ADAC (2007) 1:85–104
DOI 10.1007/s11634-007-0011-2

REGULAR ARTICLE

Fuzzy clustering based on nonconvex optimisation approaches using difference of convex (DC) functions algorithms

Hoai An Le Thi · Hoai Minh Le · Tao Pham Dinh

Received: 5 June 2007 / Revised: 26 June 2007 / Accepted: 2 July 2007 / Published online: 25 July 2007

Abstract We present a fast and robust nonconvex optimization approach for Fuzzy C-Means (FCM) clustering model. Our approach is based on DC (Difference of Convex functions) programming and DCA (DC Algorithms) that have been successfully applied in various fields of applied sciences, including Machine Learning. The FCM model is reformulated in the form of three equivalent DC programs for which different DCA schemes are investigated. For accelerating the DCA, an alternative FCM-DCA procedure is developed. Experimental results on several real world problems that include microarray data illustrate the effectiveness of the proposed algorithms and their superiority over the standard FCM algorithm, with respect to both running-time and accuracy of solutions.

Keywords Fuzzy clustering · Nonconvex optimization · DC programming · DCA

JEL Classification: C61 · C63 · C88

H. A. Le Thi (✉) · H. M. Le
Laboratory of Theoretical and Applied Computer Science (LITA EA 3097),
UFR MIM, University of Paul Verlaine-Metz,
Ile du Saulcy, 57045 Metz, France
e-mail: lethi@univ-metz.fr

H. M. Le
e-mail: lehoai@univ-metz.fr

T. Pham Dinh
Laboratory of Modelling, Optimization & Operations Research,
National Institute for Applied Sciences, Rouen, BP 08, Place Emile Blondel,
76131 Mont Saint Aignan Cedex, France
e-mail: pham@insa-rouen.fr

1 Introduction

Clustering, which aims at dividing a data set into groups or clusters containing similar data, is a fundamental problem in unsupervised learning and has many applications in various domains. In recent years, there has been significant interest in developing clustering algorithms to massive data sets (Alon and Spencer 1991; Klawonn and Höppner 2003; Le Thi et al. 2007a,b; MacQueen 1967 and references therein). Two main approaches have been used for clustering: statistical and machine learning based on mixture models (see e.g. Alon and Spencer 1991; Arora and Kanan 2001; Duda and Hart 1972) and the mathematical programming approach that considers clustering as an optimization problem (see e.g. Bradley and Mangasarian 1998; Feder and Greene 1998; Le Thi et al. 2007a,b; Mangasarian 1997 and the references therein).

Clustering algorithms are partitioned into two classes: *hard* clustering and *fuzzy* clustering. In hard clustering, data is divided into disjoint clusters, where each data element belongs to exactly one cluster. In fuzzy clustering, objects are not classified as belonging to one and only one cluster, but instead, to each object is assigned a degree of membership with each of the clusters. In real applications sharp boundaries between clusters may not exist, so fuzzy clustering is often better suited for the data.

This work presents a mathematical programming approach to fuzzy clustering. We consider the Fuzzy C-Means (FCM) clustering model that is undoubtedly the most widely used fuzzy clustering technique. It was originally introduced in Bezdek (1981) as a fuzzification of the k-Means model of hard clustering.

Let $X := \{x_1, x_2, \ldots, x_n\}$ denote n objects to be partitioned into c $(2 \leq c \leq n)$ homogeneous clusters $C_1, C_2, \ldots, C_c$ where $x_k \in \mathbb{R}^p$ $(k = 1, \ldots, n)$ represents a multidimensional data vector. Consider the matrix $U = (u_{i,k})_{c\times n}$ called the *fuzzy partition matrix* in which each element $u_{i,k}$ indicates the membership degree of the object x_k in the cluster C_i (e.g. the probability that x_k belongs to the cluster C_i). The FCM technique is based on optimizing the objective function

$$J_m(U, V) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{i,k}^m ||x_k - v_i||^2, \tag{1}$$

where $\|.\|$ is, in this paper, the Euclidean norm in $\mathbb{R}^p$, and V the $(c \times p)$—matrix whose ith row is $v_i \in \mathbb{R}^p$, the center of C_i. The fixed parameter $m \geq 1$ is called the fuzziness index of membership of each datum. The mathematical model of FCM is given by

$$\begin{cases} \min J_m(U, V) := \sum_{k=1}^{n} \sum_{i=1}^{c} u_{i,k}^m ||x_k - v_i||^2 \\ s.t. \quad u_{i,k} \in [0, 1] \text{ for } i = 1, \ldots, c \quad k = 1, \ldots, n. \\ \sum_{i=1}^{c} u_{i,k} = 1, \quad k = 1, \ldots, n \end{cases} \tag{2}$$

This is a nonconvex optimization problem for which only heuristic algorithms are available until now. From an optimization point of view, the introduction of the fuzzy partition matrix U makes problem (2) more difficult than the k-Means model in hard clustering using the squared Euclidean distance (the objective function of (2) is much

more complicated, the number of variables is increased, the presence of constraints). In fact, in real applications this is a very large scale problem (*high* dimension and *large* data set, i.e. p and n are very large), that is why global optimization approaches such as Branch & Bound, Cutting plane algorithms etc. cannot be used.

We investigate in this paper an efficient nonconvex programming approach for the FCM problem (2). Our method is based on DC (Difference of Convex functions) programming and DCA (DC Algorithms) that were introduced by Pham Dinh Tao in a preliminary form in 1985. They have been extensively developed since 1994 by Le Thi Hoai An and Pham Dinh Tao (see Le Thi 1997; Le Thi and Pham Dinh 1997, 2005; Pham Dinh and Le Thi 1998 and the references therein) and become now classic and more and more popular (see e.g. Liu et al. 2005; Liu and Shen 2006; Neumann et al. 2004; Ronan et al. 2006; Shen et al. 2003; Weber et al. 2005). To our knowledge, DCA is one of the very few algorithms of nonconvex programming approach that can solve efficiently large-scale optimisation problems. DCA has been successfully applied to many (smooth or nonsmooth) nonconvex programs in various domains of applied sciences (see Le Thi 1997; Le Thi and Pham Dinh 1997, 2005; Le Thi et al. 2007a,b; Pham Dinh and Le Thi 1998 and the references therein), in particular in Machine Learning (Le Thi et al. 2007a; Liu and Shen 2006; Neumann et al. 2004; Ronan et al. 2006; Weber et al. 2005) for which they provide quite often a global solution and proved to be more robust and efficient than the standard methods. In Le Thi et al. (2007a) DCA scheme has been developed to hard clustering using the k-Means model.

The purpose of this paper is to demonstrate that, as shown for previous studies in Machine Learning, DCA is a promising approach for fuzzy clustering.

A so-called DC program is that of minimizing a DC function

$$f = g - h \text{ (with } g \text{ and } h \text{ being convex functions)}$$

over a convex set. The construction of DCA involves the convex DC components g and h but not the DC function f itself. Moreover, a DC function f *has infinitely many DC decompositions* $g - h$ *which have a crucial impact on the quality* (speed of convergence, robustness, efficiency, globality of computed solutions,...) of DCA. We propose in this work three equivalent DC programs for the FCM model (2) and develop the corresponding DCA schemes. We show that the effect of the DC decomposition is great.

Despite its local character, DCA with a good initial solution (depending on the specific structure of treated DC programs) converges quite often to a global solution in practice. Hence, another important question when applying DCA is how to find a *good* starting point. For this purpose we investigate in this work an alternative FCM-DCA procedure. Experimental results on several biomedical data sets show clearly the effectiveness of the proposed algorithms and their superiority with respect to the standard FCM algorithm in both running-time and accuracy of solutions.

