Pattern Recognition 44 (2011) 70–77

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Solving the minimum sum-of-squares clustering problem by hyperbolic smoothing and partition into boundary and gravitational regions

Adilson Elias Xavier *, Vinicius Layter Xavier

Department of Systems Engineering and Computer Science, Graduate School of Engineering (COPPE), Federal University of Rio de Janeiro, P.O. Box 68511, Rio de Janeiro RJ 21941-972, Brazil

ARTICLE INFO

Article history:
Received 11 January 2010
Received in revised form
20 May 2010
Accepted 6 July 2010

Keywords:
Cluster analysis
Min-sum-min problems
Nondifferentiable programming
Smoothing

ABSTRACT

This article considers the minimum sum-of-squares clustering (MSSC) problem. The mathematical modeling of this problem leads to a *min-sum-min* formulation which, in addition to its intrinsic bi-level nature, has the significant characteristic of being strongly nondifferentiable. To overcome these difficulties, the proposed resolution method, called hyperbolic smoothing, adopts a smoothing strategy using a special C^{∞} differentiable class function. The final solution is obtained by solving a sequence of low dimension differentiable unconstrained optimization subproblems which gradually approach the original problem. This paper introduces the method of partition of the set of observations into two nonoverlapping groups: "data in frontier" and "data in gravitational regions". The resulting combination of the two methodologies for the MSSC problem has interesting properties, which drastically simplify the computational tasks.

1. Introduction

Cluster analysis deals with the problems of classification of a set of patterns or observations, in general represented as points in a multidimensional space, into clusters, following two basic and simultaneous objectives: patterns in the same clusters must be similar to each other (homogeneity objective) and different from patterns in other clusters (separation objective) [1–3].

Clustering is an important problem that appears in a broad spectrum of applications, whose intrinsic characteristics engender many approaches to this problem, as described by Dubes and Jain [4], Jain and Dubes [5] and Hansen and Jaumard [6].

Clustering analysis has been used traditionally in disciplines such as: biology, biometry, psychology, psychiatry, medicine, geology, marketing and finance. Clustering is also a fundamental tool in modern technology applications, such as: pattern recognition, data mining, web mining, image processing, machine learning and knowledge discovering.

In this paper, a particular clustering problem formulation is considered. Among many criteria used in cluster analysis, the most natural, intuitive and frequently adopted criterion is the minimum sum-of-squares clustering (MSSC). This criterion corresponds to the minimization of the sum-of-squares of distances of observations to their cluster means, or equivalently, to the minimization of within-group sum-of-squares. It is a criterion for both the homogeneity and the separation objectives. According to the Huygens Theorem, minimizing the within-cluster inertia of a partition (homogeneity within the cluster) is equivalent to maximizing the between-cluster inertia (separation between clusters).

The minimum sum-of-squares clustering (MSSC) formulation produces a mathematical problem of global optimization. It is both a nondifferentiable and a nonconvex mathematical problem, with a large number of local minimizers.

There are two main strategies for solving clustering problems: hierarchical clustering methods and partition clustering methods. Hierarchical methods produce a hierarchy of partitions of a set of observations. Partition methods, in general, assume a given number of clusters and, essentially, seek the optimization of an objective function measuring the homogeneity within the clusters and/or the separation between the clusters.

For the sake of completeness, we present first the Hyperbolic Smoothing Clustering Method (HSCM), Xavier [7]. Basically the method performs the smoothing of the nondifferentiable *min-sum-min* clustering formulation. This technique was developed through an adaptation of the hyperbolic penalty method originally introduced by Xavier [8]. By smoothing, we fundamentally mean the substitution of an intrinsically nondifferentiable two-level problem by a C^{∞} unconstrained differentiable single-level alternative.

Additionally, the paper presents a new, faster, methodology. The basic idea is the partition of the set of observations into two nonoverlapping parts. By using a conceptual presentation, the first set corresponds to the observation points relatively close to two or more centroids. This set of observations, named boundary

* Corresponding author.
E-mail addresses: adilson@cos.ufrj.br (A.E. Xavier), vinicius@cos.ufrj.br (V.L. Xavier).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.07.004

band points, can be managed by using the previously presented smoothing approach. The second set corresponds to observation points significantly closer to a single centroid in comparison with others. This set of observations, named gravitational points, is managed in a direct and simple way, offering much faster performance.

This work is organized in the following way. A step-by-step definition of the minimum sum-of-squares clustering problem is presented in the next section. The original hyperbolic smoothing approach and the derived algorithm are presented in Section 3. The boundary and gravitational regions partition scheme and the new derived algorithm are presented in Section 4. Computational results are presented in Section 5. Brief conclusions are drawn in Section 6.

2. The minimum sum-of-squares clustering problem

Let $S=\{s_1,\ldots,s_m\}$ denote a set of m patterns or observations from an Euclidean n-space, to be clustered into a given number q of disjoint clusters. To formulate the original clustering problem as a *min-sum-min* problem, we proceed as follows. Let x_i, $i=1,\ldots,q$ be the centroids of the clusters, where each $x_i \in \mathbb{R}^n$. The set of these centroid coordinates will be represented by $X \in \mathbb{R}^{nq}$. Given a point s_j of S, we initially calculate the Euclidian distance from s_j to the center in X that is nearest. This is given by

$$z_j = \min_{i=1,\ldots,q} \|s_j - x_i\|_2. \tag{1}$$

The most frequent measurement of the quality of a clustering associated to a specific position of q centroids is provided by the sum of the squares of these distances, which determines the MSSC problem:

$$\begin{aligned} &\text{minimize} \quad \sum_{j=1}^{m} z_j^2 \\ &\quad \text{subject to} \quad z_j = \min_{i=1,\ldots,q} \|s_j - x_i\|_2, \quad j=1,\ldots,m \end{aligned} \tag{2}$$

