Fuzzy Clustering with Partial Supervision

Witold Pedrycz, *Senior Member, IEEE,* and James Waletzky

***Abstract*—Presented here is a problem of fuzzy clustering with partial supervision, i.e., unsupervised learning completed in the presence of some labeled patterns. The classification information is incorporated additively as a part of an objective function utilized in the standard FUZZY ISODATA. The algorithms proposed in the paper embrace two specific learning scenarios of complete and incomplete class assignment of the labeled patterns. Numerical examples including both synthetic and real-world data arising in the realm of software engineering are also provided.**

***Index Terms*— Complete and incomplete class assignment, fuzzy C-means, fuzzy clustering, FUZZY ISODATA, objective function, partial supervision, software reusability.**

I. Introductory Remarks: Clustering with Objective Functions

Clustering [7] and, more specifically, fuzzy clustering (cf., [2], [8], and [10]) are aimed at organizing and revealing structures in data sets such as patterns, temporal waveforms, images, etc. Clustering is commonly viewed as an instance of unsupervised learning, viz. learning without a teacher. The grouping of the patterns is then accomplished through clustering by defining and quantifying similarities between the individual data points (patterns). The patterns that are similar to the highest extent are assigned to the same cluster. In comparison to two-valued clustering, fuzzy clustering provides an additional conceptual enhancement by allowing the pattern to be allocated to several clusters (classes) to various degrees (membership values). In this way, the patterns can be treated more realistically and the analysis is capable of identifying eventual "outliers," viz. the patterns being more difficult to assign to a single category. The idea of objective function-based clustering is to partition the patterns in such a way that this objective function becomes minimized. The objective function Q can be specified as a sum of distances between the patterns and the corresponding prototypes of the clusters. In general, it takes on the form

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{p} \|\mathbf{x}_k - \mathbf{v}_i\|^2 \tag{1}$$

where $\mathbf{x}_k,\ k = 1, 2, \ldots, N$ are the patterns in $\mathbf{R}^n, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c$ are prototypes (prototype vectors) of the clusters, $1 < p < \infty$, while $\mathbf{U} = [u_{ik}]$ is a partition matrix describing an allocation of the patterns to classes and satisfying two conditions

$$\text{(i)} \qquad \sum_{i=1}^{c} u_{ik} = 1 \quad \text{for } k = 1, 2, \ldots, N$$

$$\text{(ii)} \qquad 0 < \sum_{k=1}^{N} u_{ik} < N \quad \text{for } i = 1, 2, \ldots c.$$

The family of all partition matrices will be denoted by $\mathcal{U}$.

The minimization of the objective function Q is completed with respect to $\mathbf{U} \in \mathcal{U}$ and the prototypes of the clusters, namely

$$\min_{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_c} Q$$

subject to

$$\mathbf{U} \in \mathcal{U}.$$

The distance function in the objective function (1), $d_{ik} = \|\mathbf{x}_k - \mathbf{v}_i\|^2$, can be specified either as a straightforward Euclidean distance or, its generalization such as the Mahalanobis distance defined as

$$d_{ik} = \|\mathbf{x}_k - \mathbf{v}_i\|^T \mathbf{A} \|\mathbf{x}_k - \mathbf{v}_i\|$$

with $\mathbf{A}$ being a positive definite matrix in $\mathbf{R}^n \times \mathbf{R}^n$ (cf., [2] and [5]).

There are two key design issues to be raised when designing and applying clustering methods

1) the form of the objective function;
2) the number of the clusters specified in the problem.

Even though the clustering methods are viewed as completely unsupervised, this perception is not completely valid. Specifically, the objective function being selected in advance predefines the shape of the clusters one is interested in finding in the data set. In other words, the search for some general regularities in the collection of the patterns is focused on finding a certain class of geometrical shapes favored by the particular objective function (such as circles, lines, spheres, or hyperellipsoides). The selection of the distance function could still allow for high flexibility; for instance the Mahalanobis distance [7] (as contrasted with a standard Euclidean metric) guides the clustering process toward establishing hyperellipsoidal forms of the clusters. In fact, a vast number of methods have been developed in this vein (cf., [3], [5], [8], and [13]).

Manuscript received January 2, 1995; revised November 7, 1995 and July 20, 1996. This work was supported by the Natural Sciences and Engineering Research Council of Canada and MICRONET.

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Ont., Canada R3T 2N2 (e-mail: pedrycz@ee.umanitoba.ca).

J. Waletzky is with MPR Teltech Ltd., Burnaby, B.C., Canada V5A 4B5 (e-mail: waletzky@mprgat.mpr.ca).

Publisher Item Identifier S 1083-4419(97)05198-4.

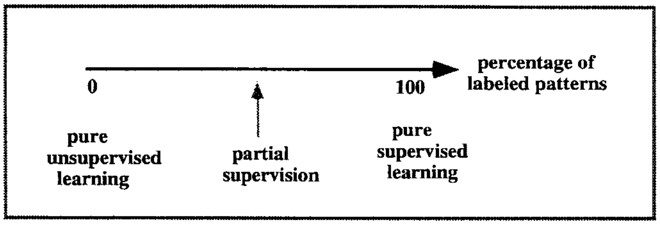


Fig. 1. Partial supervision versus supervised and unsupervised learning.

The number of clusters is usually treated as unknown or, at best, becomes restricted to a certain range of reasonable values implied by the classification problem itself. Various cluster validity indices such as cluster partition, partition entropy etc., (cf., [13] and [16]) are aimed at the determination of the most "plausible" number of the clusters.

While contrasting supervised versus unsupervised learning, it becomes evident that quite often the real-world applications call for many intermediate modes of the structural search in the data set whose efficiency could be substantially enhanced by a prudent use of the available domain knowledge about the classification problem at hand. We will be referring to these cases as clustering with partial supervision.

