J Glob Optim (2010) 48:657–669
DOI 10.1007/s10898-010-9558-0

DC models for spherical separation

A. Astorino · A. Fuduli · M. Gaudioso

Received: 28 April 2009 / Accepted: 21 May 2010 / Published online: 8 June 2010

Abstract We propose two different approaches for spherical separation of two sets. Both methods are based on minimizing appropriate nonconvex nondifferentiable error functions, which can be both expressed in a DC (Difference of two Convex) form. We tackle the problem by adopting the DC-Algorithm. Some numerical results on classical binary datasets are reported.

Keywords Spherical separation · DC functions · DCA

1 Introduction

The objective of pattern classification is to decide, on the basis of appropriate attributes, whether an object belongs to exactly one among several classes. Such a general definition has allowed to propose diverse mathematical formulations for classification purposes with application in many fields. Some well known examples are object recognition in machine vision, gene expression profile analysis, DNA and protein analysis and many others.

From the mathematical point of view, classification reduces to finding separation surfaces in the sample space, where the objects are represented through their attributes. Consequently

A. Astorino
Istituto di Calcolo e Reti ad Alte Prestazioni C.N.R., c/o Dipartimento di Elettronica Informatica e Sistemistica, Università della Calabria, 87036 Rende (CS), Italy
e-mail: astorino@icar.cnr.it

A. Fuduli
Dipartimento di Matematica, Università della Calabria, 87036 Rende (CS), Italy
e-mail: antonio.fuduli@unical.it

M. Gaudioso (✉)
Dipartimento di Elettronica Informatica e Sistemistica, Università della Calabria, 87036 Rende (CS), Italy
e-mail: gaudioso@deis.unical.it

convexity and separation properties of sets (e.g. theorems such as Hahn-Banach's) play a relevant role.

In practical cases, since it is not always easy to check whether the theoretical conditions of existence of such surfaces hold, it is necessary to tackle the classification-separation problems by means of tools which are able to provide a quantitative answer even when the sets are not separable. In this case, it is possible to adopt a numerical optimization model to minimize some misclassification measure. Among such approaches we recall the seminal contributions by Rosen [12] and Mangasarian [9], and the fundamental breakthrough due to the introduction of the kernel based approach in the framework of the Support Vector Machine (SVM) method [6,14–16,20].

In particular, the basic idea of SVM for classification problems is to map the data into a higher dimensional space (the feature space) and to separate the transformed sets by means of a hyperplane. Such a transformation allows to obtain general nonlinear separation surfaces in the original input space (see [14–16] for an extensive treatment of the subject and [8] and [10] for some effective variants).

It is possible, however, to look for nonlinear separation surfaces directly in the input space (this is the case of the polyhedral separation [2] and of the ellipsoidal separation [3]), or even in the feature space, as in the fixed-center spherical separation approach described in [4].

Our approach deals with spherical binary classification (see also [4,18,19] for some related issues), where two nonempty and disjoint finite sets of sample points in the n-dimensional space $\mathbb{R}^n$, say $\mathcal{A} = \{a_1, \ldots, a_m\}$ and $\mathcal{B} = \{b_1, \ldots, b_k\}$, are given. The objective is to find a sphere separating the set $\mathcal{A}$ from the set $\mathcal{B}$, i.e. a sphere enclosing all points of $\mathcal{A}$ and no points of $\mathcal{B}$. Throughout the paper we indicate by $\|\cdot\|$ the Euclidean norm and by $a^T b$ the inner product of the vectors a and b.

We remark that the role of the two sets $\mathcal{A}$ and $\mathcal{B}$ is not symmetric, as it may happen that $\mathcal{A}$ is separable from $\mathcal{B}$ but the reverse is not true. In fact a necessary (but not sufficient) condition for the existence of a separating sphere is that the intersection of $\mathcal{B}$ and of $\mathrm{conv}(\mathcal{A})$ is empty.

The sets $\mathcal{A}$ and $\mathcal{B}$ are defined to be spherically separated by $S(x_0, R)$, the sphere centered in $x_0 \in \mathbb{R}^n$ with radius $R \in \mathbb{R}$, if

$$(a_i - x_0)^T (a_i - x_0) \leq R^2 \tag{1.1}$$

for all points $a_i \in \mathcal{A}$ $(i = 1, \ldots, m)$ and

$$(b_l - x_0)^T (b_l - x_0) \geq R^2 \tag{1.2}$$

for all points $b_l \in \mathcal{B}$ $(l = 1, \ldots, k)$.