3. The hyperbolic smoothing clustering method

Considering its definition, each z_j must necessarily satisfy the following set of inequalities:

$$z_j - \|s_j - x_i\|_2 \le 0, \quad i=1,\ldots,q. \tag{3}$$

Substituting these inequalities for the equality constraints of problem (2), the relaxed problem is produced:

$$\begin{aligned} &\text{minimize} \quad \sum_{j=1}^{m} z_j^2 \\ &\quad \text{subject to} \quad z_j - \|s_j - x_i\|_2 \le 0, \quad j=1,\ldots,m, \; i=1,\ldots,q. \end{aligned} \tag{4}$$

Since the variables z_j are not bounded from below, the optimum solution of the relaxed problem will be $z_j=0$, $j=1$, …, m. In order to obtain the desired equivalence, we must, therefore, modify problem (4). We do so by first letting $\varphi(y)$ denote $\max\{0, y\}$ and then observing that, from the set of inequalities in (4), it follows that

$$\sum_{i=1}^{q} \varphi(z_j - \|s_j - x_i\|_2) = 0, \quad j=1,\ldots,m. \tag{5}$$

Using (5) in place of the set of inequality constraints in (4), we would obtain an equivalent problem maintaining the undesirable property that $z_j, j=1$, …, m still has no lower bound. Considering, however, that the objective function of problem (4) will force each z_j, $j=1,\ldots,m$, downward, we can think of bounding the latter variables from below by including an ε perturbation in (5). So, the following modified problem is obtained:

$$\begin{aligned} &\text{minimize} \quad \sum_{j=1}^{m} z_j^2 \\ &\quad \text{subject to} \quad \sum_{i=1}^{q} \varphi(z_j - \|s_j - x_i\|_2) \ge \varepsilon, \quad j=1,\ldots,m \end{aligned} \tag{6}$$

for $\varepsilon > 0$. Since the feasible set of problem (2) is the limit of that of (6) when $\varepsilon \to 0_+$, we can then consider solving (2) by solving a sequence of problems like (6) for a sequence of decreasing values for ε that approaches 0.

Analyzing the problem (6), the definition of function φ endows it with an extremely rigid nondifferentiable structure, which makes its computational solution very hard. In view of this, the numerical method we adopt for solving problem (1), takes a smoothing approach. From this perspective, let us define the function:

$$\phi(y,\tau) = (y + \sqrt{y^2 + \tau^2})/2 \tag{7}$$

for $y \in \mathbb{R}$ and $\tau > 0$.

Function ϕ has the following properties:

(a) $\phi(y,\tau) > \varphi(y), \forall \tau > 0$;
(b) $\lim_{\tau\to 0}\phi(y,\tau) = \varphi(y)$;
(c) $\phi(y,\tau)$ is an increasing convex C^∞ function in variable y.

By using function ϕ in the place of function φ, the problem

$$\begin{aligned} &\text{minimize} \quad \sum_{j=1}^{m} z_j^2 \\ &\quad \text{subject to} \quad \sum_{i=1}^{q} \phi(z_j - \|s_j - x_i\|_2, \tau) \ge \varepsilon, \quad j=1,\ldots,m \end{aligned} \tag{8}$$

is produced.

Now, the Euclidean distance $\|s_j - x_i\|_2$ is the single nondifferentiable component on problem (8). So, to obtain a completely differentiable problem, it is still necessary to smooth it. For this purpose, let us define the function

$$\theta(s_j, x_i, \gamma) = \sqrt{\sum_{l=1}^{n} (s_j^l - x_i^l)^2 + \gamma^2} \tag{9}$$

for $\gamma > 0$.

Function θ has the following properties:

(a) $\lim_{\gamma\to 0}\theta(s_j,x_i,\gamma) = \|s_j - x_i\|_2$;
(b) θ is a C^∞ function.

By using function θ in place of the distance $\|s_j - x_i\|_2$, the following completely differentiable problem is now obtained:

$$\begin{aligned} &\text{minimize} \quad \sum_{j=1}^{m} z_j^2 \\ &\quad \text{subject to} \quad \sum_{i=1}^{q} \phi(z_j - \theta(s_j, x_i, \gamma), \tau) \ge \varepsilon, \quad j=1,\ldots,m. \end{aligned} \tag{10}$$

So, the properties of functions ϕ and θ allow us to seek a solution to problem (6) by solving a sequence of subproblems like problem (10), produced by the decreasing of the parameters $\gamma \to 0$, $\tau \to 0$, and $\varepsilon \to 0$.

Since $z_j \ge 0, j=1,\ldots,m$, the objective function minimization process will work for reducing these values to the utmost. On the other hand, given any set of centroids x_i, $i=1,\ldots,q$, due to property (c) of the hyperbolic smoothing function ϕ, the constraints of problem (10) are a monotonically increasing function in z_j. So, these constraints will certainly be active and problem (10) will finally be equivalent to problem:

$$\text{minimize} \quad \sum_{j=1}^{m} z_j^2$$
$$\text{subject to} \quad h_j(z_j,x) = \sum_{i=1}^{q} \phi(z_j - \theta(s_j,x_i,\gamma),\tau) - \varepsilon = 0, \quad j=1,\ldots,m. \tag{11}$$

The dimension of the variable domain space of problem (11) is $(nq+m)$. As, in general, the value of the parameter m, the cardinality of the set S of the observations s_j, is large, problem (11) has a large number of variables. However, it has a separable structure, because each variable z_j appears only in one equality constraint. Therefore, as the partial derivative of $h(z_j,x)$ with respect to z_j, $j=1,\ldots,m$ is not equal to zero, it is possible to use the Implicit Function Theorem to calculate each component z_j, $j=1,\ldots,m$ as a function of the centroid variables x_i, $i=1,\ldots,q$. In this way, the unconstrained problem

$$\text{minimize} \quad f(x) = \sum_{j=1}^{m} z_j(x)^2 \tag{12}$$

is obtained, where each $z_j(x)$ results from the calculation of a zero of each equation

$$h_j(z_j,x) = \sum_{i=1}^{q} \phi(z_j - \theta(s_j,x_i,\gamma),\tau) - \varepsilon = 0, \quad j=1,\ldots,m. \tag{13}$$

Due to property (c) of the hyperbolic smoothing function, each term ϕ above is strictly increasing with variable z_j and therefore the equation has a single zero.