The remainder of the paper is organized as follows. In Sect. 2, we give a brief introduction of DC programming and DCA. Section 3 is devoted to the description of several DC formulations and DCA schemes for solving problem (2) while the acceleration of DCA by an alternative FCM-DCA procedure is presented in Sect. 4. Finally, numerical experiments on biomedical data are reported in Sect. 5.

2 DC programming and DCA

DC programming and DCA constitute the backbone of smooth/nonsmooth nonconvex programming and global optimization. They address the problem of minimizing a function f which is the difference of two convex functions on the whole space $\mathbb{R}^p$ or on a convex set $C \subset \mathbb{R}^p$. Generally speaking, a DC program is an optimisation problem of the form:

$$\alpha = \inf\{f(x) := g(x) - h(x) : x \in \mathbb{R}^p\} \qquad (P_{dc})$$

where g, h are lower semi-continuous proper convex functions on $\mathbb{R}^p$. Such a function f is called a DC function, and $g - h$ a DC decomposition of f while g and h are the DC components of f. The convexity constraint $x \in C$ can be incorporated in the objective function of (P_{dc}) by using the indicator function on C denoted by χ_C which is defined by $\chi_C(x) = 0$ if $x \in C$, and $+\infty$ otherwise:

$$\inf\{f(x) := g(x) - h(x) : x \in C\} = \inf\{\chi_C(x) + g(x) - h(x) : x \in \mathbb{R}^p\}.$$

Let (Rockfellar 1970)

$$g^*(y) := \sup\{\langle x, y\rangle - g(x) : x \in \mathbb{R}^p\}$$

be the conjugate function of a convex function g. Then, the following program is called the dual program of (P_{dc}):

$$\alpha_D = \inf\{h^*(y) - g^*(y) : y \in \mathbb{R}^p\}. \qquad (D_{dc})$$

One can prove (Le Thi and Pham Dinh 1997) that $\alpha = \alpha_D$, and there is the perfect symmetry between primal and dual DC programs: the dual to (D_{dc}) is exactly (P_{dc}). For a convex function θ, the subdifferential of θ at $x_0 \in \text{dom}\,\theta := \{x \in \mathbb{R}^p : \theta(x_0) < +\infty\}$, denoted by $\partial\theta(x_0)$, is defined by

$$\partial\theta(x_0) := \{y \in \mathbb{R}^n : \theta(x) \geq \theta(x_0) + \langle x - x_0, y\rangle, \forall x \in \mathbb{R}^p\}. \qquad (3)$$

The subdifferential $\partial\theta(x_0)$ generalizes the derivative in the sense that θ is differentiable at x_0 if and only if $\partial\theta(x_0) \equiv \{\nabla_x\theta(x_0)\}$.

DCA is based on the local optimality conditions of (P_{dc}), namely

$$\partial h(x^*) \cap \partial g(x^*) \neq \emptyset \qquad (4)$$

(such a point x^* is called a *critical point* of $g - h$), and

$$\emptyset \neq \partial h(x^*) \subset \partial g(x^*). \qquad (5)$$

Note that (5) is a necessary local optimality condition for (P_{dc}). For many classes of the DC program, it is also a sufficient optimality condition (see Le Thi and Pham Dinh 1997, 2005).

The idea of DCA is simple: each iteration l of DCA approximates the concave part $-h$ by its affine majorization (that corresponds to taking $y^l \in \partial h(x^l)$) and minimizes the resulting convex function (that is equivalent to determining a point $x^{l+1} \in \partial g^*(y^l)$).

DCA scheme
Initialization: Let $x^0 \in \mathbb{R}^p$ be a best guess, $0 \leftarrow l$.
Repeat

- Calculate $y^l \in \partial h\left(x^l\right)$
- Calculate $x^{l+1} \in \arg\min\{g(x) - h\left(x^l\right) - \langle x - x^l, y^l\rangle : x \in \mathbb{R}^p\}$ (P_l)
- $l + 1 \leftarrow l$

Until convergence of the values x^l.

Note that (P_l) is a convex optimisation problem and insofar "easy" to solve. Convergence properties of DCA and its theoretical basis can be found in (Le Thi 1997; Le Thi and Pham Dinh 1997, 2005; Pham Dinh and Le Thi 1998). For instance it is important to mention that

- DCA is a descent method: the sequences $\left\{g\left(x^l\right) - h\left(x^l\right)\right\}$ and $\left\{h^*\left(y^l\right) - g^*\left(y^l\right)\right\}$ are decreasing (*without linesearch*) (Le Thi and Pham Dinh 1997).
- If the optimal value α of problem (P_{dc}) is finite and the infinite sequences $\{x^l\}$ and $\{y^l\}$ are bounded, then every limit point x^* (resp. $\widetilde{y}$) of the sequence $\{x^l\}$ (resp. $\{y^l\}$) is a critical point of $g - h$ (resp. $h^* - g^*$), i.e. $\partial h(x^*) \cap \partial g(x^*) \neq \emptyset$ (resp. $\partial h^*(y^*) \cap \partial g^*(y^*) \neq \emptyset$) (Le Thi and Pham Dinh 1997).
- DCA has a *linear convergence* for DC programs (Le Thi and Pham Dinh 2005).

For a complete study of DC programming and DCA the reader is referred to Le Thi (1997), Le Thi and Pham Dinh (1997, 2005), Pham Dinh and Le Thi (1998) and the references therein. The solution of a nonconvex program (P_{dc}) by DCA must be composed of two stages: the search of an *appropriate* DC decomposition of f and that of a *good* initial point. We shall apply *all these DC enhancement features* to solve problem (2) with the help of an equivalent DC program given in the next section.

We note that the convex concave procedure (CCCP) for constructing discrete time dynamical systems mentioned in Yuille and Rangarajan (2003) is nothing else than a special case of DCA. In the last five years DCA has been successfully applied in several studies in Machine Learning e.g. for SVM-based Feature Selection Neumann et al. (2004), for improving boosting algorithms Krause and Singer (2004), for implementing-learning Liu et al. (2005), Shen et al. (2003), Liu and Shen (2006), for Transductive SVMs Ronan et al. (2006) and for unsupervised clustering Le Thi et al. (2007a,b).

3 DC formulations and DCA schemes for the FCM model

We will present in this section different DC formulations of the FCM model (2) and show how DCA works for these DC programs.

