A Possibilistic Fuzzy c-Means Clustering Algorithm

Nikhil R. Pal, Kuhu Pal, James M. Keller, and James C. Bezdek

***Abstract*—In 1997, we proposed the fuzzy-possibilistic c-means (FPCM) model and algorithm that generated both membership and typicality values when clustering unlabeled data. FPCM constrains the typicality values so that the sum over all data points of typicalities to a cluster is one. The row sum constraint produces unrealistic typicality values for large data sets. In this paper, we propose a new model called *possibilistic-fuzzy c-means* (PFCM) model. PFCM produces memberships and possibilities simultaneously, along with the usual point prototypes or cluster centers for each cluster. PFCM is a hybridization of possibilistic c-means (PCM) and fuzzy c-means (FCM) that often avoids various problems of PCM, FCM and FPCM. PFCM solves the noise sensitivity defect of FCM, overcomes the coincident clusters problem of PCM and eliminates the row sum constraints of FPCM. We derive the first-order necessary conditions for extrema of the PFCM objective function, and use them as the basis for a standard alternating optimization approach to finding local minima of the PFCM objective functional. Several numerical examples are given that compare FCM and PCM to PFCM. Our examples show that PFCM compares favorably to both of the previous models. Since PFCM prototypes are less sensitive to outliers and can avoid coincident clusters, PFCM is a strong candidate for fuzzy rule-based system identification.**

***Index Terms*—c-means models, fuzzy clustering, hybrid clustering, possibilistic clustering.**

I. Introduction

CLUSTERING an unlabeled data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \Re^p$ is the partitioning of X into $1 < c < n$ subgroups such that each subgroup represents "natural" substructure in X. This is done by assigning *labels* to the vectors in X, and hence, to the objects generating X. A *c-partition* of X is a set of (cn) values $\{u_{ik}\}$ that can be conveniently arrayed as a $(c \times n)$ matrix $U = [u_{ik}]$. There are three sets of partition matrices

$$M_{\rm pcn} = \{U \in \Re^{\rm cn} : 0 \le u_{ik} \le 1 \; \forall \, i, k; \; \forall \, k \exists i \ni u_{ik} > 0\} \tag{1a}$$

$$M_{\rm fcn} = \left\{ U \in M_{\rm pcn} : \sum_{i=1}^{c} u_{ik} = 1 \; \forall \, k; \sum_{k=1}^{n} u_{ik} > 0 \; \forall \, i \right\} \tag{1b}$$

$$M_{\rm hcn} = \{U \in M_{\rm fcn} : u_{ik} = 0 \text{ or } 1 \; \forall \, i \text{ and } k\}. \tag{1c}$$

Equation (1) defines, respectively, the sets of possibilistic, fuzzy *or* probabilistic, and crisp *c-partitions* of X. So, there are *four* kinds of label vectors, but fuzzy and probabilistic label vectors are mathematically identical, having entries between 0 and 1 that sum to 1 over each column. The reason these matrices are called *partitions* follows from the interpretation of their entries. If U is crisp or fuzzy, u_{ik} is taken as the *membership* of $\mathbf{x}_k$ in the i-th partitioning fuzzy subset (cluster) of X. If U in $M_{\rm fcn}$ is probabilistic, u_{ik} is usually the (posterior) probability $p(i|\mathbf{x}_k)$ that, given $\mathbf{x}_k$, it came from class i. And if U in $M_{\rm pcn}$ is possibilistic, it has entries between 0 and 1 that do not necessarily sum to 1 over any column. In this last case, u_{ik} is taken as the possibility that $\mathbf{x}_k$ belongs to class i. An alternate interpretation of possibility u_{ik} is that it measures the *typicality* of $\mathbf{x}_k$ to cluster i. Observe that $M_{\rm hcn} \subset M_{\rm fcn} \subset M_{\rm pcn}$.

A clustering algorithm $\mathbf{C}$ finds a $U \in M_{\rm hcn}(M_{\rm fcn}, M_{\rm pcn})$ which (hopefully) "best" explains and represents (unknown) structure in X with respect to the model that defines $\mathbf{C}$. For U in $M_{\rm fcn}$, $c = 1$ is represented uniquely by the hard 1-partition $\mathbf{1}_n = [\underbrace{1 \;\; 1 \;\; \ldots \;\; 1}_{n \text{ times}}]$, which unequivocally assigns all n objects to a single cluster; and $c = n$ is represented uniquely by $U = I_n$, the $n \times n$ identity matrix, up to a permutation of columns. In this case, each object is in its own singleton cluster. Choosing $c = 1$ or $c = n$ rejects the hypothesis that X contains clusters.

One of the most widely used fuzzy clustering models is *fuzzy c-means* (FCM) [1]. The FCM algorithm assigns memberships to $\mathbf{x}_k$ which are inversely related to the relative distance of $\mathbf{x}_k$ to the c point prototypes $\{\mathbf{v}_i\}$ that are cluster centers in the FCM model. Suppose $c = 2$. If $\mathbf{x}_k$ is equidistant from two prototypes, the membership of $\mathbf{x}_k$ in each cluster will be the same $(= 0.5)$, regardless of the absolute value of the distance of $\mathbf{x}_k$ from the two centroids (as well as from the other points in the data). The problem this creates is that noise points, far but equidistant from the central structure of the two clusters, can nonetheless be given equal membership in both, when it seems far more natural that such points be given very low (or even no) membership in either cluster.

To overcome this problem, Krishnapuram and Keller [2] proposed a new clustering model named *possibilistic c-means* (PCM), which relaxes the column sum constraint in (1b) so that the sum of each column satisfies the looser constraint $0 < \sum_{i=1}^{c} u_{ik} \le c$. In other words, each element of the k-th column can be any number between 0 and 1, as long as at least one of them is positive. They suggested that in this case the value u_{ik} should be interpreted as the *typicality* of $\mathbf{x}_k$ relative to cluster i (rather than its membership in the cluster). They

Manuscript received January 30, 2004; revised July 8, 2004. The work of K. Pal was supported in part by the Department of Science and Technology, Government of India, under Sanction Order SR/FTP/ET-248/2001.

N. R. Pal is with the Electronics and Communications Sciences Unit, Indian Statistical Institute, Calcutta 7000108, India (e-mail: nikhil@isical.ac.in).

K. Pal is with the Institute of Engineering and Management, Salt Lake Electronics Complex, Calcutta 700091, India (e-mail: kuhu305@hotmail.com).

J. M. Keller is with the Electrical and Computer Engineering Department, University of Missouri-Columbia, Columbia, MO 65211-2060 USA (e-mail: kellerj@missouri.edu).

J. C. Bezdek is with the Computer Science Department, University of West Florida, Pensacola, FL 32514 USA (e-mail: jbezdek@uwf.edu).

interpreted each row of U as a possibility distribution over X. The PCM algorithm they suggested for optimization of the PCM objective function sometimes helps to identify outliers (noise points). However, as pointed out by Barni *et al.* [3], the price PCM pays for its freedom to ignore noise points is that PCM is very sensitive to initializations, and it sometimes generates coincident clusters. Moreover, typicalities can be very sensitive to the choice of the additional parameters needed by the PCM model.

Timm *et al.* [13]–[15] proposed two possibilistic fuzzy clustering algorithms that can avoid the coincident cluster problem of PCM. In [13] and [14], the authors modified the PCM objective function adding an inverse function of the distances between cluster centers. This extra term acts as a repulsive force and keeps the clusters separate (avoids coincident clusters). In [14] and [15], Timm *et al.* use the same concept to modify the objective function as used in Gustafson and Kessel [16] clustering algorithm. These algorithms, although use only the typicalities (possibilities), attempt to exploit the benefits of both fuzzy and possibilistic clustering.

In [12], we justified the need for *both* possibility (i.e., typicality) and membership values, and proposed a model and companion algorithm to optimize it. Our 1997 paper called this algorithm FPCM. FPCM normalizes the possibility values, so that the sum of possibilities of all data points in a cluster is 1. Although FPCM is much less prone to the problems of both FCM and PCM just described, the possibility values are very small when the size of the data set increases. In this paper we propose a new model that hybridizes FCM and PCM, enjoys the benefits of both models, and eliminates the problem of FPCM. To avoid confusion, we call this new model *possibilistic fuzzy c-means* (PFCM). The rest of the paper is organized as follows. Section II discusses FCM. Section III does the same for PCM. In Section IV we discuss the FPCM clustering model, along with the first order necessary conditions for the FPCM functional. In Section V we present the new PFCM model, Section VI includes some numerical examples that compare FCM and PCM to PFCM. Section VII has our discussion and conclusions.

II. What's Wrong With Fuzzy and Probabilistic Partitions?

The FCM model is the constrained optimization problem

$$\underbrace{\min}_{(U,\mathbf{V})}\left\{J_m(U,\mathbf{V};X)=\sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m\|\mathbf{x}_k-\mathbf{v}_i\|_A^2\right\} \quad (2)$$

where $U \in M_{\text{fcn}}$, $\mathbf{V}=(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_c)$ is a vector of (unknown) cluster centers (weights or prototypes), $\mathbf{v}_i \in \Re^p$ for $1 \le i \le c$, and $\|\mathbf{x}\|_A=\sqrt{\mathbf{x}^tA\mathbf{x}}$ is any inner product norm. Optimal partitions U^* of X are taken from pairs $(U^*,\mathbf{V}^*)$ that are local minimizers of J_m. Approximate optimization of J_m by the FCM-AO *algorithm* is based on iteration through the following necessary conditions for its local extrema.