Since in general we are not able to know in advance whether or not the two sets are spherically separable, a classification error function has to be defined in order to find a minimal error separating sphere. In particular, according to (1.1) and (1.2), the classification error function w associated to the sphere $S(x_0, R)$ can be defined as follows:

$$\begin{aligned} w(x_0, R) \stackrel{\triangle}{=} & \sum_{i=1}^{m} \max\{0, (a_i - x_0)^T (a_i - x_0) - R^2\} \\ & + \sum_{l=1}^{k} \max\{0, R^2 - (b_l - x_0)^T (b_l - x_0)\}. \end{aligned} \tag{1.3}$$

We observe that the above function is nonsmooth and nonconvex.

In the paper, we deal with the minimization of such an error function. As previously mentioned, the problem has been already considered in [4], under the hypothesis that the center

of the sphere is prefixed; in this case it reduces to the computation of the optimal radius, which can be obtained by minimizing a convex nonsmooth function of one variable; it is worth noting that kernel transformations can be embedded into the model.

We consider here the coordinates of the center as decision variables together with the radius, and we present in the following two different approaches, according to the fact that the center of the sphere can be located in the entire space or, alternatively, it is constrained to belong to the convex hull of the set $\mathcal{A}$. We remark that the latter variant can cope with the use of kernels.

The paper is organized as follows. In Sect. 2 we recall the spherical separation algorithm proposed in [4]. In Sect. 3 we tackle the spherical separation problem in a general setting (no constraints in the location of the center), and we introduce two different decompositions of the objective function in DC (Difference of Convex) form. In Sect. 4, in order to embed kernel transformations of the SVM type, we state the spherical separation problem when the center is constrained to belong to the convex hull of set $\mathcal{A}$. In Sect. 5 we recall the DC-Algorithm [1,17], which we adopt to test our approach. The numerical results are presented in Sect. 6 and some conclusions are drawn in Sect. 7.

Finally we recall some basic concepts of convex analysis, which will be used in the sequel. The subdifferential $\partial f(x)$ of a convex function f at any point x is the set of the subgradients, i.e. the set of vectors $g \in \mathbb{R}^n$ satisfying the subgradient inequality

$$f(y) \geq f(x) + g^T(y - x) \quad \text{for all } \ y \in \mathbb{R}^n.$$

Analogously, the ϵ-subdifferential $\partial_\epsilon f(x)$, for any $\epsilon \geq 0$, of a convex function f at any point x is the set of the ϵ-subgradients, i.e. the set of vectors $g \in \mathbb{R}^n$ satisfying the ϵ-subgradient inequality

$$f(y) \geq f(x) + g^T(y - x) - \epsilon \quad \text{for all } \ y \in \mathbb{R}^n.$$

2 The fixed center case

In this section we briefly review the problem of discriminating the two sets $\mathcal{A}$ and $\mathcal{B}$ by means of a sphere, once the center x_0 is given [4]. This is a simplified case of the spherical separation proposed in [18,19].

As in [18,19] the objective is to find, in the input space or in the feature space, a minimal radius sphere separating the set $\mathcal{A}$ from the set $\mathcal{B}$. Taking into account the definition of the error function w, we come out with the following optimization problem:

$$\begin{aligned} \min_{x_0, R} \ & R^2 + C \sum_{i=1}^{m} \max\{0, (a_i - x_0)^T (a_i - x_0) - R^2\} \\ & + C \sum_{l=1}^{k} \max\{0, R^2 - (b_l - x_0)^T (b_l - x_0)\}, \end{aligned} \tag{2.1}$$

where the positive constant C states the relative importance of the two objectives (the radius and the classification error).

In general, $(n + 1)$ parameters need to be selected: the center of the sphere (a point in $\mathbb{R}^n$) and the radius of the sphere (a scalar). In case the center is given, the above problem reduces to the minimization of a function of just one variable (the radius), which is nonsmooth and convex.