Again, due to the Implicit Function Theorem, the functions $z_j(x)$ have all derivatives with respect to the variables x_i, $i=1,\ldots,q$, and therefore it is possible to calculate the gradient of the objective function of problem (12),

$$\nabla f(x) = \sum_{j=1}^{m} 2 z_j(x) \nabla z_j(x), \tag{14}$$

where

$$\nabla z_j(x) = -\nabla h_j(z_j,x) \Big/ \frac{\partial h_j(z_j,x)}{\partial z_j}, \tag{15}$$

while $\nabla h_j(z_j,x)$ and $\partial h_j(z_j,x)/\partial z_j$ are obtained from Eqs. (7), (9) and (13).

In this way, it is easy to solve problem (12) by making use of any method based on first order derivative information. Finally, it must be emphasized that problem (12) is defined on an (nq)-dimensional space, so it is a small problem, since the number of clusters, q, is, in general, very small for real applications.

The solution of the original clustering problem can be obtained by using the Hyperbolic Smoothing Clustering Algorithm, described below in a simplified form.

The simplified HSC algorithm.

```
Initialization Step:   Choose initial values: $x^0, \gamma^1, \tau^1, \varepsilon^1$.
    Choose values $0 < \rho_1 < 1, 0 < \rho_2 < 1, 0 < \rho_3 < 1$; let $k = 1$.
Main Step:    Repeat until a stopping rule is attained
    Solve problem (12) with $\gamma = \gamma^k$, $\tau = \tau^k$ and $\varepsilon = \varepsilon^k$, starting at
    the initial point $x^{k-1}$ and let $x^k$ be the solution obtained.
Let $\gamma^{k+1} = \rho_1 \gamma^k$, $\tau^{k+1} = \rho_2 \tau^k$, $\varepsilon^{k+1} = \rho_3 \varepsilon^k$, $k := k+1$.
```

Just as in other smoothing methods, the solution to the clustering problem is obtained, in theory, by solving an infinite sequence of optimization problems. In the HSC algorithm, each problem to be minimized is unconstrained and of low dimension.

Notice that the algorithm causes τ and γ to approach 0, so the constraints of the subproblems as given in (10) tend to those of (6). In addition, the algorithm causes ε to approach 0, so, in a simultaneous movement, the solved problem (6) gradually approaches the original MSSC problem (2).

4. The accelerated hyperbolic smoothing clustering method

The calculation of the objective function of the problem (12) demands the determination of the zeros of m equations (13), one equation for each observation point. This is the most relevant computational task associated to the HSC algorithm.

In this section, a faster procedure is presented. The basic idea is the partition of the set of observations into two nonoverlapping regions. By using a conceptual presentation, the first region corresponds to the observation points that are relatively close to two or more centroids. The second region corresponds to the observation points that are significantly close to a unique centroid in comparison with the other ones.

So, the first part J_B is the set of boundary observations and the second is the set J_G of gravitational observations. Considering this partition, Eq. (12) can be expressed in the following way:

$$\text{minimize } f(x) = \sum_{j=1}^{m} z_j(x)^2 = \sum_{j \in J_B} z_j(x)^2 + \sum_{j \in J_G} z_j(x)^2, \tag{16}$$

so, the objective function can be presented in the form

$$\text{minimize } f(x) = f_B(x) + f_G(x), \tag{17}$$

where the two components are completely independent.

The first part of expression (17), associated with the boundary observations, can be calculated by using the previous presented smoothing approach, see (12) and (13):

$$\text{minimize } f_B(x) = \sum_{j \in J_B} z_j(x)^2, \tag{18}$$

where each $z_j(x)$ results from the calculation of a zero of each equation

$$h_j(z_j,x) = \sum_{i=1}^{q} \phi(z_j - \theta(s_j,x_i,\gamma),\tau) - \varepsilon = 0, \quad j \in J_B. \tag{19}$$

The second part of expression (17) can be calculated by using a faster procedure, as we will show right away.

Let us define the two parts in a more rigorous form. Let $\overline{x}_i, i=1,\ldots,q$ be a referential position of centroids of the clusters taken in the iterative process.

The boundary concept in relation to the referential point $\overline{x}$ can be easily specified by defining a δ band zone between neighboring centroids. For a generic point $s \in \mathbb{R}^n$, we define the first and second nearest distances from s to the centroids:

$$d_1(s,\overline{x}) = \| s - \overline{x}_{i_1} \| = \min_i \| s - \overline{x}_i \|, \tag{20}$$

$$d_2(s,\overline{x}) = \| s - \overline{x}_{i_2} \| = \min_{i \neq i_1} \| s - \overline{x}_i \|, \tag{21}$$

where i_1 and i_2 are the labeling indexes of these two nearest centroids.

By using the above definitions, let us define precisely the δ boundary band zone:

$$Z_\delta(\overline{x}) = \{ s \in \mathbb{R}^n \,|\, d_2(s,\overline{x}) - d_1(s,\overline{x}) < 2\delta \} \tag{22}$$

and the gravity region, which is the complementary space:

$$G_\delta(\bar{x}) = \{s \in \mathbb{R}^n - Z_\delta(\bar{x})\}. \tag{23}$$

Fig. 1 illustrates in $\mathbb{R}^2$ the $Z_\delta(\bar{x})$ and $G_\delta(\bar{x})$ partitions. The central lines form the Voronoy polygon associated with the referential centroids $\bar{x}_i, i=1,\dots,q$. The region between two lines parallel to Voronoy lines constitutes the boundary band zone $Z_\delta(\bar{x})$.