Theorem FCM [1]: If $D_{ikA}=\|\mathbf{x}_k-\mathbf{v}_i\|_A>0$ for all i and k, $m>1$, and X contains at least c distinct points, then $(U,\mathbf{V}) \in M_{\text{fcn}} \times \Re^{cp}$ may minimize J_m only if

$$u_{ik}=\left(\sum_{j=1}^{c}\left(\frac{D_{ikA}}{D_{jkA}}\right)^{2/(m-1)}\right)^{-1}$$
$$1 \le i \le c;\ 1 \le k \le n \text{ and} \quad (3a)$$

$$\mathbf{v}_i=\frac{\sum_{k=1}^{n}u_{ik}^m\mathbf{x}_k}{\sum_{k=1}^{n}u_{ik}^m}, \qquad 1 \le i \le c. \quad (3b)$$

Singularity in FCM occurs when one or more of the distances $D_{ikA}=0$ at any iterate. In this case (rare in practice), assign 0's to each u_{ik} for which $D_{ikA}>0$, and distribute memberships arbitrarily across the $\mathbf{v}_i$'s for which $D_{ikA}=0$, subject to the constraints in (1b). The most popular algorithm for approximating solutions of (2) is Picard iteration through (3a) and (3b). This type of iteration is often called *alternating optimization* (AO) as it simply loops through one cycle of estimates for $\mathbf{V}_{t-1} \Rightarrow U_t \Rightarrow \mathbf{V}_t$ and then checks $\|\mathbf{V}_t-\mathbf{V}_{t-1}\|_{err} \le \varepsilon$. Equivalently, the entire procedure can be shifted one half cycle, so that initialization and termination is done on U, and the iterates become $U_{t-1} \Rightarrow \mathbf{V}_t \Rightarrow U_t$, with the alternate termination criterion $\|U_t-U_{t-1}\|_{err} \le \varepsilon$. The literature contains both specifications; the convergence theory is the same in either case. There are some obvious advantages to initializing and terminating on $\mathbf{V}$ in terms of convenience, speed and storage. The alternate form that initializes and terminates on U's is more stringent, since many more parameters must become close before termination is achieved. It can happen that different results ensue by using the same ε with both forms. The Appendix exhibits the form of FCM-AO used in our examples. A limit property of (3) that is important for this study is [1]:

$$\lim_{m\to 1^+}\{u_{ik}\}=\left\{\begin{array}{ll}1, & D_{ikA}<D_{jkA}\,\forall\, j\neq i\\ 0, & \text{otherwise}\end{array}\right\}$$
$$1 \le i \le c;\ 1 \le k \le n. \quad (4a)$$

Using this result, we take the same limit in (3b), obtaining

$$\underbrace{\lim}_{m\to 1^+}\left\{\left(\mathbf{v}_i=\frac{\sum_{k=1}^{n}(u_{ik})^m\mathbf{x}_k}{\sum_{k=1}^{n}(u_{ik})^m}\right)\right\}=\frac{\sum_{\mathbf{x}_k\in X_i}\mathbf{x}_k}{n_i}=\bar{\mathbf{v}}_i$$
$$1 \le i \le c \quad (4b)$$

where $X=X_1\cup\cdots X_i\cup\cdots X_c$ is the hard c-partition of X defined by the right side of (4a) with $\sum_{k=1}^{n}u_{ik}=n_i=|X_i|$, and $\bar{\mathbf{v}}_i$ is the mean vector of X_i. If we use these results in (2), we get $J_1(U,\mathbf{V};X)$ at (5), as shown at the bottom of the next page. $J_1(U,\mathbf{V};X)$ is the classical within-groups sum of squared errors objective function. Equation (5) is the optimization problem that defines the *hard c-means* (HCM) model. Moreover, the right

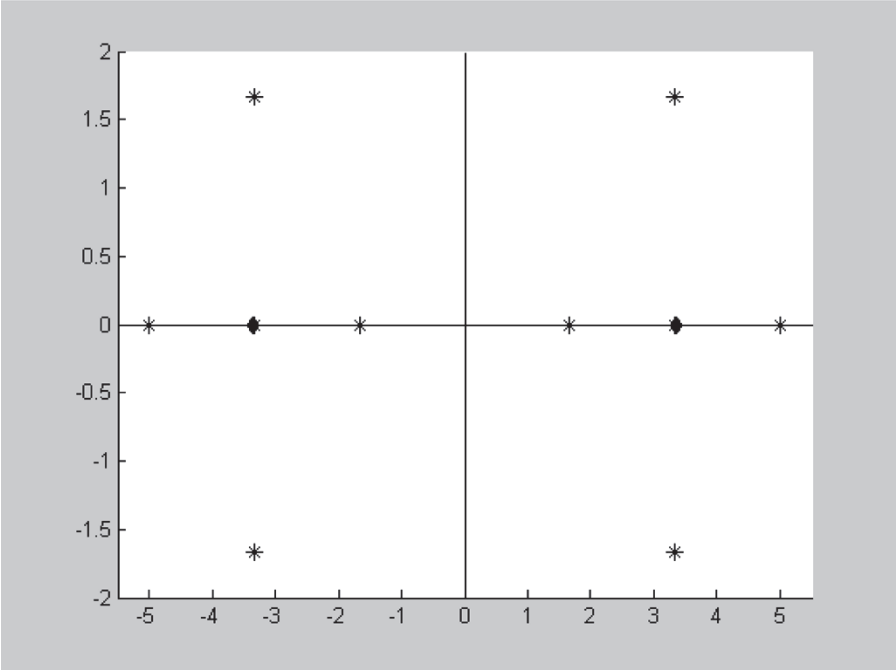

Fig. 1. Data set X_{10}, and $\mathbf{V}_{\mathrm{FCM}}^{10} = \left[\mathbf{v}_{1,\mathrm{FCM}}^{10}, \mathbf{v}_{2,\mathrm{FCM}}^{10}\right]$, shown as diamonds.

sides of (4a) and (4b) are the necessary conditions for local extrema of J_1. Taking the limit of (3) at the other extreme, we get

$$\lim_{m\to\infty}\{u_{ik}\} = \frac{1}{c} \qquad 1 \le i \le c;\ \ 1 \le k \le n \quad \text{(6a)}$$

$$\lim_{m\to\infty}\left\{\left(\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^m}\right)\right\} = \frac{\sum_{k=1}^{n}\mathbf{x}_k}{n} = \bar{\mathbf{v}} \qquad 1 \le i \le c. \quad \text{(6b)}$$

In (6b), $\bar{\mathbf{v}}$ is the grand mean of X. If $U \in M_{\mathrm{fcn}}$, the constraint $\sum_{i=1}^{c} u_{ik} = 1\ \forall\ k$ makes it difficult to interpret u_{ik} as the typicality of $\mathbf{x}_k$ to the ith cluster. To illustrate this problem of fuzzy partitions generated by FCM, Figs. 1 and 2 show two-dimensional data sets X_{10} and X_{12} with 10 and 12 points whose coordinates are given in columns 2 and 3 of Table I.

We denote $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{10}\}$ by X_{10} and $X_{10} \cup \{\mathbf{x}_{11}, \mathbf{x}_{12}\}$ as X_{12}. Some authors call $\mathbf{x}_{11}$ an "inlier" (bridge) and $\mathbf{x}_{12}$ an "outlier" (noise). X_{10} has two diamond shaped clusters with five points each on the left and right sides of the y axis. $\mathbf{x}_{11}$ and $\mathbf{x}_{12}$ are equidistant from all corresponding pairs of points in the two clusters.

Example 1:

Data set: X_{10}, X_{12}.
Algorithm: FCM.

Initialization:

$$\mathbf{V}_0 = \begin{bmatrix} 0.07 & 0.36 \\ 0.40 & 0.99 \end{bmatrix}. \quad \text{(7a)}$$

Parameters: $c = m = 2$.

To understand how inliers and outliers can affect partitions found by FCM, we applied FCM to X_{10} and X_{12} with $c = m = 2$ and other FCM protocols as listed in Section V. We initialized FCM on V using random values drawn from $[0, 1]$ as shown in (7a)- the actual values are truncated to two significant digits for display purpose. Table I shows the membership values (rows U_i of U are shown transposed as columns in the Table, rounded to two-digit accuracy) obtained for each point at termination after 12 iterations of FCM-AO. The terminal centroids $V_{\mathrm{FCM}}^{10} = \left[\mathbf{v}_{1,\mathrm{FCM}}^{10}, \mathbf{v}_{2,\mathrm{FCM}}^{10}\right]$ are shown in (7b) and also shown in Fig. 1 by the diamond symbol.

$$V_{\mathrm{FCM}}^{10} = \begin{bmatrix} -3.36 & 3.36 \\ 0.00 & 0.00 \end{bmatrix}$$
$$V_{\mathrm{FCM}}^{12} = \begin{bmatrix} -2.99 & 2.99 \\ 0.54 & 0.54 \end{bmatrix}. \quad \text{(7b)}$$

Now, suppose we add the points $\mathbf{x}_{11} = (0.0, 0.0)^T$ and $\mathbf{x}_{12} = (0.0, 10.0)^T$ to X_{10}. The point $\mathbf{x}_{12}$ is ten units directly above $\mathbf{x}_{11}$ as shown in Fig. 2. Applying FCM to X_{12} with the same parameters and initialization as before, we get the FCM partition shown in columns 6 and 7 of Table I, and the terminal cluster

$$\lim_{m\to 1^+}\left\{\underbrace{\min}_{(U,\mathbf{V})}\left\{J_m(U,\mathbf{V};X) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m \|\mathbf{x}_k - \mathbf{v}_i\|_A^2\right\}\right\} = \underbrace{\min}_{(U,\mathbf{V})}\left\{J_1(U,\mathbf{V};X) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}\|\mathbf{x}_k - \mathbf{v}_i\|_A^2\right\}. \quad \text{(5)}$$

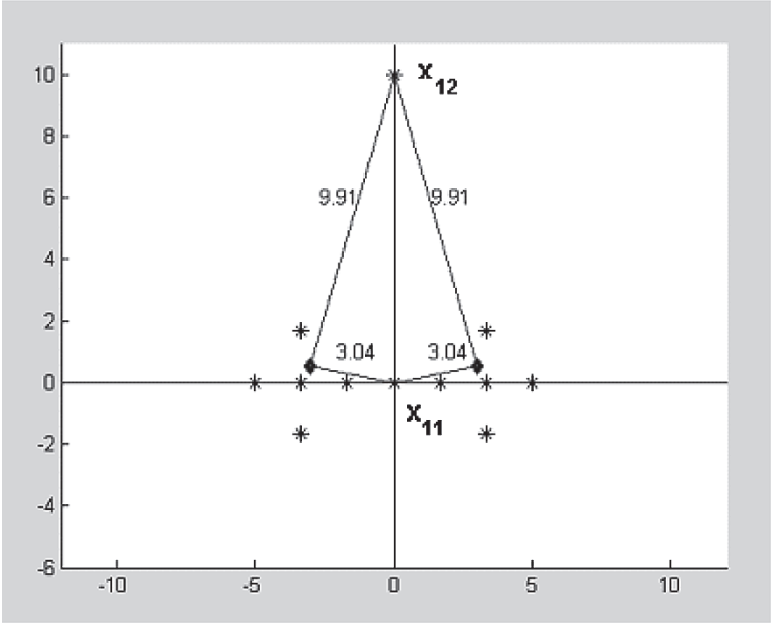

Fig. 2. Data set X_{12} (columns 1 and 2 of Table I) and $\mathbf{v}_{12}^{\mathrm{FCM}}$.