In fact, by introducing the change of variable

$$z \stackrel{\triangle}{=} R^2, \quad z \geq 0$$

and by defining

$$\xi_i \stackrel{\triangle}{=} (a_i - x_0)^T (a_i - x_0) \geq 0, \quad i = 1, \ldots, m,$$
$$\psi_l \stackrel{\triangle}{=} (b_l - x_0)^T (b_l - x_0) \geq 0, \quad l = 1, \ldots, k,$$

problem (2.1) becomes:

$$\min_{z \geq 0} z + C \left(\sum_{i=1}^{m} \max\{0, \xi_i - z\} + \sum_{l=1}^{k} \max\{0, z - \psi_l\} \right).$$

Observe that the function to be minimized depends only on the scalar (nonnegative) variable z and is convex and piecewise affine. The problem can be simply put in the form of a structured Linear Program.

In [4] an "ad hoc" algorithm that finds the optimal solution in $O(p \log p)(p \stackrel{\triangle}{=} \max\{m, k\})$ has been presented. It basically consists of two phases: the "sorting phase" and the "cutting phase". In the first one the sample points are sorted according to their distance from the center, while in the second one an optimal "cut" is found.

The adopted simplification is rather drastic, nevertheless a judicious choice of the center (e.g. the barycenter of the set $\mathcal{A}$) has allowed to obtain reasonably good separation results at a very low computational cost, thanks also to the use of kernel transformations which can be introduced in a straightforward way.

3 The unconstrained moving center case

In this section we consider the case where the center of the sphere is no longer fixed. Thus the problem becomes:

$$\min_{x_0, R} h(x_0, R), \tag{3.1}$$

where

$$h(x_0, R) \stackrel{\triangle}{=} R^2 + C \sum_{i=1}^{m} \max\{0, \|a_i - x_0\|^2 - R^2\}$$
$$+ C \sum_{l=1}^{k} \max\{0, R^2 - \|b_l - x_0\|^2\}. \tag{3.2}$$

Observe that the objective function h is nonconvex. It is well known that, for any given set of convex functions $f_i(x)$, $i = 1, \ldots, s$, $f_i : \mathbb{R}^n \mapsto \mathbb{R}$, the function

$$f(x) \stackrel{\triangle}{=} \min_{i=1,\ldots,s} f_i(x)$$

can be put in the DC form by setting:

$$f(x) = \sum_{i=1}^{s} f_i(x) - \max_{i=1,\ldots,s} \sum_{r=1; r \neq i}^{s} f_r(x). \tag{3.3}$$

By applying such transformation and putting $z \stackrel{\triangle}{=} R^2$, h can be rewritten in the form of a Difference of Convex (DC) functions as follows:

$$h(x_0, z) = \alpha_1(x_0, z) - \alpha_2(x_0, z),$$

where

$$\alpha_1(x_0, z) \stackrel{\triangle}{=} z(1 + Ck) + C\sum_{i=1}^{m} \max\{0, \|a_i - x_0\|^2 - z\} + C\sum_{l=1}^{k} \max\{0, \|b_l - x_0\|^2 - z\}$$

and

$$\alpha_2(x_0, z) \stackrel{\triangle}{=} C\sum_{l=1}^{k} \|b_l - x_0\|^2$$

are convex functions.

Then problem (3.1) becomes:

$$\begin{cases} \min\limits_{x_0, z} \ [\alpha_1(x_0, z) - \alpha_2(x_0, z)] \\ \qquad z \geq 0. \end{cases} \tag{3.4}$$

In addition we propose in the following yet another DC decomposition for h, based on the equality

$$\max\{0, f_1(x) - f_2(x)\} = \max\{f_1(x), f_2(x)\} - f_2(x), \tag{3.5}$$

which is valid for any couple of convex functions f_1 and f_2. By applying (3.5) to our case, we can write h in the form:

$$h(x_0, R) = \beta_1(x_0, R) - \beta_2(x_0, R),$$

where

$$\beta_1(x_0, R) \stackrel{\triangle}{=} R^2 + C\sum_{i=1}^{m} \max\{R^2, \|a_i - x_0\|^2\} + C\sum_{l=1}^{k} \max\{R^2, \|b_l - x_0\|^2\}$$

and

$$\beta_2(x_0, R) \stackrel{\triangle}{=} CmR^2 + C\sum_{l=1}^{k} \|b_l - x_0\|^2$$

are convex functions. Then we come out with our second reformulation of problem (3.1):

$$\min_{x_0, R}[\beta_1(x_0, R) - \beta_2(x_0, R)]. \tag{3.6}$$

We will use in our numerical experiments the DCA [1,17] approach, which will be briefly recalled in Sect. 5. DCA solves minimization problems characterized by DC objective functions. It is known that its performance, in terms of speed of convergence, globality of computed solutions, efficiency, etc., may be affected by the specific DC decomposition scheme adopted. This observation has motivated the setting of two alternative decomposition schemes.

