Outline of a New Feature Space Deformation Approach in Fuzzy Pattern Recognition

Petar B. Petrović
Associate Professor

University of Belgrade
Faculty of Mechanical Engineering

Pattern recognition ability is one of the most important features that characterize intelligent behavior of either biological or artificial systems. Mathematical pattern recognition is the way to solve this problem using transparent algorithms that are mostly based on conventional mathematics. In complex systems it shows inadequacy, primary due to the needs for extensive computation and insufficient robustness. Algorithms based on soft computing approach offer a good alternative, giving a room to design effective tools for real-time application, having in mind that relevance (significance) prevails precision in complex systems. In this article is modified and extended Subtractive Clustering Method, which is proven to be effective in real-time applications, when massive pattern sets is processed. The new understanding and new relations that connect parameters of the algorithm with the information underlying the pattern set are established, giving on this way the algorithm ability to be data driven to the maximum extent. Proposed algorithm is verified by a number of experiments and few of them are presented in this article.

Keywords: *Classification, Fuzzy clustering, Feature space deformation.*

1. INTRODUCTION

Recognition of patterns is one of the most important aspects of human perception. The inherent characteristic of human perception is its ability to recognize and classify patterns in a nondichotomous way. This process is fuzzy in its nature. The fuzziness is present in almost all levels in a pattern recognition process: the prototype description, the feature extraction and valorization, and recognition algorithm – human perception usually uses an opaque algorithm to recognize objects [7]. This is probably a natural (evolutionary) answer to the complexity, information ambiguity and information incompleteness, widely existing in the real world.

Pattern recognition has been extensively studied in various fields of engineering, including artificial intelligence as the most challenging engineering task today. In general, there are two basic approaches: mathematical pattern recognition (primary cluster analysis) and nonmathematical pattern recognition. The mathematical pattern recognition is strictly mathematically defined and it is far more context dependent than the lattr, which is primarily based on heuristic search. In this paper mathematical pattern recognition is considered only.

Mathematical pattern recognition task consists of two specific stages: the transduction stage and *classification* stage (Figure 1). These stages will be discussed in the text that follows.

Received: February 2003, revised: August 2003, accepted: October 2003.

Correspondence to: Petar B. Petrović,
Faculty of Mechanical Engineering,
27. marta 80, 11120 Belgrade, Serbia and Montenegro
E-mail: pbpetrovic@mas.bg.ac.yu

Let Ω be a set of physical objects (under the term object here are considered physical objects and processes). These objects may be characterized using the finite set of parameters, relevant to the classification task at hand. Each of these parameters, or couple of them, are specific to the particular *feature* of the object $q \in \Omega$. Since each parameter of the objects may be measured using some measurement procedure, the particular feature may be measured too, after applying the arbitrary complex measurement procedure m associated to that feature. In this way, object q may be associated to the mathematical object x=M(q)= (m_1(q), …, m_q(q))∈X, [7], where m_i(q) denotes the value of the feature i for the object q and X is the corresponding set of all associated mathematical objects. Such generated mathematical object x is called a *pattern*. Depending on the adopted set of features, many objects q may be associated to the single mathematical object x. Mapping from the physical space to the pattern space may be considered as multivalued multivalued (many to one) mapping in general. Moreover, this mapping doesn't include the information of object structure. It is simple collection of valued features that is formally organized as multidimensional

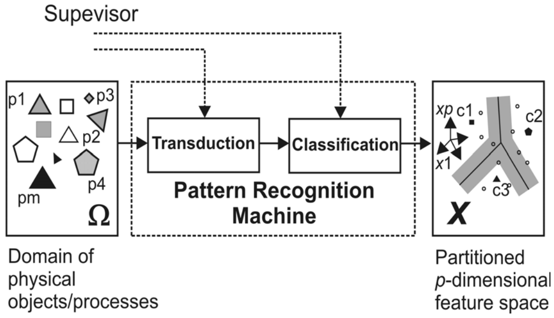


Figure 1. Structure of pattern recognition machine.

© Faculty of Mechanical Engineering, Belgrade. All rights reserved.

vector, which further may be represented as a point in multidimensional pattern space. As it is shown in Figure 2, an intelligent transducer that is capable to perform transduction of Ω to *X*, should be supported by feature extraction module and set of measurement procedures, relevant to the generated feature set for any particular pattern recognition task.