TABLE I
DATA SETS X_{10}, X_{12} AND U_{FCM} AT TERMINATION ON X_{10} AND X_{12}

	Data		FCM on X_{10}		FCM on X_{12}	
Pt.	x	y	$\mathbf{U}_1^{\mathrm{T}}$	$\mathbf{U}_2^{\mathrm{T}}$	$\mathbf{U}_1^{\mathrm{T}}$	$\mathbf{U}_2^{\mathrm{T}}$
1	-5.00	0.00	0.96	0.04	0.94	0.06
2	-3.34	1.67	0.95	0.05	0.97	0.03
3	-3.34	0.00	1.0	0.00	0.99	0.01
4	-3.34	-1.67	0.95	0.05	0.90	0.10
5	-1.67	0.00	0.90	0.10	0.92	0.08
6	1.67	0.00	0.10	0.90	0.08	0.92
7	3.34	1.67	0.06	0.94	0.03	0.97
8	3.34	0.00	0.00	1.00	0.01	0.99
9	3.34	-1.67	0.06	0.94	0.10	0.90
10	5.00	0.00	0.04	0.96	0.06	0.94
11	0.00	0.00			0.50	0.50
12	0.00	10.00			0.50	0.50

centers V_{FCM}^{12} are also shown in (7b) and in Fig. 2 by the diamond symbol. Points $\mathbf{x}_{11}$ and $\mathbf{x}_{12}$ have membership values of 0.50 in each cluster because both are equidistant from the centroids V_{FCM}^{12}, even though $\mathbf{x}_{12}$ is more than three times further away from V_{FCM}^{12} than is $\mathbf{x}_{11}$. This is illustrated in Fig. 2.

The addition of $\mathbf{x}_{11}$ and $\mathbf{x}_{12}$ to X_{10} does *not* change the terminal memberships of points $\mathbf{x}_1, \ldots, \mathbf{x}_{10}$ very much at all (the maximum change is 0.05). However, $\mathbf{x}_{11}$ seems far more typical of the overall structure of the data than $\mathbf{x}_{12}$. This illustrates how a noise point such as $\mathbf{x}_{12}$ can adversely affect data interpretation using fuzzy memberships. The problem lies with the basic notion of fuzzy (or probabilistic!) partitioning of data sets. Specifically, the summation constraint in (1b) forces this undesirable situation to occur.

III. WHAT'S WRONG WITH POSSIBILISTIC PARTITIONS?

To circumvent the counterintuitive results just displayed, Krishnapuram and Keller [2] suggest relaxing the column constraint $\sum_{i=1}^{c} u_{ik} = 1 \; \forall \; k$ for fuzzy partitions in (1b) so that u_{ik} better reflects what we expect for the typicality of $\mathbf{x}_k$ to the ith cluster. We represent typicality by t_{ik} and the typicality matrix as $T = [t_{ik}]_{c \times n}$. Krishnapuram and Keller [2] proposed the PCM model

$$\underbrace{\min}_{(T,V)} \left\{ P_m(T, \mathbf{V}; X, \gamma) = \sum_{k=1}^{n} \sum_{i=1}^{c} (t_{ik})^m ||\mathbf{x}_k - \mathbf{v}_i||_A^2 + \sum_{i=1}^{c} \gamma_i \sum_{k=1}^{n} (1 - t_{ik})^m \right\} \quad (8)$$

where $||\mathbf{x}||_A = \sqrt{\mathbf{x}^t A \mathbf{x}}$ is any inner product norm, $T \in M_{\mathrm{pcn}}$, $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c)$ is a vector of cluster centers, $\mathbf{v}_i \in \Re^p$ and $\gamma_i > 0$ is a *user-defined* constant, $1 \leq i \leq c$. Since the rows *and* columns of T are independent, minimization of $P_m(T, \mathbf{V}) = \sum_{i=1}^{c} \sum_{k=1}^{n} P_m^{ij}(T, \mathbf{V})$ can be done by minimizing the ijth term of P_m with respect to T

$$P_m^{ij}(T, \mathbf{V}) = t_{ij}^m D_{ijA}^2 + \gamma_i (1 - t_{ij})^m. \quad (9)$$

Approximate optimization of P_m by the PCM-AO *algorithm* is based on iteration through the following necessary conditions for its local extrema.

Theorem PCM [2]: If $\gamma_i > 0, 1 \le i \le c$, $m > 1$, and X contains at least c distinct points, then $(U, \mathbf{V}) \in M_{\text{pcn}} \times \Re^{cp}$ may minimize P_m only if

$$t_{ik} = \frac{1}{1 + \left(\frac{D_{ikA}^2}{\gamma_i}\right)^{1/(m-1)}}, \quad 1 \le i \le c;\ 1 \le k \le n \tag{10a}$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} t_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{n} t_{ik}^m}, \qquad 1 \le i \le c;\ 1 \le k \le n. \tag{10b}$$

Here, i) unlike its counterpart in (3a), D_{ikA} could be zero in (10a), so PCM does not suffer from the same singularity problem that FCM does, and ii) the functional form in (10b) is identical to that in (3b). The PCM-AO algorithm for solving (8) suggested by Krishnapuram and Keller simply replaced (3) with (10) in FCM-AO and added the specification of the $\{\gamma_i\}$ to the "pick list" in Appendix. PCM-AO can also be initialized with either T_0 or $\mathbf{V}_0$. Krishnapuram and Keller recommend using terminal outputs of FCM-AO as a good way to initialize PCM-AO. They also suggest choosing the $\{\gamma_i\}$ by computing

$$\gamma_i = K \frac{\sum_{k=1}^{n} u_{ik}^m D_{ikA}^2}{\sum_{k=1}^{n} u_{ik}^m}$$
$$K > 0 \text{ (the most common choice is } K = 1) \tag{11}$$

where the $\{u_{ik}\}$ are a terminal FCM partition of X. Several aspects of PCM are discussed in [4] and [5]. PCM sometimes helps when the data are noisy. However, the user needs to be very careful about the choice of $\mathbf{V}_0$, the $\{\gamma_i\}$ and K when using PCM-AO [3], [5].

The first term of P_m is just J_m, and in the absence of the second term, unconstrained optimization will lead to the trivial solution $t_{ik} = 0 \,\forall\, i, k$. The second term of P_m acts as a penalty which tries to bring t_{ik} toward 1. Thus, as pointed out by Krishnapuram and Keller [2], [5], the terminal typicality matrix is strongly dependent on the choice of the $\{\gamma_i\}$. If γ_i is low, t_{ik} will be small, and if γ_i is high, t_{ik} will be high. Some limiting properties of t_{ik} in (10a) will help explain this sensitivity of PCM-AO to the $\{\gamma_i\}$; see (12)–(14), as shown at the bottom of the page. Since the columns and rows of the typicality matrix are independent of each other, $P_m(T, \mathbf{V})$ can be split into (cn) subobjective functions as shown in (9). If the initialization of each row is not sufficiently distinct, coincident clusters may result. This is one reason for using FCM-AO to generate the initializing parameters for PCM-AO. Sometimes coincident clusters can be advantageous. For example, when we start with a large value of c, and defuzzification of the terminal PCM-AO partition results in $c' < c$ distinct clusters, this may indicate that the correct value of c is c'. However, this is useful only when the data really has c' clusters. PCM-AO can terminate with $c' < c$ clusters even when there are really $c > c'$ distinct clusters in the data. In other words, there is no guarantee that $c = c'$ is the right number of clusters even when PCM-AO seems to suggest this.

The problem of coincident cluster generation is not specific to P_m. This problem will arise for any *separable* objective function, that is, a function that can be expressed as a sum of independent subobjective functions. This problem is not caused by a poor choice of penalty terms; rather, it is due to the *lack of* constraints placed on the typicality matrix.

To summarize Sections II and III: $\sum_{i=1}^{c} u_{ik} = 1$ for fuzzy (or probabilistic) $U's \in M_{\text{fcn}}$ is *too strong*—it forces outliers to belong to one or more clusters and, therefore, unduly influence the main structure of X. On the other hand, the constraint $0 < \sum_{i=1}^{c} t_{ik} \le c$ is *too weak*—it allows data point to behave almost independently of the other data in X, resulting in $T \in (M_{\text{pcn}} - M_{\text{fcn}})$ that is very brittle to the choices of its parameters. The FPCM model was proposed to exploit the benefits of fuzzy and possibilistic modeling while circumventing their weaknesses.

IV. What's Wrong With FPCM?

We believe that memberships (or relative typicalities) and possibilities (or absolute typicalities) are *both* important for correct interpretation of data substructure. When we want to crisply label a data point, membership is a plausible choice as it is natural to assign a point to the cluster whose prototype is closest to the point. On the other hand, while estimating the centroids, typicality is an important means for alleviating the undesirable effects of outliers.

The FCM necessary condition in (3a) for u_{ik} is a function of $\mathbf{x}_k$ and all c centroids. On the other hand, the necessary condition for typicality t_{ik} in (10a) is a function of $\mathbf{x}_k$ and $\mathbf{v}_i$ alone. That is, u_{ik} is influenced by the positions of *all* c cluster centers, whereas t_{ik} is affected by *only one*. Since t_{ik} depends on only (and not relative to others) the distance from $\mathbf{x}_k$ to $\mathbf{v}_i$ and on the constant γ_i, we regard u_{ik} as the *relative typicality* and t_{ik} the *absolute typicality* (or just typicality) of $\mathbf{x}_k$ with respect to cluster i.

$$\lim_{\gamma_i \to \infty} \{t_{ik}\} = 1, \qquad 1 \le i \le c;\ 1 \le k \le n \tag{12}$$

$$\lim_{m \to 1^+} \{t_{ik}\} = \begin{cases} 1, & \text{if } D_{ikA}^2 < \gamma_i \\ \frac{1}{2}, & \text{if } D_{ikA}^2 = \gamma_i \\ 0, & \text{if } D_{ikA}^2 > \gamma_i \end{cases}, \qquad 1 \le i \le c;\ 1 \le k \le n \text{ and} \tag{13}$$

$$t_{ik} = \frac{1}{2} \text{ if } D_{ikA}^2 = \gamma_i, \qquad 1 \le i \le c;\ 1 \le k \le n. \tag{14}$$

Even if we change the penalty terms of the PCM functional in (8), we will still have the problem of coincident clusters as long as the objective function is separable. In [12], for the FPCM model, we proposed the following optimization problem:

$$\underbrace{\min}_{(U,T,\mathbf{V})}\left\{J_{m,\eta}(U,T,\mathbf{V};X)=\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik}^{m}+t_{ik}^{\eta})D_{ikA}^{2}\right\} \tag{15}$$

subject to the constraints $m>1$, $\eta>1$, $0\le u_{ik},t_{ik}\le 1$, $D_{ikA}=\|\mathbf{x}_k-\mathbf{v}_i\|_A$, and

$$\sum_{i=1}^{c}u_{ik}=1 \qquad \forall\, k,\ \text{i.e.},\ U\in M_{\text{fcn}} \text{ and} \tag{16}$$

$$\sum_{k=1}^{n}t_{ik}=1 \qquad \forall\, i,\ \text{i.e.},\ T^{t}\in M_{\text{fnc}}. \tag{17}$$

The *transpose* of admissible T's are members of the set M_{fnc}. We view T as a typicality assignment of the n objects to the c clusters. The possibilistic term $\sum_{i=1}^{c}\sum_{k=1}^{n}t_{ik}^{\eta}D_{ikA}^{2}$ will distribute the $\{t_{ik}\}$ with respect to all n data points, but not with respect to all c clusters. Under the usual conditions placed on c-means optimization problems, we obtained the first order necessary conditions for extrema of $J_{m,\eta}$, which we state here as a theorem.