We finally observe that in both decompositions of h, the first functions, α_1 and β_1, respectively, are nonsmooth.

4 The constrained moving center case

In this section we impose that the center is selected in the convex hull of any generic non-empty, finite set of points in the n-dimensional space $\mathbb{R}^n$, say $\mathcal{X} = \{x_1, \ldots, x_r\}$.

Differently from the general setting of previous section, the resulting mathematical model is able to accomodate kernel transformations of the type adopted in the SVM approach.

In particular, we introduce into problem (3.1) the following constraints:

$$x_0 = \sum_{j=1}^{r} \lambda_j x_j, \quad \sum_{j=1}^{r} \lambda_j = 1 \quad \lambda_j \geq 0, \quad j = 1, \ldots, r. \tag{4.1}$$

Substituting the above expression of x_0 in the definition of function h and letting Q be the $r \times r$ matrix whose generic entry is $q_{ij} \triangleq x_i^T x_j$, we come out with the following minimization problem:

$$\begin{cases} \min\limits_{\lambda, R} \tilde{h}(\lambda, R) \\ \quad e^T \lambda = 1 \\ \quad \lambda \geq 0, \end{cases} \tag{4.2}$$

where

$$\begin{aligned} \tilde{h}(\lambda, R) \triangleq R^2 &+ C \sum_{i=1}^{m} \max\{0, c_i - 2u_i^T \lambda + \lambda^T Q \lambda - R^2\} \\ &+ C \sum_{l=1}^{k} \max\{0, -d_l + 2v_l^T \lambda - \lambda^T Q \lambda + R^2\}, \end{aligned}$$

with

$$u_i^T \triangleq \left[x_1^T a_i, \ldots, x_r^T a_i\right], \quad c_i \triangleq \|a_i\|^2, \quad i = 1, \ldots, m,$$

and

$$v_l^T \triangleq \left[x_1^T b_l, \ldots, x_r^T b_l\right], \quad d_l \triangleq \|b_l\|^2, \quad l = 1, \ldots, k.$$

Also problem (4.2), setting $z \triangleq R^2$, can be put in DC form. In fact function $\tilde{h}$ is rewritable as

$$\tilde{h}(\lambda, z) = \tilde{\alpha}_1(\lambda, z) - \tilde{\alpha}_2(\lambda, z),$$

where

$$\begin{aligned} \tilde{\alpha}_1(\lambda, z) \triangleq C &\sum_{i=1}^{m} \max\{0, c_i - 2u_i^T \lambda + \lambda^T Q \lambda - z\} \\ &+ C \sum_{l=1}^{k} \max\{0, d_l - 2v_l^T \lambda + \lambda^T Q \lambda - z\} \\ &+ z(1 + Ck) - C \sum_{l=1}^{k} (d_l - 2v_l^T \lambda) \end{aligned}$$

and

$$\tilde{\alpha}_2(\lambda, z) \triangleq Ck\lambda^T Q \lambda$$

are both convex functions, being Q positive semidefinite. Then we come out with the following problem:

$$\begin{cases} \min\limits_{\lambda,z} \ [\tilde{\alpha}_1(\lambda, z) - \tilde{\alpha}_2(\lambda, z)] \\ \quad e^T\lambda = 1 \\ \quad \lambda, z \geq 0. \end{cases} \tag{4.3}$$

As in the unconstrained case, it is still possible to obtain the following alternative DC decomposition of problem (4.2) based on the equality (3.5):

$$\begin{cases} \min\limits_{\lambda,R} \ [\tilde{\beta}_1(\lambda, R) - \tilde{\beta}_2(\lambda, R)] \\ \quad e^T\lambda = 1 \\ \quad \lambda \geq 0, \end{cases} \tag{4.4}$$

where

$$\tilde{\beta}_1(\lambda, R) \stackrel{\triangle}{=} R^2 + C\sum_{i=1}^{m} \max\{R^2, c_i - 2u_i^T\lambda + \lambda^T Q\lambda\} + C\sum_{l=1}^{k} \max\{R^2, d_l - 2v_l^T\lambda + \lambda^T Q\lambda\}$$

and

$$\tilde{\beta}_2(\lambda, R) \stackrel{\triangle}{=} CmR^2 + C\sum_{l=1}^{k}(d_l - 2v_l^T\lambda + \lambda^T Q\lambda)$$

are convex functions.