In the problem (2) the variable U is a priori bounded in $\mathbb{R}^{c\cdot n}$. One can also find a bounding constraint for the variable V. Indeed, let $x_{k,j}$ be the jth component,

$j = 1, \ldots, p$, of the vector x_k and let

$$\alpha_j := \min_{k=1,\ldots,n} x_{k,j}, \quad \beta_j := \max_{k=1,\ldots,n} x_{k,j}.$$

Hence $v_i \in \mathcal{T}_i := \Pi_{j=1}^{p} [\alpha_j, \beta_j]$ for all $i = 1, \ldots, c$. For each $k \in \{1, \ldots, n\}$ let Δ_k be the $(c-1)$-simplex in $\mathbb{R}^c$ defined by

$$\Delta_k := \left\{ U^k := (u_{i,k})_i \in \mathbb{R}^c_+ : \sum_{i=1}^{c} u_{i,k} = 1 \right\}.$$

and $\Delta := \Pi_{k=1}^{n} \Delta_k$, $\mathcal{T} := \Pi_{i=1}^{c} \mathcal{T}_i$. The problem (2) can be rewritten as a constraint minimisation problem :

$$\min \{ J_m(U, V) : U \in \Delta, V \in \mathcal{T} \}. \tag{6}$$

3.1 The first DC program: a natural DC decomposition

Using the equation $2 f_1 f_2 = (f_1 + f_2)^2 - (f_1^2 + f_2^2)$ we can express the objective function of (2) as

$$\begin{aligned} J_m(U, V) &= \sum_{k=1}^{n} \sum_{i=1}^{c} u_{i,k}^m \|x_k - v_i\|^2 \\ &= \frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{c} \left(u_{i,k}^m + \|x_k - v_i\|^2 \right)^2 - \frac{1}{2} \left(u_{i,k}^{2m} + \|x_k - v_i\|^4 \right). \end{aligned}$$

Hence the following DC decomposition of $J_m(U, V)$ seems to be natural:

$$J_m(U, V) := G_1(U, V) - H_1(U, V) \tag{7}$$

with

$$G_1(U, V) = \frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{c} \left(u_{i,k}^m + \|x_k - v_i\|^2 \right)^2 \tag{8}$$

and

$$H_1(U, V) = \frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{c} \left(u_{i,k}^{2m} + \|x_k - v_i\|^4 \right) \tag{9}$$

being convex functions of U, V. Hence the first equivalent DC formulation of (2) is

$$\min \{ G_1(U, V) - H_1(U, V) : (U, V) \in \Delta \times \mathcal{T} \} \tag{10}$$

which can be written in the standard form as

$$\min \left\{ \chi_{\Delta \times \mathcal{T}} + G_1(U, V) - H_1(U, V) : (U, V) \in \mathbb{R}^{c \times n} \times \mathbb{R}^{c \times p} \right\}, \tag{11}$$

where χ_K is the indicator function of the set $K := \Delta \times \mathcal{T}$.

Solving (11) by DCA: According to the description of the DCA scheme in the previous section, applying DCA to (10) amounts to computing two sequences of pairs $\{(Y^l, Z^l)\}$ and $\{(U^l, V^l)\}$ in the way that $(Y^l, Z^l) \in \partial H_1(U^l, V^l)$ and $(U^{l+1}, V^{l+1}) \in \partial G_1^*(Y^l, Z^l)$ solves the convex program of the form (P_l).

It is easy to see that the function H_1 is differentiable and its gradient can be computed as

$$\begin{aligned}\nabla H_1(U^l, V^l) &= (\nabla_U H_1(U^l, V^l)), (\nabla_V H_1(U^l, V^l)) \\ &= \left(\left(m \left(u_{i,k}^l \right)^{2m-1} \right)_{i=1,\ldots,c}^{k=1,\ldots,n}, 2 \sum_{k=1}^{n} (\|x_k - V_i^l\|^2 (V_i^l - x_k))_{i=1\ldots,c} \right).\end{aligned} \tag{12}$$

Computing (U^{l+1}, V^{l+1}) amounts to solving the convex program

$$\min \left\{ G_1(U, V) - \left\langle (U, V), (Y^l, Z^l) \right\rangle : (U, V) \in \Delta \times \mathcal{T} \right\} \tag{13}$$

for which one can apply any algorithm for convex programming. Knowing that the projection of points into a simplex and/or a finite rectangle can be explicitly computed, in the implementation of our algorithm we use the Gradient Projection algorithm of Polyak (1987). This algorithm can be described as follows

Gradient Projection Algorithm GP for solving (13)
Let $\{\lambda_r\}$ be a sequence such that $\lim_{r \to +\infty} \lambda_r = 0$ and $\sum_{r=1}^{+\infty} \lambda_r = +\infty$.

1. For $r = 1$, put $(U^r, V^r) := \left(U^l, V^l\right)$, the current solution of DCA at iteration l.
2. Define $\left(U^{r+1}, V^{r+1}\right)$ as follows (Proj_S denotes the projection from $\mathbb{R}^p$ onto the set S): the kth column $\left(U^{r+1}\right)^k \in \mathbb{R}^c$ of the matrix U^{r+1} is

$$\left(U^{r+1}\right)^k := \text{Proj}_{\Delta_k} \left(\left(U^r\right)^k - \lambda_r \frac{\theta^r}{\|\theta^r\|} \right),$$

   where $\theta^r \in \mathbb{R}^c$ is the kth column of $\nabla_U G_1 (U^r, V^r) - Y^l$, and the ith row $\left(V^{r+1}\right)_i \in \mathbb{R}^p$ of the matrix V^{r+1} is

$$\left(V^{r+1}\right)_i := \text{Proj}_{\mathcal{T}_i} \left(\left(V^r\right)_i - \lambda_r \frac{\left(\nabla_V G_1(U^r, V^r) - Z^l\right)_i}{\| \left(\nabla_V G_1(U^r, V^r) - Z^l\right)_i \|} \right).$$

3. If $\| \left(U^{r+1}, V^{r+1}\right) - (U^r, V^r) \| \leq \epsilon$ then go to Step 4, otherwise increase r by 1 and go to Step 2.
4. STOP, output $(U^{l+1}, V^{l+1}) := (U^{r+1}, V^{r+1})$.

Remark 1 Although the calculation of the projection of points into a simplex and/or a rectangle is not expensive, the algorithm GP is not interesting for large-scale problems because it is an iterative method and the speed of convergence is quite sensitive

to the choice of the sequence $\{\lambda_r\}$. To overcome this difficulty we find other DC decompositions of J_m.

3.2 The second DC program: an interesting DC decomposition in the case $m \geq 2$

Let us consider now the case $m \geq 2$ that is often used in practice (m is the fuzziness index).

For finding other DC decompositions, we will determine a ball of radius r and center $0 \in \mathbb{R}^p$ containing necessarily the optimum centers v_i. The necessary first order optimality conditions for (U, V) imply that $\nabla_V J_m(U, V) = 0$, i.e.,

$$\partial_{v_i} J_m(U, V) = \sum_{k=1}^{n} u_{i,k}^m 2(v_i - x_k), \quad i = 1, \ldots, c, \quad k = 1, \ldots, n$$

or $v_i \sum_{k=1}^{n} u_{i,k}^m = \sum_{k=1}^{n} u_{i,k}^m x_k$. On the other hand, the condition that clusters should not be empty imposes that $\sum_{k=1}^{n} u_{i,k}^m > 0$ for $i = 1, \ldots, c$. Hence

$$\|v_i\|^2 \leq \frac{\left(\sum_{k=1}^{n} u_{i,k}^m \|x_k\|\right)^2}{\left(\sum_{k=1}^{n} u_{i,k}^m\right)^2} \leq \sum_{k=1}^{n} \|x_k\|^2 := r^2.$$

Let R_i $(i = 1, \ldots, c)$ be the Euclidean ball centered at the origin and with radius r in $\mathbb{R}^p$, and let $\mathcal{C} := \Pi_{i=1}^{c} R_i$. The second DC decomposition of the clustering criterion J_m is inspired by the following result.

Theorem 1 *There exists $\rho > 0$ such that the function*

$$H_2(U, V) := \frac{\rho}{2} \|(U, V)\|^2 - J_m(U, V) \tag{14}$$

is convex on $\Delta \times \mathcal{C}$.