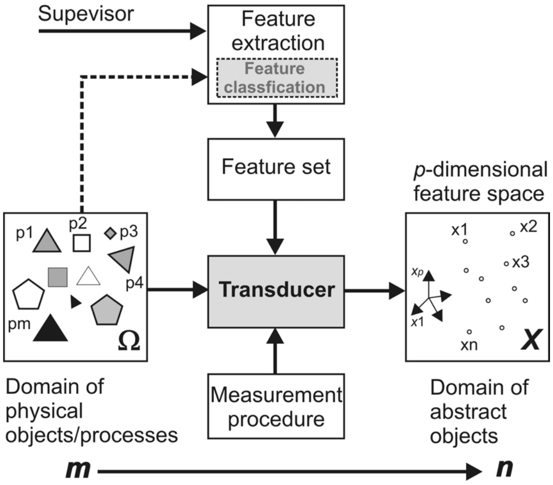


Figure 2. Structure of intelligent transducer machine that is able to perform the first stage of pattern recognition task – mapping from physical to abstract pattern space.

The second stage of pattern recognition process is classifying of pattern vectors. Classifying means that a given mathematical object *x* has to be assigned to a class of objects similar to it. This assignment may be considered either as hard (crisp) or fuzzy, depending on whether a pattern belongs exclusively to a single class or to all classes to different degree. Thus, in hard pattern recognition, a membership value of zero or one is assigned to each pattern, $\mu(x)$, whereas in fuzzy pattern recognition, a value between zero and one is assigned to each pattern by a membership function, $\mu_F(x)$. Accordingly, in fuzzy pattern recognition, a class of similar objects is a fuzzy set $\widetilde{F}$ ($\widetilde{F}$ is the label of the class). The grade of membership of mathematical object *x* in a class may be also considered as the degree of its similarity to a representative object of that class. This representative object is often named a *prototype*. In both, hard and fuzzy partition of pattern set, the following must be satisfied:

$$\forall x \in X,\ \sum_{i=1}^{c} \mu_{F_i}(x) = 1. \tag{1}$$

where *c* denotes the number of classes, i.e., the family of subsets of set *X*. This is so called orthogonality constraint that may be relaxed in cases where pattern set is noisy, allowing in this way that too noisy patterns may have low degree of membership to all of existing classes [7]. In order to formalize similarity measure, one can define positive real-valued function *d*, such that:

$$\forall x \in X,\ d(x,x) = 0\ , \tag{2}$$

$$\forall x, y \in X,\ d(x,y) = d(y,x). \tag{3}$$

and such that, similar (close) elements in *X* will have similar classification values, i.e., will be assigned to the same class, while dissimilar elements of *X* will have different classification values, i.e., will be assigned to different classes. In mathematical pattern recognition, explicit definition of the classification algorithm must be known and it is always applied to an abstract mathematical object $x = M(q) \in X$, not to the physical object $q \in \Omega$. This algorithm is said to be *transparent* and it is opposite to previously mentioned opaque algorithm, where recognition without definition is used. As it is stated in [12], opaque algorithm characterizes much intelligent behavior.

Despite unstructured pattern vector, in some cases knowledge of the physical object structure may be of great help in the recognition process [20], [10]. This leads to the *hierarchical partition* of pattern space by successive merging and splitting the complex patterns to the simpler sub-patterns, which are considered as primitives for the particular level where current recognition is performed. On each level locally optimal pattern recognition strategy should be used, without taking into account strategies used in past steps and strategies that has to be used in future steps. As it is stated in [20], hierarchical pattern recognition methods are not iterative in general. They are rather recursive [10] and cannot change assignments of objects to the classes made on preceding levels. In this article, only nonhierarchical pattern recognition will be considered, but this doesn't imply that the results presented may not be used in hierarchical clustering.

As it is shown in Figure 3, a pattern classifying machine that is capable to perform classification of abstract set *X* to *c* classes of similar elements, should be supported by appropriate similarity measure function, transparent classification algorithm and prototype set or appropriate estimator capable to identify the number of classes existing in a given pattern set *X*.

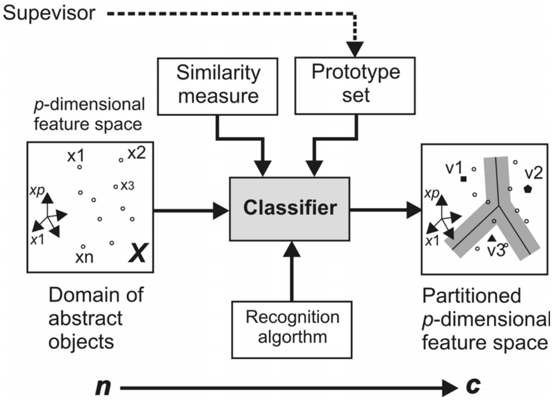


Figure 3. Structure of classifying machine that is able to perform the second stage of pattern recognition task – classification of abstract set *X* into the *c* classes of similar patterns.

