Noname manuscript No.
(will be inserted by the editor)

Polyhedral Separation via Difference of Convex (DC) Programming

Annabella Astorino ·
Massimo Di Francesco* ·
Manlio Gaudioso ·
Enrico Gorgone ·
Benedetto Manca

Received: date / Accepted: date

Abstract We consider polyhedral separation of sets as a possible tool in supervised classification. In particular we focus on the optimization model introduced by Astorino and Gaudioso [5] and adopt its reformulation in Difference of Convex (DC) form. We tackle the problem by adapting the algorithm for DC programming known as DCA. We present the results of the implementation of DCA on a number of benchmark classification datasets.

Keywords Classification · Machine Learning · DC optimization

1 Introduction

The classification of an object is a decision making process whose outcome is the assignment of a specific class membership to the object under observation. Medical diagnosis [21], chemistry [18], cybersecurity [1], image processing [19] are only some of the possible application areas of classification.

A. Astorino
ICAR - National Research Council, Rende, Italy
E-mail: annabella.astorino@icar.cnr.it

M. Di Francesco
*corresponding author
Dipartimento di Matematica e Informatica, Università di Cagliari, Cagliari, Italy
E-mail: mdifrance@unica.it

M. Gaudioso
ICAR - National Research Council, Rende, Italy
E-mail: gaudioso@dimes.unical.it

E. Gorgone
Dipartimento di Matematica e Informatica, Università di Cagliari, Cagliari, Italy
E-mail: egorgone@unica.it

B. Manca
Dipartimento di Matematica e Informatica, Università di Cagliari, Cagliari, Italy
E-mail: bmanca@unica.it

Each object (*sample*) is characterized by a finite number of (quantitative and/or qualitative) attributes, usually referred to as the *features*.

Construction of a classifier is a supervised learning activity, where a dataset of samples, whose class membership is known in advance, is given as the input. The objective is to gather a mathematical model capable to correctly classify newly incoming samples whose class membership is, instead, unknown.

Classification deals, mainly, with separation of sets of samples in the feature space, which is assumed to be $\mathbb{R}^n$. Whenever classes are two, we are faced with a *binary* classification problem. In this paper, in fact, the training dataset is partitioned into two subsets, say $\mathcal{A}$ and $\mathcal{B}$, thus the problem consists in finding a separation surface, if any, between them.

Most of the models rely on setting an appropriate optimization problem whose output is either a separating surface or a nearly-separating one, resulting in the minimization of some measure of the classification error.

Starting from the pioneering works by Mangasarian [10] and Vapnik [31], the hyperplane has been considered as the election surface to be looked for, although the use of nonlinear separation surfaces has been pursued too [26], [25], [7], [6].

The literature on classification is huge. We cite [31], [13], [27], [28] as basic references in Support Vector Machine (SVM) framework and [30] as a recent approach in the Deep Learning.

It is well known (see, e.g., [13]) that, if the convex hulls of the two sets $\mathcal{A}$ and $\mathcal{B}$ do not intersect, there exists a separating hyperplane such that the set $\mathcal{A}$ is on one side of such hyperplane and $\mathcal{B}$ is on the other side. It can be calculated by Linear Programming [10] and the two sets are referred to as *linearly separable*. On the other hand, if $\text{conv}(\mathcal{A}) \cap \text{conv}(\mathcal{B}) \neq \emptyset$, a number of algorithms can be adopted to determine a quasi-separating hyperplane such that the related error functions are minimized (see for example [20]).

In this paper we deal with binary classification based on the use of a polyhedral surface. The concept of polyhedral separability was introduced in [22] and applied within the classification framework in [5], [2].

Whenever, in fact, the two sets $\mathcal{A}$ and $\mathcal{B}$ are not linearly separable, it is possible to resort to polyhedral separation, that is to determine $h > 1$ hyperplanes such that $\mathcal{A}$ is in the convex polyhedron given by the intersection of h half-spaces and $\mathcal{B}$ lies outside such polyhedron.

In [5] an optimization model was proposed to calculate a set of h hyperplanes generating a polyhedral separation, whenever possible, for the sets $\mathcal{A}$ and $\mathcal{B}$. The model consists, as usual in classification, in minimizing an error function to cope with the case when the two sets are not h-polyhedrallly separable. Parallel to SVM, the model was extended in [2] to accommodate for margin maximization.

The error function adopted in [5] is neither convex nor concave and it was dealt with by means of successive linearizations.

In this paper we focus on the numerical treatment of the optimization problem to be solved in order to get a polyhedral separation surface. In particular we fully exploit the DC (Difference of Convex) [16] nature of the objective function and thus, differently from [5], we adopt an algorithm designed to treat DC functions. In fact, the literature provides a wide set of efficient algorithms in this area nowadays.