Proof Let us consider the function $h : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ defined by

$$h(x, y) = \frac{\rho}{2} x^2 + \frac{\rho}{2n} y^2 - x^m y^2. \tag{15}$$

The Hessian of h is given by:

$$J(x, y) = \begin{pmatrix} \rho - m(m-1) y^2 x^{m-2} & -2m x^{m-1} y \\ -2m x^{m-1} y & \dfrac{\rho}{n} - 2x^m \end{pmatrix}. \tag{16}$$

For all (x, y), $0 \le x \le 1$, $| y | \le \alpha$, we have for the determinant $| J(x, y) |$ of $J(x, y)$:

$$\begin{aligned} | J(x, y) | &= \left(\rho - m(m-1)y^2x^{m-2}\right)\left(\frac{\rho}{n} - 2x^m\right) - 4m^2x^{2m-2}y^2 \\ &= \frac{1}{n}\rho^2 - \left[\frac{m}{n}(m-1)y^2x^{m-2} + 2x^m\right]\rho - 4m^2x^{2m-2}y^2 \\ &\ge \frac{1}{n}\rho^2 - \left(\frac{m}{n}(m-1)\alpha^2 + 2\right)\rho - 4m^2\alpha^2. \end{aligned}$$

provided that $\alpha > 0$ and ρ is larger than the upper zero of this quadratic function, i.e., for

$$\rho \ge \frac{n}{2}\left[\frac{m}{n}(m-1)\alpha^2 + 2 + \sqrt{\left[\frac{m}{n}(m-1)\alpha^2 + 2\right]^2 + \frac{16}{n}m^2\alpha^2}\right]. \tag{17}$$

Note that $\rho > 0$ because $m > 1$. With ρ as large as in (17), the function h is convex on $[0, 1] \times [-\alpha, \alpha]$. Therefore, for $x \to u_{i,k}$ and $y \to \|x_k - v_i\|^2$, the functions

$$\theta_{i,k}(u_{i,k}, v_i) := \frac{\rho}{2}u_{i,k}^2 + \frac{\rho}{2n}\|x_k - v_i\|^2 - u_{i,k}^m\|x_k - v_i\|^2$$

are convex on $\{0 \le u_{i,k} \le 1, \|v_i\| \le r\}$ with ρ as in (17) and

$$\alpha = r + \max_{1\le k\le n}\|x_k\|. \tag{18}$$

As a consequence, the function $h_{i,k}$ defined by

$$h_{i,k}(u_{i,k}, v_i) = \theta_{i,k}(u_{i,k}, v_i) + \frac{\rho}{n}\langle x_k, v_i\rangle - \frac{\rho}{2n}\|x_k\|^2$$

is convex w.r.t. $(u_{i,k}, v_i)$. Finally, since $H_2(U, V) := \frac{\rho}{2}\|(U, V)\|^2 - J_m(U, V) = \sum_{k=1}^n\sum_{i=1}^c h_{i,k}(u_{i,k}, v_i)$, the function $H_2(U, V)$ is convex on $\Delta \times \mathcal{C}$ with ρ as in (17) and $\alpha = r + \max_{1\le k\le n}\|x_k\|$. □

In the sequel, the function H_2 is defined with a ρ satisfying the condition (17). According to the proposition above we can express our second DC decomposition of as follows:

$$J_m(U, V) := G_2(U, V) - H_2(U, V) \tag{19}$$

with the convex function $G_2(U, V) := \frac{\rho}{2}\|(U, V)\|^2$. Now, the basic optimisation problem (2) can be written as

$$\min\left\{\chi_{\Delta\times\mathcal{C}}(U, V) + \frac{\rho}{2}\|(U, V)\|^2 - H_2(U, V) : (U, V) \in (U, V) \in \mathbb{R}^{c\times n} \times \mathbb{R}^{c\times p}\right\}. \tag{20}$$

Solving (19) by DCA: For designing a DCA according to the general DCA scheme from Sect. 2, we first need the computation of $(Y^l, Z^l) \in \partial H_2(U^l, V^l)$ and then have

to solve the convex program

$$\min\left\{\frac{\rho}{2}\|(U,V)\|^2 - \langle (U,V),(Y^l,Z^l)\rangle : (U,V)\in \Delta\times\mathcal{C}\right\}. \tag{21}$$

The function H_2 is differentiable and its gradient at the point (U^l, V^l) is given by:

$$\begin{aligned}(Y^l,Z^l) &= \nabla H_2(U^l,V^l)\\ &= \left(\left(\rho u_{i,k}^l - m(u_{i,k}^l)^{m-1}\|x_k - V_i^l\|^2\right)_{i=1,\ldots,c}^{k=1,\ldots,n},\right.\\ &\quad \left.\left(\rho V_i^l - 2\sum_{k=1}^{n}(V_i^l - x_k)(u_{i,k}^l)^m\right)_{i=1,\ldots,c}\right).\end{aligned} \tag{22}$$

The solution of the auxiliary problem (21) is explicitly computed as

$$\begin{aligned}\left(U^{l+1}\right)^k &= \mathrm{Proj}_{\Delta_k}\left(\left(Y^l\right)^k\right) \quad k=1,\ldots,n,\\ V_i^{l+1} &= \mathrm{Proj}_{R_i}\left(\frac{1}{\rho}(Z^l)_i\right) \quad i=1,\ldots,c.\end{aligned} \tag{23}$$

Remark 2 The DC decomposition (19) is interesting because the resulting DCA is simple: each iteration of DCA consists of computations of the projection of points onto a simplex and/or onto a ball, that all are given in the explicit form (23). So DCAs do not require an iterative method at each iteration as in the first DCA scheme.

3.3 The third DC program: a nice DC reformulation of FCM model

Inspired by Theorem 1 we develop our third approach that allows to solve efficiently the FCM model for any value $m \geq 1$. This approach is more sophisticated than the two former ones, and the resulting DCA seems to be the best.

In an elegant way we introduce a nice DC reformulation of the problem (2) for which the resulting DCA is explicitly determined via a very simple formula.

Let us consider new variables $t_{i,k}$ such that $u_{i,k} = t_{i,k}^2$. The constraint $\sum_{i=1}^c u_{i,k} = 1$ becomes

$$\sum_{i=1}^{c} t_{i,k}^2 = 1 \text{ or } \|t_k\|^2 = 1 \text{ with } t_k = (t_{1,k}, t_{2,k}, \ldots, t_{c,k}) \in \mathbb{R}^c. \tag{24}$$

Let S_k be the Euclidean sphere centered at the origin and of radius 1 in $\mathbb{R}^c$. We can reformulate the problem (2) as:

$$\begin{cases} \min \quad J_{2m}(T, V) := \sum_{k=1}^{n} \sum_{i=1}^{c} t_{i,k}^{2m} ||x_k - v_i||^2 \\ s.t. \quad T = (t_{i,k})_{c\times n} \in \mathcal{S} := \Pi_{k=1}^{n} S_k, \quad V = (v_1, v_2, \ldots, v_c) \in \mathcal{C} := \Pi_{i=1}^{c} R_i \end{cases}. \tag{25}$$

For finding a DC decomposition of the clustering criterion (25) we express it in the form

$$J_{2m}(T, V) = \frac{\rho}{2}(\|T\|^2 + \|V\|^2) - \left[\frac{\rho}{2}\|(T, V)\|^2 - J_{2m}(T, V)\right] \tag{26}$$

with some $\rho > 0$, and from (24) we have for all $(T, V) \in \mathcal{S} \times \mathcal{C}$:

$$J_{2m}(T, V) = \frac{\rho}{2} n + \frac{\rho}{2} \|V\|^2 - H_3(T, V) \tag{27}$$

with

$$H_3(T, V) := \frac{\rho}{2}\|(T, V)\|^2 - J_{2m}(T, V). \tag{28}$$

The next statement is similar to Theorem 1

Theorem 2 *Let $\mathcal{B} := \Pi_{k=1}^{n} B_k$ where B_k is the ball of center 0 and radius 1 in $\mathbb{R}^c$. The function $H_3(T, V)$ is convex on $\mathcal{B} \times \mathcal{C}$ for all values of ρ satisfying*

$$\rho \geq \frac{m}{n}(2m-1)\alpha^2 + 1 + \sqrt{\left[\frac{m}{n}(2m-1)\alpha^2 + 1\right]^2 + \frac{16}{n} m^2 \alpha^2} \tag{29}$$

where α is given by (18).