The phenomenon of partial supervision occurs when in addition to a vast number of unlabeled patterns one is also furnished with some (usually, few) labeled patterns. Definitely, these few already classified patterns, when carefully exploited, could provide some general guidance to the clustering mechanism. As indicated in [10] (in which the very idea was studied), the labeled patterns serve as "anchor" (reference) elements that shape the clusters. For instance, many problems of character recognition lend themselves to a concept of partial supervision. Assume, owing to classification costs, that only a fraction of the characters are labeled. Depending upon the number of these patterns, it is evident that we are faced with a continuity of intermediate schemes of partial supervision gravitating either to the algorithms of supervised or unsupervised learning, refer to Fig. 1.

In this study we will confine ourselves to the objective function (performance index) defined by (1) with the distance function specified as the Mahalanobis or Euclidean distance. The detailed algorithm of FUZZY ISODATA is readily accessible in the literature (cf., [2] and [3]) and will not be summarized here. While the very concept of the method we are about to study here has been introduced in [10], the thrust of this paper is to look carefully at some avenues leading to the optimization of the mechanisms of partial supervision. These will be investigated in presence of various geometric configurations of patterns (shapes of clusters). In what follows we elaborate on several extensions of the method by considering an unknown number of clusters and admitting various levels of classification credibility associated with the labeled patterns.

Surprisingly, limited attention has been paid to the mechanisms of partial supervision. The recent paper by Bensaid *et al.* [1] addresses these issues albeit very differently than approached in this study. In particular, the objective functions are distinct—thus the method developed here a generalization of FUZZY ISODATA whereas the algorithm presented in [1] is a *relative* of the generic clustering scheme.

TABLE I
FUZZY ISODATA WITH PARTIAL SUPERVISION

(1) Given is the number of clusters (c), **b** and **F**. Select the distance function ||.|| and initialize partition matrix $\mathbf{U} \in \mathfrak{U}$, including known membership values

(2) Calculate centers (prototypes) of the clusters and the fuzzy covariance matrices

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^2 \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^2}$$

$$\mathbf{P}_i = \frac{\sum_{k=1}^{N} u_{ik}^2 (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} u_{ik}^2}$$

i=1, 2,..., c with the distances defined below

$$d_{ik} = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{M}_i (\mathbf{x}_k - \mathbf{v}_i)$$

and

$$\mathbf{M}_i^{-1} = \left[\frac{1}{\rho_i \det(\mathbf{P}_i)}\right]^{\frac{1}{n}} \mathbf{P}_i$$

where typically ρ_i,=1, i=1,2, ..., c

(3) Update partition matrix:

$$u_{ij} = \frac{1}{1+\alpha}\left\{\frac{1+\alpha\left(1 - b_j \sum_{l=1}^{c} f_{lj}\right)}{\sum_{l=1}^{c}\left(\frac{d_{ij}}{d_{lj}}\right)^2} + \alpha f_{ij} b_j\right\}$$

(4) Compare **U**' to **U**, if || **U** - **U**'|| < δ (with δ being a tolerance limit) then stop, else go to (2) with **U** = **U**'

II. Clustering Under Partial Supervision

The key idea of the clustering algorithms with partial supervision is to take advantage of the available classification information and apply it actively as part of the optimization procedures. To distinguish between labeled and unlabeled patterns we will introduce a two-valued (Boolean) indicator vector $\mathbf{b} = [b_k]$, $k = 1, 2, \ldots N$ with 0-1 entries

$$b_k = \begin{cases} 1, & \text{if pattern } \mathbf{x}_k \text{ is labeled} \\ 0, & \text{otherwise.} \end{cases}$$

Similarly the membership values of the labeled patterns are arranged in a matrix form, say $\mathbf{F} = [f_{ik}]$, $i = 1, 2, \ldots, c$, $k = 1, 2, \ldots, N$. The component of supervised learning encapsulated in the form of $\mathbf{b}$ and $\mathbf{F}$ contributes additively to the modified objective function

$$J = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^p d_{ik}^2 + \alpha \sum_{i=1}^{c}\sum_{k=1}^{N} (u_{ik} - f_{ik} b_k)^p d_{ik}^2.$$

Here α $(\alpha \geq 0)$ denotes a scaling factor whose role is to maintain a balance between the supervised and unsupervised component within the optimization mechanism. To clarify, let us rewrite the above objective function in the form

$$J = Q + \alpha \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik} - f_{ik}b_k)^p d_{ik}^2. \tag{2}$$

The problem as formulated above with $p = 2$ has been studied in [10].

The choice of α depends very much on the relative sizes of the families of labeled and unlabeled patterns. Usually the cardinality of the latter is much higher than the population of labeled patterns. To ensure that the impact of the labeled patterns is not ignored, the value of α should produce approximately equal weighting of the two additive components of J. This suggests that α be proportional to the rate N/M where M denotes the number of labeled patterns.

Let us now minimize J. First we fix the values of the prototypes of the clusters. Noting that the columns of $\mathbf{U}$ are independent, the clustering task reads as the following constrained optimization problem,

$$\min J_k$$

subject to

$$\sum_{i=1}^{c} u_{ik} = 1$$

$k = 1, 2, \ldots, N$ where

$$J_k = \sum_{i=1}^{c} u_{ik}^p d_{ik}^2 + \alpha \sum_{i=1}^{c}(u_{ik} - f_{ik}b_k)^p d_{ik}^2$$

and $\mathbf{u}_k$ stands for the kth column of $\mathbf{U}$.