Theorem FPCM [12]: If $D_{ikA}=\|\mathbf{x}_k-\mathbf{v}_i\|_A>0$ for all i and $k,m,\eta>1$, and X contains at least c distinct data points, then $(U,T^{t},\mathbf{V})\in M_{\text{fcn}}\times M_{\text{fcn}}\times\Re^{p}$ may minimize $J_{m,\eta}$ only if

$$u_{ik}=\left(\sum_{j=1}^{c}\left(\frac{D_{ikA}}{D_{jkA}}\right)^{2/(m-1)}\right)^{-1}$$
$$1\le i\le c;\ \ 1\le k\le n \tag{18a}$$

$$t_{ik}=\left(\sum_{j=1}^{n}\left(\frac{D_{ikA}}{D_{ijA}}\right)^{2/(\eta-1)}\right)^{-1}$$
$$1\le i\le c;\ \ 1\le k\le n \ \text{ and} \tag{18b}$$

$$\mathbf{v}_i=\frac{\sum_{k=1}^{n}(u_{ik}^{m}+t_{ik}^{\eta})\,\mathbf{x}_k}{\sum_{k=1}^{n}(u_{ik}^{m}+t_{ik}^{\eta})},\qquad 1\le i\le c. \tag{18c}$$

Proof: The expressions in (18a) and (18b) both have the functional form in (3a) and are obtained exactly in the same manner. These two equations follow immediately with the Lagrange multiplier theorem. Actually, (18a) is obtained by solving the reduced problem $\underbrace{\min}_{U_k\in N_{\text{fcn}}}\left\{J_{m,\eta}^{k}(U_k)=\sum_{i=1}^{c}(u_{ik}^{m}+t_{ik}^{\eta})D_{ikA}^{2}\right\}$ with T and $\mathbf{V}$ fixed for the k-th column U_k of U. The set over which $J_{m,\eta}^{k}$ is minimized is N_{fcn}, the set of all fuzzy label vectors in $\Re^{c}$. Formula (18b) is obtained in exactly the same way by solving the other half problem for the i-th row $\mathbf{T}^{i}$ of T, i.e., $\underbrace{\min}_{\mathbf{T}^{i}\in N_{\text{fnc}}}\left\{J_{m,\eta}^{i}(\mathbf{T}^{i})=\sum_{k=1}^{n}(u_{ik}^{m}+t_{ik}^{\eta})D_{ikA}^{2}\right\}$. The set over which $J_{m,\eta}^{i}$ is minimized is N_{fnc}, the set of all fuzzy label vectors in $\Re^{n}$. Both decompositions of $J_{m,\eta}$ are possible because $J_{m,\eta}$ is a sum of nonnegative terms, so the sum of the minimums is the minimum of the sums.

If one or more $D_{ikA}=\|\mathbf{x}_k-\mathbf{v}_i\|_A=0$, we can use the usual singularity conditions to continue (18a) and (18b). That is, assign zero weights to all u_{ik} and t_{ik} such that $D_{ikA}=\|\mathbf{x}_k-\mathbf{v}_i\|_A>0$, and distribute nonzero memberships and possibilities arbitrarily subject to $\sum_{i=1}^{c}u_{ik}=1$ and $\sum_{k=1}^{n}t_{ik}=1$. The correctness of these assignments follows exactly as it does in the proof of Theorem FCM.

For (18c), the reduced problem $\underbrace{\min}_{V\in\Re^{cp}}\left\{\hat{J}_{m,\eta}(\mathbf{V})=\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik}^{m}+t_{ik}^{\eta})D_{ikA}^{2}\right\}$ can be solved by fixing U and T, and zeroing the gradient of $\hat{J}_{m,\eta}(\mathbf{V})$ with respect to each $\mathbf{v}_i$. This results in (18c), as long as $D_{ikA}=\|\mathbf{x}_k-\mathbf{v}_i\|_A$ is an inner product norm, and hence, differentiable in each $\mathbf{v}_i$.

Equation (18a) is identical in form to (3a), the FCM membership formula. This does *not* mean that FCM and FPCM will generate the same membership values, even if both algorithms are started with the same initialization. Why? Because (18c) is different from (3b); in FPCM, U and T are both used for the update to $\mathbf{V}$, so the succeeding estimate of U will differ from the one produced by FCM. We next state without proof some limiting results for FPCM

$$\lim_{m\to 1^{+}}\{u_{ik}\}=\left\{\begin{array}{ll}1, & D_{ikA}<D_{jkA}\,\forall\, j\ne i\\ 0, & \text{otherwise}\end{array}\right\},$$
$$1\le i\le c;\ \ 1\le k\le n \tag{19a}$$

$$\lim_{m\to\infty}\{u_{ik}\}=\frac{1}{c},\qquad 1\le i\le c;\ \ 1\le k\le n \tag{19b}$$

$$\lim_{\eta\to 1^{+}}\{t_{ik}\}=\left\{\begin{array}{ll}1, & D_{ikA}<D_{ijA}\,\forall\, j\ne k\\ 0, & \text{otherwise}\end{array}\right\},$$
$$1\le i\le c;\ \ 1\le k\le n \tag{19c}$$

$$\lim_{\eta\to\infty}\{t_{ik}\}=\frac{1}{n},\qquad 1\le i\le c;\ \ 1\le k\le n \tag{19d}$$

$$\lim_{\substack{m\to\infty\\ \eta\to\infty}}\{\mathbf{v}_i\}=\frac{\sum_{k=1}^{n}\mathbf{x}_k}{n}=\bar{\mathbf{v}},\qquad 1\le i\le c. \tag{19e}$$

FPCM has the same type of singularity as FCM. FPCM does not suffer from the sensitivity problem that PCM seems to exhibit [because of the limit result in (12)]. Unfortunately, when the number n of data points is large, the typicality values computed by (18b) will be very small. Thus, after the FPCM-AO algorithm for approximating solutions to (15) based on iteration through (18) terminates, the typicality values may need to be scaled up. Conceptually, this is no different than scaling typicalities with respect to γ_i as is done in PCM. In PCM, γ_i is used to scale t_{ik} such that at $D_{ikA}=\sqrt{\gamma_i}$, the typicality is 0.5. While scaling seems to "solve" the small value problem (which is caused by the row sum constraint on T), the scaled values do not possess any additional information about points in the data. Thus scaling the $\{t_{ik}\}$ is an artificial fix for a mathematical defect of FPCM. We can avoid the scaling step and make FPCM more useful for large data sets, with the new PFCM model.

V. A New PFCM

The apparent problem of FPCM is that it imposes a constraint on the typicality values (sum of the typicalities over all data points to a particular cluster is 1). We relax the constraint (row $\text{sum} = 1$) on the typicality values but retain the column constraint on the membership values. This leads to the following optimization problem:

$$\underbrace{\min}_{(U,T,V)} \left\{ J_{m,\eta}(U,T,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{c} (au_{ik}^{m} + bt_{ik}^{\eta}) \times \|\mathbf{x}_k - \mathbf{v}_i\|_A^2 + \sum_{i=1}^{c}\gamma_i \sum_{k=1}^{n}(1-t_{ik})^{\eta} \right\} \quad (20)$$

subject to the constraints $\sum_{i=1}^{c} u_{ik} = 1 \,\forall\, k$, and $0 \le u_{ik}, t_{ik} \le 1$. Here, $a > 0$, $b > 0$, $m > 1$ and $\eta > 1$. In (20), the $\gamma_i > 0$ are user defined constants. The constants a and b define the relative importance of fuzzy membership and typicality values in the objective function. Note that, in (20) u_{ik} has the same meaning of membership as that in FCM. Similarly, t_{ik} has the same interpretation of typicality as in PCM. At this point, a natural question comes: should we constrain ourselves to $a + b = 1$? Doing so, we can eliminate one parameter, yet we can assign different relative importance to u_{ik} and t_{ik}. However, this has an undesirable effect. If we increase the importance (weight) of membership then that necessarily forces us to reduce the importance of typicality by the same amount. This is too restrictive. Also, we will see later that the optimal typicality values depend on the magnitude of b. So by constraining $a + b = 1$, we lose modeling flexibility.

If $b = 0$, and $\gamma_i = 0$ for all i, then (20) reduces to the FCM optimization problem in (2); while $a = 0$ converts it to the usual PCM model in (8). Later, we will see that when $b = 0$, even if we do not set $\gamma_i = 0$ for all i, (20) implicitly becomes equivalent to the FCM model. Like FPCM, under the usual conditions placed on c-means optimization problems, we get the first-order necessary conditions for extrema of $J_{m,\eta}$.

Theorem PFCM: If $D_{ikA} = \|\mathbf{x}_k - \mathbf{v}_i\|_A > 0$ for all i and $k, m, \eta > 1$, and X contains at least c distinct data points, then $(U,T,V) \in M_{\text{fcn}} \times M_{\text{pcn}} \times \Re^p$ may minimize $J_{m,\eta}$ only if

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{D_{ikA}}{D_{jkA}} \right)^{2/(m-1)} \right)^{-1} \quad 1 \le i \le c;\ \ 1 \le k \le n \quad (21)$$

$$t_{ik} = \frac{1}{1 + \left(\frac{b}{\gamma_i} D_{ikA}^2 \right)^{1/(\eta-1)}} \quad 1 \le i \le c;\ \ 1 \le k \le n \quad (22)$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} (au_{ik}^{m} + bt_{ik}^{\eta})\,\mathbf{x}_k}{\sum_{k=1}^{n} (au_{ik}^{m} + bt_{ik}^{\eta})} \quad 1 \le i \le c. \quad (23)$$

Proof: Equations (21) and (22) follow immediately with the Lagrange multiplier theorem. Equation (21) is obtained by solving the reduced problem $\underbrace{\min}_{U_k \in N_{\text{fcn}}} \left\{ J_{m,\eta}^{k}(U_k) = \sum_{i=1}^{c} (au_{ik}^{m} + bt_{ik}^{\eta}) D_{ikA}^2 \right\}$ with T and $\mathbf{V}$ fixed for the kth column U_k of U. The function $J_{m,\eta}^{k}$ is minimized over N_{fcn}. The membership expression is independent of the constant a. The influence of a on the memberships comes via the centroids (23). Equation (22) is obtained solving the problem $\min \left\{ J_{m,\eta}^{ik} = (au_{ik}^{m} + bt_{ik}^{\eta}) D_{ikA}^2 + \gamma_i (1 - t_{ik}) \right\}$. The constant b has a direct influence on the typicality values. These decompositions of $J_{m,\eta}$ are possible because $J_{m,\eta}$ is a sum of nonnegative terms, so the sum of the minimums is the minimum of the sums.