In [24, 23] an iterative algorithm was introduced to minimize functions of the form $f = f_1 - f_2$, with g and h convex functions. The algorithm, called DCA, considers at iteration t the linearization $f_2^{(t)}$ of function f_2 at point x^t and determines the next iterate x^{t+1} as an optimal solution of the convex problem

$$\min_x f_1(x) - f_2^{(t)}(x) \tag{1}$$

DCA has proven to be an efficient method to tackle DC problems, even nonsmooth and different artificial intelligence problems have been approached by means of the DCA [3, 4, 8, 9, 19].

Several other methods have been more recently proposed in the literature (see [14], [17]) which allow to solve large scale DC programs too.

In this paper we build on the model [5] and adopt a decomposition of the error function for the h-polyhedral separability problem as the difference of two convex functions. We then apply the DCA to carry out an extensive experimentation on several classes of benchmark instances.

The paper is organized as follows. In Section 2 we describe the h-polyhedral classification model and its reformulation as a Difference of Convex optimization problem. In Section 3 we describe how the DCA has been adapted to the DC reformulations. In Section 4 we present the results of our implementation on a number of benchmark classification problems. Some conclusions are drawn in Section 5

2 The polyhedral separability model

Let $\mathcal{A} = \{a_1, \ldots, a_m\}$ and $\mathcal{B} = \{b_1, \ldots, b_k\}$ be two finite sets of $\mathbb{R}^n$.

Definition 1 The sets $\mathcal{A}$ and $\mathcal{B}$ are *h-polyhedrally separable* if there exist a set of h hyperplanes $\{(v^j, \eta^j)\}$, $v^j \in \mathbb{R}^n$, $\eta^j \in \mathbb{R}$, $j = 1, \ldots, h$, such that

$$\begin{array}{ll} a_i^T v^j < \eta^j & \forall i = 1, \ldots, m, j = 1, \ldots, h \\ b_l^T v^j > \eta^j & \forall l = 1, \ldots, k, \text{and at least one } j \in \{1, \ldots, h\} \end{array} \tag{2}$$

The following proposition gives an equivalent characterization of h-polyhedral separability:

Proposition 1 *The sets $\mathcal{A}$ and $\mathcal{B}$ are h-polyhedrally separable if and only if there exist h hyperplanes $\{(w^j, \gamma^j)\}$ such that*

$$\begin{array}{ll} a_i^T w^j \le \gamma^j - 1 & \forall i = 1, \ldots, m, j = 1, \ldots, h \\ b_l^T w^j \ge \gamma^j + 1 & \forall l = 1, \ldots, k, \textit{and at least one } j \in \{1, \ldots, h\}. \end{array} \tag{3}$$

Proof [5, Proposition 2.1]

Moreover, in [5, Proposition 2.2] it is proven that a necessary and sufficient condition for the sets $\mathcal{A}$ and $\mathcal{B}$ to be h-polyhedrally separable (for some $h \leq |\mathcal{B}|$) is given by

$$\text{conv}(\mathcal{A}) \cap \mathcal{B} = \emptyset. \tag{4}$$

Remark 1 *The roles of $\mathcal{A}$ and $\mathcal{B}$ in (4) are not symmetric.*

According to Proposition 1, a point $a_i \in \mathcal{A}$ is well classified by the set of hyperplanes $\{w^j, \gamma^j\}$ if $a_i^T w^j - \gamma^j + 1 \leq 0$ for all $j = 1, \dots, h$. Therefore, we can compute the classification error of the point a_i with respect to $\{w^j, \gamma^j\}$ as

$$\max_{1 \leq j \leq h} \{0, a_i^T w^j - \gamma^j + 1\}. \tag{5}$$

Analogously, if $\min_{1 \leq j \leq h}\{-b_l^T w^j + \gamma^j + 1\} \leq 0$, a point $b_l \in \mathcal{B}$ is well classified by the set $\{w^j, \gamma^j\}$. Thus, the error of classification of the point b_l is

$$\max\{0, \min_{1 \leq j \leq h} \{-b_l^T w^j + \gamma^j + 1\}\}. \tag{6}$$

Given a set of h hyperplanes $\{w^j, \gamma^j\}$ we denote with $W = [w^1 : \cdots : w^h]$ the matrix whose j-th column is the vector w^j and with $\Gamma = (\gamma^1, \dots, \gamma^h)$ the vector whose components are the γ^j's. The classification error function for the h-polyhedral separability problem for the sets $\mathcal{A}$ and $\mathcal{B}$, with respect to the hyperplanes $\{w^j, \gamma^j\}$, is then given by

$$e(W, \Gamma) := e_1(W, \Gamma) + e_2(W, \Gamma), \tag{7}$$

where

$$e_1(W, \Gamma) := \frac{1}{m} \sum_{i=1}^{m} \max_{1 \leq j \leq h} \{\max\{0, a_i^T w^j - \gamma^j + 1\}\} \tag{8}$$