Using the standard technique of Lagrange multipliers, the optimization problem becomes converted into the form of unconstrained minimization (here J_k is written down explicitly to underline the variables taken into account in the optimization)

$$J_k(\lambda, \mathbf{u}_k) = \sum_{i=1}^{c} u_{ik}^p d_{ik}^2 + \alpha \sum_{i=1}^{c}(u_{ik} - f_{ik}b_k)^p d_{ik}^2 - \lambda\left(\sum_{i=1}^{c} u_{ik} - 1\right) \tag{3}$$

with λ denoting the Lagrange multiplier. The pair $(\lambda, \mathbf{u}_k)$ forms a stationary point of the optimized functional if and only if $\frac{\partial J_k}{\partial \lambda} = 0$ and $\frac{\partial J_k}{\partial \mathbf{u}_k} = 0$. These two derivatives yield the following relationships:

$$\frac{\partial J_k}{\partial \lambda} = \sum_{i=1}^{c} u_{ik} - 1 = 0 \tag{4}$$

$$\frac{\partial J_k}{\partial u_{st}} = p u_{st}^{p-1} d_{st}^2 + \alpha p (u_{st} - f_{st}b_k)^{p-1} d_{st}^2 - \lambda = 0$$
$$s = 1, 2, \ldots, c, \; t = 1, 2, \ldots, N. \tag{5}$$

Now two avenues can be pursued:

i) by setting the fuzziness coefficient or fuzzifier (p) to 2, we can derive explicit formulae needed to update the partition matrix;

ii) for any value of "p" different from 2 the optimization conditions call for some extra computational effort as now u_{st} (and λ) are linked together in the form of a certain polynomial equation whose roots need to be determined numerically.

In what follows, we confine ourselves to $p = 2$. We start with solving (5) for u_{st}

$$2d_{st}^2[u_{st} + \alpha(u_{st} - f_{st}b_t)] = \lambda$$

and

$$[u_{st} + \alpha(u_{st} - f_{st}b_t)] = \frac{\lambda}{2d_{st}^2}$$
$$u_{st}(1+\alpha) = \frac{\lambda}{2d_{st}^2} + \alpha f_{st} b_t \tag{6}$$
$$u_{st} = \frac{1}{1+\alpha}\left\{\frac{\lambda}{2d_{st}^2} + \alpha f_{st} b_t\right\}.$$

The sum of the membership values, $\sum_{j=1}^{c} u_{jt} = 1$, implies

$$1 = \frac{1}{1+\alpha}\left\{\left(\frac{\lambda}{2}\right)\sum_{j=1}^{c}\frac{1}{2d_{jt}^2} + \alpha b_t \sum_{j=1}^{c} f_{jt}\right\}.$$

In the sequel,

$$\frac{\lambda}{2} = \frac{1 + \alpha\left(1 - b_t \sum_{j=1}^{c} f_{jt}\right)}{\sum_{j=1}^{c}\left(\frac{1}{d_{jt}^2}\right)}. \tag{7}$$

Since (6) contains a factor $\frac{\lambda}{2}$, the left hand side of (7) may be directly substituted into (6) producing a final expression for u_{st}

$$u_{st} = \frac{1}{1+\alpha}\left\{\frac{1 + \alpha\left(1 - b_t \sum_{j=1}^{c} f_{jt}\right)}{\sum_{j=1}^{c}\left(\frac{d_{st}^2}{d_{jt}^2}\right)} + \alpha f_{st} b_t\right\}.$$

The complete algorithm is summarized in Table I.

At this point, a few remarks are indispensable. As a quick check one may note that if $\mathbf{b} = \mathbf{0}$ (no supervision at all), the proposed extension of FUZZY ISODATA reduces to the standard algorithm with the objective function

$$J = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 d_{ik}^2$$
$$= (1+\alpha)\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 d_{ik}^2.$$

Obviously, the term $(1+\alpha)$ plays a role of the scaling factor that could be easily ignored as not having any impact on the

TABLE II
CLASSIFICATION STATISTICS FOR XOR DATA SET

Algorithm	$\|\|U' - U\|\|$	Number of Correctly Classified Patterns	Number of Incorrectly Classified Patterns	Percent of Correctly Classified Patterns
Original	175.5229	62	88	41.33%
Fuzzy Covariance	222.6592	40	110	26.67%
Original (w/ Partial Sup)	153.7032	75	75	50.00%
Fuzzy Cov. (w/ Partial Sup)	26.4022	140	10	93.33%

TABLE III
CLASSIFICATION STATISTICS FOR IRIS DATA SET

Algorithm	$\|\|U' - U\|\|$	Number of Correctly Classified Patterns	Number of Incorrectly Classified Patterns	Percent of Correctly Classified Patterns
Original	167.4087	67	83	44.67%
Fuzzy Covariance	45.1437	129	21	86.00%
Original (w/ Partial Sup)	18.1891	145	5	96.67%
Fuzzy Cov. (w/ Partial Sup)	12.2168	150	0	100.00%

clustering results. Analogously, by assigning $\mathbf{b} = \mathbf{0}$ in the expression for u_{st} this update formula for the partition matrix reduces to the well-known form

$$u_{st} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{st}}{d_{jt}}\right)^2}.$$

The proposed method may be augmented in several ways. Two options are described below.

i) The number of assumed clusters is not fixed in advance. While the labeled patterns are assigned by considering *a priori* given number of classes, in general, the patterns in the entire set may eventually be clustered in more classes. To accommodate this situation, we replace the previously used indicator vector $\mathbf{b}$ by an indicator matrix $\mathbf{B} = [b_{ik}]$, $i = 1, 2, \ldots, c$, $k = 1, 2, \ldots, N$. The Boolean entries of $\mathbf{B}$ are interpreted as

$$b_{ik} = \begin{cases} 1, \text{ if membership of } \mathbf{x}_k \text{ to the } i\text{th class is known} \\ 0, \text{ otherwise.} \end{cases}$$