The above mentioned stages in pattern recognition are not independent of each other, they are rather mutually interconnected. This fact is observed in [20], where is stated: "If we could chose "optimal" features, clustering and classification would be trivial; on the other hand, we often attempt to discover the optimal features by clustering feature variables!"

It is important to understand that the pattern recognition is hierarchically organized problem that includes classification of object features into two classes – one relevant for the task at hand (sufficiently representative and discriminative in the same time) and the other which is not, and after that, passing to the lower level, classification of physical objects based on previously classified set of relevant features. Classification itself is present on all pattern recognition levels and accordingly, it seams that meta-problem of everything in pattern recognition is classification.

In complex systems, the systems that are large size and/or where gathering precise information to describe their behavior is not possible, the classification becomes extremely difficult task. Mathematical pattern recognition, based on precise transparent algorithms is often mathematically intractable, making them intrinsically unsuitable for real-world application. This inadequacy can be expressed in what Zadeh called the *principle of incompatibility* (incompatibility between information precision and significance or relevance) [19]. Translating this principle into pattern recognition frame, one can state that as complexity of the system increases, our ability to formulate transparent and yet effective algorithm for recognition of patterns diminishes until precision and heuristic search become almost mutually exclusive. That's why soft computing prevails conventional mathematical approach in case of complex systems.

In this article the theoretical background of a new algorithm for semantic classifying patterns as well as some practical implementation details verified by selected computer simulation experiments is presented. The algorithm is unsupervised and independent from *apriori* defined parameters to the maximum extent (data driven classification) and it is suitable for real-time application with presence of massive pattern data. In section 2 basic iterative (subsection 2.1) and non iterative clustering algorithms (subsection 2.2) are formulated, where special attention is paid to objective functions and relation of their parameters to the pattern data that has to be clustered. In this section is given basic formulation of proposed non-iterative clustering algorithm (2.3), together with relations between the pattern set and algorithm related parameters. In section 3 results of performed computer simulation experiments, which show the effectiveness of proposed non-iterative clustering algorithm are given. In section 4 summary and conclusions are presented.

2. THEORETICAL FORMULATION

Let R^p be a p-dimensional real Euclidean feature space and $X=\{x_1, x_2,..., x_n\}\in R^p$ be a set of n pattern vectors, where $x_k=M(q)=(m_1(q),\ldots,m_p(q))=(x_k^1, x_k^2, \ldots, x_k^p)$, $1 \le j \le n$, and $m_i(q)$ denotes the value of the feature i for the physical object $q\in\Omega$.

Assume further, that the matrix:

$$\tilde{U}=[\mu_{ik}]\in R^{cxn},\ 1 \le k \le n,\ 1 \le i \le c, \qquad (4)$$

is the fuzzy partition matrix, where its member μ_{ik} is the membership degree of pattern vector x_k to the i-th cluster, and

$$v_i = (v_i^1,\ x_i^2,\ \ldots,\ x_i^p),\ 1 \le i \le c, \qquad (5)$$

is the center, i.e., the most representative pattern (*prototype*) of the cluster i. The set of all cluster centers is denoted by $V = \{v_1, v_2, ..., v_c\} \in R^p$, and in general is allowed to be $V \cap X \neq \emptyset$.

2.1. Iterative algorithms

According to Ruspini [17] who has introduced the notion fuzzy partition to represent the clusters in a pattern vector set, the problem of fuzzy clustering is to find a fuzzy partition matrix defined as (4), where number of fuzzy clusters, c, is a priori known, such that close patterns (in sense of (2) and (3)) will have similar classification vector, while dissimilar patterns will have different classification vector. The classification vector $\tilde{U}_k$ of a pattern x_k is k-th column vector of partition matrix $\tilde{U}$, i.e., $\tilde{U}_k = [\mu_{1k},\ldots,\mu_{ck}]^T \in R^{cx1}$. The way to solve above defined task is to select $\tilde{U}_k$ such that suitable defined functional will be minimized. As it is stated in [17], generally this functional has no solution, so it has to be relaxed into a minimization problem with suitable constructed objective function J, that is the function of two variables, $\tilde{U}$ and V.

