Probability in the Engineering and Informational Sciences, **22**, 2008, 603–621. Printed in the U.S.A.
doi:10.1017/S0269964808000351

PROBABILISTIC DISTANCE CLUSTERING ADJUSTED FOR CLUSTER SIZE

CEM IYIGUN AND ADI BEN-ISRAEL
Rutgers Center for Operations Research, and
Department of Management Science and Information Systems
School of Business
Rutgers University
E-mail: iyigun@business.rutgers.edu; adi.benisrael@gmail.com

The probabilistic distance clustering method of [1] works well if the cluster sizes are approximately equal. We modify that method to deal with clusters of arbitrary size and for problems where the cluster sizes are themselves unknowns that need to be estimated. In the latter case, our method is a viable alternative to the expectation-maximization (EM) method.

1. INTRODUCTION

A method for probabilistic clustering of data, proposed by the authors [1], is based on the assumption that the probability of a point belongimg to a cluster is inversely proportional to its distance from the cluster center. The resulting clustering algorithm is fast and efficient and works best if the cluster sizes are about equal.

In cases in which the cluster sizes differ greatly or the cluster sizes themselves are unknowns that need to be estimated (as in demixing problems), the above assumption can be modified to take into account the cluster sizes. This modification is the objective of this article.

We take data points to be vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and consider a *dataset* $\mathcal{D}$ consisting of N data points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. A *cluster* is a set of data points that are similar, in some sense, and *clustering* is a process of partitioning a dataset into disjoint clusters.

 0269-9648/08 $25.00

In *distance clustering* (or d-clustering), "similarity" is interpreted in terms of a *distance* function $d(\mathbf{x}, \mathbf{y})$ in $\mathbb{R}^n$; for example,

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|, \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \,,$$

where $\|\cdot\|$ is a norm. A common choice is the *Mahalanobis distance* with the norm

$$\|\mathbf{u}\| = \langle \mathbf{u}, \Sigma^{-1}\mathbf{u} \rangle^{1/2} \,,$$

where Σ is the covariance matrix of the data in question.

Example 1: A dataset in $\mathbb{R}^2$ with $N = 1100$ data points is shown in Figure 1. The data on the left was simulated from a Normal distribution $N(\boldsymbol{\mu}, \Sigma)$, with

$$\boldsymbol{\mu}_1 = (2, 0), \qquad \Sigma_1 = \begin{pmatrix} 0.0005 & 0 \\ 0 & 0.05 \end{pmatrix}, \qquad (100 \text{ points}),$$

and the data on the right consist of 1000 points simulated in a circle of diameter 1 centered at $\boldsymbol{\mu}_2 = (3, 0)$, from a radially symmetric distribution with Prob $\{\|\mathbf{x} - \boldsymbol{\mu}_2\| \le r\} = 2\,r$. These data will serve as an illustration in Examples 2 and 3.

Points are assigned to clusters using a *clustering criterion*. In d-clustering, each point is assigned to the cluster with the nearest center. After each assignment, the cluster centers may change, resulting in further reclassifications. A d-clustering algorithm

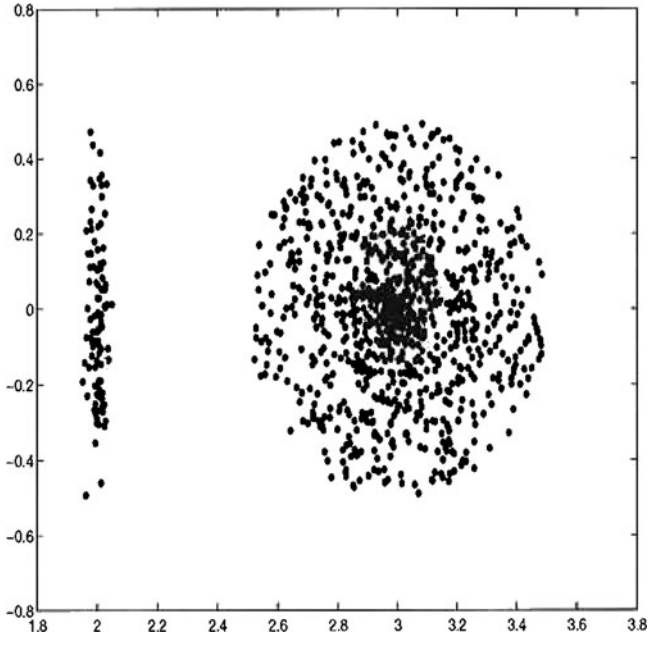


FIGURE 1. A dataset in $\mathbb{R}^2$.

will therefore iterate between centers and reassignments. The best known such method is the *k-means clustering algorithm* (see Hartigan [2]).

In *probabilistic clustering*, the assignment of points to clusters is "soft," and cluster membership is replaced by probabilities $p_k(\mathbf{x}) = \text{Prob}\{\mathbf{x} \in \mathcal{C}_k\}$ that a data point $\mathbf{x}$ belongs to the cluster $\mathcal{C}_k$. Probabilistic d-clustering is when the probabilities depend on the relevant distances.

Probabilistic d-clustering adjusted for the cluster size is called *probabilistic dq-clustering*, or *PDQ clustering* for short.

An algorithm for probabilistic dq-clustering is presented in Section 3. The centers are updated as optimal solutions of the extremal problem in Section 2.3. These centers are also stationary points of the *joint distance function*, a function that approximates the data in its lowest-level sets; see Section 2.2. The cluster sizes (if not given) are updated using the extremal problem of Section 2.4

In Section 4 we apply the algorithm to the estimation of the parameters of Gaussian mixtures and compare it to the EM method. Some numerical results are given in Section 5.

For other approaches to probabilistic clustering, see the surveys in Höppner, Klawonn, Kruse, and Runkler [4] and Tan. Steinbach, and Kumar [9].

2. PROBABILISTIC dq-CLUSTERING

Let a dataset $\mathcal{D} \subset \mathbb{R}^n$ be partitioned into K clusters $\{\mathcal{C}_k : k = 1, \ldots, K\}$,

$$\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{C}_k,$$

and let $\mathbf{c}_k$ be the *center* (in some sense) of the cluster $\mathcal{C}_k$. The *size* q_k of $\mathcal{C}_k$ is known in some applications and is an unknown to be estimated in others. Here, the cluster size, or its estimate, is assumed given wherever it appears in the right-hand side of a formula.