As previously, the available membership values are organized in the matrix form $\mathbf{F}$. Due to the format of the available classification information, the successive columns of $\mathbf{F}$ may not sum up to 1, therefore leaving room for the formation of some additional clusters

$$\sum_{i=1}^{c} f_{ik} b_{ik} < 1.$$

The difference in the membership values, $1 - \sum_{i=1}^{c} f_{ik} b_{ik}$, can be then distributed among additional clusters.

ii) The second generalization is aimed at discriminating between different levels of confidence attached to the labeled patterns. The confidence factor, $\text{conf}_k \in [0, 1]$ describes a level of confidence we assign to the actual membership grades coming with the kth pattern. The confidence factors appear as a part of the objective function $(p = 2)$

$$J_k = \sum_{i=1}^{c} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{c} (u_{ik} - f_{ik} b_k)^2 \text{conf}_k d_{ik}^2.$$

The higher the confidence level, the more significant the contribution of the corresponding pattern to the objective function. We assume that if $b_k = 0$ then $\text{conf}_k = 0$. Functionally, this extension of the method is similar to the mechanism used in [1].

III. SIMULATION STUDIES

This section summarizes several numerical experiments involving synthetic and real data including the well known IRIS data set benchmark and an object oriented software library consisting of C++ container classes. The patterns were clustered using four variations of FUZZY ISODATA, that is, FUZZY ISODATA, FUZZY ISODATA with a covariance matrix, FUZZY ISODATA with partial supervision, and partially supervised FUZZY ISODATA with a covariance matrix. The performance of each of the algorithms was monitored by expressing a distance between the partition matrices, $\mathbf{U}$ and $\mathbf{U}'$, obtained in successive iterations

$$||\mathbf{U}' - \mathbf{U}|| = \sum_{i=1}^{c} \sum_{k=1}^{N} (u'_{ik} - u_{ik})^2.$$

A stabilization of this index is usually a suitable stopping criterion.

Experiment 1: The two-dimensional set of patterns shown in Fig. 2 has been generated manually.

It consists of four overlapping clusters—two of them are ellipsoidal while the remaining ones form a visible cross of points resembling the standard EXCLUSIVE-OR problem. Overall, the diversity of the forms of the clusters along with their distribution makes the problem quite challenging for unsupervised learning. While the convergence of all the algorithms is fairly similar, Fig. 3, (one can eventually underline that the method with a covariance matrix and partial supervision exhibited the highest rate of convergence among all the

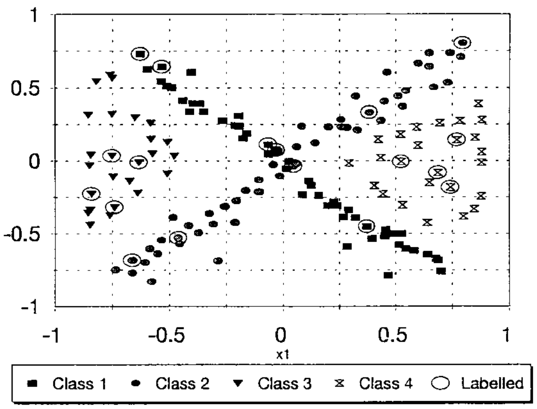


Fig. 2. XOR data set.

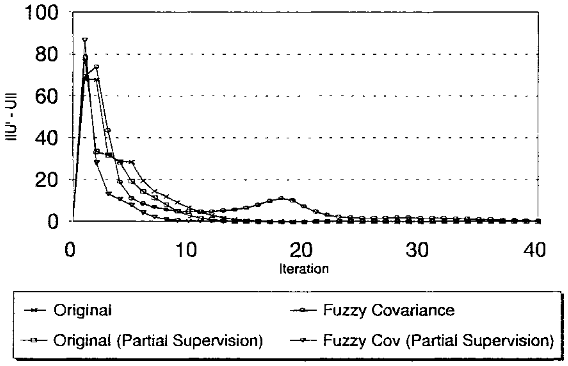


Fig. 3. $||U - U'||$ for XOR data set.

TABLE IV
SELECTED CLASSES OF C++ COMPILER

features	TArrayAsVector	TMDIAssociation	TMBagAsVectoriter	TBinarySearchTreimp	TMDIDictionaryAsHashTable	TIQueueAsVectorIter
Array	1.0	0.0	0.0	0.0	0.0	0.0
LIFO	0.0	0.0	0.0	0.0	0.0	0.0
FIFO	0.0	0.0	0.0	0.0	0.0	1.0
Bag	0.0	0.0	1.0	0.0	0.0	0.0
Tree	0.0	0.0	0.0	1.0	0.0	0.0
Dictionary	0.0	0.0	0.0	0.0	1.0	0.0
List	0.0	0.0	0.0	0.0	0.0	0.0
Vector	0.7	0.0	0.7	0.0	0.0	0.7
Hash Table	0.0	0.0	0.0	0.0	0.7	0.0
Double List	0.0	0.0	0.0	0.0	0.0	0.0
Key	0.0	1.0	0.0	0.0	0.0	0.0
Value	0.0	0.5	0.0	0.0	0.0	0.0
Sorted	0.0	0.0	0.0	0.0	0.0	0.0
Counted	0.0	0.0	0.0	0.0	0.0	0.0
Managed	0.0	1.0	1.0	0.0	1.0	0.0
Indirect	0.0	0.0	0.0	0.0	1.0	1.0
Iterator	0.0	0.0	0.0	0.0	0.0	1.0

variants of the clustering methods used in the experiment), the obtained classification results vary significantly between the methods. Fig. 4(a)–(d) shows the misclassified patterns and the classification results are illustrated numerically in Table II.