and

$$e_2(W, \Gamma) := \frac{1}{k} \sum_{l=1}^{k} \max\{0, \min_{1 \leq j \leq h} \{-b_l^T w^j + \gamma^j + 1\}\} \tag{9}$$

represent the errors for points of $\mathcal{A}$ and $\mathcal{B}$, respectively. Function $e(W, \Gamma)$ is non negative and piecewise affine; $e_1(W, \Gamma)$ is convex and $e_2(W, \Gamma)$ is quasiconcave; moreover in [5] it has also been proven that the sets $\mathcal{A}$ and $\mathcal{B}$ are h-polyhedrally separable if and only if there exists a set of h hyperplanes (W^*, Γ^*) such that $e(W^*, \Gamma^*) = 0$ and, in that case, $w^j = 0$ for all $j = 1, \dots, h$ cannot be the optimal solution.

In [5] the problem of minimizing the error function (7) is tackled by solving, at each iteration, a linear program providing a descent direction. Here, instead, we rewrite $e(W, \Gamma)$ as difference of convex functions and then we address its minimization through ad hoc DC techniques.

To obtain such reformulation it is useful the following identity, valid for any set of h affine functions $z^j(x)$, $j = 1, \dots, h$:

$$\begin{aligned} \max(0, \min_j z^j(x)) &= \max(0, -\max_j(-z^j(x))) \\ &= \max(0, \max_j(-z^j(x))) - \max_j(-z^j(x)). \end{aligned} \tag{10}$$

By applying (10) to $e_2(W, \mathit{\Gamma})$ we obtain the following DC decomposition of $e(W, \mathit{\Gamma})$:

$$e(W, \mathit{\Gamma}) = \hat{e}_1(W, \mathit{\Gamma}) - \hat{e}_2(W, \mathit{\Gamma}), \tag{11}$$

where both

$$\hat{e}_1(W, \mathit{\Gamma}) = e_1(W, \mathit{\Gamma}) + \frac{1}{k} \sum_{l=1}^{k} \max[0, \max_j (b_l^T w^j - \gamma^j - 1)]\} \tag{12}$$

and

$$\hat{e}_2(W, \mathit{\Gamma}) = \frac{1}{k} \sum_{l=1}^{k} \{\max_j (b_l^T w^j - \gamma^j - 1)\} \tag{13}$$

are convex.

The DC decomposition (11) has been already discussed in [29], where the authors suggested an algorithm that combines a local and a global search in order to find a global minimum of the error function. In the numerical experience we are going to discuss in the next Sections we confine ourselves to find just local minima of the error functions involved.

3 Exploiting the function structure in the DCA implementation

We have applied DCA to the minimization of the error function (11). Before discussing our experiment setting, we describe how we have adapted DCA to deal with polyhedral separation applied to a number of datasets from the classification literature.

At iteration t, in any possible configuration $(W_t, \mathit{\Gamma}_t)$ of the h hyperplanes we can calculate for each $l = 1, \dots, k$ the index j_l where the maximum in (13) is achieved:

$$j_l = \arg \max_{1 \le j \le h} (b_l^T w_{(t)}^j - \gamma_{(t)}^j - 1), \quad l = 1, \dots, k \tag{14}$$

and we define consequently the linearization of function $\hat{e}_2$ at iteration t:

$$\hat{e}_2^t(W, \mathit{\Gamma}) := \frac{1}{k} \sum_{l=1}^{k} (b_l^T w^{j_l} - \gamma^{j_l} - 1) \tag{15}$$

which satisfies $\hat{e}_2^t(W, \mathit{\Gamma}) \le \hat{e}_2(W, \mathit{\Gamma})$.

Then, we consider the convexification of the original DC function:

$$e^t(W, \mathit{\Gamma}) = \hat{e}_1(W, \mathit{\Gamma}) - \hat{e}_2^t(W, \mathit{\Gamma}), \tag{16}$$

so that next configuration $(W_{t+1}, \mathit{\Gamma}_{t+1})$ is obtained by solving the convex program

$$\min_{W, \mathit{\Gamma}} e^t(W, \mathit{\Gamma}),$$

which in turn can be put in form of the following linear program, thanks to the introduction of the additional variables ξ_i, $i = 1, \dots, m$ and ζ_l, $l = 1, \dots, k$:

$$
\begin{array}{rlll}
(W_{t+1}, \mathit{\Gamma}_{t+1}) = \arg\min & \multicolumn{3}{l}{\dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \xi_i + \dfrac{1}{k}\displaystyle\sum_{l=1}^{k} (\zeta_l - b_l^T w^{j_l} + \gamma^{j_l} + 1)} \\
& \xi_i \geq 0 & i = 1, \dots, m & \\
& \xi_i \geq a_i^T w^j - \gamma^j + 1, & j = 1, \dots, h, & i = 1, \dots, m \\
& \zeta_l \geq 0 & l = 1, \dots, k & \\
& \zeta_l \geq b_l^T w^j - \gamma^j - 1, & j = 1, \dots, h, & l = 1, \dots, k.
\end{array}
\qquad (17)
$$

