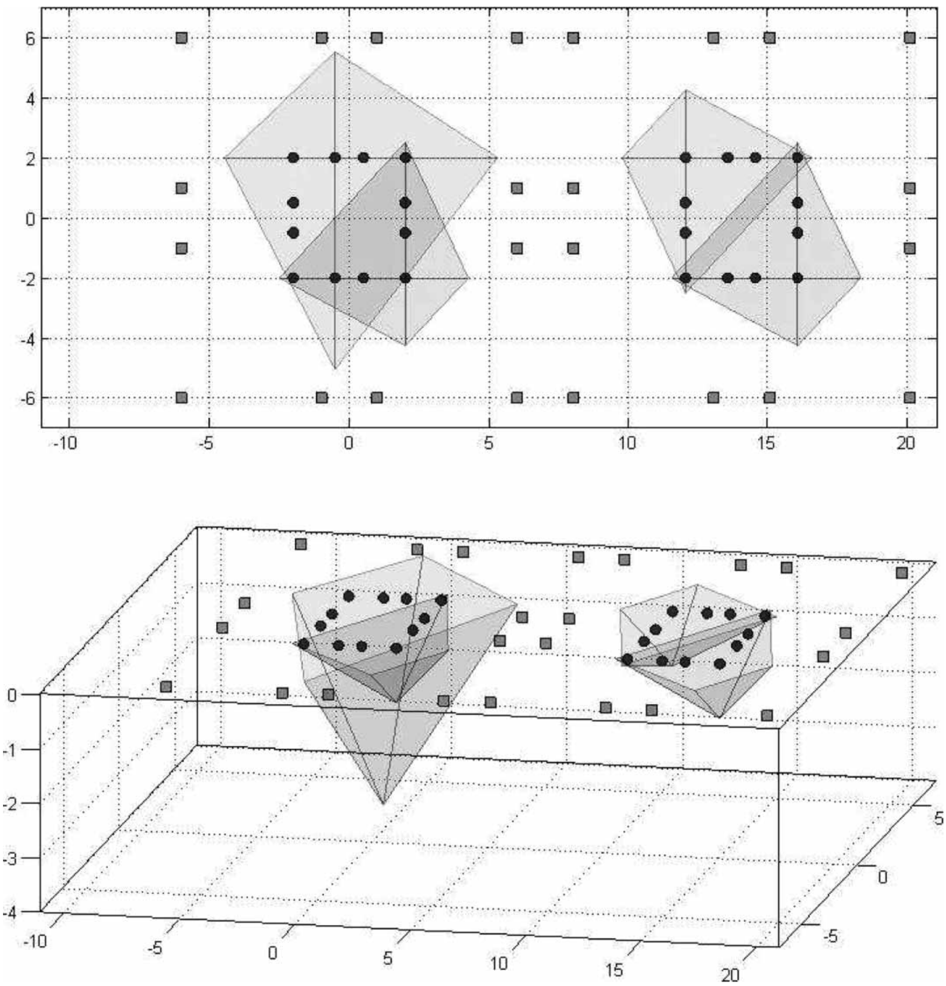

Figure 4. Two-dimensional and 3-dimensional views of polyhedral functions obtained by the second way.

2.1.3 Second way for the solution of Example 2.5. By using the modified version of Step 1 in the PCF algorithm, the separation problem for Example 2.5 has been solved in four iterations. The subsets $\tilde{A}_l$, $l = 1, \ldots, 4$ partitioning the set A and the corresponding polyhedral conic functions g_l separating these sets from B at each iteration are presented below:

$$\tilde{A}_1(x) = \{(16, 2), (14.5, 2), (12, 2), (13.5, 2), (12, 0.5), (12, -2), (12, -0.5)\}$$

$$\tilde{A}_2(x) = \{(2, 2), (2, 0.5), (2, -2), (2, -0.5), (0.5, -2), (-2, -2), (-0.5, -2)\}$$