With each data point $\mathbf{x} \in \mathcal{D}$ and a cluster $\mathcal{C}_k$, we associate the following:

- a *distance* $d_k(\mathbf{x}, \mathbf{c}_k)$, also denoted $d_k(\mathbf{x})$
- a *probability* of membership in $\mathcal{C}_k$, denoted $p_k(\mathbf{x})$.

The distance functions $d_k(\cdot)$, associated with different clusters, are different in general. In particular, we may use a different Mahalanobis distance for each cluster:

$$d_k(\mathbf{x}) = \langle \mathbf{x} - \mathbf{c}_k, \Sigma_k^{-1}(\mathbf{x} - \mathbf{c}_k) \rangle^{1/2}, \tag{1}$$

where Σ_k is an estimate of the cluster covariance.

There are several ways to model the relationship between distances and probabilities; see [1]. The following assumption is our basic principle.

Principle 1: For each $\mathbf{x} \in \mathcal{D}$ and cluster $\mathcal{C}_k$, the probability $p_k(\mathbf{x})$ satisfies

$$\frac{p_k(\mathbf{x})\, d_k(\mathbf{x})}{q_k} = \text{constant}, \quad \text{say } D(\mathbf{x}), \text{ depending on } \mathbf{x}. \tag{2}$$

Cluster membership is thus more probable the closer the data point is to the cluster center and the larger the cluster.

2.1. Probabilities

From Principle 1 and the fact that probabilities add to one, we get the following Theorem.

THEOREM 1: *Let the cluster centers* $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ *be given, let* $\mathbf{x}$ *be a data point, and let* $\{d_k(\mathbf{x}) : k = 1, \ldots, K\}$ *be its distances from the given centers. Then the membership probabilities of* $\mathbf{x}$ *are*

$$p_k(\mathbf{x}) = \left(\prod_{j\neq k} \frac{d_j(\mathbf{x})}{q_j}\right)\left(\sum_{i=1}^{K}\prod_{j\neq i}\frac{d_j(\mathbf{x})}{q_j}\right)^{-1}, \qquad k = 1, \ldots, K. \tag{3}$$

PROOF: Using (2), we write, for i, k,

$$p_i(\mathbf{x}) = \frac{p_k(\mathbf{x}) d_k(\mathbf{x})/q_k}{d_i(\mathbf{x})/q_i}.$$

Since $\sum_{i=1}^{K} p_i(\mathbf{x}) = 1$,

$$p_k(\mathbf{x}) \sum_{i=1}^{K}\left(\frac{d_k(\mathbf{x})/q_k}{d_i(\mathbf{x})/q_i}\right) = 1.$$

$$\therefore\ p_k(\mathbf{x}) = \left[\sum_{i=1}^{K}\left(\frac{d_k(\mathbf{x})/q_k}{d_i(\mathbf{x})/q_i}\right)\right]^{-1} = \left(\prod_{j\neq k} d_j(\mathbf{x})/q_j\right)\left(\sum_{i=1}^{K}\prod_{j\neq i} d_j(\mathbf{x})/q_j\right)^{-1}.$$

■

In particular, for $K = 2$,

$$p_1(\mathbf{x}) = \frac{d_2(\mathbf{x})/q_2}{d_1(\mathbf{x})/q_1 + d_2(\mathbf{x})/q_2}, \qquad p_2(\mathbf{x}) = \frac{d_1(\mathbf{x})/q_1}{d_1(\mathbf{x})/q_1 + d_2(\mathbf{x})/q_2}, \tag{4}$$

and for $K = 3$,

$$p_1(\mathbf{x}) = \frac{d_2(\mathbf{x})d_3(\mathbf{x})/q_2q_3}{d_1(\mathbf{x})d_2(\mathbf{x})/q_1q_2 + d_1(\mathbf{x})d_3(\mathbf{x})/q_1q_3 + d_2(\mathbf{x})d_3(\mathbf{x})/q_2q_3}, \quad \text{etc.} \tag{5}$$

2.2. The Joint Distance Function

We denote the constant in (2) by $D(\mathbf{x})$, a function of $\mathbf{x}$. Since the probabilities

$$p_k(\mathbf{x}) = \frac{D(\mathbf{x})}{d_k(\mathbf{x})/q_k}, \qquad k = 1, \ldots, K,$$

add to 1, we get

$$D(\mathbf{x}) = \left(\prod_{j=1}^{K} \frac{d_j(\mathbf{x})}{q_j} \right) \left(\sum_{i=1}^{K} \prod_{j \neq i} \frac{d_j(\mathbf{x})}{q_j} \right)^{-1}. \tag{6}$$

$D(\mathbf{x})$ is called the *joint distance function* of $\mathbf{x}$ and is, up to a constant, the harmonic mean of the K weighted distances $\{d_k(\mathbf{x})/q_k\}$. $D(\mathbf{x})$ has the dimension of distance.

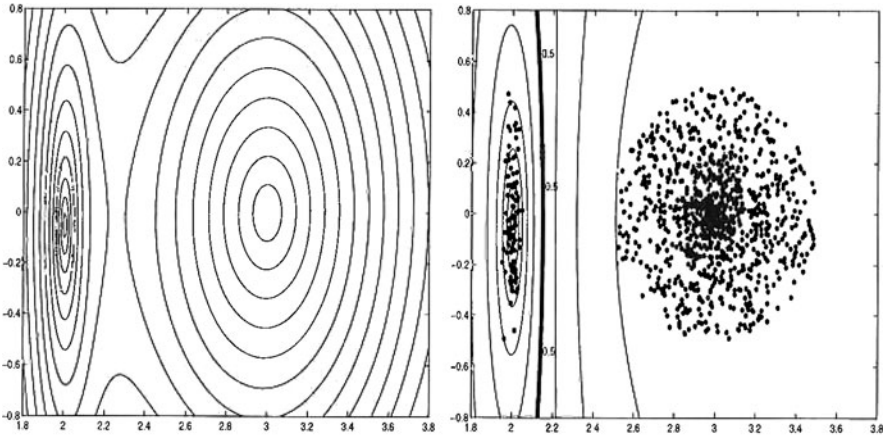
Special cases: For $K = 2$,

$$D(\mathbf{x}) = \frac{d_1(\mathbf{x})\, d_2(\mathbf{x})/q_1 q_2}{d_1(\mathbf{x})/q_1 + d_2(\mathbf{x})/q_2}, \tag{7}$$

and for $K = 3$,

$$D(\mathbf{x}) = \frac{d_1(\mathbf{x})\, d_2(\mathbf{x})\, d_3(\mathbf{x})/q_1 q_2 q_3}{d_1(\mathbf{x})\, d_2(\mathbf{x})/q_1 q_2 + d_1(\mathbf{x})\, d_3(\mathbf{x})/q_1 q_3 + d_2(\mathbf{x})\, d_3(\mathbf{x})/q_2 q_3}. \tag{8}$$

Example 2: Figure 2a shows level sets of the joint distance function (7) for the data of Example 1.