Proof First, we note that $\rho > 0$ because $m \geq 1$. Otherwise, the proof is similar to the proof of Theorem 1 in which $u_{i,k}$ is now replaced by $t_{i,k}$ and m is replaced by $2m$. □

In the sequel we work with values of ρ and α fulfilling (29). It is clear that for all $T \in \mathcal{B}$ and a given matrix $V \in \mathcal{C}$, the function $J_{2m}(T, V)$ is concave in T (since $H_3(T, V)$ is convex in T due to Theorem 1). Therefore the minimum for $T \in \mathcal{B}$ is obtained at the boundary $\mathcal{S}$ of B_i, i.e. $\mathcal{S}$ contains the minimizers of $J_{2m}(T, V)$ on $\mathcal{B}$ such that

$$\begin{aligned} &\min\left\{\frac{\rho}{2}\|V\|^2 - H_3(T, V) : (T, V) \in \mathcal{B} \times \mathcal{C}\right\} \\ &= \min\left\{\frac{\rho}{2}\|V\|^2 - H_3(T, V) : (T, V) \in \mathcal{S} \times \mathcal{C}\right\}. \end{aligned}$$

Neglecting the first term in (27) and writing $G_3(T, V) = \frac{\rho}{2}\|V\|^2$ (a convex function), we see that the problem (25) can now be reformulated in the form of the DC program

$$\min\{G_3(T, V) - H_3(T, V) : (T, V) \in \mathcal{B} \times \mathcal{C}\},$$

or, equivalently, as :

$$\min\left\{\chi_{\mathcal{B}\times\mathcal{C}}(T,V)+G_3(T,V)-H_3(T,V):(T,V)\in\mathbb{R}^{c\times n}\times\mathbb{R}^{c\times p}\right\} \tag{30}$$

Solving (30) by DCA: The general DCA scheme applied to (30) consists in the construction of two sequences $\left(Y^l, Z^l\right) \in \partial H_3\ \left(T^l, V^l\right)$ and

$$\left(T^{l+1}, V^{l+1}\right) \in \arg\min\left\{\frac{\rho}{2}\|(T,V)\|^2-\langle(T,V),\left(Y^l,Z^l\right)\rangle:\mathcal{B}\times\mathcal{C}\right\}. \tag{31}$$

The function H_3 is differentiable and its gradient at the point (T^l, V^l) is given by:

$$\begin{aligned}\left(Y^l,Z^l\right)&=\nabla H_3\left(U^l,V^l\right)\\&=\left(\left(\rho t_{i,k}^l-2m\left(t_{i,k}^l\right)^{2m-1}\|x_k-v_i^l\|^2\right)_{i=1,\ldots,c}^{k=1,\ldots,n},\right.\\&\qquad\left.\left(\rho V_i^l-2\sum_{k=1}^{n}\left(V_i^l-x_k\right)t_{i,k}^{2m}\right)_{i=1,\ldots,c}\right).\end{aligned} \tag{32}$$

The solution of (31) can be explicitly computed as

$$\begin{aligned}\left(T^{l+1}\right)^k&=\ \mathrm{Proj}_{B_k}\left(\left(Y^l\right)^k\right)\quad k=1,\ldots,n,\\V_i^{l+1}&=\mathrm{Proj}_{R_i}\left(\frac{1}{\rho}\left(Z^l\right)_i\right)\quad i=1,\ldots,c.\end{aligned} \tag{33}$$

Remark 3 The change of variables $t_{i,k}^2 = u_{i,k}$ has crucial advantages. Firstly, working with the function $J_{2m}(T, V)$ the algorithm can treat all cases of $m \geq 1$. Secondly, the feasible domain of T is first transformed into $\mathcal{S}$ (the product of the spheres S_k) and $\mathcal{S}$ is again replaced by $\mathcal{B}$ (the product of the balls B_k) due to the concativity of the function $J_{2m}(T, V)$. This transformation is very useful for DCA: determining the sequence T^l amounts to computing the projection of points onto a ball that is much simpler than the projection of points onto a simplex required in the first two DCA schemes. Therefore, the third DCA scheme is the best one.

3.4 Algorithms

We can now describe in detail the three DCA algorithms for solving the FCM model (2).

DCA1: DCA applied to the first DC program (11)
Initialization: Choose the memberships U^0 and the cluster centers V^0. Let $\epsilon > 0$ be sufficiently small, $0 \leftarrow l$.

Repeat

– Set $(Y^l, Z^l) := \left(\left(mu_{i,k}^{2m-1}\right)_{i=1,\ldots,c}^{k=1,\ldots,n}, 2\sum_{k=1}^{n}(\|x_k - v_i\|^2(v_i - x_k))_{i=1,\ldots,c}\right)$.
– Apply algorithm **GP** until its convergence to get (U^{l+1}, V^{l+1}).
– $l + 1 \leftarrow l$

Until $\| \left(U^{l+1}, V^{l+1}\right) - \left(U^l, V^l\right) \| \leq \epsilon$.

DCA2: DCA applied to the second DC program (20)
*Initialization***:** Choose the memberships U^0 and the cluster centers V^0. Let $\epsilon > 0$ be sufficiently small, $0 \leftarrow l$.
Repeat

– Set $Y^l := \left(\rho u_{i,k}^l - m\left(u_{i,k}^l\right)^{m-1}\|x_k - V_i^l\|^2\right)_{i=1,\ldots,c}^{k=1,\ldots,n}$,
$Z^l := \left(\rho V_i^l - 2\sum_{k=1}^{n}\left(V_i^l - x_k\right)\left(u_{i,k}^l\right)^m\right)_{i=1,\ldots,c}$
– Define (U^{l+1}, V^{l+1}) by setting:

$$\left(U^{l+1}\right)^k = \mathrm{Proj}_{\Delta_k}\left(\left(Y^l\right)^k\right) \quad \text{for } k = 1, \ldots, n,$$

$$V_i^{l+1} = \Pr_{R_i}\left(\frac{1}{\rho}\left(Z^l\right)_i\right) = \begin{cases} \frac{(Z^l)_{i,.}}{\rho} & \text{if } \| \left(Z^l\right)_{i,.} \| \leq \rho r \\ \frac{(Z^l)_{i,.}\, r}{\|(Z^l)_i\|} & \text{otherwise} \end{cases}, \quad (i = 1, \ldots, c).$$

– $l + 1 \leftarrow l$

Until $\| \left(U^{l+1}, V^{l+1}\right) - \left(U^l, V^l\right) \| \leq \epsilon$.