In the final assignment procedure the pattern is classified based upon the maximal membership value encountered in the partition matrix. It is noticeable that the standard algorithm leads to high number of misclassified cases meaning that the results of clustering deviate very far from those anticipated. The mechanism of partial supervision was implemented by randomly selecting a few patterns from each class and labeling them accordingly. These are also shown in Fig. 2. The prototype vectors generated by all four algorithms are given in Fig. 5.

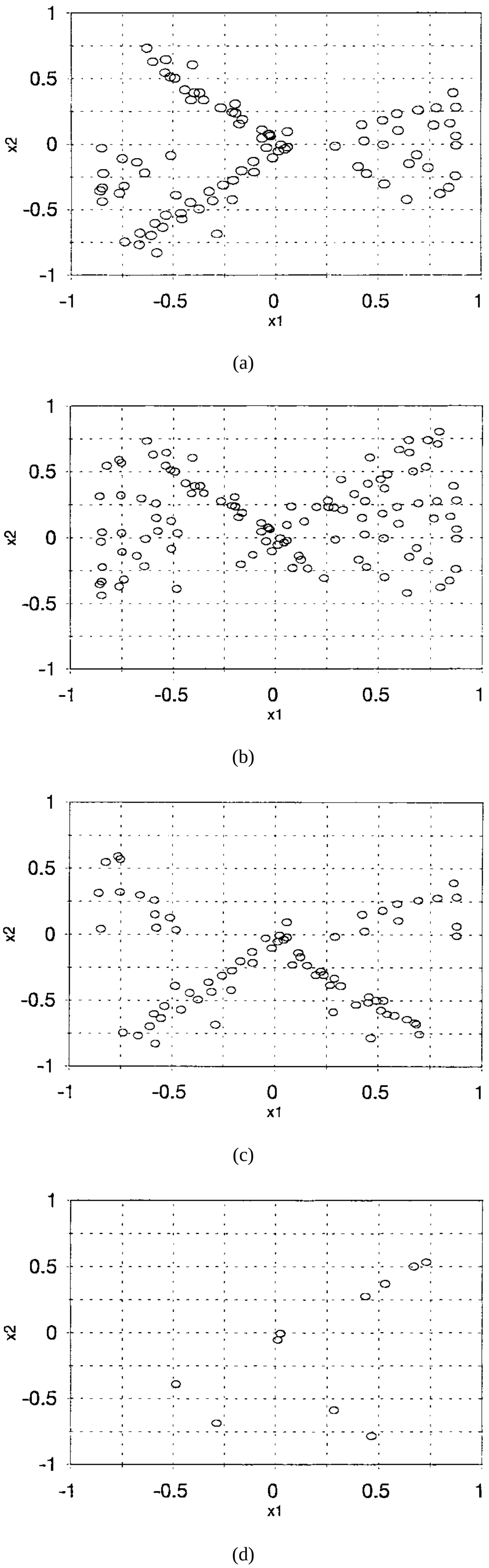
Notice that partially supervised FUZZY ISODATA, with covariance matrix, generated prototypes which were most representative of each class. Two of the prototypes are situated near the origin where the two crosses intersect, and the remaining two are located in the center of their respective ellipsoids.

The effect of partial supervision is profound significantly improving the outcomes of clustering. Obviously, to take full advantage, the mechanism of partial supervision should be implemented very prudently implying that the labeled patterns need to be somewhat representative of the entire data set. For instance, using all labeled patterns from the same class is not very instrumental to the improvement of the overall classification results. The intent of the series of experiments is to quantify this issue. In these runs a certain portion (5%, 10%, 15%, and 25%) of the patterns from each class was randomly selected and labeled. Subsequently these patterns constitute a guidance mechanism for further clustering. The labeled patterns constitute a certain percentage of all the patterns. To report visible and stable tendencies, the experiments were repeated 20 times for each percentage level (in fact, this led to significantly stable and meaningfully reproducible results). The results summarized in terms of classification errors and standard deviations of these are given in Fig. 6.

As expected, the classification error decreases with the increase of the percentage of the labeled patterns, l. The improvement becomes visible for some low values of l and the improvement is not as significant for its higher values. Interestingly, the standard deviation of the error is high for the low values of l—a visible indicator of how the quality of the labeled patterns could affect the quality of clustering.

Example 2: This experiment is concerned with the IRIS data set being commonly viewed as a standard benchmark for testing supervised and unsupervised classification schemes. Unsupervised learning algorithms have traditionally had a difficult time classifying the patterns owing to the considerable overlap between two classes. Again the same four clustering methods were used, see Fig. 7.

As before, the FUZZY ISODATA with a covariance matrix and partial supervision yielded the best results as far as labeling of patterns is concerned. Among the 21 patterns being completely misclassified by the FUZZY ISODATA with a covariance matrix, ten coming from all the classes were selected and labeled. The method equipped with this element of supervision produced a zero classification error as shown in Table III.



(a)

(b)

(c)

(d)

Fig. 4. Misclassified patterns for XOR data set: (a) original FUZZY ISODATA, (b) FUZZY ISODATA with covariance matrix, (c) FUZZY ISODATA with partial supervision, and (d) original FUZZY ISODATA with partial supervision and covariance matrix.