Finally a possible choice in constraining the center is setting $\mathcal{X} = \mathcal{A}$. Such assumption appears to reflect the spirit of the spherical separation model and it is the one we have implemented in our numerical tests.

4.1 Using kernel transformations

Function $\tilde{h}$ is indeed suitable for using kernel transformations of the SVM type.

Our kernel approach consists in mapping the data into a higher dimensional space (the feature space F) and in separating the two transformed sets by means of a sphere. In particular, given the map

$$\phi : \ y \in Y \subseteq \mathbb{R}^n \ \rightarrow \ \phi(y) \in F \subseteq \mathbb{R}^N,$$

a function $K : \ Y \times Y \ \rightarrow \ \mathbb{R}$ is a kernel if it satisfies the following condition:

$$K(y_1, y_2) = \phi(y_1)^T\phi(y_2), \quad \text{for all } y_1, y_2 \in Y.$$

Consequently the use of a kernel function K allows us to compute the inner products in the feature space without explicitly computing the map ϕ.

Once transformation ϕ has been introduced and the two sets $\mathcal{A}$ and $\mathcal{B}$ have been recoded, respectively, as

$$\hat{\mathcal{A}} = \{\phi(a_1), \ldots, \phi(a_m)\} \ \text{ and } \ \hat{\mathcal{B}} = \{\phi(b_1), \ldots, \phi(b_k)\},$$

we look for a sphere in $\mathbb{R}^N$, centered in $\phi(x_0) \in \mathbb{R}^N$ and with radius $\hat{R} \in \mathbb{R}$. The objective is again the minimization of both the radius and the classification error in the feature space.

Parallel to the approach adopted for the non-kernelized version, we constrain the center to be in the convex hull of $\hat{\mathcal{A}}$.

In stating the problem in the feature space, a relevant role is played by the $m \times m$ matrix $\hat{Q}$, whose entries $\hat{q}_{ij} = \phi(a_i)^T \phi(a_j)$ can be expressed, through the kernel function, in the form $\hat{q}_{ij} = K(a_i, a_j)$.

More precisely, introducing the map ϕ and the kernel function K, the problem can be stated as follows:

$$\begin{cases} \min\limits_{\lambda, \hat{R}} \hat{h}(\lambda, \hat{R}) \\ \qquad e^T \lambda = 1 \\ \qquad \lambda \geq 0, \end{cases} \tag{4.5}$$

where

$$\begin{aligned} \hat{h}(\lambda, \hat{R}) \triangleq \hat{R}^2 + C \sum_{i=1}^{m} \max\{0, \hat{c}_i - 2\hat{u}_i^T \lambda + \lambda^T \hat{Q} \lambda - \hat{R}^2\} \\ + C \sum_{l=1}^{k} \max\{0, -\hat{d}_l + 2\hat{v}_l^T \lambda - \lambda^T \hat{Q} \lambda + \hat{R}^2\} \end{aligned}$$

with

$$\begin{aligned} \hat{u}_i^T &\triangleq \left[\phi(a_1)^T \phi(a_i), \ldots, \phi(a_m)^T \phi(a_i)\right] = \left[K(a_1, a_i), \ldots, K(a_m, a_i)\right], \quad i = 1, \ldots, m, \\ \hat{v}_l^T &\triangleq \left[\phi(a_1)^T \phi(b_l), \ldots, \phi(a_m)^T \phi(b_l)\right] = \left[K(a_1, b_l), \ldots, K(a_m, b_l)\right], \quad l = 1, \ldots, k, \\ \hat{c}_i &\triangleq \phi(a_i)^T \phi(a_i) = K(a_i, a_i), \quad i = 1, \ldots, m, \end{aligned}$$

and

$$\hat{d}_l \triangleq \phi(b_l)^T \phi(b_l) = K(b_l, b_l), \quad l = 1, \ldots, k.$$

We observe that problem (4.5) has the same structure as problem (4.2) and can be put in a DC form as well, according to the guidelines discussed in the previous sections.