Now, the sets J_B and J_G can be defined in a precise form

$$J_B(\bar{x}) = \{j=1,\dots,m \mid s_j \in Z_\delta(\bar{x})\}, \tag{24}$$

$$J_G(\bar{x}) = \{j=1,\dots,m \mid s_j \in G_\delta(\bar{x})\}. \tag{25}$$

Proposition 1. *Let s be a generic point belonging to the gravity region $G_\delta(\bar{x})$, with nearest centroid i_1. Let x be the current position of the centroids. Let $\Delta x = \max_i \|x_i - \bar{x}_i\|$ be the maximum displacement of the centroids.*

If $\Delta x < \delta$ then s will continue to be nearer to centroid x_{i_1} than to any other one, so

$$\min_{i \neq i_1} \|s - x_i\| - \|s - x_{i_1}\| \geq 0. \tag{26}$$

Proof.

$$\min_{i \neq i_1} \|s - x_i\| - \|s - x_{i_1}\| = \min_{i \neq i_1} \|s - \bar{x}_i + \bar{x}_i - x_i\| - \|s - \bar{x}_{i_1} + \bar{x}_{i_1} - x_{i_1}\| \tag{27}$$

$$\geq \min_{i \neq i_1} \|s - \bar{x}_i\| - \|\bar{x}_i - x_i\| - \|s - \bar{x}_{i_1}\| - \|\bar{x}_{i_1} - x_{i_1}\| \tag{28}$$

$$\geq 2\delta - 2\Delta x \geq 0. \quad \square \tag{29}$$

Since $\delta \geq \Delta x$, Proposition 1 makes it possible to calculate exactly expression (16) in a very fast way. First, let us define the subsets of gravity observations associated with each referential centroid:

$$J_i(\bar{x}) = \left\{ j \in J_G \mid \min_{l=1,\dots,q} \|s_j - \bar{x}_l\| = \|s_j - \bar{x}_i\| \right\}. \tag{30}$$

The center of the observations in each nonempty subset is given by

$$v_i = \frac{1}{|J_i|} \sum_{s_j \in J_i} s_j, \quad \forall i = 1,\dots,q. \tag{31}$$

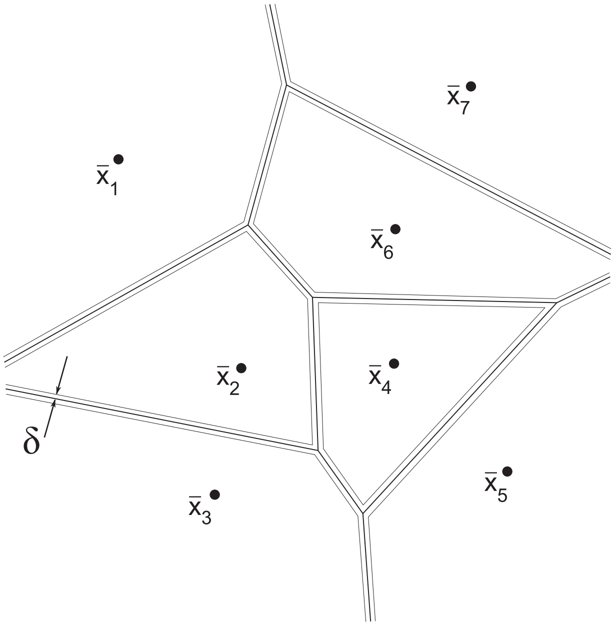

Fig. 1. The $Z_\delta(\bar{x})$ and $G_\delta(\bar{x})$ partitions.

Let us consider the second sum in expression (16). It will be computed by taking into account the centers defined above.

$$\text{minimize } f_G(x) = \sum_{j \in J_G} z_j(x)^2 = \sum_{i=1}^{q} \sum_{j \in J_i} \|s_j - x_i\|^2 \tag{32}$$

$$= \sum_{i=1}^{q} \sum_{j \in J_i} \|s_j - v_i + v_i - x_i\|^2 \tag{33}$$

$$= \sum_{i=1}^{q} \left[\sum_{j \in J_i} \|s_j - v_i\|^2 + 2(v_i - x_i) \sum_{j \in J_i} (s_j - v_i) + \sum_{j \in J_i} \|x_i - v_i\|^2 \right]. \tag{34}$$

Eq. (31) implies that

$$\sum_{j \in J_i} (s_j - v_i) = 0, \tag{35}$$

and then:

$$\text{minimize} f_G(x) = \sum_{i=1}^{q} \sum_{j \in J_i} \|s_j - v_i\|^2 + \sum_{i=1}^{q} |J_i| \|x_i - v_i\|^2. \tag{36}$$

When the position of centroids x_i, $i=1,\dots,q$ moves during the iterative process, the value of the first sum in (36) assumes a constant value, since the vectors s and v are fixed. On the other hand, for the calculation of the second sum, it is only necessary to calculate q distances, $\|v_i - x_i\|$, $i=1,\dots,q$.

The gradient of the second part of the objective function is easily calculated by

$$\nabla f_G(x) = \sum_{i=1}^{q} 2|J_i|(x_i - v_i), \tag{37}$$

where the vector $(v_i - x_i)$ must be in $\mathbb{R}^{nq}$, so it has the first $(i-1)q$ components and the last $l = iq+1,\dots,nq$ components equal zero.

Therefore, if it is observed that $\delta \geq \Delta x$ within the iterative process, the calculation of the expression $\sum_{j \in J_G} z_j(x)^2$ and its gradient can be done exactly by very fast procedures, Eqs. (36) and (37).

By using the above results, it is possible to construct a procedure, the Accelerated Hyperbolic Smoothing Clustering Algorithm, which has conceptual properties that offer the ideal conditions for a faster computational performance for solving the clustering problem given by formulation (16).

A fundamental question is the proper choice of the boundary parameter δ. Moreover, there are two main options for updating the boundary parameter δ, inside the internal minimization procedure or after it. For simplicity sake, the AHSC-L2 algorithm, the hyperbolic smoothing approach connected with the partition scheme presented below, adopts the second option, which offers a better computational performance, in spite of an eventual violation of the $\delta \geq \Delta x$ condition, which is naturally corrected in the next partition update.