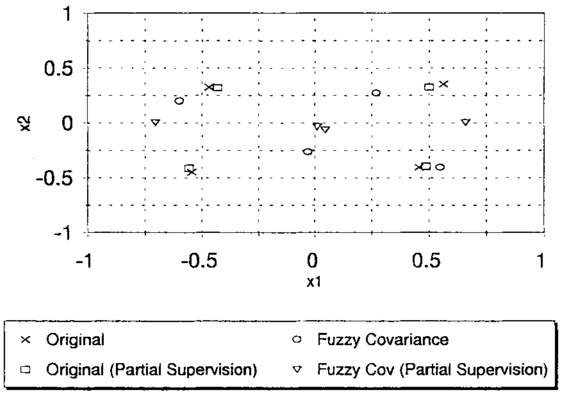


Fig. 5. Prototypes produced by each of the four algorithms for XOR data set.

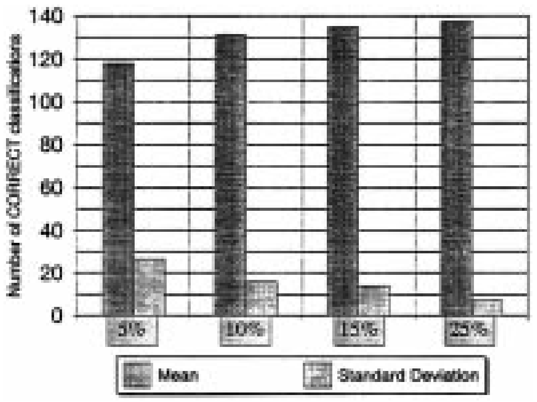


Fig. 6. Means an standard deviations of the number of correct classifications for XOR data set obtained for varying percentages of labeled patterns.

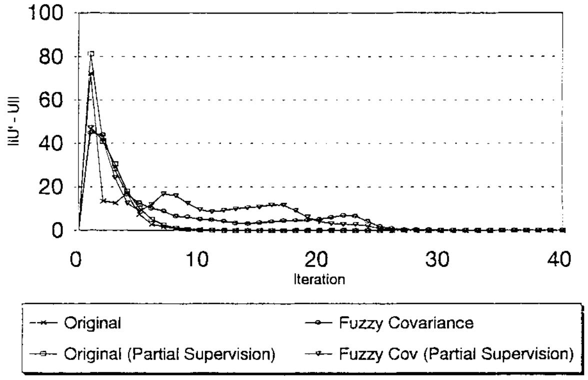


Fig. 7. $||U - U'||$ for IRIS data set.

Example 3: The idea of software reusability [12], [15] is predominantly associated with the arrangement of any reusable software modules into some coherent categories based upon their resemblance. This organization of the modules allows for their efficient retrieval and reuse. Once the specifications (required features) of a certain module are provided, we should be able to retrieve the most relevant (similar) pieces of software from the repository and re-use them rather than develop the required modules from scratch. The retrieval mechanism becomes secondary to the organization of the modules. The data set used in this experiment consists of