DCA3: DCA applied to the third DC program (30)
*Initialization***:** Choose T^0 and the cluster centers V^0. Let $\epsilon > 0$ be sufficiently small, $0 \leftarrow l$.
Repeat

– Set $Y^l := \left(\rho t_{i,k}^l - 2mt_{i,k}^{2m-1}\|x_k - v_i\|^2\right)_{i=1,\ldots,c}^{k=1,\ldots,n}$,
$Z^l := \left(\rho V_i^l - 2\sum_{k=1}^{n}\left(V_i^l - x_k\right)t_{i,k}^{2m}\right)_{i=1,\ldots,c}$
– Define $\left(T^{l+1}, V^{l+1}\right)$ by setting:

$$V_i^{l+1} = \begin{cases} \frac{(Z^l)_i}{\rho} & \text{if } \| \left(Z^l\right)_i \| \leq \rho r \\ \frac{(Z^l)_{i,.}\, r}{\|(Z^l)_i\|} & \text{otherwise} \end{cases}, \quad (i = 1, \ldots, c), \tag{34}$$

$$\left(T^{l+1}\right)^k = \begin{cases} \left(Y^l\right)^k & \text{if } \| \left(Y^l\right)^k \| \leq 1 \\ \frac{(Y^l)^k}{\|(Y^l)^k\|} & \text{otherwise} \end{cases}, \quad (k = 1, \ldots, n). \tag{35}$$

Until $\| \left(T^{l+1}, V^{l+1}\right) - \left(T^{l}, V^{l}\right) \| \leq \epsilon$.
Output: $u^*_{i,k} = t^l_{i,k}.t^l_{i,k}$ for $i = 1, .., c$ and $k = 1, .., n$.

4 Finding a good starting point of DCA by an alternative FCM-DCA procedure

Finding a good starting point is a challenge in designing solution methods of DC programs by DCA. The research of such a point depends on the structure of the problem being considered and can be done by, for example, a heuristic procedure. Generally speaking a good starting point for DCA must not be a *local minimizer*, because DCA is stationary from such a point. Nevertheless, we observe that from any initial point which is not a local minimizer, the objective function is decreasing rapidly during some first iterations of DCA. The same remark holds for the classical FCM algorithm. That is why we propose an alternative FCM-DCA procedure for (2) which is desribed as follows.

FCM-DCA procedure:
Input: Choose the initial membership matrix U^0 and the cluster center matrix V^0 randomly. Let $maxiter$ be a given integer.
Repeat

- *One iteration of FCM:* Compute the cluster centers $V^l = \left(v^l_1, v^l_2, \ldots, v^l_c\right)$ by the formula

$$v^l_i = \sum_{k=1}^{n} u^m_{ik} x_k \Big/ \sum_{k=1}^{n} u^m_{ik} \quad \forall i = 1, \ldots, c. \tag{36}$$

  Compute the membership matrix U^l by setting

$$u^l_{ik} = \left[\sum_{k=1}^{c} \frac{\|x_k - v_i\|^{2/(m-1)}}{\|x_k - v_j\|^{2/(m-1)}} \right]^{-1}. \tag{37}$$

- *One iteration of DCA:* perform one iteration of **DCA1** or **DCA2** (resp. **DCA3**) from $\left(U^l, Z^l\right)$ (resp. from $\left(T^l, V^l\right)$ with $t^l_{ik} = \sqrt{u_{ik}}$, for $i = 1, \ldots, c$ and for $k = 1, .., n$) to obtain $\left(U^{l+1}, V^{l+1}\right)$ (resp. $\left(T^{l+1}, V^{l+1}\right)$ and set $u^{l+1}_{ik} = \left(t^{l+1}_{ik}\right)^2$).
- $l + 1 \leftarrow l$
- *Until* $l = maxiter$.

If we use the combined FCM-DCA procedure until its convergence, the efficiency of DCA may not be well exploited. An efficient algorithm based on DCA may be a two-phase DCA algorithm in which the first phase deals with some iterations of the combined FCM-DCA procedure and the second phase consists of applying a pure DCA as DCA1, DCA2 or DCA3 starting from the point given by the first phase.

5 Numerical results

The algorithms has been coded in C++ and run on PC Pentium 4 CPU 2.8GHz 1.00Go RAM. We have tested our code on two collections of real data. The first one is composed of 5 problems:

- "PAPILLON" is a well known dataset called "jeux de papillon". Several articles on clustering have discussed this dataset (see Revue Modulad, Le Monde Des Utilisateurs de L'Analyse de Données (1993) no. 11, 7–44).
- "IRIS" is the classical IRIS dataset which is perhaps the best known dataset found in pattern recognition literature. The dataset consists of 3 classes, 50 instances each and 4 numeric attributes where each class refers to a type of iris plant, namely Iris Setosa, Iris Versicolor, Iris Verginica. The first class is linearly separable from the other ones while that latter ones are not linearly separable. The measurements consist of the sepal and petal lengths and widths in centimeters.
- "GENE" is a Gene Expression dataset containing 384 genes that we get from http://faculty.washington.edu/kayee/cluster/.
- "VOTE" is the Congressional Votes dataset (Congressional Quarterly Almanac, 98th Congress, 2nd session 1984, Volume XL: Congressional Quarterly Inc. Washington 1985), it consists of the votes for each of the U.S. House of Representative Congressmen, for 16 key votes, on different subjects (handicap, religion, immigration, army, education ...). For each vote, three answers are possible: yes, nay, and unknown. The individuals are separated into two clusters: democrats (267) and republicans (168).
- "ADN" is the ADN dataset (ftp://genbank.bio.net) that consists of 3, 186 genes, described by 60 DNA sequel elements, called nucleotides or base pairs, with 4 possible modalities (A, C, G or T). These genes are divided into three different clusters: "intron → exon" or "ie" (sometimes called donors, 767 objects), "exon → intron" or "ei" (sometimes called acceptors, 765 objects), and "neither", noted as "n objects").

Table 1 Comparative results for the first collection of data

Data	DCA1			DCA2			DCA3			FCM		
(n, p, c)	N^{o}it	Time	PWPO	N^{o}it	Time	PWPO	N^{o}it	Time	PWPO	N^{o}it	Time	PWPO
PAPILON												
(23,4,5)	20	0.003	91.3	10	0.002	91.3	2	0.001	91.3	18	0.002	91.3
IRIS												
(150,4,3)	23	0.03	91.77	5	0.01	91.77	4	0.01	91.77	15	0.03	89.33
VOTE												
(435,16,2)	16	0.05	87.9	3	0.01	87.9	3	0.01	89.8	19	0.06	83.7
GENE												
(384,17,5)	16	0.67	88.3	8	0.20	90.7	8	0.20	96.2	35	0.73	85.8
ADN												
(3186,60,3)	8	0.78	92	8	0.62	92.4	6	0.55	94	25	1.95	89.8

Table 2 Comparative results for the microarray data "Yeast"

Yeast data set ($n = 2,945$, $p = 15$, $c = 16$)												
m	DCA1			DCA2			DCA3			FCM		
	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time
1.1	64831	31	314	N/A	N/A	N/A	**64061**	120	34	65868	179	564
1.3	64144	357	64	N/A	N/A	N/A	**62593**	114	30	62681	543	1886
1.5	43398	54	301	N/A	N/A	N/A	**43341**	155	68	44367	123	213
1.7	44981	42	187	N/A	N/A	N/A	**43792**	94	30	43939	48	118
1.9	45012	65	84	N/A	N/A	N/A	**43956**	72	29	44643	42	87
2	**43687**	61	70	**43687**	37	32	**43687**	44	24	43738	37	57
3	43710	45	34	**43687**	32	24	**43687**	42	22	43738	24	19
4	45012	20	19	43722	23	18	**43676**	17	16	43738	21	18
5	**43687**	21	18	**43687**	19	15	**43687**	15	14	43738	19	16

The second collection of data consists of five microarray data sets:

- "Yeast" is composed of 2, 945 genes in the space of 15 dimensions and can be downloaded from http://genomics.stanford.edu.
- "Serum" is composed of 517 genes in the space of 12 dimensions; the entire dataset is available at http://genome-www.stanford.edu/serum.
- "Human cancer" is composed of 60 human cancer cell lines in the space of 768 dimensions and is available at http://discover.nci.nih.gov/nature2000/.
- "Breast Cancer" is composed of 84 human cancer cell lines in the space of 1753 dimensions and is available at http://bioinformatics.upmc.edu/GE2/GEDA.html.
- "Ovarian Cancer" is composed of 39 human cancer cell lines in the space of 7039 dimensions and is available at http://bioinformatics.upmc.edu/GE2/GEDA.html.