(a) Level sets of the joint distance function

(b) Probability level sets

FIGURE 2. Results for the data of Example 1.

2.3. An Extremal Principle

Equation (2) can be derived from an extremal principle. For notational simplicity, we consider the case of two clusters, with analogous results readily available for several clusters.

Let $\mathbf{x}$ be a given data point with distances $d_1(\mathbf{x})$ and $d_2(\mathbf{x})$ to the cluster centers and assume the cluster sizes q_1 and q_2 are known. Then the probabilities in (4) are the optimal solutions of the extremal problem

$$\min\left\{\frac{d_1(\mathbf{x})\,p_1^2}{q_1}+\frac{d_2(\mathbf{x})\,p_2^2}{q_2}\,:\,p_1+p_2=1,\,p_1,p_2\geq 0\right\}. \tag{9}$$

Indeed, the Lagrangian of this problem is

$$L(p_1,p_2,\lambda)=\frac{d_1(\mathbf{x})\,p_1^2}{q_1}+\frac{d_2(\mathbf{x})\,p_2^2}{q_2}-\lambda(p_1+p_2-1), \tag{10}$$

and zeroing the partials $\partial L/\partial p_i$ gives the principle (2).

Substituting the probabilities (4) in the Lagrangian (10), we get the optimal value of (9):

$$L^*(p_1(\mathbf{x}),p_2(\mathbf{x}),\lambda)=\frac{d_1(\mathbf{x})\,d_2(\mathbf{x})/q_1q_2}{d_1(\mathbf{x})/q_1+d_2(\mathbf{x})/q_2}, \tag{11}$$

which is, again, the joint distance function (7).

The corresponding extremal problem for the dataset $\mathcal{D}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N\}$ is

$$\begin{aligned}\min\quad & \sum_{i=1}^{N}\left(\frac{d_1(\mathbf{x}_i)\,p_1(\mathbf{x}_i)^2}{q_1}+\frac{d_2(\mathbf{x}_i)\,p_2(\mathbf{x}_i)^2}{q_2}\right)\\ \text{s.t.}\quad & p_1(\mathbf{x}_i)+p_2(\mathbf{x}_i)=1,\\ & p_1(\mathbf{x}_i),p_2(\mathbf{x}_i)\geq 0,\ i=1,\ldots,N,\end{aligned} \tag{12}$$

where $p_1(\mathbf{x}_i)$ and $p_2(\mathbf{x}_i)$ are the cluster probabilities at $\mathbf{x}_i$ and $d_1(\mathbf{x}_i)$ and $d_2(\mathbf{x}_i)$ are the corresponding distances. The problem separates into N problems like (9), and its optimal value is

$$\sum_{i=1}^{N}\frac{d_1(\mathbf{x}_i)\,d_2(\mathbf{x}_i)/q_1q_2}{d_1(\mathbf{x}_i)/q_1+d_2(\mathbf{x}_i)/q_2}, \tag{13}$$

the sum of the joint distance functions of all points.

Note: An explanation for the terms p_k^2 (squares of probabilities) in problem (9) is that this problem is a smoothed version of the "real" problem, $\min\{d_1,d_2\}$, which is nonsmooth; see [10] for this and other smoothing schemes.

2.4. An Extremal Principle for the Cluster Sizes

Taking the cluster sizes as variables in the extremal principle (12),

$$\min\left\{\sum_{i=1}^{N}\left(\frac{d_1(\mathbf{x}_i)\,p_1(\mathbf{x}_i)^2}{q_1}+\frac{d_2(\mathbf{x}_i)\,p_2(\mathbf{x}_i)^2}{q_2}\right):\ q_1+q_2=N, q_1,q_2\geq 0\right\}$$

with $p_1(\mathbf{x}_i)$ and $p_2(\mathbf{x}_i)$ assumed known, we have the Lagrangian

$$L(q_1,q_2,\lambda)=\sum_{i=1}^{N}\left(\frac{d_1(\mathbf{x}_i)\,p_1(\mathbf{x}_i)^2}{q_1}+\frac{d_2(\mathbf{x}_i)\,p_2(\mathbf{x}_i)^2}{q_2}\right)+\lambda(q_1+q_2-N).$$

Zeroing the partials $\partial L/\partial q_k$ gives

$$q_k^2=\frac{1}{\lambda}\left(\sum_{i=1}^{N} d_k(\mathbf{x}_i)\,p_k(\mathbf{x}_i)^2\right),\qquad k=1,2, \tag{14}$$

showing that the cluster size q_k is proportional to $\left[\sum_{i=1}^{N} d_k(\mathbf{x}_i)\,p_k(\mathbf{x}_i)^2\right]^{1/2}$. This holds for any number of clusters. In particular, for two clusters we have

$$q_1=N\left(\sum_{i=1}^{N} d_1(\mathbf{x}_i)\,p_1(\mathbf{x}_i)^2\right)^{1/2}$$

$$\times\left[\left(\sum_{i=1}^{N} d_1(\mathbf{x}_i)\,p_1(\mathbf{x}_i)^2\right)^{1/2}+\left(\sum_{i=1}^{N} d_2(\mathbf{x}_i)\,p_2(\mathbf{x}_i)^2\right)^{1/2}\right]^{-1}, \tag{15a}$$

$$q_2=N-q_1 \tag{15b}$$

since $q_1+q_2=N$.

2.5. Centers

Dealing first with the case of two clusters, we rewrite (12) as a function of the cluster centers,

$$f(\mathbf{c}_1,\mathbf{c}_2)=\sum_{i=1}^{N}\left(\frac{d_1(\mathbf{x}_i,\mathbf{c}_1)\,p_1(\mathbf{x}_i)^2}{q_1}+\frac{d_2(\mathbf{x}_i,\mathbf{c}_2)\,p_2(\mathbf{x}_i)^2}{q_2}\right) \tag{16}$$

and look for centers $\mathbf{c}_1$ and $\mathbf{c}_2$ minimizing f.