Dunn [8] and Bezdek [1], [2], have proposed the following objective function based on *variance criterion*:

$$J_A(\tilde{U},V)=\sum_{i=1}^{c}\sum_{k=1}^{n}(\mu_{ik})^w d(x_k\text{-}v_i)^{\,2}, \qquad w>1 \qquad (6)$$

where dissimilarity measure between pattern vector and cluster center $d(x_k\text{-}v_i)$ is adopted to be a norm, which is defined as:

$$d(x_k\text{-}v_i)^2=\|x_k-v_i\|_M^2=(x_k\text{-}v_i)^T M(x_k\text{-}v_i) \qquad (7)$$

and satisfies conditions (2) and (3). Matrix $G\in R^{pxp}$ is called simple covariance matrix and it must be symmetric and positive-definite. In case that feature space is isotropic, G should be identity matrix, that leads to Euclidean norm as a dissimilarity measure function (clusters can be seen as equally sized hypersfere). In case of anisotropic feature space one can select other types of covariance matrix G, for instance, diagonal norm or Mahalanobis norm, as it is discussed in [20] and [6]. Non-identity covariance matrix introduce into the feature space anisotropic dissimilarity measure which create hyperelipsoidal shape of clusters.

Differentiating the objective function (6) with respect to V (for fixed $\tilde{U}$) and to μ_{ik} (for fixed V), and keeping the restriction (1), on can define location of cluster centers:

$$\mathrm{v}_i=\frac{1}{\sum_{k=1}^{n}(\mu_{ik})^w}\sum_{k=1}^{n}(\mu_{ik})^w x_k,\ w>1,\ (i=1,\ldots,c) \qquad (8)$$

and membership degree for each pattern vector:

$$\mu_{ki} = \left\| x_k - v_i \right\|^{-2/(w-1)} \sum_{j=1}^{c} \left\| x_k - v_j \right\|^{2/(w-1)}, \qquad (9)$$
$$i = 1,\dots,c, \quad k = 1,\dots,n.$$

where w is the exponential weight factor that shapes fuzzy partition matrix (the larger w, the fuzzier partition matrix; no theoretically justified procedure exist for choosing w; usually is chosen $w = 2$).

The nonlinear optimization problem described by equations (8) and (9) cannot be solved analytically. There exist various iterative algorithms, which obtain a local minimum of objective function (6). The best known is *fuzzy c-means* algorithm as an extension of IZODATA algorithm [3] that consists of four steps:

```
Step I      Given an input pattern $\boldsymbol{x}$;
            Set all initial parameters ($w, c, M, \varepsilon, \Delta$)
            Set randomly initial partition matrix $\tilde{U}^{(l)}$,
            $l = 0$, keeping restriction (1)
    WHILE ($\Delta > \varepsilon$)
Step II     Calculate cluster centers set $V^{(l)}$ using (8)
Step III    Calculate the new partition matrix $\tilde{U}^{(l+1)}$ using (9)
Step IV     Calculate $\Delta = \left\| \tilde{U}^{(l+1)} - \tilde{U}^{(l)} \right\|_M$
    END WHILE
```

IZODATA Algorithm

Quality of the solution depends strongly on the choice of the number of fuzzy clusters, c, and initial partition matrix, $\tilde{U}^{(0)}$, both as a priori data.

Recently, as in [6], [13], [14], or [15], additional objective functions are proposed, modifying original idea (6) by introducing new elements that control clustering process and that are more sensitive to a given pattern set X. In all cases it is an iterative process that searches for the global minimum of objective function.

Starting from the concept of possibilistic clustering, the following objective function is proposed in [13], [14] and [6]:

$$J_B(\tilde{U},V) = \sum_{i=1}^{c}\sum_{k=1}^{n} (\mu_{ik})^w d(x_k - v_i)^2 +$$
$$+ \sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1 - \mu_{ik})^w, \quad w > 1 \qquad (11)$$

where η_i is a positive parameter. It is important to note that in possibilistic clustering partition matrix $\tilde{U}$ is, strictly speaking, not longer a “partition matrix”, since its partition vectors doesn’t satisfy the constraint (1). Instead, there is clamed that members of matrix $\tilde{U}$ can be interpreted as degree of typicality (not sharing). This gives rise to good performance in the presence of noise and outliers.