Summing up, the DCA based algorithm for the minimization of (11) can be stated as follows:

```
0. Choose $W_0 \in \mathbb{R}^{n \times h}, \mathit{\Gamma}_0 \in \mathbb{R}^h$ and a tolerance $\epsilon$. Set $t = 0$;
1. Set $g_t \in \partial \hat{e}_2^t(W_t, \mathit{\Gamma}_t)$;
2. Set $(W_{t+1}, \gamma_{t+1})$ as a solution of (17);
3. If $|e(W_{t+1}, \mathit{\Gamma} t + 1) - e(W_t, \mathit{\Gamma} t)| \leq \epsilon$ terminate.
   Otherwise, increase $t$ by 1 and return to step 1.
```

The above algorithm is a descent method. It is easy to verify that, if $e(W_{t+1}, \mathit{\Gamma}_{t+1}) = e(W_t, \mathit{\Gamma}_t)$, then $(W_t, \mathit{\Gamma}_t)$ is a critical point, i.e.

$$\partial \hat{e}_1(W_t, \mathit{\Gamma}_t) \cap \partial \hat{e}_2(W_t, \mathit{\Gamma}_t) \neq \emptyset\,.$$

Hence, this result provides the stopping criterion at step 3. For more theoretical details and the convergence theorem see [23,24].

Following the SVM paradigm, aimed at obtaining a good generalization capability, we have added to $\hat{e}_1(W, \mathit{\Gamma})$ in (11) the margin term:

$$\frac{1}{2} \sum_{j=1}^{h} \|w^j\|^2, \qquad (18)$$

thus coming out with the following DC model:

$$\bar{e}(W, \mathit{\Gamma}) = \left(\frac{C}{2} \sum_{j=1}^{h} \|w^j\|^2 + \hat{e}_1(W, \mathit{\Gamma}) \right) - \hat{e}_2(W, \mathit{\Gamma}), \qquad (19)$$

where $C > 0$ is a tradeoff parameter between the two objectives of maximizing the margin and minimizing the classification error. The minimization of (19) can be addressed by DCA, too. In this case, at each iteration we have to solve a quadratic program that differs from the linear program (17) only for the quadratic margin term (18). Consequently, the algorithmic scheme is unchanged except for the step 2 where a quadratic program has to be solved.

4 Numerical experiments

In the numerical experiments we have implemented two DCA codes:

– h-PolSepDC, where we minimize (11) (the separation problem with no separation margin) by solving at each iteration the linear program (17);
– h-PolSepDC-QP, where we minimize (19) (a margin has been accounted for) by solving at each iteration a quadratic program.

In particular, we have chosen $h = 2$ according to the hyperparameter tuning performed in [5].

Moreover, since the role of the sets $\mathcal{A}$ and $\mathcal{B}$ is not symmetric in the definition of polyhedral separability, in the numerical experiments one has to define who is $\mathcal{A}$ and $\mathcal{B}$ in any dataset. So, we have called set $\mathcal{A}$ the one with less number of points, following, also for this issue, the rule adopted in [5].

We have used MATLAB R2015b calling CPLEX library, under a 2,6 GHz Intel Core i7 processor, on an OS X 10.12.6 operating system.

To evaluate the impact of the DC decomposition of the error function (7) with respect to the classic nonsmooth optimization approach, we have reimplemented, in MATLAB, the algorithm proposed in [5] (2-PolSep code). Finally, for sake of completeness we have also used the standard MATLAB SVM package to run the linear separability classification problem (SVM-LINEAR code).

We have considered several test problems drawn from the binary classification literature which are described in Table 1. In particular, all datasets are taken from the LIBSVM (LIBrary for Support Vector Machines) repository [11], except for g50c and g10n, which are described in [12].

Table 1 Datasets

#	Dataset	Space dimension	#Samples
1	Cancer	9	699
2	Diagnostic	30	569
3	Heart	13	297
4	Pima	8	769
5	Ionosphere	34	351
6	Sonar	60	208
7	Galaxy	14	4192
8	g50c	50	550
9	g10n	10	550
10	Mushrooms	22	8124
11	Prognosis	32	110
12	Tic Tac Toe	9	958
13	Votes	16	435
14	Letter-a	16	20000
15	a9a	123	1605

For all datasets we have performed a standard ten-fold cross-validation protocol and in Table 2 we summarize the LP/QP problems solved at each fold, in terms of number of variables and constraints.