5 DC functions and the DC-Algorithm (DCA)

Both problems stated in Sects. 3 and 4 involve minimization of DC functions.

We recall in the following the DC-Algorithm [1,17], which is the one we have implemented in our numerical experiments.

Suppose we want to solve the following problem:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{5.1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $f(x) = f_1(x) - f_2(x)$, with f_1 and f_2 being proper, lower semi-continuous, convex and not necessarily differentiable functions.

We remind that, if a point x^* is a local minimum of f, then the following inclusion holds:

$$\partial f_2(x^*) \subset \partial f_1(x^*), \tag{5.2}$$

which implies

$$\partial f_1(x^*) \cap \partial f_2(x^*) \neq \emptyset. \tag{5.3}$$

A point x^* satisfying condition (5.3) is named a critical point of f.

Note that condition (5.2) is also sufficient for local optimality, whenever f_2 is a convex polyhedral function.

Moreover point x^* is a global minimum of f if and only if $\partial_\epsilon f_2(x^*) \subset \partial_\epsilon f_1(x^*)$, for all $\epsilon \geq 0$.

We recall that the theory of conjugate functions plays a role in the statement of DCA [1,17]. In fact it consists in linearizing, at the current point, function f_2 and minimizing a convex approximation of function f obtained by replacing f_2 with its linearization: the minimum of such approximation becomes the next iterate. The algorithm, aimed at satisfying local optimality conditions, provides critical points, by using only tools from convex analysis.

In particular, given the current point x_k at iteration k, we define the following linearization of f_2 rooted at x_k:

$$f_2^{(k)}(x) \stackrel{\triangle}{=} f_2(x_k) + g_k^T(x - x_k),$$

with g_k being an arbitrary subgradient of f_2 at x_k. Then x_{k+1} is obtained as a minimizer of the convex function

$$f^{(k)}(x) \stackrel{\triangle}{=} f_1(x) - f_2^{(k)}(x).$$

Note that $f^{(k)}$ interpolates f at x_k; moreover, since $f_2^{(k)}(x) \leq f_2(x)$ for all $x \in \mathbb{R}^n$, we have

$$f^{(k)}(x) \geq f(x), \quad \text{for all } x \in \mathbb{R}^n, \tag{5.4}$$

and consequently, taking into account that x_{k+1} is a minimizer of $f^{(k)}$, it holds

$$f(x_{k+1}) \leq f(x_k).$$

It is easy to verify that, if $f(x_{k+1}) = f(x_k)$, then x_k is a critical point. This result provides a stopping criterion in designing a descent method. We finally remark that, when f_2 is differentiable, any critical point satisfies the necessary optimality condition (5.2).

6 Numerical results

In our numerical experiments we have implemented the DC-Algorithm described in Sect. 5, to tackle the spherical separation problem in both the unconstrained and the constrained moving center cases (UMC and CMC, respectively). For each such case we have implemented both DC decomposition schemes of the objective functions introduced in Sect. 3.

The corresponding Fortran 77 codes are, respectively, UMC-1 and UMC-2 to solve problems (3.4) and (3.6), and CMC-1 and CMC-2 to solve (4.3) and (4.4). In particular, kernel transformations are embedded into codes CMC-1 and CMC-2. All programs have been run on a 2.40 GHz Intel Core Duo, under Microsoft Windows XP operating system.

We recall that the DCA scheme requires, at each iteration, to solve a convex program. To this aim we have used the subroutine NCVX [7], which implements a bundle type approach

Table 1 Datasets

#	Dataset	Dimension	Points
1	Cancer	9	699
2	Diagnostic	30	569
3	Heart	13	297
4	Pima	9	769
5	Ionosphere	34	351
6	Sonar	60	208
7	Galaxy	14	4192
8	g50c	50	550
9	g10n	10	550

enabling solution of nonsmooth unconstrained optimization problems, either convex or nonconvex. A new version of the NCVX code has been implemented, which allows treatment of the linear constraints.

The stopping criterion adopted is based on the value of the objective function f. In particular we stop at the iteration k whenever at the current point x_k the following condition holds:

$$|f(x_k) - f(x_{k-1})| \leq \epsilon,$$

with $\epsilon = 10^{-3}$.