Another objective function that introduce entropy as a measure for fuzzyfication of pattern data set [15] is:

$$J_C(\tilde{U},V) = \sum_{i=1}^{c}\sum_{k=1}^{n} \mu_{ik} d(x_k - v_i)^2 + \varsigma^{-1} \sum_{i=1}^{c}\sum_{k=1}^{n} \mu_{ik} \log \alpha_i^{-1} \mu_{ik} \qquad (12)$$

where α_i and ζ are positive parameters. The first term in (12) refers to objective function that was used for ordinary crisp c-partition, while the entropy term introduces fuzziness (the concept of entropy of fuzzy set as a measure of fuzziness is discussed in [20]).

2.2. Non-iterative algorithms

Convergence speed and stability of iterative algorithms in the presence of high dimensionality of feature space and large size pattern set may appear as a critical problem in their practical application, especially in case of real-time application. Moreover, these algorithms become unusable in case when pattern set is not present in advance, but the patterns arrive sequentially in time.

Opposite to the iterative algorithms, non-iterative algorithms are one-pass algorithms that possess high speed and straightforward convergence as inherent property. Accordingly, they appear as a good candidate for real-time and real-world applications where complexity starts to rule, diminishing requirements for precision and emphasizing the search for approximate but yet significant solution. The principle of incompatibility leads to something what was termed in [9] as *admissible* algorithms instead of precise algorithms.

Following idea stated above, in [18] simple non-iterative algorithm, named the *Mountain Method* was proposed. This method was based on the relaxation of iterative fuzzy clustering methods by allowing that only previously specified points located in feature space may be selected as cluster center. The feature space is discredited by regularly spaced grid. These grid points are further used for calculation potential field. A total potential for each grid point is calculated on such a way that partial potential is calculated in respect to any particular pattern point, and then, total potential is obtained by summing of calculated partial potentials for all pattern points. Repeating this procedure for all grid points one can calculate the complete potential field in considered feature space. After that, the grid point having the highest accumulated potential is selected as the first cluster center. After the first cluster center is selected, the potential field of the feature space is reduced by applying the negative potential field centered in the grid point that is selected as a first cluster. As a consequence, the grid points close to the first cluster center will have greatly reduced potential, while the distant one will preserve their initially formed potential. The next cluster center is determined by selection of the grid point with highest remaining potential. Search for the other cluster centers is obtained by repeating the procedure until the remaining potential field drops below the previously defined level. Algorithm itself is simple and stable in term of convergence, but very sensitive for dimensionality of pattern space and selected grid density. For instance, if feature space has dimensionality dim=3 and if resolution per axis of R=0.1 is selected, then the feature space is covered by $(1/R)$^d =10^3 grid points. It shows that computational requirements grow exponentially with dimensionality and resolution, which is extremely unsuitable for real-time applications.

As a solution for observed problem, the *Subtractive Clustering Method* (SCM) is proposed in [5]. Instead the restriction that only grid points may be selected as cluster centers, the problem is relaxed to the maximum extent, allowing that only pattern points itself may be selected as a cluster center. In this way, SCM may be interpreted as a search for prototypes within the pattern data set X, i.e., $V \subseteq X$. This approach drastically reduces computing burden, especially in case of high dimensionality of feature space. Moreover, it seams quite logical that typical represent of the cluster is not imaginary pattern, but existing pattern $x \in X$ that reflects further to the existing physical object $q \in \Omega$.

As it is shown below, SCM has no a priori prescribed number of clusters that has to be identified, instead, heuristic criteria is established to accept/reject new cluster and halt the clustering process. This criteria is based on a priori defined set of parameters that governs clustering process in such a way that should be avoided generation of too small or too many clusters, as well as, too close clusters.

```
Step I      Given an input pattern $X$; normalize input pattern
            Set all initial parameters ($r_a$, $r_b$, $\varepsilon_{upper}$, $\varepsilon_{lower}$)
            For each $x_k$ calculate potential $P_k^{(1)} = P(x_k, x_j)$, $k, j = 1, \ldots, n$
            Select the first cluster center using
            $({}^*P^{(1)}, x_i) = \max(P_1^{(1)}, \ldots, P_n^{(1)}) \rightarrow v_1 = x_i$
Step II     Reduce potential for each $x_k$ using
            $P_j^{(2)} = P_j^{(1)} - {}^*P^{(1)} P(v_1, x_k)$, $k = 1, \ldots, n$
            Identify candidate for the second cluster center using
            $({}^*P^{(2)}, x_i) = \max(P_1^{(2)}, \ldots, P_n^{(2)}) \rightarrow v_2 = x_i$, set $l=2$
    WHILE ( ${}^*P^{(l)} > \varepsilon_{upper} \, {}^*P^{(1)}$ and ${}^*P^{(l)} < \varepsilon_{lower} \, {}^*P^{(1)}$ )
            Calculate the $d_{min}$ = shortest distance between
            current candidate cluster and all previously
            idetfied clusters
            IF $(d_{min}/r_a) + ({}^*P^{(l)} / {}^*P^{(1)}) \geq 1$
                accept candidate cluster and set $v_l = x_i$
            ELSE
                set ${}^*P^{(l)} = 0$ and identify new candidate for
                the cluster center using
                $({}^*P^{(l)}, x_i) = \max(P_1^{(l)}, \ldots, P_n^{(l)}) \rightarrow v_l = x_i$
            END IF
Step IV         Reduce potential for each $x_k$ using
                $P_j^{(l+1)} = P_j^{(l)} - {}^*P^{(l)} P(v_l, x_k)$, $k = 1, \ldots, n$
                Identify candidate for the next cluster center using
                $({}^*P^{(l+1)}, x_i) = \max(P_1^{(l)}, \ldots, P_n^{(l)})_i$
    END WHILE
```