THEOREM 2: *Let the distance functions d_1 and d_2 in* (16) *be elliptic,*

$$d(\mathbf{x}, \mathbf{c}_k) = \left\langle (\mathbf{x} - \mathbf{c}_k), Q_k(\mathbf{x} - \mathbf{c}_k) \right\rangle^{1/2}, \qquad k = 1, 2, \tag{17}$$

where Q_1 and Q_2 are positive definite, so that

$$f(\mathbf{c}_1, \mathbf{c}_2) = \sum_{i=1}^{N} \left(\sqrt{\left\langle (\mathbf{x}_i - \mathbf{c}_1), Q_1(\mathbf{x}_i - \mathbf{c}_1) \right\rangle} \frac{p_1(\mathbf{x}_i)^2}{q_1} + \sqrt{\left\langle (\mathbf{x}_i - \mathbf{c}_2), Q_2(\mathbf{x}_i - \mathbf{c}_2) \right\rangle} \frac{p_2(\mathbf{x}_i)^2}{q_2} \right), \tag{18}$$

and let the probabilities $p_k(\mathbf{x}_i)$ and cluster sizes q_k be given. If the minimizers $\mathbf{c}_1$ and $\mathbf{c}_2$ of (18) *do not coincide with any of the data points $\mathbf{x}_i$, they are given by*

$$\mathbf{c}_1 = \sum_{i=1}^{N} \left(\frac{u_1(\mathbf{x}_i)}{\sum_{t=1}^{N} u_1(\mathbf{x}_t)} \right) \mathbf{x}_i, \qquad \mathbf{c}_2 = \sum_{i=1}^{N} \left(\frac{u_2(\mathbf{x}_i)}{\sum_{t=1}^{N} u_2(\mathbf{x}_t)} \right) \mathbf{x}_i, \tag{19}$$

where

$$\begin{aligned} u_1(\mathbf{x}_i) &= \left[\left(\frac{d_2(\mathbf{x}_i, \mathbf{c}_2)}{q_2} \right)^2 \frac{1}{d_1(\mathbf{x}_i, \mathbf{c}_1)} \right] \left(\frac{d_1(\mathbf{x}_i, \mathbf{c}_1)}{q_1} + \frac{d_2(\mathbf{x}_i, \mathbf{c}_2)}{q_2} \right)^{-2}, \\ u_2(\mathbf{x}_i) &= \left[\left(\frac{d_1(\mathbf{x}_i, \mathbf{c}_1)}{q_1} \right)^2 \frac{1}{d_2(\mathbf{x}_i, \mathbf{c}_2)} \right] \left(\frac{d_1(\mathbf{x}_i, \mathbf{c}_1)}{q_1} + \frac{d_2(\mathbf{x}_i, \mathbf{c}_2)}{q_2} \right)^{-2}, \end{aligned} \tag{20}$$

or, equivalently, in terms of the probabilities (4),

$$u_1(\mathbf{x}_i) = \frac{p_1(\mathbf{x}_i)^2}{d_1(\mathbf{x}_i, \mathbf{c}_1)}, \qquad u_2(\mathbf{x}_i) = \frac{p_2(\mathbf{x}_i)^2}{d_2(\mathbf{x}_i, \mathbf{c}_2)}. \tag{21}$$

PROOF: The gradient of $d(\mathbf{x}, \mathbf{c}) = \left\langle (\mathbf{x} - \mathbf{c}), Q(\mathbf{x} - \mathbf{c}) \right\rangle^{1/2}$ with respect to $\mathbf{c}$ is

$$\nabla_{\mathbf{c}} \left\langle (\mathbf{x} - \mathbf{c}), Q(\mathbf{x} - \mathbf{c}) \right\rangle^{1/2} = -\frac{Q(\mathbf{x} - \mathbf{c})}{\left\langle (\mathbf{x} - \mathbf{c}), Q(\mathbf{x} - \mathbf{c}) \right\rangle^{1/2}} = -\frac{Q(\mathbf{x} - \mathbf{c})}{d(\mathbf{x}, \mathbf{c})}, \tag{22}$$

assuming $\mathbf{x} \neq \mathbf{c}$. Therefore, if $\mathbf{c}_1$ and $\mathbf{c}_2$ do not coincide with any of the data points $\mathbf{x}_i$, we have

$$\nabla_{\mathbf{c}_k} f(\mathbf{c}_1, \mathbf{c}_2) = -Q_k \sum_{i=1}^{N} \frac{(\mathbf{x}_i - \mathbf{c}_k)}{d_k(\mathbf{x}_i, \mathbf{c}_k)} \frac{p_k(\mathbf{x}_i)^2}{q_k}. \tag{23}$$

Setting the gradient equal to zero, "canceling" the matrix Q_k and the common factor q_k, and summing like terms, we get

$$\sum_{i=1}^{N} \left(\frac{p_k(\mathbf{x}_i)^2}{d_k(\mathbf{x}_i, \mathbf{c}_k)} \right) \mathbf{x}_i = \left(\sum_{i=1}^{N} \frac{p_k(\mathbf{x}_i)^2}{d_k(\mathbf{x}_i, \mathbf{c}_k)} \right) \mathbf{c}_k,$$

proving (19) and (21). Substituting (4) in (21) then gives (20). ■

Note: The theorem holds also if a center coincides with a data point, if we interpret ∞/∞ as 1 in (19).

Theorem 2 applies, in particular, to the Mahalanobis distance (1)

$$d(\mathbf{x}, \mathbf{c}_k) = \sqrt{(\mathbf{x} - \mathbf{c}_k)^T \Sigma_k^{-1} (\mathbf{x} - \mathbf{c}_k)},$$

where Σ_k is the (given or computed) covariance matrix of the cluster $\mathcal{C}_k$.

For the general case of K clusters it is convenient to use the probabilistic form (21).