We have run our codes on some test problems drawn from the binary classification literature which are described in Table 1. The first six datasets are taken from the UCI Machine Learning Repository [11], Galaxy is the dataset used in galaxy discrimination with neural networks [13], while the last two test problems are described in [5].

We have adopted the tenfold cross-validation protocol, which consists in splitting the dataset of interest into ten equally sized pieces. Nine of them are in turn used as training set and the remaining one as testing set. By correctness we intend the total percentage of well classified points (of both $\mathcal{A}$ and $\mathcal{B}$) when the algorithm stops. It is worth noting, in evaluating the numerical results, that such correctness measure is not coincident with the adopted error function, even though, of course, a good correlation between them is expected.

In Table 2, for each dataset, the testing correctness and the CPU time are obtained as the average of the tenfold cross-validation results. No kernel transformation is involved.

For comparison purposes we report in the same table the results obtained by applying the method summarized in Sect. 2 (see [4] for a complete treatment), where spherical separation is performed by predefining the center of the sphere as the barycenter of $\mathcal{A}$. The corresponding code is referred to as FC (Fixed Center).

It is well known that a good starting point can improve the performance of DCA. In our case we have adopted the barycenter of $\mathcal{A}$ as the starting center, whereas the starting radius has been selected as the one provided in output by FC.

Preliminary tuning for the parameter C has been performed on the grid 0.1, 1, 10, 100, 1000. In particular, for a fair comparison, we have selected, for each dataset and for each code, the value of C optimizing the performance on the testing set.

In Table 2 the best result for each dataset has been underlined.

In terms of correctness, of course, the moving center approach in general overcomes FC. As expected, the best performance is achieved, except for Cancer, Pima and g10n datasets, by the unconstrained moving center approach (UMC-1 and UMC-2); the exceptions can be

Table 2 Average percentage of testing correctness and average CPU time in seconds (without kernel)

Dataset	FC		UMC-1		UMC-2		CMC-1		CMC-2	
	%	s	%	s	%	s	%	s	%	s
1	97.00	0.06	95.86	1.5	95.71	35.4	97.00	10.7	97.14	9.1
2	84.03	0.05	88.25	91.3	89.82	176.0	87.02	0.5	87.02	2.7
3	75.00	0.004	80.33	22.2	80.33	13.5	77.33	33.4	79.00	27.9
4	69.35	0.1	68.57	103.0	68.70	55.5	59.61	6.6	60.00	2.3
5	51.14	0.01	70.86	129.6	72.00	90.1	61.71	293.0	62.00	431.0
6	59.52	0.01	69.05	27.7	69.05	16.8	60.95	35.7	60.00	52.2
7	80.19	2.2	93.82	497.6	93.79	502.3	85.11	9.9	85.18	23.1
8	67.76	0.003	73.22	3.3	72.96	1.7	71.26	36.8	71.14	11.5
9	54.02	0.000	80.54	12.9	81.04	42.1	73.58	48.1	81.10	150.1

explained by considering that minimization of the classification error does not necessarily correspond to minimization of the number of misclassified points (which contribute to the definition of correctness).

As for the CPU time, FC is the natural winner. In fact such approach is very fast because the center of the separating sphere is fixed and the optimal radius is computed by a simple polynomial time algorithm (see [4]). As for the moving center approach, no superiority of any between the two tested methods is exhibited, possibly as consequence of the fact that no tuning of the input parameters of the nonsmooth optimization subroutine NCVX has been performed.

Kernel transformations (see Sect. 4.1) can cope with both the fixed and the constrained moving center approaches. We have tested the following kernel functions:

- polynomial function: $K(y_1, y_2) = (y_1^T y_2 + 1)^\sigma$;
- radial basis function: $K(y_1, y_2) = \exp(-\|y_1 - y_2\|^2/(2\sigma^2))$;
- exponential radial basis function: $K(y_1, y_2) = \exp(-\|y_1 - y_2\|/(2\sigma^2))$;

where σ is a parameter to be set.

We have run our algorithms for the different kernel functions, adopting several combinations of values of the parameters σ and C. For each dataset and for each code, we report in Table 3 the results giving the best performance in terms of average testing correctness.

We note, by comparing the results of both Tables 2 and 3, that in general the performance of the unconstrained center approach remains better than both those of the kernelized versions of FC and of the constrained center approach. However the use of kernels in CMC-1 and CMC-2, except for g50c and g10n, provides better results than the corresponding non-kernelized versions.