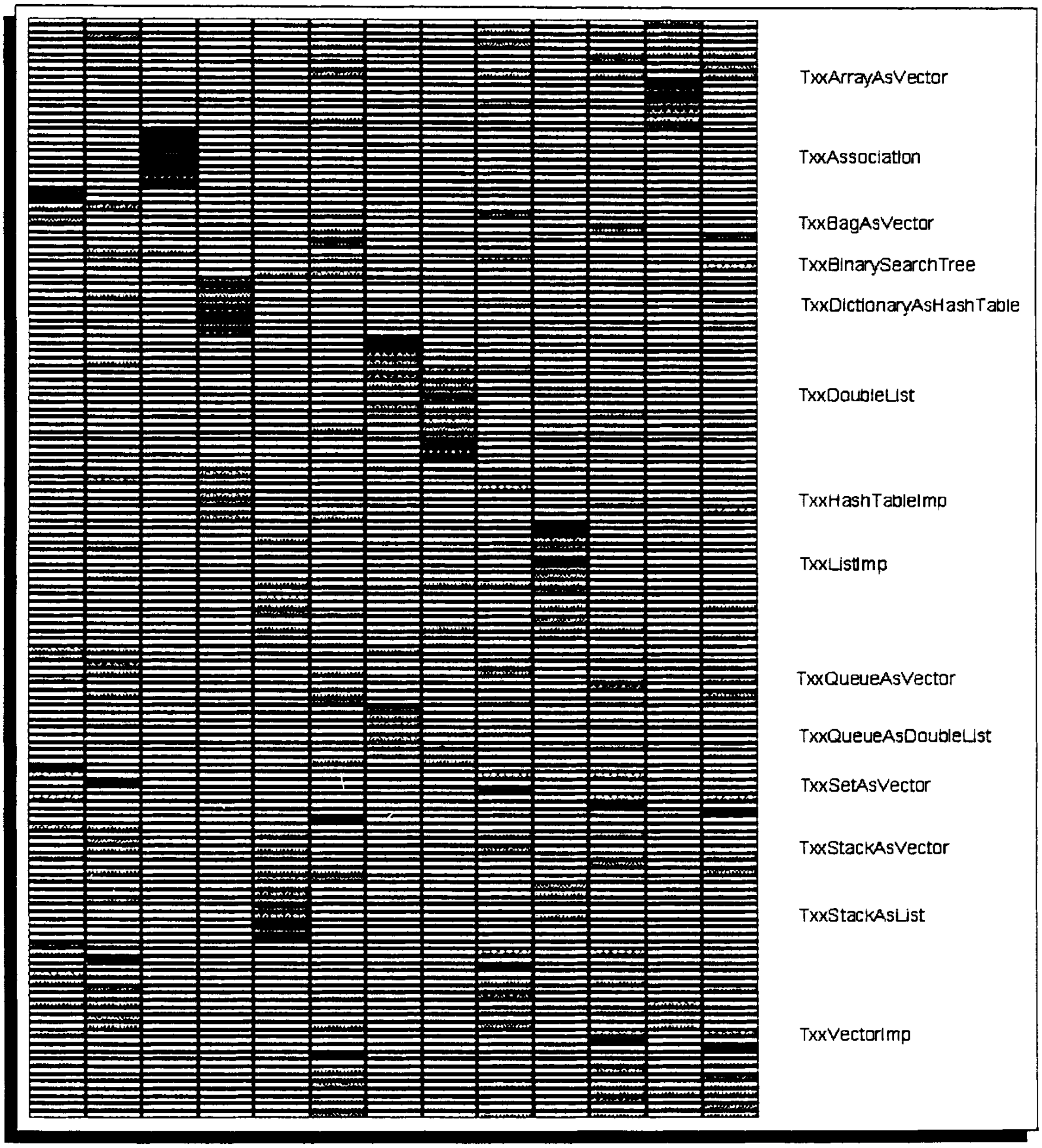


Fig. 8. FUZZY ISODATA with no supervision.

149 C++ classes provided within the Borland C++ v.4.0 compiler [4]. It is important to stress that the term class used in this context denotes a software module; this, in fact, corresponds to the notion of pattern used throughout the paper. Each class is described by 17 features which are essentially keywords occurring in the description of the classes as found in the reference manual of the software. A subset of selected classes are shown in Table IV. Each class is characterized by several of the keywords-the numbers standing in successive columns describe an extent to which the specific keyword describes the class; say Array (1.0) and Vector (0.7) for TArrayAsVector.

For clustering purposes we consider 13 classes as they are outlined in the manual [4, p. 355], namely, Array, Association, Bag, Binary Tree, Dequeue, Dictionary, Double-Linked List, Hash Table, List, Queue, Set, Stack, and Vector.

The clustering was first run without supervision. After 85 iterations the results were intuitively undesirable. In all trials, there was a significant overlap between the categories. This overlap created illogical groupings building up difficulties in distinguishing between categories. The resulting partition matrix given in Fig. 8 visualizes this effect.

As seen, very few classes exhibited membership values close to one with the exception of some classes coming from the category of the Stack, Association, and Dictionary. It should be stressed that the membership values around one are preferred.

Using partial supervision lead to good results. The component of partial supervision was formed by randomly picking up two classes (patterns) from each of the 13 categories (that, in total, makes 26 patterns). The overlap between the categories was evidently reduced, see Fig. 9.

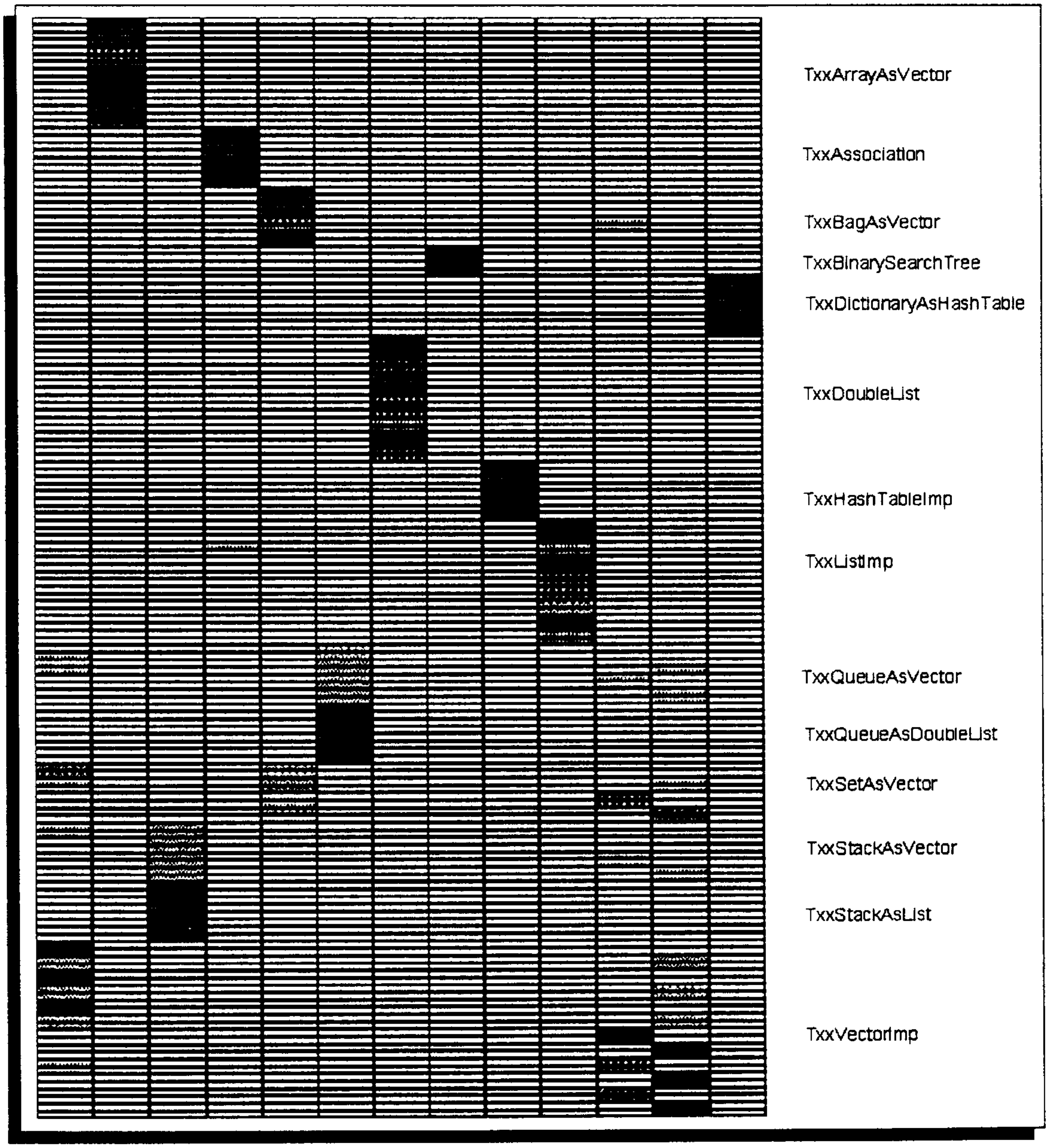


Fig. 9. FUZZY ISODATA with partial supervision.