COROLLARY 1: *Consider a function of K centers*

$$f(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K) = \sum_{k=1}^{K} \sum_{i=1}^{N} \left(\frac{d_k(\mathbf{x}_i, \mathbf{c}_k)\, p_k(\mathbf{x}_i)^2}{q_k} \right), \tag{24}$$

an analog of (16). Then, under the hypotheses of Theorem 2, the minimizers of f are

$$\mathbf{c}_k = \sum_{i=1}^{N} \left(\frac{u_k(\mathbf{x}_i)}{\sum_{t=1}^{N} u_k(\mathbf{x}_t)} \right) \mathbf{x}_i, \quad \textit{with } u_k(\mathbf{x}_i) = \frac{p_k(\mathbf{x}_i)^2}{d_k(\mathbf{x}_i, \mathbf{c}_k)}, \tag{25}$$

for $k = 1, \ldots, K$.

PROOF: Same as the proof of Theorem 2. ■

Note: Formula (25) is an optimality condition for the centers $\mathbf{c}_k$, expressing them as convex combinations of the data points $\mathbf{x}_i$, with weights $u_k(\mathbf{x}_i)$ depending on the centers $\mathbf{c}_k$. It is used iteratively in Step 3 of Algorithm 1 in Section 3 to update the centers and is an extension to several facilities of the well-known Weiszfeld iteration for facility location; see [7,12]. This formula and the corresponding formulas (15) for the cluster sizes are applied in [5] for solving multifacility location problems, subject to capacity constraints.

2.6. The Centers and the Joint Distance Function

The centers obtained in Theorem 2 are stationary points for the joint distance function (13), written as a function of the cluster centers $\mathbf{c}_1$ and $\mathbf{c}_2$,

$$F(\mathbf{c}_1, \mathbf{c}_2) = \sum_{i=1}^{N} \left(\frac{d_1(\mathbf{x}_i, \mathbf{c}_1)\, d_2(\mathbf{x}_i, \mathbf{c}_2)}{q_1 q_2} \right) \left(\frac{d_1(\mathbf{x}_i, \mathbf{c}_1)}{q_1} + \frac{d_2(\mathbf{x}_i, \mathbf{c}_2)}{q_2} \right)^{-1}. \tag{26}$$

THEOREM 3: *Let the distances* $d_k(\mathbf{x}_i, \mathbf{c}_k)$ *in* (26) *be elliptic. Then the stationary points of the function F are* $\mathbf{c}_1$ *and* $\mathbf{c}_2$ *given by* (19)–(21).

PROOF: Using (22) we derive,

$$\nabla_{\mathbf{c}_1} F(\mathbf{c}_1, \mathbf{c}_2) = \frac{1}{q_1 q_2} \sum_{i=1}^{N} \left[\left(\frac{d_1(\mathbf{x}_i)}{q_1} + \frac{d_2(\mathbf{x}_i)}{q_2} \right) d_2(\mathbf{x}_i) \left(-\frac{Q_1(\mathbf{x}_i - \mathbf{c}_1)}{d_1(\mathbf{x}_i)} \right) \right.$$
$$\left. + d_1(\mathbf{x}_i)\, d_2(\mathbf{x}_i) \frac{1}{q_1} \left(\frac{Q_1(\mathbf{x}_i - \mathbf{c}_1)}{d_1(\mathbf{x}_i)} \right) \right] \left(\frac{d_1(\mathbf{x}_i)}{q_1} + \frac{d_2(\mathbf{x}_i)}{q_2} \right)^{-2}$$
$$= \sum_{i=1}^{N} \left[\frac{d_2(\mathbf{x}_i)^2}{q_2} \left(-\frac{Q_1(\mathbf{x}_i - c_1)}{d_1(\mathbf{x}_i)} \right) \right] \left(\frac{d_1(\mathbf{x}_i)}{q_1} + \frac{d_2(\mathbf{x}_i)}{q_2} \right)^{-2}. \qquad (27)$$

Setting $\nabla_{\mathbf{c}_1} F(\mathbf{c}_1, \mathbf{c}_2)$ equal zero and summing like terms, we obtain the center $\mathbf{c}_1$ as in (19)–(21). The statements about $\mathbf{c}_2$ are proved similarly. ■

3. THE PDQ ALGORITHM

The above results are used in an algorithm for unsupervised clustering of data, called the *PDQ Algorithm* (*P* for probability, *D* for distance, and *Q* for the cluster sizes). For simplicity, we describe the algorithm for the case of two clusters.

Algorithm 1. The PDQ Algorithm

Initialization:	given dataset $\mathcal{D}$ with N points, any two centers $\mathbf{c}_1, \mathbf{c}_2$, any two cluster sizes $q_1, q_2,\ q_1 + q_2 = N$, $\epsilon > 0$
Iteration:	
Step 1	**compute** distances from $\mathbf{c}_1, \mathbf{c}_2$ for all $\mathbf{x} \in \mathcal{D}$
Step 2	**update** the cluster sizes $\mathbf{q}_1^+, \mathbf{q}_2^+$ (using (15))
Step 3	**update** the centers $\mathbf{c}_1^+, \mathbf{c}_2^+$ (using (19)–(20))
Step 4	**if** $\|\mathbf{c}_1^+ - \mathbf{c}_1\| + \|\mathbf{c}_2^+ - \mathbf{c}_2\| < \epsilon$ **stop** **return** to Step 1

The algorithm iterates among the *cluster size estimates* (15), the cluster *centers* (19) expressed as minimizers of the objective function (18), and the *distances* of the data points to these centers.

Notes:

1. The distances used in Step 1 are elliptic and may be different functions depending on the cluster.

2. In particular, if the Mahalanobis distance (1)

$$d(\mathbf{x}, \mathbf{c}_k) = \sqrt{(\mathbf{x} - \mathbf{c}_k)^T \Sigma_k^{-1} (\mathbf{x} - \mathbf{c}_k)}$$

is used, the covariance matrix Σ_k of the kth cluster can be estimated at each iteration by

$$\Sigma_k = \frac{\sum_{i=1}^{N} u_k(\mathbf{x}_i)(\mathbf{x}_i - \mathbf{c}_k)(\mathbf{x}_i - \mathbf{c}_k)^T}{\sum_{i=1}^{N} u_k(\mathbf{x}_i)}, \tag{28}$$

with $u_k(\mathbf{x}_i)$ given by (20).