If one or more $D_{ikA} = \|\mathbf{x}_k - \mathbf{v}_i\|_A = 0$, assign zero values to all u_{ik} and t_{ik} such that $D_{ikA} = \|\mathbf{x}_k - \mathbf{v}_i\|_A > 0$, and distribute nonzero memberships arbitrarily subject to $\sum_{i=1}^{c} u_{ik} = 1$ and assign $t_{ik} = 1$ to those $\mathbf{v}_i$ for which $D_{ikA} = 0$. The correctness of these assignments follows exactly as it does in the proof of Theorem FCM.

Finally, the reduced problem

$$\underbrace{\min}_{\mathbf{V} \in R^{cp}} \left\{ J_{m,\eta}(\mathbf{V}) = \sum_{i=1}^{c}\sum_{k=1}^{n} (au_{ik}^{m} + bt_{ik}^{\eta}) D_{ikA}^2 \right\}$$

is solved by fixing U and T, and zeroing the gradient of $J_{m,\eta}(\mathbf{V})$ with respect to each $\mathbf{v}_i$. This results in (23), as long as $D_{ikA} = \|\mathbf{x}_k - \mathbf{v}_i\|_A$ is an inner product induced norm.

We state some interesting properties of PFCM; see (24a)–(g), as shown at the bottom of the next page. Property P6 shows that PFCM behaves like FCM as the exponents grow without bound. That is, irrespective of the values of the constants a and b, all c centroids approach the overall (grand) mean as $m \to \infty$ and $\eta \to \infty$. Equation (23) shows that if we use a high value of b compared to a, then the centroids will be more influenced by the typicality values than the membership values. On the other hand, if we use a higher value of a then the centroids will be more influenced by the membership values. Thus, to reduce the effect of outliers, we should use a bigger value for b than a. Similar effects can also be obtained by controlling the choice of m and η. For example, if we use a large value of m and a smaller value for η, then the effect of outliers on the centroids will be reduced. However, a very large value of m will reduce the effect of memberships on the prototypes and the model will behave more like the PCM model, resulting from (24b). The PFCM algorithm is also included in Appendix.

VI. Numerical Examples

We compare various aspects of FCM, PCM, and PFCM with five data sets: X_{10}, X_{12}, X_{400}, X_{550} and IRIS.

For all data sets we use the following *Computational protocols*: $\varepsilon = 0.000\,01$, $T_{\max}$ = maximum number of iterations $= 100$, and $\| * \|_A$ is the Euclidean norm. The number of clusters c is 3 for IRIS and for all other data sets it is fixed at $c = 2$. For both PCM and PFCM we first run FCM to termination and use (11) with $K = 1$ to find the values of γ_i. All trials terminated with the convergence criteria after a few iterations.

TABLE II
TERMINAL PROTOTYPES PRODUCED BY FCM, PCM, AND PFCM ON X_{10} AND X_{12}

	FCM : m=2		PCM : η= 2		PFCM : a=1, b=1, m=2, η= 2	
X_{10}	-3.36	3.36	-3.24	3.24	-3.34	3.34
	0.00	0.00	0.00	0.00	0.00	0.00
X_{12}	-2.99	2.99	-2.15	2.15	-2.84	2.84
	0.54	0.54	0.02	0.02	0.36	0.36

Example 2:

Data sets: X_{10}, X_{12}.
Algorithm: FCM, PCM, PFCM.
Initialization:

$$V_0 = \begin{bmatrix} -3.34 & 1.67 \\ 1.67 & 0.00 \end{bmatrix}. \tag{25}$$

The initialization in (25) is obtained by randomly selecting two data points from X_{10} (each column of V_0 is a data point). Table II shows the centroids produced by FCM, PCM and PFCM. For X_{10} all three algorithms produce good prototypes. However, a closer look reveals that PFCM produces prototypes that match exactly (up to rounding) with the ideal (true) centroids $V_{\text{ideal}} = \begin{bmatrix} -3.34 & 3.34 \\ 0 & 0 \end{bmatrix}$ that would be obtained in absence of outliers.

If $\mathbf{x}_{11}$ and $\mathbf{x}_{12}$ are to be "ignored," we hope to find the true centroids of X_{10} when X_{12} is processed. From Table II, we see that $V^{12}_{\text{FCM}} = \begin{bmatrix} -2.99 & 2.99 \\ 0.54 & 0.54 \end{bmatrix}$, $V^{12}_{\text{PCM}} = \begin{bmatrix} -2.15 & 2.15 \\ 0.02 & 0.02 \end{bmatrix}$ and $V^{12}_{\text{PFCM}} = \begin{bmatrix} -2.84 & 2.84 \\ 0.36 & 0.36 \end{bmatrix}$. To make a crude assessment how each method has accounted for the inlier and outlier, we compute $E_X = \left\| V_{\text{ideal}} - V^{12}_X \right\|^2$, where X is FCM/PCM/PFCM: $E_{\text{FCM}} = 0.4141$, $E_{\text{PCM}} = 1.3161$ and $E_{\text{PFCM}} = 0.3796$. Although PCM does assign different typicality values to $\mathbf{x}_{11}$ and $\mathbf{x}_{12}$, the centroids produced by PCM are not very good compared to those generated by FCM and PFCM. With a different initialization and different choice of η, PCM should produce better results.

Table III exhibits the terminal U and T from FCM, PCM and PFCM with $a = 1.0$, $b = 1.0$, $m = 2.0$ and $\eta = 2.0$ (corresponding to Table II). Table III shows that both U_{PFCM} and U_{FCM} result in the same max-membership hard partition. The relative ordering of points (in terms of membership values) also remains the same.

PFCM provides a more informative description of the data than FCM, since it provides roughly the same membership information but also shows, via the typicalities, for example, that $\mathbf{x}_{12}$ is much more atypical than $\mathbf{x}_{11}$ for either cluster. PFCM assigns the highest typicality in the left cluster to $\mathbf{x}_3$. This is very reasonable because $\mathbf{x}_3$ is located at the center of the left cluster. Similarly, $\mathbf{x}_8$ is most typical of the right cluster. Here, $\mathbf{x}_{12}$ is least typical to either cluster indicating that it is an outlier. The typicality for $\mathbf{x}_{12}$ in both clusters is an order of magnitude smaller than the typicality value for $\mathbf{x}_{11}$. This enables us to prune outliers from the data to reduce the effects of noise (like PCM did in [2]). For this initialization, PCM assigns the highest typicality values to $\mathbf{x}_5$ and $\mathbf{x}_6$ because the PCM centroids are located at $V^{12}_{\text{PCM}} = \begin{bmatrix} -2.15 & 2.15 \\ 0.02 & 0.02 \end{bmatrix}$. Both T_{PCM} and T_{PFCM} result in the same (hardened) partition. Although for $\mathbf{x}_{12}$ both PFCM and PCM assign almost the same typicality values, for $\mathbf{x}_{11}$ the typicality values assigned by PFCM are significantly smaller than those assigned by PCM. Hence, the PFCM prototypes are less influenced by the noise points.

Example 3:

Data set: X_{10}.
Algorithm: PFCM.
Initialization: $\mathbf{V}_0 = \begin{bmatrix} 0.07 & 0.36 \\ 0.40 & 0.99 \end{bmatrix}$.

We now investigate the impact of various parameters on the performance of the PFCM algorithm. The initialization from (7a) is not a good one, because the two vectors in $\mathbf{V}_0$ are very close to each other with respect to the smallest hyperbox containing the data. However, we use it to show the "robustness" of different algorithms on the initializations. With initialization (7a) PCM generates coincident clusters. Krishnapuram and Keller [2], [5] suggest using the terminal FCM centroids to initialize

$$\text{P1}: \lim_{m\to 1^+}\{u_{ik}\} = \begin{cases} 1, & D_{ikA} < D_{jkA}\ \forall\, j \neq i \\ 0, & \text{otherwise} \end{cases}, 1 \le i \le c;\ 1 \le k \le n \tag{24a}$$

$$\text{P2}: \lim_{m\to\infty}\{u_{ik}\} = \frac{1}{c}, \qquad 1 \le i \le c;\ 1 \le k \le n \tag{24b}$$

$$\text{P3}: \lim_{\eta\to\infty}\{t_{ik}\} = 0.5, \qquad 1 \le i \le c;\ 1 \le k \le n \tag{24c}$$

$$\text{P4}: \lim_{\eta\to 1^+}\{t_{ik}\} = \begin{cases} 1, & \text{if } bD^2_{ikA} < \gamma_i \\ \frac{1}{2}, & \text{if } bD^2_{ikA} = \gamma_i \\ 0, & \text{if } bD^2_{ikA} > \gamma_i \end{cases}, 1 \le i \le c;\ 1 \le k \le n \tag{24d}$$

$$\text{P5}: t_{ik} = \frac{1}{2} \text{ if } bD^2_{ikA} = \gamma_i, \qquad 1 \le i \le c;\ 1 \le k \le n \tag{24e}$$

$$\text{P6}: \lim_{\substack{m\to\infty \\ \eta\to\infty}}\{\mathbf{v}_i\} = \sum_{k=1}^{n} \mathbf{x}_k = \bar{\mathbf{v}}, \qquad 1 \le i \le c \tag{24f}$$

$$\text{P7}: \text{ if } b = 0, \text{ then } t_{ik} = 1, \qquad 1 \le i \le c;\ 1 \le k \le n. \tag{24g}$$

TABLE III
TERMINAL U AND T FROM FCM ($m = 2$), PCM ($\eta = 2$), AND PFCM ($a = 1, b = 1, m = 2, \eta = 2$) FOR X_{12} WITH INITIALIZATION PERFORMED AS IN (25)

Point	$U_{FCM} \in M_{fcn}$ (from FCM)		$U_{PFCM} \in M_{fcn}$ (from PFCM)		$T_{PFCM} \in M_{pcn}$ (from PFCM)		$T_{PCM} \in M_{pcn}$ (from PCM)	
1	0.936	0.064	0.928	0.072	0.621	0.113	0.492	0.134
2	0.967	0.033	0.953	0.047	0.801	0.165	0.655	0.194
3	0.99	0.01	0.990	0.01	0.953	0.171	0.847	0.208
4	0.899	0.101	0.906	0.094	0.642	0.157	0.648	0.193
5	0.916	0.084	0.932	0.068	0.840	0.278	0.972	0.351
6	0.084	0.916	0.068	0.932	0.278	0.840	0.351	0.972
7	0.033	0.967	0.047	0.953	0.165	0.801	0.194	0.655
8	0.01	0.99	0.01	0.990	0.171	0.953	0.208	0.847
9	0.101	0.899	0.094	0.906	0.157	0.642	0.193	0.648
10	0.064	0.936	0.072	0.928	0.113	0.621	0.134	0.492
11	0.500	0.500	0.500	0.500	0.490	0.490	0.631	0.631
12	0.500	0.500	0.500	0.500	0.072	0.072	0.070	0.070

PCM, so we also report a few results generated by PCM (in Example 5) and PFCM when the initialization is done by the FCM prototypes.