The simplified AHSC-L2 algorithm.

```
Initialization Step:
    Choose initial start point: $x^0$;
    Choose smoothing parameter values: $\gamma^1, \tau^1, \varepsilon^1$;
    Choose reduction factors: $0 < \rho_1 < 1, 0 < \rho_2 < 1, 0 < \rho_3 < 1$;
    Specify the boundary band width: $\delta^1$;
    Let $k = 1$.
Main Step:    Repeat until an arbitrary stopping rule is attained
    For determining the $Z_\delta(\bar{x})$ and $G_\delta(\bar{x})$ partitions, given by (22)
    and (23), use $\bar{x} = x^{k-1}$ and $\delta = \delta^k$.
    Calculate the centers $v_i$, $i=1,\dots,q$ of gravitational regions by
    using (31).
    Solve problem (17) starting at the initial point $x^{k-1}$ and let
    $x^k$ be the solution obtained:
        For solving Eqs. (19), associated to the first part given by (19), take the smoothing parameters: $\gamma=\gamma^k$, $\tau=\tau^k$ and $\varepsilon=\varepsilon^k$;
        For solving the second part, given by (36), use the above calculated centers of the observations.
    Updating procedure:
        Let $\gamma^{k+1}=\rho_1\gamma^k$, $\tau^{k+1}=\rho_2\tau^k$, $\varepsilon^{k+1}=\rho_3\varepsilon^k$
        Redefine the boundary value: $\delta^{k+1}$
        Let $k$ :=$k$ + 1.
```

The algorithm above does not take into consideration the occurrence of empty gravitational regions. This possibility can be overcome by simply splitting the clusters with greater inertia.

The efficiency of the AHSC-L2 (HSC Method Connected with the Boundary and Gravitational Regions Partition Scheme) depends naturally on parameter δ, since it defines the partition of the set of observations. A choice of a large value will imply a decrease in the number of gravitational observation points and, therefore, the computational advantages given by formulation (36) will be reduced. In an extreme situation, the set $J_G(\overline{x})$ (25) can become empty, which implies a return to formulation (12) with the hard task of determining zeros of m equations (13). Otherwise, a choice of a small value for it will imply an inadequate specification of the set $G_\delta(\overline{x})$ and frequent violation of the basic condition $\Delta x<\delta$, for the validity of Proposition 1. In an extreme situation, the set $J_B(\overline{x})$ (24) can become empty, which implies the complete isolation of each one of q gravitational regions.

As a general strategy, within the first iterations, larger δ values are used, because of more expressive centroid displacements. The δ values would be dynamically updated considering the sizes of these displacements. In any case, the inexpensive partition procedure is always performed before the hard optimization one. So, it is possible to identify a priori any deficiency in the resulting partition and correct it, by increasing or decreasing δ, according to the diagnosed case.

5. Computational results

The computational results presented below were obtained from a particular implementation of the AHSC-L2 algorithm. The numerical experiments have been carried out on a PC Intel Celeron with 2.7 GHz CPU and 512 MB RAM. The programs are coded with Compac Visual FORTRAN, Version 6.1. The unconstrained minimization tasks were carried out by means of a Quasi-Newton algorithm employing the BFGS updating formula from the Harwell Library, obtained in the site: (http://www.cse.scitech.ac.uk/nag/hsl/).

In order to exhibit the distinct performance of the AHSC-L2 algorithm, the first three tables show the computational results obtained by solving the 13 largest problems of the symmetric TSP collection, Reinelt [9] (http://www.iwr.uni-heidelberg.de/groups/comopt/software). This collection constitutes the orthodox benchmark for the traveling salesman problem, where each city has two components. The last two tables show the results obtained by solving the page-blocks data set, an instance with 5473 observations with 10 components, from the UCI [10] repository of machine learning (http://www.ics.uci.edu./mlearn/MLRepository.html).

The AHSC-L2 is a general framework that bears a broad numbers of implementations. In the initialization steps the following choices were made for the reduction factors: $\rho_1=\frac{1}{4}$, $\rho_2=\frac{1}{4}$ and $\rho_3=\frac{1}{4}$. The specification of initial smoothing and perturbation parameters was automatically tuned to the problem data. So, the initial max function smoothing parameter (7) was specified by $\tau^1=\sigma/10$ where σ^2 is the variance of set of observation points: $S=\{s_1,\ldots,s_m\}$. The initial perturbation parameter (6) was specified by $\varepsilon^1=4\tau^1$ and the Euclidian distance smoothing parameter by $\gamma^1=\tau^1/100$.

The boundary width parameter at the beginning of each iteration k was specified by using the average distance between all pairs of centroids

$$\delta^k=\alpha\frac{\sum_{i=1}^{q-1}\sum_{p=i+1}^{q}\|x_i^{k-1}-x_p^{k-1}\|_2}{(n-1)n/2}, \tag{38}$$

where α is a constant value, $0\le\alpha\le 1$. For all instances, this parameter was fixed at $\alpha=0.05$ in all iterations, without an updating procedure. In a complementary analysis, the influence of the boundary width is illustrated by comparing the results for the page-blocks instance for values $\alpha=0.05$, 0.1, 0.2 and 0.5.

The adopted stopping criterion was the execution of the main step for the ASHC-L2 algorithm a fixed number of six iterations. In this way, the final values of the τ, ε, and γ parameters were reduced to 1/1024 of the initial values. The adopted stopping criteria for the unconstrained minimization procedure was fixed in all iterations, supplying precise solutions with 11 significant digits. This simple choices show well adequate for producing robust solutions for a broad class of clustering test problems.