3. If the cluster sizes q_1 and q_2 are known, they are used as the initial estimates and are not updated thereafter, in other words, Step 2 is absent.
4. The computations stop (in Step 4) when the centers stop moving, at which point the cluster membership probabilities can be computed by (4). These probabilities are not needed by the algorithm and can be used afterward for classifying the data.
5. Having the probabilities corresponding to the final centers, rigid clusters can be determined and used to refine the estimates of the covariance matrixes.
6. Step 3 of the algorithm is a generalization of the Weiszfeld iteration [12] to several centers. As in the classical case, to establish convergence it is necessary to modify the gradient in question; if a center coincides with one of the data points, see [6,8]. However, the set of initial centers for which such a modification ever becomes necessary is denumerable, and this issue can be safely ignored in practice.

Example 3: Figure 2b shows probability level sets for the data of Example 1 as determined by (4), using the centers and covariances computed by Algorithm 1.

4. ESTIMATION OF PARAMETERS IN MIXTURES OF DISTRIBUTIONS

The PDQ Algorithm of Section 3 is an alternative to the well-known EM method for demixing distributions. Given observations from a density $\phi(\mathbf{x})$, which is itself a mixture of two densities,

$$\phi(\mathbf{x}) = \pi\, \phi_1(\mathbf{x}) + (1 - \pi)\, \phi_2(\mathbf{x}), \tag{29}$$

it is required to estimate the weight π and the relevant parameters of the distributions ϕ_1 and ϕ_2.

A common situation is when the distribution ϕ is a mixture of Normal distributions ϕ_k, each with its mean $\mathbf{c}_k$ and covariance Σ_k that need to be estimated.

For the purpose of comparison with Algorithm 1, we present the EM method for a Gaussian mixture (29) of two distributions:

$$\phi_k(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_k|}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}-\mathbf{c}_k)^T \Sigma_k^{-1}(\mathbf{x}-\mathbf{c}_k)\right\}, \qquad k = 1, 2. \tag{30}$$

For further details, see, for example, Hastie et al. [3].

Algorithm 2. The EM Method

Initialization: given dataset $\mathcal{D}$ with N points,
initial guesses for the parameters $\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \hat{\Sigma}_1, \hat{\Sigma}_2, \hat{\pi}$

Iteration:

Step 1 For all $\mathbf{x}_i \in \mathcal{D}$ **compute** the "responsibilities":

$$p_1(\mathbf{x}_i) = \frac{\hat{\pi}\phi_1(\mathbf{x}_i)}{\hat{\pi}\phi_1(\mathbf{x}_i) + (1-\hat{\pi})\phi_2(\mathbf{x}_i)},$$

$$p_2(\mathbf{x}_i) = 1 - p_1(\mathbf{x}_i)\ .$$

Step 2 **update** the centers and covariances:

$$\hat{\mathbf{c}}_k = \sum_{i=1}^{N} \left(\frac{p_k(\mathbf{x}_i)}{\sum_{j=1}^{N} p_k(\mathbf{x}_j)} \right) \mathbf{x}_i,$$

$$\hat{\Sigma}_k = \sum_{i=1}^{N} \left(\frac{p_k(\mathbf{x}_i)}{\sum_{j=1}^{N} p_k(\mathbf{x}_j)} \right) (\mathbf{x}_i - \hat{\mathbf{c}}_k)(\mathbf{x}_i - \hat{\mathbf{c}}_k)^T, \qquad k = 1, 2$$

Step 3 **update** the mixing probabilities (weights):

$$\hat{\pi} = \frac{\sum_{i=1}^{N} p_1(\mathbf{x}_i)}{N}$$

Step 4 **stop** or **return** to Step 1

Notes:

1. The "responsibilities" in Step 1 correspond to the cluster membership probabilities in Algorithm 1.
2. Step 1 requires both the Mahalanobis distance (1) and the evaluation of the density (30).
3. Step 2 is computationally similar to Step 3 of Algorithm 1.
4. The stopping rule (Step 4) is again the convergence of centers as in Algorithm 1.

TABLE 1. Comparison of Methods for the Data of Example 1

	True Parameters	The PDQ Algorithm (Algorithm 1)	The EM Method (Algorithm 2)
Centers	$\boldsymbol{\mu}_1 = (2, 0)$ $\boldsymbol{\mu}_2 = (3, 0)$	$\hat{\mathbf{c}}_1 = (2.0036, -0.0542)$ $\hat{\mathbf{c}}_2 = (2.9993, -0.0010)$	$\hat{\mathbf{c}}_1 = (2.0011, -0.0284)$ $\hat{\mathbf{c}}_2 = (3.0033, -0.0018)$
Covariance Matrices	$\Sigma_1 = \begin{pmatrix} 0.0005 & 0 \\ 0 & 0.5 \end{pmatrix}$ $\Sigma_2 = \begin{pmatrix} 0.0402 & 0.0014 \\ 0.0014 & 0.0430 \end{pmatrix}$	$\hat{\Sigma}_1 = \begin{pmatrix} 0.0004 & -0.0001 \\ -0.0001 & 0.0446 \end{pmatrix}$ $\hat{\Sigma}_2 = \begin{pmatrix} 0.0399 & -0.0020 \\ -0.0020 & 0.0432 \end{pmatrix}$	$\hat{\Sigma}_1 = \begin{pmatrix} 0.0004 & -0.0001 \\ -0.0001 & 0.0442 \end{pmatrix}$ $\hat{\Sigma}_2 = \begin{pmatrix} 0.0398 & -0.0020 \\ -0.0020 & 0.0431 \end{pmatrix}$
Weights	(0.0909, 0.9090)	(0.0932, 0.9068)	(0.0909, 0.9091)

4.1. A Comparison of the PDQ Algorithm (Algorithm 1) and the EM Method (Algorithm 2.)

1. The EM Algorithm is based on maximum likelihood and therefore depends on the density functions in the mix, requiring different computations for different densities. The PDQ Algorithm is parameter-free, making no assumptions about the densities and using the same formulas in all cases.
2. In each EM iteration, the density functions must be evaluated, requiring (in Step 1) KN function evaluations, where K is the number of densities in the mixture. In comparison, the PDQ iterations are less expensive, requiring no function evaluations.
3. Because the EM iterations are costly, it is common to use another method (e.g., the K-means method), as a preprocessor, to get closer to the centers before starting EM. The PDQ Algorithm need no preprocessing and works well from a cold start.
4. If correct assumptions are made about the mixing distributions, then the EM method has an advantage over the PDQ method, as will be illustrated in Example 6.
5. Whereas the numerical comparison of the two algorithms should best be done by others, our preliminary tests show the two algorithms to be roughly equivalent in terms of the returned results, with the PDQ Algorithm somewhat faster.