Moreover the overlapping regions are reasonable. For instance, the class TQueueasDoubleList is shared with TDoubleList which is intuitively acceptable.

IV. Conclusion

The proposed clustering method is aimed at finding structure in presence of some labeled patterns. The way in which the mechanism of partial supervision has been incorporated as a part of the objective function allows us to cover all learning scenarios distributed between complete supervision (all labeled patterns) and its complete lack (no labeled patterns). Even a small percentage of the labeled patterns does substantially improve the results of clustering; obviously these patterns should be representative to the classification problem. This observation is fully supported by the completed simulation studies. Similarly, the convergence of the algorithm equipped with the supervision mechanism is, on average, slightly faster than that exhibited by its unsupervised counterparts. This observation requires some clarification. Note that the speed of the algorithm refers to the number of iterations the clustering algorithm needs to converge. The computational overhead per iteration is (or could be) higher for the introduced versions of FUZZY ISODATA than that encountered in its generic version. On the whole, this is not crucial; even though a single iteration of FUZZY ISODATA is faster, the algorithm may never converge or could require a significant number of iterations to reach this stage. Additionally, the produced results may not be acceptable.

REFERENCES

[1] A. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke, “Partially supervised clustering for image segmentation,” submitted for publication.

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

[3] J. C. Bezdek, R. J. Hathway, M. J. Sabin, and W. T. Tucker, “Convergence theory for fuzzy C-means: Counterexamples and repairs,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, pp. 837–877, Sept./Oct. 1987.

[4] *Borland C++ Version 4.0, Library Reference*, Borland Int., Inc., Scotts Valley, CA, 1993.

[5] I. Gatha and A. Oeva, “Unsupervised optimal fuzzy clustering,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 773–781, 1989.

[6] D. E. Gustafson and W. C. Kessel, “Fuzzy clustering with a fuzzy covariance matrix,” in *Proc. IEEE CDC*, 1978, pp. 761–766.

[7] R. J. Hathaway and J. C. Bezdek, “Recent convergence results for the fuzzy C-means clustering algorithms,” *J. Classification*, vol. 5, pp. 237–247, 1988.

[8] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[9] R. Krishnapuram and J. Keller, “A possibilistic approach to clustering,” *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 98–110, 1993.

[10] W. Pedrycz, “Algorithms of fuzzy clustering with partial supervision,” *Pattern Recognit. Lett.*, vol. 3, pp. 13–20, 1985.

[11] ——, “Fuzzy sets in pattern recognition: Methodology and methods,” *Pattern Recognit.*, vol. 23, pp. 121–146, 1990.

[12] R. Prieto-Diaz and P. Freeman, “Classifying software for reusability,” *IEEE Softw.*, vol. 1, pp. 6–16, 1987.

[13] M. Roubens, “Fuzzy clustering algorithms and their cluster validity,” *Eur. J. Oper. Res.*, vol. 10, pp. 294–301, 1982.

[14] R. Sarukkai, “Supervised learning without output labels,” *TR-510*, Univ. Rochester, Rochester, NY, May 1994.

[15] W. Tracz, *Software Reuse: Emerging Technology*. Los Alamitos, CA: IEEE Computer Soc. Press, 1990.

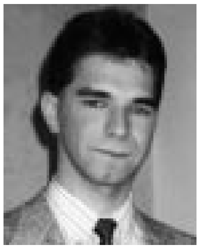
[16] M. Trivedi and J. C. Bezdek, “Low level segmentation of serial images with fuzzy clustering,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, pp. 580–598, 1986.

[17] M. P. Windham, “Cluster validity for fuzzy clustering algorithms,” *Fuzzy Sets Syst.*, vol. 3, pp. 177–183, 1981.

[18] X. L. Xie and G. A. Beni, “Validity measure for fuzzy clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 3, pp. 841–846, 1991.

Witold Pedrycz (M’88–SM’94) is Professor of Computer Engineering, Department of Electrical and Computer Engineering, University of Manitoba, Man., Ont., Canada. He is actively pursuing research in computational intelligence, fuzzy modeling, data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, and relational computation. He has published numerous papers in the area of applied fuzzy sets as well three research monographs (*Fuzzy Control and Fuzzy Systems*, 1988 and 1993; *Fuzzy Relation Equations and Their Applications to Knowledge Engineering*, 1988; and *Fuzzy Sets Engineering*, 1995). He is also one of the Editors-in-Chief of the *Handbook of Fuzzy Computation*.

Dr. Pedrycz is a member of many program committees of international conferences and serves on editorial boards of journals on fuzzy sets (including the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON NEURAL NETWORKS, and *Fuzzy Sets and Systems*) and pattern recognition (*Pattern Recognition Letters*).



James Waletzky received the B.Sc. and M.Sc. degrees in computer engineering from the University of Manitoba, Winnipeg, Canada, in 1993 and 1995, respectively.

He is currently with MPR Teltech. Ltd., Burnaby, B.C., Canada, designing software relating to telecommunications. He is also the founder of Cynthesis Software Inc., a company specializing in Internet-related PC software. His research interests include clustering algorithms, software reusability, and neural networks.