Table 2 The number of variables/constraints

Dataset	# of vars	# of constrs
1	649	1887
2	574	1536
3	295	801
4	710	2076
5	386	948
6	309	561
7	3803	11319
8	597	1485
9	517	1485
10	7358	21936
11	165	297
12	882	2586
13	426	1176
14	18034	54000
15	1693	4335

For each approach, in the columns Train and Test of Table 3 we report the average percentage of training and testing correctness, respectively. The best results in terms of testing correctness have been underlined.

A preliminary tuning for the parameter C in 2-PolSepDC-QP and SVM-LINEAR codes has been performed and we have selected, for each dataset, that value optimizing the performance on the testing set.

Table 3 Training and testing correctness percentage

#	2-PolSep		2-PolSepDC		2-PolSepDC-QP		SVM-LINEAR	
	Train	Test	Train	Test	Train	Test	Train	Test
1	97.77	97.36	97.77	97.37	97.72	97.66	97.72	97.21
2	98.54	97.18	99.53	95.78	98.22	97.89	98.79	97.53
3	86.31	84.18	86.34	83.84	85.52	84.84	86.16	84.51
4	76.62	75.12	76.98	75.12	76.68	75.64	76.65	75.38
5	96.11	87.76	97.97	87.23	92.24	88.94	93.38	87.20
6	99.04	67.16	100.00	69.61	88.35	70.99	94.50	62.84
7	94.56	94.08	95.76	95.35	95.09	94.63	95.59	95.30
8	100.00	94.38	100.00	91.63	96.61	95.29	99.01	92.38
9	100.00	94.38	100.00	91.63	96.61	95.29	99.01	92.38
10	80.06	72.66	85.80	77.23	83.47	80.28	80.07	72.85
11	71.92	69.11	90.39	63.65	68.39	69.18	78.79	64.65
12	61.89	53.22	81.00	81.04	81.00	81.04	63.02	55.43
13	96.68	94.25	96.91	94.95	96.14	95.19	96.25	94.25
14	98.30	98.24	96.31	96.23	97.24	97.24	96.31	96.24
15	–	–	78.83	78.47	77.99	77.88	79.83	79.56

The numerical results indicate the good performance, in terms of correctness, of the DC-based approaches w.r.t. both the algorithm [5] and standard SVM. In particular, the DC model, equipped with margin maximization (Code

Table 4 CPU time (secs)

#	2-PolSep	2-PolSepDC	2-PolSepDC-QP	SVM-LINEAR
	Time	Time	Time	Time
1	0.82	0.09	1.39	0.05
2	2.13	0.46	1.16	0.12
3	0.48	0.02	0.19	0.22
4	4.74	0.17	1.62	0.32
5	1.21	0.18	1.41	0.31
6	1.41	0.05	0.47	0.28
7	266.47	54.25	19.00	1.81
8	4.61	0.42	0.85	0.43
9	4.63	0.41	0.82	0.42
10	303.69	5.93	7.06	4.82
11	0.13	0.02	0.08	0.15
12	2.90	0.13	1.73	0.25
13	0.18	0.08	0.95	0.11
14	696.62	287.96	573.72	29.22
15	–	954.32	381.12	1230.74

2-PolSepDC-QP) has provided the best testing correctness in 13 out of 15 datasets. By comparing 2-PolSepDC and 2-PolSepDC-QP we can observe that the addition of margin provides a better testing correctness except for galaxy and a9a datasets. On the contrary, 2-PolSepDC has a better performance in terms of training correctness except for Tic Tac Toe and letter-a datasets. This means that the classifier coming out from 2-PolSepDC-QP has a higher generalization capability.

As for computation time (see Table 4), the DC decomposition has been much more effective than 2-PolSep method. Moreover, the increase in computation time with respect to single-hyperplane separation model SVM-LINEAR has not been particularly severe.

2-PolSep and the 2-PolSepDC are different algorithms to solve the same problem, i.e. the minimization of (11). By comparing the objective function values obtained by the two codes starting from the same initial point, we note that the second approach provides better solutions but the difference is not so significant.

Since in the definition of polyhedral separability the role of the sets $\mathcal{A}$ and $\mathcal{B}$ is not symmetric, we compare the results also in terms of recall, specificity, precision and F1score (see Table 5). The trend of these key performance indicators confirms the goodness of the DC-based approaches.