5. NUMERICAL EXAMPLES

In Examples 4–6 the PDQ and EM Algorithms were applied to the same data, in order to compare their performance. The results are reported in Tables 1–4. These examples are typical representatives of the many numerical tests we did.

Both programs used here were written in MATLAB, the EM code by Tsui [11], and the PDQ code by the first author.

The comparison is subject to the following limitations:

1. The EM program code [11] uses the K-means method (Hartigan [2]) as a preprocessor to get a good start. The number of iterations and the running time reported for this program (in Table 4) are just for the EM part, not including the preprocessing by the K-means part.

TABLE 2. Comparison of Methods for the Data of Example 5

	True Parameters	The PDQ Algorithm (Algorithm 1)	The EM Method (Algorithm 2)
Centers	$\boldsymbol{\mu}_1 = (0, 0)$ $\boldsymbol{\mu}_2 = (1, 0)$	$\hat{\mathbf{c}}_1 = (0.0023, -0.0022)$ $\hat{\mathbf{c}}_2 = (1.0080, 0.0063)$	$\hat{\mathbf{c}}_1 = (0.5429, -0.0714)$ $\hat{\mathbf{c}}_2 = (1.0603, 0.02451)$
Weights	(0.0476, 0.9524)	(0.0534, 0.9466)	(0.1851, 0.8149)

TABLE 3. Comparison of Methods for the Data of Example 6

	True Parameters	The PDQ Algorithm (Algorithm 1)	The EM Method (Algorithm 2)
Centers	$\boldsymbol{\mu}_1 = (0, 1)$ $\boldsymbol{\mu}_2 = (1, 0.7)$ $\boldsymbol{\mu}_3 = (1, 1.3)$	$\hat{\mathbf{c}}_1 = (0.0053, 1.0239)$ $\hat{\mathbf{c}}_2 = (0.9604, 0.7146)$ $\hat{\mathbf{c}}_3 = (1.0735, 1.2748)$	$\hat{\mathbf{c}}_1 = (0.0049, 0.9916)$ $\hat{\mathbf{c}}_2 = (0.9855, 0.6939)$ $\hat{\mathbf{c}}_3 = (1.0376, 1.3083)$
Covariance Matrices	$\Sigma_1 = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.1 \end{pmatrix}$	$\hat{\Sigma}_1 = \begin{pmatrix} 0.0134 & -0.0006 \\ -0.0006 & 0.1074 \end{pmatrix}$	$\hat{\Sigma}_1 = \begin{pmatrix} 0.0091 & -0.0018 \\ -0.0018 & 0.1059 \end{pmatrix}$
	$\Sigma_2 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.01 \end{pmatrix}$	$\hat{\Sigma}_2 = \begin{pmatrix} 0.0828 & 0.0023 \\ 0.0023 & 0.0117 \end{pmatrix}$	$\hat{\Sigma}_2 = \begin{pmatrix} 0.1012 & 0.0053 \\ 0.0053 & 0.0122 \end{pmatrix}$
	$\Sigma_3 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.01 \end{pmatrix}$	$\hat{\Sigma}_3 = \begin{pmatrix} 0.0907 & -0.0040 \\ -0.0040 & 0.0123 \end{pmatrix}$	$\hat{\Sigma}_3 = \begin{pmatrix} 0.0981 & -0.0005 \\ -0.0005 & 0.0090 \end{pmatrix}$
Weights	(0.333, 0.333, 0.333)	(0.3297, 0.3345, 0.3358)	(0.3318, 0.3351, 0.3331)

Table 4. Summary of Computation Results for Three Examples

		PDQ Algorithm		EM Algorithm	
Example	ϵ	Iterations	Time (s)	Iterations	Time (s)
Example 4	0.01	5	3.32	1	1.783
	0.1	2	1.42	1	1.682
Example 5	0.01	8	3.89	55	37.73
	0.1	2	1.02	9	7.28
Example 6	0.01	11	2.29	7	3.28

Note: See Section 5, item 1 for explanation of the EM running time and iterations count.

2. Our PDQ code is the first, unfinessed version, a verbatim implementation of Algorithm 1.
3. The number of iterations depends on the stopping rule. In the PDQ Algorithm, the stopping rule is Step 4 of Algorithm 1, and the number of iterations will increase the smaller ϵ is. In the EM Algorithm, the stopping rule does involve also the convergence of the likelihood function, and the effect of the tolerance ϵ is less pronounced.
4. The number of iterations depends also on the initial estimates; the better the estimates, the fewer iterations will be required. In our PDQ code, the initial solutions can be specified or are randomly chosen. The EM program gets its initial solution from its K-means preprocessor.

Example 4: Algorithms 1 and 2 were applied to the data of Example 1. Both algorithms give good estimates of the true parameters; see Table 1. The comparison of running time and iterations is inconclusive, see Table 4.

Example 5: Consider the dataset shown in Figure 3. The points of the right cluster were generated in a circle of diameter 1.5 centered at $\boldsymbol{\mu}_1 = (1, 0)$, using a radially symmetric distribution function, $\text{Prob}\{\|\mathbf{x} - \boldsymbol{\mu}_1\| \le r\} = (4/3)\, r$, and the smaller cluster on the left was similarly generated in a circle of diameter 0.1 centered at $\boldsymbol{\mu}_2 = (0, 0)$. The ratio of sizes is 1:20.

The EM method gives bad estimates of the left center and of the weights; see Table 2 and the right panel of Figure 4. The estimates provided by the PDQ Algorithm are better; see Figure 4, left panel.

The EM method also took longer; see Table 4. In repeated trials, it did not work for $\epsilon = 0.1$, and sometimes for $\epsilon = 0.01$.