In our first experiment we compare the performance of three versions of DCA and the standard FCM algorithm for the first data collection. The program of FCM is available at http://www-igbmc.u-strasbg.fr/projets/fcm/. We take the fuzziness parameter $m = 2$ for this experiment. The tolerance ϵ is equal to 10^{-7}. All the algorithm are started from the same randomly generated point. The numerical results are reported in Table 1.

In the second experiment we consider the second data collection with different choices of the fuzziness parameter m. A major problem in applying the FCM method for clustering microarray data is the choice of the parameter m. It has been shown in Dembélé and Kastner (2003) that the commonly used value $m = 2$ is not appropriate for FCM in some datasets, and that optimal values for m vary widely from one dataset to another. The numerical results on several values of $m \in [1.1, 5]$ of three DCA schemes (with FCM-DCA procedure for finding a starting point) and the FCM algorithm are reported in Tables 2, 3, 4, 5 and 6. We take $maxiter = 5$ in the FCM-DCA procedure. To emphasize the comparison we display in Fig. 1 the diagram representing the performance of the four algorithms for the "Yeast" dataset.

Table 3 Comparative results for the microarray data "Serum"

Serum data set ($n = 517$, $p = 12$, $c = 8$)												
m	DCA1			DCA2			DCA3			FCM		
	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time
1.1	12237	56	2.2	N/A	N/A	N/A	**12234**	47	0.92	13262	191	1.98
1.3	**9572**	78	5.2	N/A	N/A	N/A	**9572**	95	2.0	10231	176	5.3
1.5	8642	52	6.7	N/A	N/A	N/A	**8019**	109	2.4	9431	71	3.1
1.7	6034	41	4.1	N/A	N/A	N/A	**6013**	56	1.2	6068	29	1.4
1.9	6079	85	2.2	N/A	N/A	N/A	**6009**	55	1.3	6081	11	0.7
2	6023	17	0.76	**6001**	19	0.68	**6001**	19	0.67	6083	13	0.6
3	**6069**	12	0.61	**6069**	8	0.52	**6069**	10	0.43	6199	11	0.6
4	6079	17	0.67	**6069**	15	0.60	**6069**	15	0.52	6201	15	0.7
5	6092	18	0.87	6092	13	0.57	**6073**	11	0.49	6201	19	0.8

Table 4 Comparative results for the "Human cancer" data

Human cancer data set ($n = 60$, $p = 728$, $c = 9$)												
m	DCA1			DCA2			DCA3			FCM		
	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time
1.1	70300	156	25.1	N/A	N/A	N/A	**72235**	87	15.5	70315	118	22.4
1.3	**77599**	178	30.2	N/A	N/A	N/A	**77599**	94	19.4	77638	124	25.1
1.5	107792	237	39.6	N/A	N/A	N/A	**107824**	123	29.3	107858	280	52.4
1.7	**124698**	35	8.4	N/A	N/A	N/A	**124698**	42	7.9	**124698**	50	10.2
1.9	124602	85	13.4	N/A	N/A	N/A	**124496**	36	5.9	124698	36	8.6
2	124662	55	10.5	**124654**	28	5.2	**124654**	36	5.8	124697	31	8.4
3	**124654**	19	5.1	**124654**	17	4.6	**124654**	15	3.5	124697	21	6.3
4	124692	23	5.8	**124654**	19	4.7	**124654**	19	4.5	124699	19	5.9
5	**124654**	22	5.8	**124654**	22	4.8	**124654**	19	4.5	124699	18	5.2

In the third experiment we are interested in the effect of FCM-DCA procedure for finding the starting point of DCA. We compare two variants of DCA3 (with and without FCM-DCA procedure) on the "Human cancer" dataset. The comparative results are presented in Fig. 2.

In these tables we use the following notations:

- N^{o}it: number of iterations
- Time: CPU times in seconds
- PWPO: the percent of the well placed objects
- J_c is given by $J_c = \sum_{i=1..n} \min_{k=1..c} \|x_i - v_k\|^2$, the obtained overall quadratic deviation between data points and corresponding cluster centers, termed "cluster cost" (Note : this is not the clustering criterion (1) !).

Table 5 Comparative results for the "Breast cancer" data

Breast cancer data set ($n = 84$, $p = 1{,}753$, $c = 2$)												
m	DCA1			DCA2			DCA3			FCM		
	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time
1.1	**1878.3**	18	1.45	N/A	N/A	N/A	**1878.3**	15	1.12	1892.5	22	1.42
1.3	**1897.8**	16	1.45	N/A	N/A	N/A	**1897.8**	15	1.10	1902.8	19	1.40
1.5	1909.8	13	1.1	N/A	N/A	N/A	**1909.5**	12	1.0	1929.1	19	1.4
1.7	**2011.3**	20	1.6	N/A	N/A	N/A	**2011.3**	17	1.2	2040.4	50	3.8
1.9	**2302.3**	25	1.9	N/A	N/A	N/A	**2302.3**	18	1.3	2317.8	141	13.8
2	**2302.3**	20	1.6	**2302.3**	16	1.3	**2302.3**	18	1.3	2317.9	85	6.2
3	**2302.3**	19	1.6	**2302.3**	17	1.4	**2302.3**	15	1.2	2317.9	30	1.98
4	**2302.3**	20	1.8	**2302.3**	19	1.4	**2302.3**	19	1.4	2317.9	24	1.57
5	**2302.3**	16	1.4	**2302.3**	13	1.2	**2302.3**	12	1.2	2317.9	22	1.53

Table 6 Comparative results for the "Ovarian cancer" data

Ovarian cancer data set ($n = 39$, $p = 7{,}039$, $c = 2$)												
m	DCA1			DCA2			DCA3			FCM		
	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time	J_c	N^{o}it	Time
1.1	**253.67**	18	2.65	N/A	N/A	N/A	**253.67**	17	2.17	263.54	47	5.9
1.3	263.67	19	2.65	N/A	N/A	N/A	**253.59**	15	2.10	263.15	31	3.9
1.5	254.85	18	2.64	N/A	N/A	N/A	**253.59**	13	2.0	263.25	26	3.2
1.7	**253.77**	20	2.70	N/A	N/A	N/A	**253.77**	17	2.0	263.72	25	3.1
1.9	**255.21**	22	2.75	N/A	N/A	N/A	**255.21**	15	1.9	264.6	20	2.5
2	**255.24**	20	2.6	**255.24**	16	2.0	**255.24**	18	2.1	265.3	19	2.4
3	**271.43**	18	2.5	**271.43**	17	1.9	**271.43**	15	2.0	282.9	30	3.98
4	**303.82**	19	2.5	**303.82**	14	1.6	**303.82**	14	1.5	311.76	58	7.4
5	**345.26**	18	2.4	**345.26**	14	1.6	**345.26**	13	1.4	354.58	182	23.13

- N/A means that DCA2 cannot be applied to this case ($m < 2$).
- The bold face means "minimum value in the line".