Table 1 presents the results for the TSPLIB-3038 data set with 3038 observations. It exhibits the results produced by the AHSC-L2 algorithm and, for comparison, those of three algorithms presented by Bagirov [11] that used a similar computer: PC Pentium-4 with CPU 2.4 GHz and RAM 512 MB. The first two columns show the number of clusters (q), and the best known value for the global optimum (f_{opt}) taken from Bagirov [11]. The next columns show the error (E) for the best solution produced and the mean CPU time (T) given in seconds associated to four algorithms: multi-start k-means (MS k-means), global k-means (GKM), modified global k-means (MGKM) and the proposed AHSM-L2. The errors are calculated in the following

Table 1
Results for the TSPLIB-3038 Instance.

q	f_{opt}	MS k-means		GKM		MGKM		AHSC-L2	
		E	T	E	T	E	T	E	T
2	0.31688E10	0.00	12.97	0.00	1.38	0.00	0.86	0.05	0.07
10	0.56025E09	0.00	11.52	2.78	8.41	0.58	3.30	0.01	0.28
20	0.26681E09	0.42	14.53	2.00	16.63	0.48	5.77	0.05	0.59
30	0.17557E09	1.16	19.09	1.45	25.00	0.67	8.25	0.31	0.86
40	0.12548E09	2.24	22.28	1.35	33.23	1.35	10.70	−**0.11**	1.09
50	0.98400E08	2.60	23.55	1.19	41.52	1.41	13.23	0.44	1.36
60	0.82006E08	5.56	27.64	0.00	49.75	0.98	15.75	−**0.80**	1.91
80	0.61217E08	4.84	30.02	0.00	66.42	0.63	20.94	−**0.73**	6.72
100	0.48912E08	5.99	33.59	0.59	83.16	0.00	26.11	−**0.60**	9.79

way: $E = 100\ (f_{Best} - f_{opt})\ /\ f_{opt}$, where f_{Best} represents the value of best solution obtained.

The multi-start k-means algorithm is the traditional k-means algorithm with multiple initial starting points. In this experiment, to find q clusters, 100 times q starting points were randomly chosen in the MS k-means algorithm. The global k-means algorithm, introduced by Likas et al. [12], is a significant improvement of the k-means algorithm. The MGKS is an improved version of the Likas algorithm proposed by Bagirov [11]. The AHSC-L2 solutions were produced by using randomly 10 starting points in all cases, except $q = 40$ and 50, where 20 and 40 starting points were taken, respectively.

It is possible to observe in each row of Table 1 that the best solution produced by the new AHSC-L2 algorithm becomes significantly smaller than that by MS k-means when the number of clusters q increases. In fact, this algorithm does not perform well for big instances, despite being one of the most used algorithms. Wu et al. [13] present the top 10 data mining algorithms identified by the IEEE International Conference on Data Mining in December 2006. The k-means is in the second place in this list. The comparison between AHSC-L2 with GKM and MGKM solutions demonstrates similar superiority of the proposed algorithm. In the same way, the comparison of the time columns shows a consistent speed advantage of the proposed algorithm over the older ones. For all cases $q \geq 20$, AHSC-L2 obtains the best solutions between the four considered algorithms.

On the other hand, the best solution produced by the AHSC-L2 algorithm is very close to the putative global minimum presented by Bagirov [11], the best known solution of the TSPLIB-3038 instance up to that time. Moreover, in this experiment, by using a relatively small number of initial starting points, four new putative global minimum results have been established in comparison with Bagirov values, for $q = 40$, 60, 80 and 100.

Table 2 presents the results for the Pla85900 data set. It is an instance with 85 900 observations originated from a VLSI application that arose at Bell Laboratories in the late 1980s. Ten different randomly chosen starting points were used. The first column presents the specified number of clusters (q). The second column presents the best objective function value ($f_{Calculated}$) produced by the HSC and AHSC-L2 algorithms, with both alternatives obtaining the same results, within a five decimal digit precision criterion. The next three columns present data associated with the original HSC algorithm: the number of occurrences of the best solution (*Occ.*), the average error of the 10 solutions (E_{Mean}) in relation to the best solution obtained and CPU mean time given in seconds (T_{Mean}). The next three columns present the same items, produced by the newly proposed AHSC-L2 algorithm. The last column exhibits the speed-up factors provided by the new algorithm.

The results presented in Table 2 show a coherent performance of both algorithms. We could not find any recorded solution for this instance. Indeed, the clustering literature seldom considers instances with such number of observations. The high number of occurrences of the best solution (*Occ.*) and the low values presented in columns (E_{Mean}) show a consistent performance for both algorithms. Now, taking into account the robustness criteria, the comparison between the number of occurrences shows a slightly better performance for the original HSC algorithm. It is still more accurate, although it handles a harder task.

The most important analysis associated with Table 2 is the comparison between the mean CPU times, given by the last column speed-up factor. It shows clearly the performance gain of the new proposed AHSC-L2 algorithm, resulting from the very fast procedures associated with Eqs. (36) and (37). For the Pla85900 instance, on cases from $q = 2$ to 10, the observed speed-up factors present an increasing behavior, varying from 6.3 to 37.4. The same comparison performed between the CPU times of the new AHSC-L2 for the TSPLIB-3038 instance, the last column of Table 1, and the corresponding value observed for the HSC algorithm presented in Xavier [7], shows even more expressive factors, more than two decimal orders of magnitude, reaching an extraordinary value of 549 at the case $q = 60$.

Table 3 presents the computational results produced by the AHSC-L2 algorithm for the largest instances of the Symmetric Traveling Salesman Problem (TSP) collection: FL3795, FNL4461, RL5915, RL5934, Pla7397, RL11849, USA13509, BRD14051, D15112, BRD18512 and Pla33810. The numerical complement of the name of each instance corresponds to its number of observations. For each instance, two cases are presented: $q = 5$ and 10. Ten different randomly chosen starting points were used. For each case, the table presents: the best objective function value produced by the AHSC-L2 algorithm ($f_{AHSC\text{-}L2_{Best}}$), the average error of the 10 solutions in relation to the best solution obtained (E_{Mean}) and CPU mean time given in seconds (T_{Mean}).

It was impossible to perform any comparison, given the lack of recorded solutions for these large instances. Indeed, the clustering literature seldom considers instances with such number for observations. We should remark that the low values presented in columns (E_{Mean}) show the consistent performance of the proposed algorithm.