For completeness, we have have launched both the codes h-PolSepDC and h-PolSepDC-QP with $h > 2$. The running time is not dramatically larger but the numerical experiments show that there is no significant improvement in terms of correctness. Even worse, in some cases the improvement of training performance is not accompanied with an improvement of testing one. This proves that a high value of h -number of hyperplanes- provides classifiers with

Table 5 Numerical Results

#	Code	Training set				Testing set			
		Rec.	Sp.	Pr.	F1s.	Rec.	Sp.	Pr.	F1s.
1	2-PolSep	0.98	0.97	0.95	0.97	0.98	0.97	0.95	0.96
	2-PolSepDC	0.99	0.97	0.95	0.97	0.98	0.97	0.95	0.96
	2-PolSepDC-QP	0.98	0.98	0.96	0.97	0.98	0.97	0.95	0.97
	SVM-LINEAR	0.98	0.97	0.95	0.97	0.97	0.97	0.95	0.96
2	2-PolSep	0.99	0.98	0.99	0.99	0.97	0.97	0.98	0.98
	2-PolSepDC	0.99	1.00	1.00	1.00	0.96	0.96	0.98	0.97
	2-PolSepDCQP	0.99	0.97	0.98	0.99	0.99	0.96	0.98	0.98
	SVM-LINEAR	0.99	0.98	0.99	0.99	0.98	0.97	0.98	0.98
3	2-PolSep	0.85	0.90	0.96	0.90	0.84	0.85	0.94	0.88
	2-PolSepDC	0.85	0.90	0.96	0.90	0.84	0.84	0.93	0.88
	2-PolSepDC-QP	0.85	0.88	0.95	0.89	0.84	0.87	0.94	0.89
	SVM-LINEAR	0.85	0.90	0.96	0.90	0.84	0.85	0.94	0.89
4	2-PolSep	0.73	0.79	0.65	0.68	0.69	0.79	0.64	0.66
	2-PolSepDC	0.73	0.79	0.65	0.69	0.69	0.78	0.63	0.66
	2-PolSepDC-QP	0.73	0.79	0.65	0.69	0.71	0.78	0.64	0.67
	SVM-LINEAR	0.72	0.79	0.65	0.68	0.69	0.79	0.64	0.66
5	2-PolSep	0.96	0.97	0.98	0.97	0.92	0.81	0.90	0.90
	2-PolSepDC	0.98	0.98	0.99	0.98	0.90	0.82	0.90	0.90
	2-PolSepDC-QP	0.96	0.86	0.92	0.94	0.96	0.75	0.88	0.92
	SVM-LINEAR	0.95	0.91	0.95	0.95	0.93	0.76	0.88	0.90
6	2-PolSep	1.00	0.99	0.98	0.99	0.64	0.70	0.66	0.64
	2-PolSepDC	1.00	1.00	1.00	0.75	1.00	0.65	0.68	0.70
	2-PolSepDC-QP	0.90	0.87	0.86	0.88	0.79	0.65	0.70	0.72
	SVM-LINEAR	0.96	0.94	0.93	0.94	0.63	0.63	0.61	0.61
7	2-PolSep	0.94	0.95	0.95	0.95	0.94	0.95	0.95	0.94
	2-PolSepDC	0.96	0.95	0.95	0.96	0.95	0.96	0.96	0.95
	2-PolSepDC-QP	0.95	0.95	0.95	0.95	0.94	0.95	0.95	0.94
	SVM-LINEAR	0.96	0.95	0.95	0.96	0.95	0.96	0.96	0.95
8	2-PolSep	1.00	1.00	1.00	0.93	1.00	0.95	0.95	0.94
	2-PolSepDC	1.00	1.00	1.00	0.92	1.00	0.92	0.92	0.92
	2-PolSepDC-QP	0.96	0.97	0.97	0.97	0.95	0.96	0.96	0.95
	SVM-LINEAR	0.99	0.99	0.99	0.99	0.93	0.91	0.92	0.92
9	2-PolSep	1.00	1.00	1.00	0.93	1.00	0.95	0.95	0.94
	2-PolSepDC	1.00	1.00	1.00	0.92	1.00	0.92	0.92	0.92
	2-PolSepDC-QP	0.96	0.97	0.97	0.97	0.95	0.96	0.96	0.95
	SVM-LINEAR	0.99	0.99	0.99	0.99	0.93	0.91	0.92	0.92
10	2-PolSep	0.75	0.85	0.85	0.80	0.64	0.82	0.83	0.66
	2-PolSepDC	0.83	0.89	0.89	0.86	0.76	0.79	0.84	0.76
	2-PolSepDC-QP	0.74	0.93	0.92	0.82	0.70	0.92	0.92	0.77
	SVM-LINEAR	0.75	0.85	0.85	0.80	0.65	0.82	0.83	0.66
11	2-PolSep	0.70	0.73	0.61	0.65	0.68	0.70	0.67	0.62
	2-PolSepDC	0.93	0.89	0.84	0.88	0.62	0.65	0.51	0.62
	2-PolSepDC-QP	0.66	0.70	0.57	0.61	0.68	0.70	0.64	0.62
	SVM-LINEAR	0.79	0.79	0.69	0.73	0.61	0.67	0.54	0.61
12	2-PolSep	0.62	0.62	0.76	0.68	0.49	0.62	0.69	0.56
	2-PolSepDC	0.71	1.00	1.00	0.83	0.71	1.00	1.00	0.82
	2-PolSepDC-QP	0.71	1.00	1.00	0.83	0.71	1.00	1.00	0.82
	SVM-LINEAR	0.63	0.64	0.77	0.69	0.51	0.64	0.72	0.58
13	2-PolSep	0.98	0.96	0.93	0.96	0.95	0.94	0.91	0.93
	2-PolSepDC	0.98	0.96	0.94	0.96	0.95	0.95	0.92	0.94
	2-PolSepDC-QP	0.99	0.94	0.92	0.95	0.98	0.94	0.91	0.94
	SVM-LINEAR	0.98	0.95	0.92	0.95	0.95	0.94	0.91	0.93
14	2-PolSep	0.96	0.95	1.00	0.98	0.96	0.95	1.00	0.98
	2-PolSepDC	0.96	0.95	1.00	0.98	0.96	0.95	1.00	0.98
	2-PolSepDC-QP	0.97	0.94	1.00	0.99	0.97	0.94	1.00	0.99
	SVM-LINEAR	0.96	0.95	1.00	0.98	0.96	0.95	1.00	0.98
15	2-PolSepDC	0.89	0.75	0.54	0.67	0.89	0.75	0.53	0.67
	2-PolSepDC-QP	0.89	0.74	0.53	0.66	0.89	0.74	0.52	0.66
	SVM-LINEAR	0.87	0.78	0.55	0.67	0.86	0.78	0.55	0.67