$$\tilde{A}_3(x) = \{(16, 0.5), (16, -2), (16, -0.5), (14.5, -2), (13.5, -2)\}$$

$$\tilde{A}_4(x) = \{(0.5, 2), (-2, 2), (-0.5, 2), (-2, 0.5), (-2, -0.5)\}$$

$$g_1(x) = -0.11(x_1 - 12) + 0.11(x_2 - 2) + 0.33(|x_1 - 12| + |x_2 - 2|) - 1$$
$$g_2(x) = 0.11(x_1 - 2) - 0.11(x_2 + 2) + 0.33(|x_1 - 2| + |x_2 + 2|) - 1$$
$$g_3(x) = 0.11(x_1 - 16) - 0.11(x_2 + 2) + 0.33(|x_1 - 16| + |x_2 + 2|) - 1$$
$$g_4(x) = -0.15(x_1 + 0.5) + 0.27(x_2 - 2) + 0.81(|x_1 + 0.5| + |x_2 - 2|) - 3.79$$

It can be directly checked that $g_l(x) < 0, \forall x \in \tilde{A}_l$, $g_l(y) > 0, \forall y \in B$ which implies, $g(x) < 0, \forall x \in A$ and $g(y) > 0, \forall y \in B$, where $g(x) = \min\{g_1(x), g_2(x), g_3(x), g_4(x)\}$. The sets A and B and graphs of functions g_1, g_2, g_3 and g_4 are presented in figure 4. This example and obtained solutions explain that a more efficient separation can be realized by using the modified PCF algorithm.

It follows from this example that the PCF algorithm is able to construct separation functions without any conditions on the sets under consideration with 100 percent accuracy. Figure 1 presents geometrical interpretations for four separation types such as linear, h-polyhedral, max–min separation and PCF separation.

3. Application in classification problems

In this section the PCF algorithm developed in the previous section is applied to data classification problems. A data classification is the supervised assignment of data points to predefined and known classes. Classification problems have three common characteristics [9, 10]:

- learning is supervised;
- the dependent variable is categorical; and
- the emphasis is on building models able to assign new instances to one of a set of well-defined classes.

Supervised data classification problems have wide variety of application areas from banking to spam categorization and from the diagnosis of disease to chemistry. The aim of supervised data classification is to establish rules for the classification of some observations assuming that the classes of data are known. To find these rules, known training subsets of the given classes are used. During the last few decades many algorithms have been proposed and studied to solve data classification problems. These algorithms are mainly based on statistical, machine learning and neural networks approaches.

One promising approach to data classification problems is based on mathematical programming techniques. There are two main approaches to apply mathematical programming techniques for solving supervised data classification problems. The first is an outer approach and is based on the separation of the given training sets by means of a certain, not necessarily linear, function. The second is an inner approach. In this approach the given training sets are approximated by cluster centers. The new data vectors are assigned to the closest cluster and correspondingly to the set which contains this cluster, Bagirov *et al.* [11, 12].

The PCF algorithm presented in this paper is based on an outer approach. It is assumed that the data set under consideration contains two classes that are two nonempty finite disjoint point sets A and B in $\mathbb{R}^n$.

3.1 *Numerical results*

In this section we present results of numerical experiments with some real-world data sets. The data sets used are the Wisconsin Breast Cancer Diagnosis (WBCD), the Wisconsin Breast Cancer Prognosis (WBCP), the Cleveland Heart Disease (Heart), the Pima Indians Diabetes (Diabetes), the BUPA Liver Disorders (Liver) and the Ionosphere. All data sets contain two classes. The description of these data sets can be found in [13].

The PCF algorithm has been implemented by using different GAMS solvers on a notebook with Intel Centrino 1.6 GHz processor. First we consider entire data sets and check their separability applying the new algorithm. Results of these experiments are presented in table 2.