We observe that the DC decomposition scheme adopted in problems (3.6) and (4.4) provides slightly better results in terms of correctness than the one used in (3.4) and (4.3).

Finally, we remark that the spherical separation model does not appear to provide a valid alternative to classical SVM approach on the considered benchmark datasets. This, of course, does not prevent spherical separation from being a valuable tool for structured datasets coming from real world applications.

Table 3 Average percentage of testing correctness and average CPU time in seconds (with kernel)

Dataset	FC		CMC-1		CMC-2	
	%	s	%	s	%	s
1	97.14	0.2	97.28	3.2	97.28	4.1
2	85.44	0.06	87.54	18.2	88.07	10.3
3	75.33	0.02	78.67	16.9	78.33	64.6
4	71.82	0.2	67.66	2.1	67.27	2.6
5	64.85	0.000	66.86	101.1	70.86	48.2
6	61.43	0.04	70.00	19.2	70.95	4.6
7	83.77	4.2	90.55	219.1	90.43	175.6
8	67.76	0.003	67.76	1.0	68.00	13.2
9	54.34	0.001	66.74	7.6	70.58	16.5

7 Conclusions

The results of our numerical experiments suggest that both the unconstrained and constrained moving center approaches provide a substantial improvement in classification correctness with respect to the fixed center method. The increase of computational effort on small and moderate size problems appears affordable.

As for parameters tuning and selection of the appropriate kernel function, there is no evidence of any optimal rule.

Acknowledgments We express our gratitude to the referees for a number of significant suggestions which helped us in preparation of a revised version of the paper.

References

1. An, L.T.H., Tao, P.D.: The DC (Difference of Convex Functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. Ann. Oper. Res. **133**, 23–46 (2005)
2. Astorino, A., Gaudioso, M.: Polyhedral separability through successive LP. J. Optim. Theory Appl. **112**, 265–293 (2002)
3. Astorino, A., Gaudioso, M.: Ellipsoidal separation for classification problems. Optim. Methods Softw. **20**, 261–270 (2005)
4. Astorino, A., Gaudioso, M.: A fixed-center spherical separation algorithm with kernel transformations for classification problems. Computat. Manage. Sci. **6**, 357–372 (2009)
5. 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)
6. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
7. Fuduli, A., Gaudioso, M., Giallombardo, G.: Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. SIAM J. Optim. **14**, 743–756 (2004)
8. Fung, G., Mangasarian, O.L.: Proximal support vector machine classifiers. in Proceedings KDD-2001, pp. 77–86. San Francisco (2001)
9. Mangasarian, O.: Linear and nonlinear separation of patterns by linear programming. Oper. Res. **13**, 444–452 (1965)
10. Mangasarian, O.L., Musicant, D.R.: Lagrangian support vector machines. J. Mach. Learn. Res. **1**, 161–177 (2001)
11. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases. In http://www.ics.uci.edu/~mlearn/MLRepository.html (1992)

12. Rosen, J.B.: Pattern separation by convex programming. J. Math. Anal. Appl. **10**, 123–134 (1965)
13. Odewahn, S., Stockwell, E., Pennington, R., Humphreys, R., Zumach, W.: Automated star/galaxy discrimination with neural networks. Astron. J. **103**, 318–331 (1992)
14. Schölkopf, B., Smola, A.J.: Learning with kernels. MIT Press, Cambridge (2002)
15. Schölkopf, B., Burges, C.J.C., Smola, A.J.: Advances in kernel methods. Support vector learning. MIT Press, Cambridge (1999)
16. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge University Press, Cambridge (2004)
17. Tao, P.D., An, L.T.H.: A D.C. optimization algorithm for solving the trust-region subproblem. SIAM J. Con. Opt. **8**, 476–505 (1998)
18. Tax, D.M.J., Duin, R.P.W.: Data domain description using support vectors. In: ESANN'1999 proceedings Bruges, pp. 251–256. Belgium (1999)
19. Tax, D.M.J., Duin, R.P.W.: Uniform object generation for optimizing one-class classifiers. J. Mach. Learn. Res. **2**, 155–173 (2001)
20. Vapnik, V.: The nature of the statistical learning theory. Springer, New York (1995)