a smaller generalization capability. For instance, we report some results (see Tables 6-7).

5 Conclusions

We have adopted a Difference of Convex decomposition of the error function in polyhedral separation and have tackled the resulting optimization problem via DCA algorithm.

Table 6 Percentage of correctness with h-PolSepDC ($h \geq 2$)

#	$h = 2$		$h = 3$		$h = 5$		$h = 10$	
	Train	Test	Train	Test	Train	Test	Train	Test
1	97.77	97.37	97.91	97.07	97.91	97.22	97.91	97.22
2	99.53	95.78	99.92	94.55	99.92	94.39	99.92	94.39
3	86.34	83.84	86.34	83.84	–	–	–	–
4	76.98	75.12	74.86	73.43	–	–	74.83	73.56
5	97.97	87.23	97.63	87.49	100.00	86.33	100.00	87.76
6	100.00	69.61	100.00	70.61	100.00	70.61	100.00	70.61
7	95.76	95.35	95.76	95.35	–	–	95.74	95.30
8	100.00	94.03	100.00	89.46	100.00	89.46	–	–
9	100.00	94.03	100.00	89.46	–	–	–	–
10	85.80	77.23	86.00	77.38	86.93	77.97	86.93	77.97
11	90.39	63.65	90.39	64.65	90.39	64.65	90.39	64.65
12	81.00	81.04	81.00	81.04	–	–	–	–
13	96.91	94.95	96.99	94.49	97.09	94.49	97.09	94.49
14	96.31	96.23	96.31	96.23	–	–	–	–

Table 7 Percentage of correctness with h-PolSepDC-QP ($h \geq 2$)

#	$h = 2$		$h = 3$		$h = 5$		$h = 10$	
	Train	Test	Train	Test	Train	Test	Train	Test
1	97.72	97.66	97.72	97.65	–	–	–	–
2	98.22	97.89	98.26	97.89	–	–	–	–
3	85.52	84.84	85.52	84.84	–	–	–	–
4	76.68	75.64	75.39	73.69	76.53	74.22	76.79	73.43
5	92.24	88.94	93.26	88.35	93.19	88.10	–	–
6	88.35	70.99	88.56	71.01	–	–	–	–
7	95.59	95.30	95.07	94.58	–	–	–	–
8	96.61	95.29	96.61	95.29	–	–	–	–
9	96.61	95.29	96.61	95.29	–	–	–	–
10	83.47	80.28	83.47	80.28	–	–	–	–
11	68.39	69.18	68.39	69.18	68.39	69.18	68.39	69.18
12	81.00	81.04	81.00	81.04	–	–	–	–
13	96.14	95.19	96.09	95.19	96.07	95.19	–	–
14	97.24	97.24	97.21	97.11	–	–	–	–

The numerical results we have obtained demonstrate the good performance of the approach both in terms of classification correctness and computation time.

Future research would investigate the integration between feature selection [15] and polyhedral separation aimed at detecting a possibly smaller subsets of significant attributes in terms of classification correctness.

Compliance with ethical standards

Funding: The fifth author was supported by KASBA - funded by *Regione Autonoma della Sardegna*. This research was also Supported by *Fondazione di Sardegna* through the project *Algorithms for Approximation with Applications*.