Table IV shows the results produced by PFCM on X_{10} for different values of a, b, m and η. The centroids corresponding to run **1** in Table IV are identical (up to rounding) to the PFCM centroids in Table II, which are generated by PFCM using a different initialization. The first four results in Table IV demonstrate that if we vary b from 1 to 7, keeping all other parameters fixed, the terminal PFCM prototypes are essentially unchanged. Why? Increasing b assigns more importance to typicality, but the results do not change much since X_{10} has nice well separated clusters, However, if the data have outliers then, we shall see that giving more importance to typicality by increasing b improves the prototypes. (This property can be used to get a better understanding of the data.).

Comparing **5** with **6**, we see that increasing the value of m from 5.0 to 7.0, when the value of η is low (1.5), degrades the prototypes because the membership values become more uniform. That is, the membership component of (23) tries to push the prototypes toward the grand mean, and this degrades the prototypes, making typicality more important for computation of centroids. Run **7** shows very poor centroids. Comparing the choice of parameters in **5** with those in **7**, we find that only b changes from 1.0 to 5.0. The high value of m removes the skewness in the membership values (membership values become close to 0.5 for all data points). Hence, as just explained, the FCM part of the objective function tries to move the centroids toward the grand mean vector and a low value of η makes typicality assignments almost crisp. So the cluster centers will drift toward the grand mean and typicality will be high for only a few points close to the centroids. Inspection of T reveals that for just one point in each cluster the typicality is almost 1.0, while for all other points in the same cluster the typicality values are very small. Hence, the poor results. If there are a few noise points, the results could even be worse depending on the locations the prototypes. However, if we increase the value of η, we can recover from this situation.

The centroids generated by run **8** are much better than those in run **7**. Between runs **7** and **8** only η changes from 1.5 to 10. For this choice, the typicality values are close to 0.5 for all data points. Consequently, typicality values do not have much influence on the prototypes. So, for this choice of parameters, PFCM behaves more like FCM. Run **11** in Table IV corresponds to FCM with $m = 5.0$. It is interesting to see that we get the same (up to rounding) centroids as in **8**. For this data set even with $m = 2$ and $\eta = 7$, we get reasonably good results, see run **9**. This is practically the FCM result. Since $\eta = 7.0$, the effect of typicality is drastically reduced and we get results similar to the FCM. Table IV includes the PFCM prototypes for a few other choices of parameters, which are also quite good.

We discussed earlier that constraining $a + b = 1$ is not expected to result in good prototypes. To illustrate this we report three more results. The ratio a/b for run **12** is the same as that of Run **1**, yet the centroids produced by run **12** are different from those produced by run **1**, though all other protocols including

TABLE IV
RESULTS PRODUCED BY PFCM FOR DIFFERENT VALUES OF THE PARAMETERS WITH X_{10}

	a	b	m	η	$\mathbf{v}_1$	$\mathbf{v}_2$
1	1	1	2	2	-3.34 0.00	3.34 0.00
2	1	3	2	2	-3.35 0.00	3.35 0.00
3	1	6	2	2	-3.35 0.00	3.35 0.00
4	1	7	2	2	-3.35 0.00	3.35 0.00
5	1	1	5	1.5	-3.33 0.00	3.33 0.00
6	1	1	7	1.5	-2.01 0.00	2.01 0.00
7	1	5	5	1.5	-1.76 0.00	1.76 0.00
8	1	5	5	10	-3.32 0.00	3.32 0.00
9	1	1	2	7	-3.35 0.00	3.35 0.00
10	1	4	3	2	-3.34 0.00	3.34 0.00
11	1	0 (≡FCM)	5	-----	-3.32 0.00	3.32 0.00
12	0.5	0.5	2	2	-3.297 0.00	3.297 0.00
13	0.25	0.75	2	2	-3.289 0.00	3.289 0.00
14	0.1667	0.8333	5	1.5	2.71 0.00	2.71 0.00

initialization remain the same as before. Similarly, run **13** has the same ratio of a/b as that of run **2**. In this case too, there is a noticeable difference between the two sets of centroids. Run **14** with $a/b = 1/6$ corresponds to run **3**. However, run **14** results in coincident clusters. Because the absolute weight $(a = 0.1667)$ assigned to the membership component of the objective function in (20) is quite small making the algorithm behave almost like PCM.

Lessons From Example.3: We must not use a very large value of η, since as (24) suggests, larger η's reduce the effect of typicality. Also, it seems that b should be larger than a so that typicality values influence the computation of prototypes, thereby reducing the effects of outliers. But b should not be "too large," for then the effect of membership values will be practically eliminated.

Example 4:

Data set: X_{12}.
Algorithm: PFCM.
Initialization: $\mathbf{V}_0 = \begin{bmatrix} 0.07 & 0.36 \\ 0.40 & 0.99 \end{bmatrix}$.

Table V depicts the centroids generated by PFCM on X_{12} using the same parameters as in Table IV. While we did not see any change in the prototypes when b was increased keeping all other parameters fixed for X_{10}, the centroids do improve for X_{12} (see Runs **1**–**4** in Table V). This happens because larger values of b increase the relative importance of the typicalities in determining the prototypes. Except for run **7**, PFCM generated good prototypes. Run **7** generates coincident clusters. Why? We mentioned earlier that we can choose the parameters of PFCM so that it behaves more like PCM. If a is very small, and the initial centroids are not well placed (separated), then PFCM, like PCM, can generate centers that are very close to each other (i.e., practically coincident clusters). Run **7** in Table V corresponds to this situation. To further illustrate this point, we ran PFCM using the same set of parameters as in **7**, but initialized with the terminal FCM centers. In this case PFCM generates $\mathbf{V}^{12}_{\mathrm{PFCM}} = \begin{bmatrix} -3.27 & 3.27 \\ 0.09 & 0.09 \end{bmatrix}$. This indicates that in run **7** we obtained coincident clusters because the initial prototypes were very close to each other. Run **11** shows the prototypes produced by FCM with $m = 5.0 (a = 1, b = 0)$. In the context of Table IV, we argued and numerically demonstrated that the choice of parameters in run **8** will make PFCM behave like FCM. This is also true for X_{12}, run **8** and run **11** generate almost the same prototypes. For X_{12}, like X_{10}, we made three runs (**12**–**14**) constraining $a + b = 1$, but maintaining the same ratios a/b corresponding to runs **1**–**3**. The conclusions drawn from runs **12**–**14** in Table IV are equally applicable here.

Example 5:

Data set: X_{12}.
Algorithm: PCM.
Initialization: FCM centroids.

Runs **1** through **5** of Table VI show the PCM centers for different values of η when the PCM algorithm is initialized with the terminal FCM prototypes. With an increase in η beyond 2, PCM generates almost coincident clusters. Except for one point the typicality values become close to 0.5. In other words, with a moderately large value of η the limit property of typicality in (14) is attained. However, using our PFCM framework to realize PCM, we can overcome this situation by increasing the value of b. Run **6** and run **7** in Table VI represent two such cases for which we use respectively $b = 3.0$ and $b = 10.0$ with $a = 0.0$ (PFCM is equivalent to PCM). With $b = 10.0$, PCM (realized by PFCM) can produce excellent prototypes even with $\eta = 3.0$. Using PCM also one can get results like run **7** with appropriately scaled values of γ_i.

TABLE V
RESULTS PRODUCED BY PFCM FOR DIFFERENT VALUES OF THE PARAMETERS WITH X_{12}

	a	b	m	η	$\mathbf{v}_1$	$\mathbf{v}_2$
1	1	1	2	2	-2.84 0.36	2.84 0.36
2	1	3	2	2	-3.00 0.33	3.00 0.33
3	1	6	2	2	-3.11 0.54	3.11 0.54
4	1	7	2	2	-3.13 0.26	3.13 0.26
5	1	1	5	1.5	-3.03 0.08	3.03 0.08
6	1	1	7	1.5	-3.03 0.03	3.03 0.03
7	1	5	5	1.5	-0.00 0.05	0.00 0.05
8	1	5	5	10	-2.93 0.47	2.93 0.47
9	1	1	2	7	-2.97 0.54	2.97 0.54
10	1	4	3	2	-3.11 0.21	3.11 0.21
11	1	0 (=FCM)	5	----	-2.96 0.47	2.96 0.47
12	0.5	0.5	2	2	-2.53 0.32	2.53 0.32
13	0.25	0.75	2	2	-2.48 0.20	2.48 0.20
14	0.1667	0.8333	5	1.5	2.82 0.02	2.82 -0.02

Next, we discuss results on X_{400} and X_{550}. X_{400} is a mixture of two 2-variate normal distributions with mean vectors $\begin{pmatrix} 5.0 \\ 6.0 \end{pmatrix}$ and $\begin{pmatrix} 5.0 \\ 12.0 \end{pmatrix}$. Each cluster has 200 points, while X_{550} is an augmented version of X_{400} with an additional 150 points uniformly distributed over $[0, 15] \times [0, 11]$. Fig. 3 shows the scatterplot of X_{400} and X_{550}.

Example 6:

Data Sets: X_{400} and X_{550}.
Algorithms: FCM, PFCM, and PCM.
Initialization:

$$\mathbf{V}_0 = \begin{pmatrix} 5.97 & 4.75 \\ 6.12 & 5.14 \end{pmatrix}. \tag{26}$$

The initialization in (26) (which is displayed with only two significant digits) is obtained by two randomly selected data points from X_{400}. Table VII depicts the prototypes generated by the three algorithms when the initialization in (26) is used. The true mean of X_{400} is $V_{\mathrm{true}} = \begin{bmatrix} 4.97 & 4.97 \\ 6.15 & 12.15 \end{bmatrix}$. The PCM algorithm did not produce good prototypes although the initial centroids

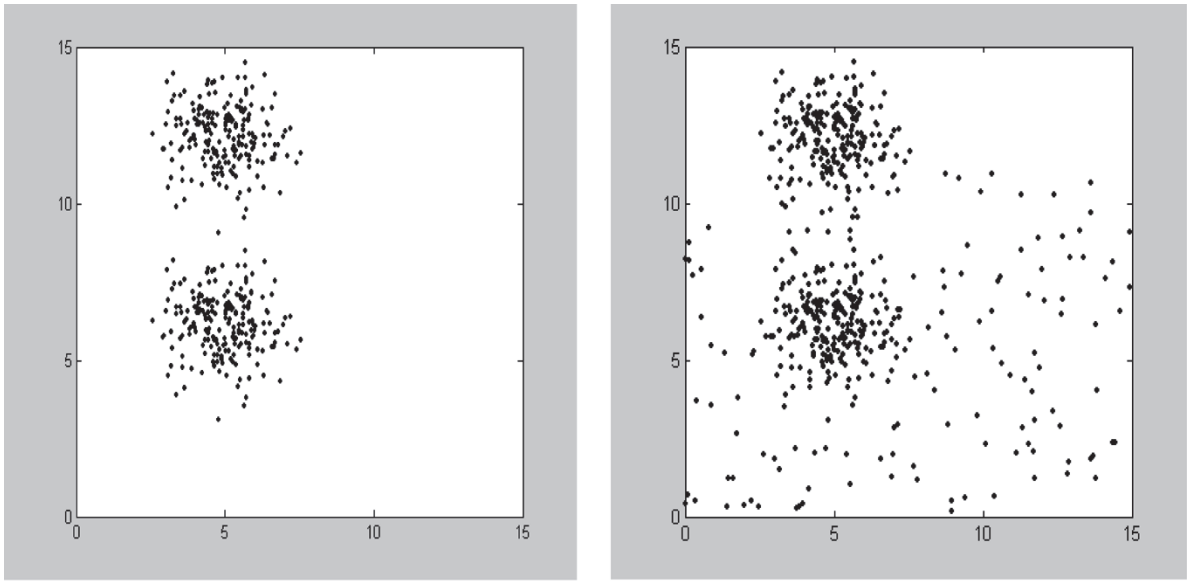

Fig. 3. (a) Scatter-plot of X_{400}. (b) Scatter-plot of X_{550}.