Table 2. Results of numerical experiments for entire data sets using linear, h-polyhedral, max–min, PCF separation methods.

					h-Polyhedral		Max–min		PCF separation	
Data set	m	p	n	Linear	h	acc.	$r \times j$	acc.	acc.	Time (s)
1 Liver	145	200	6	68.41	12	74.20	10×2	87.83	100	0.440
2 WBCD	239	444	9	97.36	7	98.98	5×2	100	100	0.097
3 WBCP	46	148	32	76.80	4	100	3×2	100	100	0.117
4 Ionosphere	126	225	34	93.73	4	97.44	2×2	100	100	0.089
5 Heart	137	160	13	84.19	10	100	2×5	100	100	0.792
6 Diabetes	268	500	8	76.95	12	80.60	7×4	89.45	100	1.335

This table contains also results of numerical experiments made by Bagirov [6], for the same data sets obtained using the h-polyhedral and max–min separabilities. In this table the following notation is used: m is the number of instances in the first class, p is the number of instances in the second class, n is the number of attributes, h is the number of hyperplanes in h-polyhedral separability, $(r \times j)$ is the number of hyperplanes in max–min separability. The accuracy is defined as the ratio between the number of well-classified points of both A and B and the total number of points in the whole data set. As was pointed out in [5] and [6], the h-polyhedral and max–min separation algorithms require significantly greater computational efforts than algorithms based on linear programming techniques. As can be seen from table 2, the PCF algorithm allows classification of any data set with 100% accuracy in the shortest time.

3.2 *Tenfold cross-validation*

In this subsection we test the validity of the classification functions obtained using the PCF algorithm. For this purpose we perform tenfold cross-validation tests.

When the PCF algorithm is applied to obtain a classification function by using the K-fold cross-validation, the whole data set is divided to K subsets or parts with an almost equal number of elements. To generate each training/validation set pair, we keep one of the K parts out as the validation set, and combine the remaining K–1 parts to form the training set. Doing this K times, each time leaving out another one of the K parts, we get K pairs. One of these subsets is sequentially used as a test set for calculating the accuracy of the function constructed by applying the PCF algorithm on the remaining part of the whole set which is called a training set [14].

1. $A^1 \cup B^1$ and $(A^2 \cup B^2) \cup (A^3 \cup B^3) \cup \cdots \cup (A^K \cup B^K)$;
2. $A^2 \cup B^2$ and $(A^1 \cup B^1) \cup (A^3 \cup B^3) \cup \cdots \cup (A^K \cup B^K)$;

$\vdots$

K. $A^K \cup B^K$ and $(A^1 \cup B^1) \cup (A^2 \cup B^2) \cup \cdots \cup (A^{K-1} \cup B^{K-1})$.

Then the overall accuracy is calculated as the ratio between the number of well-classified points of both A and B and the data set size. The overall classification function for every part is defined as a pointwise minimum of all functions g_l, where l varies between 1 and L, and L is a number of iterations required for obtaining $y_l = 0$ at Step 1 of the PCA (see (7) and (8)). Thus the function $g(x)$ defined by (8) becomes a function with maximum classification accuracy for training set.

Table 3. Tenfold cross-validation results obtained using the PCF algorithm.

Data sets	Training accuracy	Test accuracy	Time	Average number of LP solved
1 Liver	100	68.40	16.84	114
2 WBCD	100	100	0.58	1
3 WBCP	100	75.77	4.44	18
4 Ionosphere	100	88.03	0.84	5
5 Heart	100	79.12	10.23	74
6 Diabetes	100	71.48	44.17	216

Table 4. Tenfold cross-validation results obtained using modified PCF algorithm.

Data sets	Training accuracy	Test accuracy	Average number of LP solved
1 Liver	100	77.97	72
3 WBCP	100	86.21	12
4 Ionosphere	100	95.73	3
5 Heart	100	88.89	27
6 Diabetes	100	80.47	124

The tenfold cross-validation results are presented in table 3.