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Astorino, A., Chiarello, A., Gaudioso, M., Piccolo, A.: Malicious URL detection via spherical classification. Neural Computing and Applications **28**, 699–705 (2017)
2. Astorino, A., Fuduli, A.: Support vector machine polyhedral separability in semisupervised learning. Journal of Optimization Theory and Applications **164**(3), 1039–1050 (2015)
3. Astorino, A., Fuduli, A., Gaudioso, M.: DC models for spherical separation. Journal of Global Optimization **48**(4), 657–669 (2010)
4. Astorino, A., Fuduli, A., Gaudioso, M.: Margin maximization in spherical separation. Computational Optimization and Applications **53**(2), 301–322 (2012)
5. Astorino, A., Gaudioso, M.: Polyhedral separability through successive LP. Journal of Optimization theory and applications **112**(2), 265–293 (2002)
6. Astorino, A., Gaudioso, M.: Ellipsoidal separation for classification problems. Optimization Methods and Software **20**(2-3), 261–270 (2005)
7. Astorino, A., Gaudioso, M.: A fixed-center spherical separation algorithm with kernel transformations for classification problems. Computational Management Science **6**(3), 357–372 (2009)
8. Astorino, A., Gaudioso, M., Seeger, A.: Conic separation of finite sets. I. The homogeneous case. Journal of Convex Analysis **21**(1), 1–28 (2014)
9. Astorino, A., Gaudioso, M., Seeger, A.: Conic separation of finite sets. II. The nonhomogeneous case. Journal of Convex Analysis **21**(3), 819–831 (2014)
10. Bennett, K., Mangasarian, O.: Robust linear programming discrimination of two linearly inseparable sets. Optimization methods and software **1**(1), 23–34 (1992)
11. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27 (2011)
12. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pp. 57–64 (2005)
13. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press (2000)
14. Gaudioso, M., Giallombardo, G., Miglionico, G., Bagirov, A.: Minimizing nonsmooth DC functions via successive DC piecewise-affine approximations. Journal of Global Optimization **71**(1), 37–55 (2018)
15. Gaudioso, M., Gorgone, E., Labbe, M., Rodriguez-Chia, A.: Lagrangian relaxation for svm feature selection. Computers and Operations Research **87**, 137–145 (2017)
16. Hiriart-Urruty, J.B.: Generalized differentiability duality and optimization for problems dealing with differences of convex functions. Lecture Notes in Economic and Mathematical Systems **256**, 37–70 (1986)
17. Joki, K., Bagirov, A., Karmitsa, N., Makela, M.: A proximal bundle method for nonsmooth DC optimization utilizing nonconvex cutting planes. Journal of Global Optimization **68**(3), 501–535 (2017)

18. Jurs, P.: Pattern recognition used to investigate multivariate data in analytical chemistry. Science **232**(4755), 1219–1224 (1986)
19. Khalaf, W., Astorino, A., D'Alessandro, P., Gaudioso, M.: A DC optimization-based clustering technique for edge detection. Optimization Letters **11**(3), 627–640 (2017)
20. Mangasarian, O.: Arbitrary-norm separating plane. Operations Research Letters **24**(1-2), 15–23 (1999)
21. Mangasarian, O., Street, W., Wolberg, W.: Breast cancer diagnosis and prognosis via linear programming. Operations Research **43**(4), 570–577 (1995)
22. Megiddo, N.: On the complexity of polyhedral separability. Discrete & Computational Geometry **3**(4), 325–337 (1988)
23. Pham Dinh, T., Le Thi Hoai, A.: Recent advances in DC programming and DCA. In: Transactions on computational intelligence XIII, pp. 1–37 (2014)
24. Pham Dinh, T., et al.: The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals of Operations Research **133**(1-4), 23–46 (2005)
25. Plastria, F., Carrizosa, E., Gordillo, J.: Multi-instance classification through spherical separation and VNS. Computers and Operations Research **52**, 326–333 (2014)
26. Rosen, J.B.: Pattern separation by convex programming. Journal of Mathematical Analysis and Applications **10**, 123–134 (1965)
27. Schölkopf, B., Burges, C., Smola, A.: Advances in kernel methods. Support vector learning. MIT Press, Cambridge, MA (1999)
28. Sra, S., Nowozin, S., Wright, S.: Optimization for Machine Learning. The MIT Press (2011)
29. Strekalovsky, A., Gruzdeva, T., Orlov, A.: On the problem polyhedral separability: A numerical solution. Automation and Remote Control **76**(10), 1803–1816 (2015)
30. Thongsuwan, S., Jaiyen, S., Padcharoen, A., Agarwal, P.: Convxgb: A new deep learning model for classification problems based on cnn and xgboost. Nuclear Engineering and Technology (2020)
31. Vapnik, V.: The nature of the statistical learning theory. Springer Verlag, New York (1995)