From our numerical results we see that:

- DCA is an efficient approach for fuzzy clustering: all three algorithms DCA are better than the FCM standard algorithm.
- DCA3 is the best among these three DCA schemes: the cluster cost and the CPU time are always smallest.
- The combined DCA-FCM procedure is efficient for finding a good starting point of DCA.
- Both FCM and DCA are quite sensitive to the fuzziness parameter m, and the optimal values for m vary widely from one dataset to another.

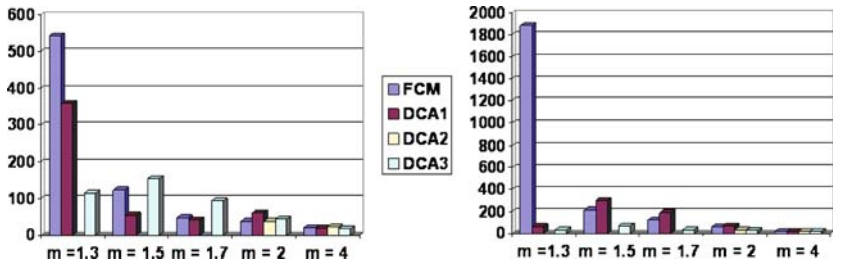

Fig. 1 Comparative results on the "Yeast" data: the number of iterations (*left*) and the CPU time (*right*)

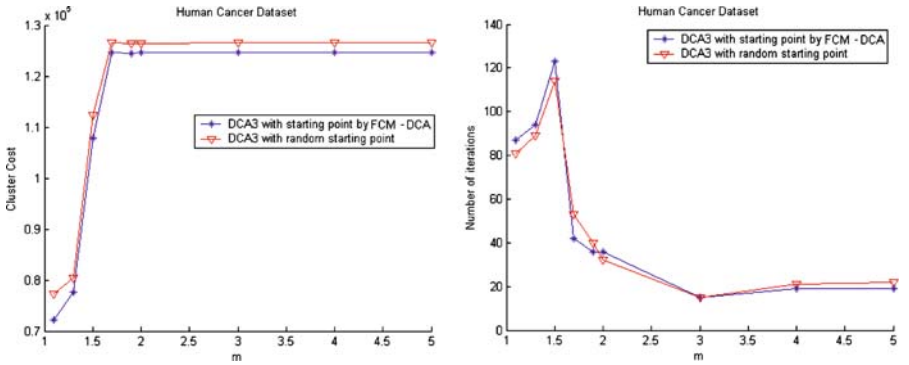

Fig. 2 DCA3 with and without FCM-DCA procedure, the cluster cost (*left*) and the number of iterations (*right*)

6 Conclusion

We have rigorously studied the DC programming and DCA for fuzzy clustering. The classical FCM model has been reformulated as DC programs with different DC decompositions. The effect of DC decomposition and the starting point are well exploited for obtaining fast and robust algorithms. The results obtained by DCA2 and DCA3 are interesting: they require only the projection of points onto simplices and/or balls that are explicitly computed. In particular, the DCA3 algorithm is very simple and inexpensive: the gain of CPU time with respect to FCM algorithm is up to 62 times (see Table2, $m = 1.3$). The numerical results on several real data sets show that DCA is an efficient approach for fuzzy clustering in large data sets of high dimension and it is superior to the FCM algorithm in both running-time and quality of solutions.

References

Alon N, Spencer JH (1991) The probabilistic method. Wiley, New York

Arora S, Kannan R (2001) Learning mixtures of arbitrary Gaussians. In: Proceedings of 33rd annual ACM symposium on theory of computing, pp 247–257

Bradley BS, Mangasarian OL (1998) Feature selection via concave minimization and support vector machines. In: Shavlik J (ed) Machine learning. Proceedings of the 15th international conferences (ICML 1998). Morgan Kaufman, San Francisco, pp 82–90

Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithm. Plenum Press, New York
Dhilon IS, Korgan J, Nicholas C (2003) Feature selection and document clustering. In: Berry MW (ed) A comprehensive survey of text mining. Springer, Berlin, pp 73–100
Dembélé D, Kastner P (2003) Fuzzy c-means clustering method for clustering microarray data. Bioinformatics 19(8):573–580
Duda RO, Hart PE (1972) Pattern classification and scene analysis. Wiley, New York
Feder T, Greene D (1988) Optimal algorithms for approximate clustering. Proc STOC Chicago, Illinois, pp 434–444, ISBN: 0-89791-264-0
Fisher D (1987) Knowledge acquisition via incremental conceptual clustering. Mach Learn 2:139–172
Fukunaga K (1990) Statistical pattern recognition. Academic, New York
Krause N, Singer Y (2004) Leveraging the margin more carefully. In: Proceedings of International conference on machine learning ICML, V Banff, Alberta, Canada pp. 63–70, ISBN: 1-58113-828-5
Klawonn F, Höppner F (2003) What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier. In: Berthold MR, Lenz H-J, Bradley E, Kruse R, Borgelt C (eds) Advances in intelligent data analysis. Springer, Berlin, pp 254–264
Le Thi HA (1997) Contribution à l'optimisation non convexe et l'optimisation globale: théorie, algorithmes et applications. Habilitation à Diriger des Recherches, Université de Rouen, France
Le Thi HA, Pham Dinh T (1997) Solving a class of linearly constrained indefinite quadratic problems by DC algorithms. J Global Optim 11(3):253–285
Le Thi HA, Pham Dinh T (2005) 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
Le Thi HA, Belghiti T, Pham Dinh T (2007) A new efficient algorithm based on DC programming and DCA for Clustering. J Global Optim 37:593–608
Le Thi HA, Le Hoai M, Pham Dinh T (2007) Optimization based DC programming and DCA for Hierarchical Clustering. Eur J Oper Res 183:1067–1085
Liu Y, Shen X, Doss H (2005) Multicategory ψ-learning and support vector machine: computational tools. J Comput Graph Stat 14:219–236
Liu Y, Shen X (2006) Multicategoryψ-learning. J Am Stat Assoc 101:500–509
Mangasarian OL (1997) Mathematical programming in data mining. Data Min Knowl Discov 1:183–201
MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley symposium on mathematical statistics and probability. University of California Press, Berkeley 1:281–297
Neumann J, Schnörr C, Steidl G (2004) SVM-based feature selection by direct objective minimisation. Pattern Recognition. In: Proceedings of the 26th DAGM symposium, pp 212–219
Pham Dinh T, Le Thi HA (1998) DC optimization algorithms for solving the trust region subproblem. SIAM J Optim 8:476–505
Polyak B (1987) Introduction to optimization. Optimization Software, Inc., Publication Division, New York
Rajapakse JC, Giedd JN, Rapoport JL (2004) Statistical approach to segmentation of single-channel cerebral MR images. IEEE Trans Medical Imaging 16: 176–186
Rockfellar RT (1970) Convex Analysis. Princeton: Princeton University
Ronan C, Fabian S, Jason W, Léon B (2006) Trading convexity for scalability. In: International conference on machine learning (ICML), Pittsburgh, Pennsylvania, pp. 201–208, ISBN: 1-59593-383-2
Shen X, Tseng GC, Zhang X, Wong WH (2003) ψ-learning. J Am Stat Assoc 98:724–734
Weber S, Schüle T, Schnörr C (2005) Prior learning and convex–concave regularization of binary tomography. Electron Notes Discr Math 20:313–327
Yuille AL, Rangarajan A (2003) The convex conCave procedure (CCCP). Neural Comput 15:915-936