Example 6: Consider the dataset shown in Figure 5 (left). It consists of three clusters of equal size, 200 points each, generated from Normal distributions $N(\boldsymbol{\mu}_i, \Sigma_i)$, with parameters $\boldsymbol{\mu}_i$ and Σ_i given in the left column of Table 3. A similar example appears as Fig. 9.6 in Tan et al. [9, p. 593]

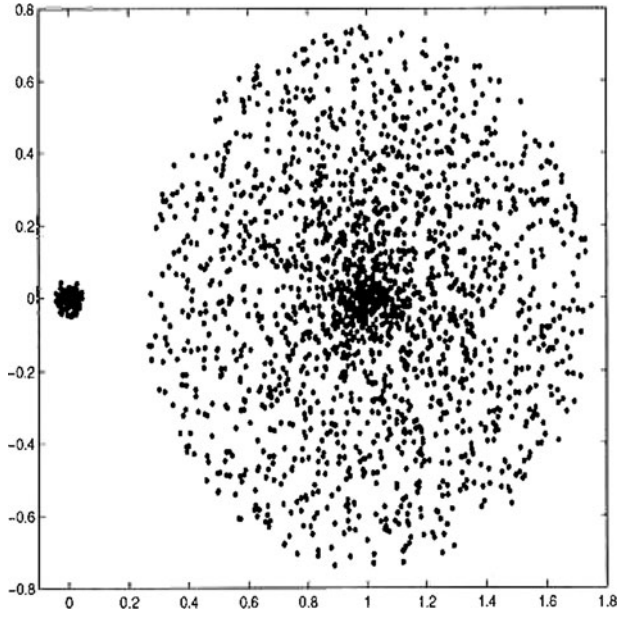


FIGURE 3. Dataset of Example 5.

As noted in Section 4.1, item 4, if the assumptions on the mixing distributions are justified, the EM method gives good estimates of the relevant parameters. The PDQ Algorithm does not require or depend on such assumptions and still gives decent estimates. This is illustrated in Table 3.

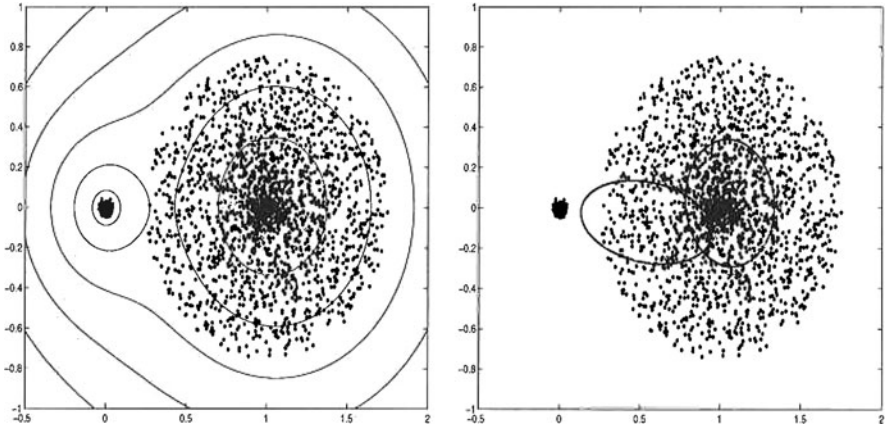


FIGURE 4. Comparison of the PDQ Algorithm (left), and the EM method (right).

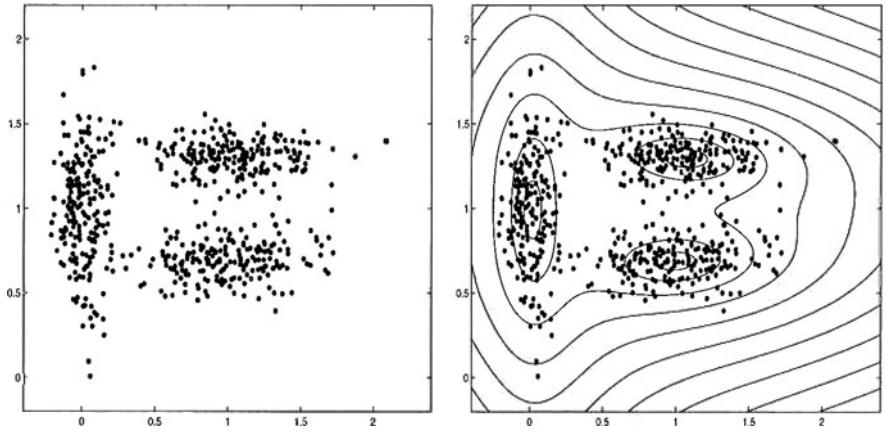


FIGURE 5. Data of Example 6 (left) and level sets of the joint distance function (right).

6. CONCLUSIONS

The PDQ Algorithm is a probabilistic clustering method based on distances (of data points from cluster centers) and on the cluster sizes. At each iteration, the method updates the cluster centers and the cluster sizes (if unknown). The method uses inexpensive iterations and converges fast.

An important application is estimating the parameters of a mixture of distributions. In this problem, the PDQ method might serve as an alternative to the EM method or as a preprocessor giving the EM method a good start. Further numerical tests, by disinterested parties, are required.

References

1. Ben-Israel, A. & Iyigun, C. (2008). Probabilistic distance clustering, *Journal of Classification* 25: 5–26.
2. Hartigan, J. (1975). *Clustering algorithms*. New York: Wiley.
3. Hastie, T., Tibshirani, R. & Friedman, J.H. (2003). *The elements of statistical learning*. New York: Springer-Verlag.
4. Höppner, F., Klawonn, F., Kruse, F. & Runkler, T. (1999). *Fuzzy cluster analysis*. New York: Wiley.
5. Iyigun, C. & Ben-Israel, A. (2008). A distance clustering method for multifacility location problems (in preparation).
6. Kuhn, H.W. (1973). A Note on Fermat's problem. *Mathematical Programming* 4: 98–107.
7. Love, R., Morris, J. & Wesolowsky, G. (1988). *Facilities location: Models and methods*. Amsterdam: North-Holland.
8. Ostresh, L.M., Jr. (1978). On the convergence of a class of iterative methods for solving the weber location problem. *Operations Research* 26: 597–609.
9. Tan, P., Steinbach, M. & Kumar, V. (2006). *Introduction to data mining*. Reading, MA: Addison Wesley.

10. Teboulle, M. (2007). A unified continuous optimization framework for center-based clustering methods. *Journal of Machine Learning* 8: 65–102.
11. Tsui, P. (2006). *EM-GM Algorithm MATLAB Code*. Waterloo, Canada: PAMI Research Group, University of Waterloo.
12. Weiszfeld, E. (1937). Sur le point par lequel la somme des distances de *n* points donnés est minimum. *Tohoku Mathematics Journal* 43: 355–386.