By using the modified PCF algorithm, the tenfold cross-validation has been performed for Heart, Ionosphere, WBCP, Diabetes, Liver data sets and results are presented in table 4. Since the PCF algorithm solves the classification problem for the data set WBCD with 100% accuracy in a single iteration, the modified algorithm was not applied to this dataset. As can be seen from the results presented in table 4, the test accuracy obtained by using the modified PCF algorithm for all the data sets is essentially increased and the average number of LP solved for each data set is diminished.

4. Conclusions

A simple algorithm for separating the given two finite point disjoint sets A and B in the n-dimensional space is suggested. By solving a linear subproblem, this algorithm generates a so-called polyhedral conic function which separates a certain part of the set A from B at each iteration. These functions are defined as an augmented l_1 norm – with a linear part added, whose graph is a polyhedral cone. A sublevel set of such a function includes at the utmost an intersection of 2^n half-spaces. A final separating function is defined as a point-wise minimum of these functions. It has been shown that algorithm terminates in a finite number of iterations and arbitrary two finite point disjoint sets can be separated by using this algorithm. An application on some illustrative example and on classification problems with some real-world benchmark data sets has been implemented. The obtained results demonstrate a high performance of the presented separation method from both the separation accuracy and running time viewpoints. However, the large differences between accuracies on training and test sets show that the PCF algorithm may not generalize well due to over-fitting the classification problem. The analysis done in the paper gives motivation for future research on improving the classification quality by using the new separation method in Global Tree Optimization approach and support vector machine classifiers and in feature selection problems.

Acknowledgements

The authors thank Alex Rubinov and Adil Bagirov for discussions. Their valuable comments were very helpful in improving the quality of the manuscript.

References

[1] Bennett, K.P. and Mangasarian, O.L., 1992, Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, **1**, 23–34.
[2] Chen, C. and Mangasarian, O.L., 1995, Hybrid misclassification minimization. Mathematical Programming Technical Report, 95–05, University of Wisconsin.
[3] Bennett, K.P. and Bredersteiner, E.J., 1997, A parametric optimization method for machine learning. *INFORMS Journal on Computing*, **9**, 311–318.
[4] Bradley, P.S., Fayyad, U.M. and Mangasarian, O.L., 1999, Data mining: overview and optimization opportunities. *INFORMS Journal on Computing*, **11**, 217–238.
[5] Astorino, A. and Gaudioso, M., 2002, Polyhedral separability through successive LP. *Journal of Optimization Theory and Applications*, **112**(2), 265–293.
[6] Bagirov, A.M., 2005, Max–min separability. *Optimization Methods and Software*, **20**(2–3), 271–290.
[7] Megiddo, N., 1988, On the complexity of polyhedral separability. *Discrete and Computational Geometry*, **3**, 325–337.
[8] Gasimov, R.N., 2001, Characterization of the Benson proper efficiency and scalarization in nonconvex vector optimization. In: Murat Koksalan and Stanley Zionts (Eds) *Multiple Criteria Decision Making in the New Millennium*, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag Berlin Heidelberg, **507**, 189–198.
[9] Bagirov, A.M., Rubinov, A.M., Soukhoroukova, N.V. and Yearwood, J., 2003, Unsupervised and supervised data classification via nonsmooth and global optimization. *Top*, **11**(1), 1–93.
[10] Roiger, R.J. and Geatz, M.W., 2003, *Data Mining: A Tutorial-based Primer* (New York: Addison Wesley).
[11] Bagirov, A.M., Rubinov, A.M. and Yearwood, J., 2001, Using global optimization to improve classification for medical diagnosis and prognosis. *Topics in Health Information Management*, **22**, 65–74.
[12] Bagirov, A.M., Rubinov, A.M. and Yearwood, J., 2002, A global optimization approach to classification. *Optimization and Engineering*, **3**(2), 129–155.
[13] Murphy, P.M. and Aha, D.W., 1992, UCI repository of machine learning databases. Technical report, Department of Information and Computer Science, University of California, Irvine. Available online at: www.ics.uci.edu/mlearn/MLRepository.html.
[14] Alpaydin, E., 2005, *Introduction to Machine Learning* (Massachusetts: MIT Press).