TABLE VI
RESULTS PRODUCED BY PCM FOR DIFFERENT VALUES OF THE PARAMETERS WITH X_{12} WHEN INITIALIZED WITH THE FCM CENTROIDS.

	a	b	m	η	$\mathbf{v}_1$	$\mathbf{v}_2$
1	0	1	-----	2	-2.146 0.019	2.146 0.019
2	0	1	-----	1.5	-2.96 0.54	2.96 0.54
3	0	1	-----	2.5	-0.00 0.04	0.00 0.04
4	0	1	-----	3	-0.00 0.06	0.00 0.06
5	0	1	-----	5	-0.00 0.07	0.00 0.07
6	0	5	-----	3	-1.67 0.01	1.67 0.01
7	0	10	-----	3	-3.20 0.01	3.20 0.01

are reasonably separated, but coming from the same cluster. Both FCM and PFCM result in good prototypes, which are very close to V_{true}. For X_{550}, like X_{12}, the effect of the noise points is smaller on the PFCM prototypes than it is on the FCM centroids. Column 4 of Table VII shows the PFCM centroids for both data sets with $a = 0.5$, $b = 4.0$, $m = 2.0$, and $\eta = 1.5$. Since b is comparatively larger than a, we expect the centroid computation in (23) to be more influenced by the typicality values and, hence, the centroids are not expected to be affected much by the noise points. Column 4 reveals that it is indeed the case. The PFCM centroids obtained for X_{550} in column 4 are closer to V_{true} than the centroids in column 3. Although, PCM produces coincident clusters with (26), if we use the terminal FCM prototypes to initialize PCM, it results in good prototypes as shown in column 5 of Table VII.

Example 7:

Data Sets : X_{400} and X_{550}.
Algorithms: FCM, PFCM and PCM.
Initialization:

$$\mathbf{V}_0 = \begin{pmatrix} 7.19 & 5.02 \\ 12.37 & 5.50 \end{pmatrix}. \tag{27}$$

We now compare FCM and PCM with PFCM with initialization in (27). Like Example 6, the initialization in (27) is obtained by randomly selecting two data points, but the two initial centroids are well separated and they come from two different clusters. Column 1 shows the FCM centroids while columns 2 and 3 depict the PCM and PFCM centroids. In this case, both PCM and PFCM algorithms result in good prototypes for X_{400}. The PCM and PFCM centroids on X_{400} are very close to each other. For X_{550}, both PCM and PFCM generate better prototypes than FCM.

Example 8:

Data Set: IRIS.
Algorithms: FCM, PFCM, and PCM.
Initialization: Stated in appropriate places.

TABLE VII
PROTOTYPES PRODUCED BY FCM, PCM AND FPCM WITH INITIALIZATION AS IN (26)

Results on X_{400}									
FCM (m=2.0)		PCM (η=2.0)		PFCM (a=1.0, b=1.0, m=2.0, η=2.0)		PFCM (a=0.5 b=4.0, m=2.0, η=1.5)		PCM (η=2.0, initialized with FCM centroids)	
4.97	4.97	5.01	5.01	4.98	4.97	5.02	5.01	4.98	4.99
6.14	12.18	6.28	6.28	6.16	12.18	6.18	12.22	6.30	12.09
Results on X_{550} (with 150 noise points)									
FCM (m=2.0)		PCM (η=2.0)		PFCM (a=1.0, b=1.0, m=2.0, η=2.0)		PFCM (a=0.5 b=4.0, m=2.0, η=1.5)		PCM (η=2.0, initialized with FCM centroids)	
5.68	5.25	5.03	5.03	5.49	5.21	5.21	5.12	5.07	5.02
5.45	11.65	6.41	6.33	5.70	11.71	5.97	11.99	6.74	11.51

TABLE VIII
PROTOTYPES PRODUCED BY FCM, PCM AND FPCM WHEN THEY ARE INITIALIZED BY (27)

Results on X_{400}							
FCM (m=2.0)		PCM (η=2.0)		PFCM (a=1.0 b=1.0, m=2.0, η=2.0)		PCM (η=2.0)	
4.97	4.97	4.98	4.99	4.98	4.97	4.98	4.99
6.14	12.18	6.30	12.09	6.17	12.17	6.30	12.09
Results on X_{550} (with 150 noise points)							
FCM (m=2.0)		PCM (η=2.0)		PFCM (a=1.0 b=1.0, m=2.0, η=2.0)		PCM (η=2.0)	
5.25	5.68	5.07	5.02	5.44	5.19	5.07	5.02
11.65	5.45	6.74	11.51	5.87	11.63	6.74	11.51

TABLE IX
RESULTS ON IRIS DATA WHEN ALGORITHMS ARE INITIALIZED WITH THREE RANDOMLY SELECTED DATA POINTS

Initial centroids			FCM (m=2, M_{FCMU}=16)			PCM (η=2, M_{PCMT}=50)			PFCM (a=1.0, b=1.0, m=2.0 η=2.0, M_{PFCMU}=13 M_{PFCMT}=14)		
5.80	5.00	5.10	6.78	5.89	5.00	6.17	6.17	5.00	6.62	5.92	5.00
2.70	2.00	3.50	3.05	2.76	3.41	2.88	2.88	3.40	3.01	2.79	3.41
5.10	3.50	1.40	5.65	4.36	1.48	4.76	4.76	1.48	5.46	4.40	1.48
1.90	1.00	0.30	2.05	1.40	0.25	1.61	1.61	0.25	1.99	1.41	0.25
Initial centroids			FCM (m=2, M_{FCMU}=16)			PCM (η=2, M_{PCMT}=50)			FPCM (a=1.0, b=1.0, m=2.0 η=2.0, M_{PFCMU}=13 M_{PFCMT}=14)		
7.20	5.50	5.10	6.77	5.89	5.00	6.17	6.17	5.00	6.62	5.92	5.00
3.00	2.40	3.30	3.05	2.76	3.41	2.88	2.88	3.40	3.01	2.79	3.41
5.80	3.70	1.70	5.65	4.36	1.48	4.76	4.76	1.48	5.46	4.40	1.48
1.60	1.00	0.50	2.05	1.40	0.25	1.61	1.61	0.25	1.99	1.41	0.25

We now demonstrate PFCM on a real data set, IRIS, with three clusters. IRIS [7], [8] is a four-dimensional data set containing 50 samples each of three types of Iris flowers. One of the three clusters (class 1) is well separated from the other two, while classes 2 and 3 have some overlap. The typical result of comparing hardened FCM or HCM partitions to the physically correct labels of Iris is 14–17 errors. We made several runs of PFCM on IRIS with different initializations and different choices of parameters. First, we report results of a few runs when FCM, PCM, and PFCM are initialized with three randomly selected data points (Table IX). We have made several such runs and in each case FCM and PFCM generated good prototypes, but PCM, even when the three initial centroids come from three different classes, resulted in coincident clusters (i.e., obtained two distinct clusters). Table IX displays some typical results with initializations shown in the first column of the same table. The resubstitution errors (M) with U_{PFCM} and T_{PFCM} are better than that with U_{FCM}. PCM gets two coincident clusters making 50 resubstitution errors.

We also made a few runs of PCM and PFCM when the algorithm is initialized with FCM terminal prototypes. In Table X, column 1, row 2 shows the FCM terminal prototypes that are used to initialize the PCM and PFCM algorithms (results shown in columns 2–3 of row 2). In this case, as expected PFCM produced good prototypes, but PCM again resulted in two coincident clusters (Table X). In row 3, column 1, we show another set of FCM terminal prototypes that are obtained with $m = 3$. When these FCM prototypes are used to initialize, PCM again produced two coincident clusters that we do not show in Table X. The PFCM algorithm in this

TABLE X
RESULTS ON IRIS DATA WHEN ALGORITHMS ARE INITIALIZED WITH FCM TERMINAL PROTOTYPES

FCM terminal prototypes (m=2, M_{FCMU}=16)			PCM (η=2, M_{FCMU}=50)			PFCM (a=1.0, b=1.0, m=2.0, η=2.0 M_{PFCMU}=13, M_{PFCMT}= 14)		
6.77	5.00	5.89	6.17	5.00	6.17	6.62	5.92	5.00
3.05	3.41	2.76	2.88	3.40	2.88	3.01	2.79	3.41
5.65	1.48	4.36	4.76	1.48	4.76	5.46	4.40	1.48
2.05	0.25	1.40	1.61	0.25	1.61	1.99	1.41	0.25
FCM terminal prototype (m=3, M_{FCMU}=16)			FPCM (a=1.0,b=1.0, m=3.0, η=3.0, M_{FPCMU}= 11, M_{FPCMT}= 10)			FPCM (a=1.0,b=3.0, m=2.0, η=1.5, M_{PFCMU}= 13, M_{PFCMT}= 11)		
6.69	5.00	5.91	6.41	5.01	6.05	6.67	5.02	5.83
3.04	3.40	2.79	2.95	3.40	2.85	3.04	3.42	2.79
5.55	1.49	4.38	5.14	1.51	4.54	5.55	1.47	4.29
2.04	0.25	1.40	1.83	0.26	1.48	2.07	0.25	1.34

case too, produced low resubstitution error both with respect to membership and typicality. In IRIS since class 2 and class 3 overlap, one can argue that IRIS has two clusters. So, in this regard, PCM does an excellent job in finding only two clusters even when the algorithm is asked to look for three clusters. However, there are applications where even when there is no clear cluster substructure, we want to find clusters. For example, although IRIS may be thought of having two clusters, if we want to extract rules or prototypes for classifier design, we need to find at least *three* clusters. With IRIS data PCM could not do this with all initializations that we tried but PFCM could. Thus for such applications, PFCM prototypes will be more useful than the FCM and PCM prototypes because PFCM prototypes are not sensitive to outliers and PFCM can avoid coincident clusters.