Table 4 presents the results for the Page-blocks data set. This instance has 5437 observations with 10 components and is the biggest problem presented by Bagirov [11]. Ten different randomly chosen starting points were used. The first column presents the specified number of clusters (q). The second column presents the putative global optimum (f_{opt}) taken from Bagirov [11]. The next two columns show a comparison between the error of the best solution (E) in relation to the putative global solution and the mean CPU time (T) associated with modified global

Table 2
Results for the Pla85900 instance.

q	$f_{Calculated}$	Algorithm HSC			Algorithm AHSC-L2			Speed up
		Occ.	E_{Mean}	T_{Mean}	Occ.	E_{Mean}	T_{Mean}	
2	0.37491E16	4	0.86	23.07	5	0.58	3.65	6.3
3	0.22806E16	10	0.00	47.41	7	0.04	4.92	9.6
4	0.15931E16	10	0.00	76.34	10	0.00	5.76	13.3
5	0.13397E16	1	0.80	124.32	1	1.35	7.78	16.0
6	0.11366E16	8	0.12	173.44	2	1.25	7.87	22.0
7	0.97110E15	4	0.42	254.37	1	0.87	9.33	27.3
8	0.83774E15	8	0.55	353.61	4	0.37	12.96	27.3
9	0.74660E15	3	0.68	438.71	1	0.25	13.00	33.8
10	0.68294E15	4	0.29	551.98	3	0.46	14.75	37.4

Table 3
Results for larger instances of the TSPLIB collection.

Instance	$q=5$			$q=10$		
	$f_{AHSC\text{-}L2_{Best}}$	E_{Mean}	T_{Mean}	$f_{AHSC\text{-}L2_{Best}}$	E_{Mean}	T_{Mean}
FL3795	0.368283E09	6.18	0.18	0.106394E09	2.30	0.26
FNL4461	0.181667E10	0.43	0.31	0.853304E09	0.36	0.52
RL5915	0.379585E11	1.01	0.45	0.187794E11	0.41	0.74
RL5934	0.393650E11	1.69	0.39	0.191761E11	2.35	0.76
Pla7397	0.506247E14	1.94	0.34	0.243486E14	2.10	0.80
RL11849	0.809552E11	1.11	0.83	0.369192E11	0.53	1.55
USA13509	0.329511E14	0.01	1.01	0.149816E14	1.39	1.69
BRD14051	0.122288E11	1.20	0.82	0.593928E10	1.17	2.00
D15112	0.132707E12	0.00	0.88	0.644901E11	0.71	2.27
BRD18512	0.233416E11	1.30	1.25	0.105912E11	1.05	2.24
Pla33810	0.335680E15	0.22	3.54	0.164824E15	0.68	5.14

Table 4
Results for the page-blocks instance.

q	f_{opt}	MGKM		$f_{AHSC\text{-}L2_{Best}}$	E_{Best}	E_{Mean}	T_{Mean}
		E	T				
2	0.57937E11	0.00	6.92	0.579368E11	**0.00**	0.21	0.12
10	0.45662E10	0.00	34.09	0.453301E10	−**0.72**	2.95	0.56
20	0.17139E10	0.19	62.09	0.169102E10	−**0.01**	3.30	1.86
30	0.94106E09	0.00	89.42	0.952516E09	1.21	3.64	4.23
40	0.62570E09	0.00	118.55	0.614456E09	−**1.79**	2.27	7.74
50	0.42937E09	0.00	149.77	0.423180E09	−**1.44**	3.38	14.31
60	0.31185E09	0.33	184.06	0.311755E09	−**0.03**	2.02	27.78
80	0.20576E09	0.00	258.69	0.203424E09	−**1.13**	1.07	50.56
100	0.14545E09	0.10	346.94	0.147175E09	1.18	1.30	90.31

Table 5
Influence of the band zone width.

q	$\alpha=0.05$		$\alpha=0.1$		$\alpha=0.2$		$\alpha=0.5$	
	E_{Best}	T_{Mean}	E_{Best}	T_{Mean}	E_{Best}	T_{Mean}	E_{Best}	T_{Mean}
2	**0.00**	0.12	**0.00**	0.13	**0.00**	0.14	**0.00**	0.22
10	−**0.72**	0.56	−**0.72**	0.75	−**0.72**	0.97	−**0.72**	3.47
20	−**0.01**	1.86	−**2.30**	2.21	−**2.54**	3.58	−**2.42**	12.61
30	1.21	4.23	1.23	5.31	0.23	8.49	0.14	26.64
40	−**1.79**	7.74	−**1.80**	10.22	−**1.79**	14.45	−**1.97**	50.63
50	−**1.44**	14.31	−**1.37**	20.07	−**1.93**	27.30	−**1.63**	87.99
60	−**0.03**	27.78	−**0.03**	33.01	−**0.14**	45.53	−**0.18**	125.95
80	−**1.13**	50.56	−**1.20**	64.33	−**1.20**	86.86	−**1.21**	276.20
100	1.18	90.31	1.70	105.39	1.47	129.76	3.36	490.46

k-means (MGKM), which presented the best performance among the three algorithms used by Bagirov [11]. The next four columns show the data associated to the ASHC-L2 algorithm: the best obtained solution ($f_{AHSC\text{-}L2_{Best}}$), the error in relation to the putative global solution (E_{Best}), the average error of the 10 solutions (E_{Mean}) in relation to the best one and the mean CPU time (T_{Mean}).

In this experiment, the AHSC-L2 algorithm, using only 10 initial starting points, obtained six new putative global solutions, for cases $q=10$, 20, 40, 50, 60 and 80, as shown in column (E_{Best}). For case $q=2$, it tied with the existing putative solution. Only for cases $q=30$ and 100, it obtained worse solutions. However, by using 50 initial starting points for case $q=30$, the AHSC-L2 algorithm obtained three different solutions that improved upon the old putative global minimum. The best one was $f_{AHSC\text{-}L2_{Best}}=0.930723$, with relative error $(E_{Best}) = -\mathbf{1.10}$. The low values presented in the column E_{Mean} show once again the consistent performance of the proposed algorithm. Column T_{Mean} presents smaller CPU times than the values associated to MGKM algorithm, column E. For case $q=10$, the AHSC-L2 is 60 times faster.