VII. CONCLUSION

We have argued the need for both membership and typicality values in clustering, and have proposed a possibilistic-fuzzy clustering model named PFCM. Unlike most fuzzy and possibilistic clustering algorithms, PFCM produces *three* outputs: a $c \times n$ fuzzy partition or membership matrix U of X; a $c \times n$ possibility matrix T of typicalities in X; and a set of c point prototypes $\mathbf{V}$ that compactly represents the clusters in X. PFCM has been tested on five data sets with many runs (not all discussed here) and its initial performance indicates that it does ameliorate problems suffered by FCM, PCM and FPCM. (U, T) pairs from PFCM are *not* the same as U from FCM and T from PCM, but they seem to share the same qualitative properties as the individually estimated matrices. PFCM has two additional parameters, a and b that define the relative importance of membership and typicality in the computation of centroids. By suitable combination of these parameters we can make PFCM behave more like FCM or PCM. Further investigation is required before much can be asserted about a good range of choices for the parameters (a, b, m, η).

The necessary conditions in (21)–(23) for the PFCM model hold for any inner product norm, e.g., for the scaled Mahalanobis norm [6], so the formulation is quite general. The two main branches of generalization for c-means models are locally "adaptive" schemes such as those of Gustafson and Kessel [9] or Dave and Bhaswan [10]; and extensions of the prototypes to shell-like surfaces, see for example Krishnapuram *et al.* [11]. The basic architecture of the PFCM -AO algorithm will clearly remain the same, but the update equations for extensions in either direction will need to be modified by the appropriate necessary conditions. Many of these extensions will be straightforward, and we hope to write about some of them soon.

APPENDIX

TABLE XI
FCM-AO, PCM-AO, FPCM-AO, AND PFCM-AO ALGORITHMS FOR INNER PRODUCT NORMS

Store	Unlabeled Object Data $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \subset \Re^p$
	$*\ 1 < c < n$ $*$ m > 1 : [for PCM-AO, FPCM-AO and PFCM -AO : η >1] $*$ a, b [for PFCM-AO] $*\ T_{max}$ = iteration limit
Pick	$*$ Norm for J_m : $\|\mathbf{x}\|_A = \sqrt{\mathbf{x}^T A \mathbf{x}}$ $*\ E_t = \|\mathbf{V}_t - \mathbf{V}_{t-1}\|_l / pc$ $*\ 0 < \varepsilon$ = termination criterion $*$ For PFCM -AO and PCM-AO : $(\gamma_1, ..., \gamma_c)$, $\gamma_i > 0$
	Guess $\mathbf{V}_0 = (\mathbf{v}_{1,0}, \mathbf{v}_{2,0}, ..., \mathbf{v}_{c,0}) \in \Re^{cp}$

Do	While ($T \le T_{max}$ and $E_t > \varepsilon$)
	Calculate U_t with $\mathbf{V}_{t-1}$ and (3a) or (10a) or (18a) or (21)
	[For FPCM-AO : Calculate T_t with $\mathbf{V}_{t-1}$ and (18b)]
	[For PFCM -AO : Calculate T_t with $\mathbf{V}_{t-1}$ and (22)]
	Calculate $\mathbf{V}_t$ with (3b) or (10b) or (18c) or (23)
	End While

	$(U, T, \mathbf{V}) \Leftarrow (U_t, T_t, \mathbf{V}_t)$

REFERENCES

[1] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[2] R. Krishnapuram and J. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, Apr. 1993.

[3] M. Barni, V. Cappellini, and A. Mecocci, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 393–396, Jun. 1996.

[4] O. Nasraoui and R. Krishnapuram, "Crisp interpretations of fuzzy and possibilistic clustering algorithm," in *Proc. EUFIT*, Aachen, Germany, 1995, pp. 1312–1318.

[5] R. Krishnapuram and J. Keller, "The possibilistic c-Means algorithm: Insights and recommendations," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 385–393, Jun. 1996.

[6] J. C. Bezdek and S. K. Pal, *Fuzzy Models for Pattern Recognition*. New York: IEEE Press, 1992.

[7] E. Anderson, "The irises of the GASPE peninsula," in *Bull. Amer. Iris Soc.*, vol. 59, 1935, pp. 2–5.

[8] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1992.

[9] E. E. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. 1978 IEEE Conf. Decision and Control*, Piscataway, NJ, 1979, pp. 761–766.

[10] R. N. Davé and K. Bhaswan, "Adaptive fuzzy c-shells clustering and detection of ellipses," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 643–662, Sep. 1992.

[11] R. Krisnapuram, H. Frigui, and O. Nasroui, "Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation," *IEEE Trans. Fuzzy Syst.*, pt. 1, 2 , vol. 3, no. 1, pp. 29–43–44–60, Feb. 1995.

[12] N. R. Pal, K. Pal, and J. C. Bezdek, "A mixed c-means clustering model," in *IEEE Int. Conf. Fuzzy Systems*, Spain, 1997, pp. 11–21.
[13] H. Timm, C. Borgelt, C. Doring, and R. Kruse, "Fuzzy cluster analysis with cluster repulsion," presented at the Euro. Symp. Intelligent Technologies (EUNITE), Tenerife, Spain, 2001.
[14] H. Timm and R. Kruse, "A modification to improve possibilistic fuzzy cluster analysis," presented at the IEEE Int. Conf. Fuzzy Systems, FUZZ-IEEE' 2002, Honolulu, HI, 2002.
[15] H. Timm, C. Borgelt, C. Doring, and R. Kruse, "An extension to possibilistic fuzzy cluster analysis," *Fuzzy Sets Syst.*, vol. 2004, pp. 3–16.
[16] E. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Conf. Decision and Control*, San Diego, CA, 1979, pp. 761–766.

Nikhil R. Pal received the Master of Business Management degree from the University of Calcutta, Calcutta, IN, in 1982, and the M.Tech. and Ph.D. degrees, both in computer science, from the Indian Statistical Institute, Calcutta, in 1984 and 1991, respectively.

Currently, he is a Professor in the Electronics and Communication Sciences Unit of the Indian Statistical Institute, Calcutta. His research interest includes image processing, pattern recognition, fuzzy sets theory, neural networks, evolutionary computation, and bioinformatics. He coauthored a book titled *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing* (Norwell, MA: Kluwer, 1999), coedited two volumes titled *Advances in Pattern Recognition and Digital Techniques* (ICAPRDT'99, Narosa) and *Advances in Soft Computing* (AFSS 2002, New York: Springer-Verlag), and edited a book titled *Pattern Recognition in Soft Computing Paradigm* (Singapore: World Scientific, 2001).

Prof. Pal serves on the Editorial/Advisory Board of the *International Journal of Neural Systems, International Journal of Approximate Reasoning, International Journal of Hybrid Intelligent Systems, Neural Information Processing—Letters and Reviews, International Journal of Knowledge-Based Intelligent Engineering Systems, Iranian Journal of Fuzzy Systems*, *Fuzzy Sets and Systems*, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS B. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and a Steering Committee Member of the journal *Applied Soft Computing*, Elsevier Science. He was the President of the *Asia Pacific Neural Net Assembly*. He was the Program Chair of the *4th International Conference on Advances in Pattern Recognition and Digital Techniques*, Dec. 1999, Calcutta, India, and the General Chair of *2002 AFSS International Conference on Fuzzy Systems*, Calcutta, India, 2002. He is the General Chair of the *11th International Conference on Neural Information Processing*, ICONIP 2004, and a Co-Program Chair of the *2005 IEEE International Conference on Fuzzy Systems*, Reno, NV, 2005.

Kuhu Pal obtained the B.Sc. degree (with honors) in physics from the University of Burdwan, India, in 1984, and the M.Sc. and Ph.D. degrees in physics from Banaras Hindu University, India, in 1987 and 1993, respectively.

After that, she worked as a Research Associate first in the Physics Department of Banaras Hindu University, and then from September 1995, in the Machine Intelligence Unit of the Indian Statistical Institute, Calcutta. In September 1999, she joined the MCKV Institute of Engineering as a Lecturer and later that for a visiting position with the Computer Science Department, the University of West Florida, Pensecola, from January 2000 for a period of six months. Her research interest includes pattern recognition, fuzzy sets theory, fuzzy logic controllers, neural networks, and computational material science. She was a Researcher at the Institute of Engineering and Management, Calcutta, India.

James M. Keller received the Ph.D. in mathematics in 1978.

He has had faculty appointments in the Bioengineering/Advanced Automation Program, the Computer Engineering and Computer Science Department, and the Electrical and Computer Engineering Department at the University of Missouri-Columbia, where he currently holds the rank of Professor. He is also the R. L. Tatum Research Professor in the College of Engineering. His research interests include computer vision, pattern recognition, fuzzy set theory and fuzzy logic, fractal geometry, and neural networks. He has been funded by several industrial and government institutions, including the Electronics and Space Corporation, Union Electric, Geo-Centers, the National Science Foundation, NASA/JSC, the Air Force Office of Scientific Research, the Army Research Office, the Office of Naval Research, and the Army Night Vision and Electronic Sensors Directorate. He has coauthored over 225 technical publications

Prof. Keller is a National Lecturer for the Association for Computing Machinery (ACM), an IEEE Neural Networks Council Distinguished Lecturer, and is a Past President of the North American Fuzzy Information Processing Society (NAFIPS). He is the former Editor-in-Chief of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, an Associate Editor of the *International Journal of Approximate Reasoning*, and is on the Editorial Board of *Pattern Analysis and Applications, Fuzzy Sets and Systems, International Journal of Fuzzy Systems*, and the *Journal of Intelligent and Fuzzy Systems*. He is currently serving a three-year term as an Elected Member of the IEEE SMC Society Administrative Committee. He was the Conference Chair of the 1991 NAFIPS Workshop, Program Co-Chair of the 1996 NAFIPS Meeting, Program Co-Chair of the 1997 IEEE International Conference on Neural Networks, and the Program Chair of the 1998 IEEE International Conference on Fuzzy Systems. He was the General Chair for the 2003 IEEE International Conference on Fuzzy Systems.

James C. Bezdek received the Ph.D. degree from Cornell University, Ithaca, NY, in 1973.

His interests include woodworking, optimization, motorcycles, pattern recognition, gardening, fishing, image processing, computational neural networks, blues music, and computational medicine.

Dr. Bezdek is the Founding Editor of the *International Journal of Approximate Reasoning* and the IEEE TRANSACTIONS ON FUZZY SYSTEMS, a Fellow of the IFSA, and recipient of the IEEE 3rd Millennium and Fuzzy Systems Pioneer medals