Table 5 presents the influence of the boundary width (38) on the performance of the ASHC-L2 algorithm, according to the robustness and efficiency criteria. The first column presents the specified number of clusters (q). The next eight columns give the error of the best solution obtained in relation to the putative minimum global solution (E_{Best}) and CPU mean time given in seconds (T_{Mean}) for four specifications of the width: $\alpha=0.05$, 0.1, 0.2 and 0.5. A fixed set of 10 different randomly chosen starting points were used for the four width specifications.

In this experiment, by increasing the boundary width, the AHSC-L2 algorithm obtained, in general, smaller best solutions, as shown by columns E_{Best}. The few exceptions are explained by the stochastic characteristic of the sequence of points generated by the algorithm. On the other hand, the columns T_{Mean} show a systematic increase of CPU times. Both behaviors are coherent since the model becomes more accurate and more complex when α increases. For the page-blocks and remaining instances, the range $\alpha = 0.05 - 0.2$ seems adequate, reconciling the two effects that are present: accuracy and velocity.

6. Conclusions

In this paper, a new method for the solution of the minimum sum-of-squares clustering problem is proposed. It is a natural development that improves the global performance of the original HSC method presented by Xavier [7]. The robustness of the performance of the AHSC-L2 algorithm can be attributed to the complete differentiability of the approach. The high speed of the AHSC-L2 algorithm can be attributed to the partition of the set of observations into two nonoverlapping parts. This approach offers a drastic simplification of computational tasks.

The computational experiments presented in this paper were obtained by using a particular and simple set of criteria for all specifications. The AHSC-L2 algorithm is a general framework that can support different implementations.

There are many potential improvements that can be made by considering better strategies in the specification of smoothing and stopping criteria parameters and in the selection of the starting point. For the latter, it is possible to adopt the incremental approach successfully used by Likas [12] and Bagirov [11] for generating a good initial point. This initialization step can also be implemented by using a kd-tree structure as presented by Pelleg and Moore [14] and Redmond and Henegan [15].

The most relevant computational task associated with the AHSC-L2 algorithm remains the determination of the zeros of Eqs. (19), for each observation in the boundary region, with the purpose of calculating the first part of the objective function (18). However, since these calculations are completely independent, they can be easily parallelized.

It must be observed that the AHSC-L2 algorithm, as presented here, is firmly linked to the MSSC problem formulation. Thus, each different problem formulation requires a specific methodology to be developed, in order to apply the partition into boundary and gravitational regions.

Finally, it must be remembered that the MSSC problem is a global optimization problem with several local minima, so that both algorithms can only produce local minima. The obtained computational results exhibit a deep local minima property, which is well suited to the requirements of practical applications.

Acknowledgments

The authors would like to thank Claudio Joaquim Martagão Gesteira and Geraldo Veiga of the Federal University of Rio de Janeiro for the helpful review of the work and constructive comments. The authors also must thank one anonymous referee for comments that improved the quality of the paper.

References

[1] M.R. Anderberg, Cluster Analysis for Applications, Academic Press Inc. New York, 1973.

[2] J.A. Hartingan, Clustering Algorithms, John Wiley and Sons, Inc., New York, NY, 1975.

[3] H. Späth, Cluster Analysis Algorithms for Data Reduction and Classification, Ellis Horwood, Upper Saddle River, NJ, 1980.

[4] R.C. Dubes, A.K. Jain, Cluster techniques: the user's dilemma, Pattern Recognition 8 (1976) 247–260.

[5] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall Inc., Upper Saddle River, NJ, 1988.

[6] P. Hansen, B. Jaumard, Cluster analysis and mathematical programming, Math. Programming 79 (1997) 191–215.

[7] A.E. Xavier, The hyperbolic smoothing clustering method, Pattern Recognition 43 (2010) 731–737.

[8] A.E. Xavier, Penalização Hiperbólica: Um Novo Método para Resolução de Problemas de Otimização, M.Sc. Thesis, COPPE, UFRJ, Rio de Janeiro, 1982.

[9] G. Reinelt, TSP-LIB—A traveling salesman library, ORSA J. Comput. (1991) 376–384.

[10] UCI, UCI repository of machine learning databases, 2010 ⟨http://www.ics.uci.edu./mlearn/MLRepository.html⟩.

[11] A.M. Bagirov, Modified global k-means algorithm for minimum sum-of-squares clustering problems, Pattern Recognition 41 (10) (2008) 3192–3199.

[12] A. Likas, M. Vlassis, J. Verbeek, The global k-means clustering algorithm, Pattern Recognition 36 (2003) 451–461.

[13] X. Wu, et al., Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (2008) 1–37.

[14] D. Pelleg, A. Moore, Accelerating exact k-means algorithms with geometric reasoning, in: KDD '99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, 1999, pp. 277–281.

[15] S.J. Redmond, C. Heneghan, A method for initialising the k-means clustering algorithm using kd-trees, Pattern Recognition Lett. 28 (2007) 965–973.

Adilson Elias Xavier is a Professor of the Federal University of Rio de Janeiro (UFRJ), whose main interests rely on Mathematical Programming, particularly Nonlinear Programming and Augmented Lagrangian Methods. He earned his D.Sc. on Systems Engineering and Computing at UFRJ on 1992. He is the author of the Hyperbolic Penalty method for Nonlinear Programming and the Hyperbolic Smoothing modeling technique. He has been working as consultant in many project with some of the most important Brazilian companies, such as Petrobras, CEPEL, Eletrobras, ONS, Furnas and Embratel. He earned prizes from SOBRAPO (Brazilian Operations Research Society) and IFORS (International Federation of Operational Research Societies).

Vinicius Layter Xavier is graduated on statistics and now he is a Master Degree Student in the Systems Engineering and Computer Science Department of the Federal University of Rio de Janeiro (UFRJ).