SCM Algorithm

Although the SCM algorithm is simple and efficient there are some drawbacks that avoid its real potentials to be exploit to the maximum extent. First, cluster generation process is governed by a priori given set of parameters, and no relation with pattern data set is established at all. According to that, there are no inherent abilities of the clustering process to adjust itself to the specific properties of the pattern set that has to be clustered. Second, generation of partition matrix is not well related to the given pattern set in term as it was made in various iterative clustering algorithms (6), (11), (12). In order to overcome these drawbacks, SCM algorithm is modified and new non-iterative algorithm is generated, following the basic idea of original SCM algorithm.

2.3 . Further extension of SCM

Suppose the given pattern data set: $X = (x_1, ..., x_n)$ is normalized i.e., fitted into a unit p-dimensional hypercube that represents feature space. The presence of a pattern deforms that feature space. Measure of this deformation may be any real-valued positive function defined as:

$$P_k = P(d^2(x_k, x), \theta) \ , \tag{13}$$

where θ is the set of real valued parameters. The relation (13) defines potential field around the pattern x_k..

Taking into account whole pattern set, one can define a total deformation of the feature space as a function of the pattern set distribution:

$$\mathbf{\Phi} = \sum_{k=1}^{n} P\left[d^2(x_k, x), \theta\right], \quad k = 1, \ldots, n \ , \tag{14}$$

This is an additive function that superimposes partial fields generated by each pattern existing in the feature space.

In general, any radial basis function may be a candidate for potential function P (13). Here Gaussian exponential function is proposed:

$$P_G = \exp\left[-d^2(x_k - x)/\sigma^2\right], \quad (k = 1, \ldots, n) \ , \tag{15}$$

and generalized bell function:

$$P_B = \left\{1 + \left[d^2(x_k - x)/\sigma^2\right]^{\frac{1}{m-1}}\right\}^{-1}, \ (k = 1, \ldots, n); \ m \geq 1 \, . \tag{16}$$

Both are radially symmetric functions of the squared distance between the pattern x_i and x_k and valued in the interval $\Phi \in [0,1]$. As it is well known, Gaussian function has one parameter, σ, that shapes decay rate and wideness in the same time (see Fig. 4a). Generalized bell function has two parameters, σ and m. Parameter σ defines wideness, exactly; it determines the distance at which potential has a half of the maximum value. Opposite to Gaussian function, parameter σ controls wideness of the generalized bell function only. Decay rate is controlled by another parameter, m. When $m = 1$ decay rate has its maximum value and function takes a rectangular pulse shape (all points grater then σ will have zero value). For $m \approx 1.5$ generalized bell function is approximately the same as Gaussian function. For grater values of m the top area become more concentrated while decay rate becomes lower, that results in the widening of basis area (see Fig. 4b). In both cases parameter σ is related to the resolution parameter of the potential functions (16) and (17), but its real nature may not be considered as the same. In general, the generalized bell function is more controllable and appears better suited for the problem at hand.

Intensity of the potential field will follow the pattern distribution so the high potential intensity will occur in dense areas, while low intensity will be in the areas where density of patterns are low or there are no patterns at all. Following this property, the maximum intensity of potential field will coincide with center of the densest region in feature space. The highest deformation of the feature space is located where the most of the presented patterns are concentrate.

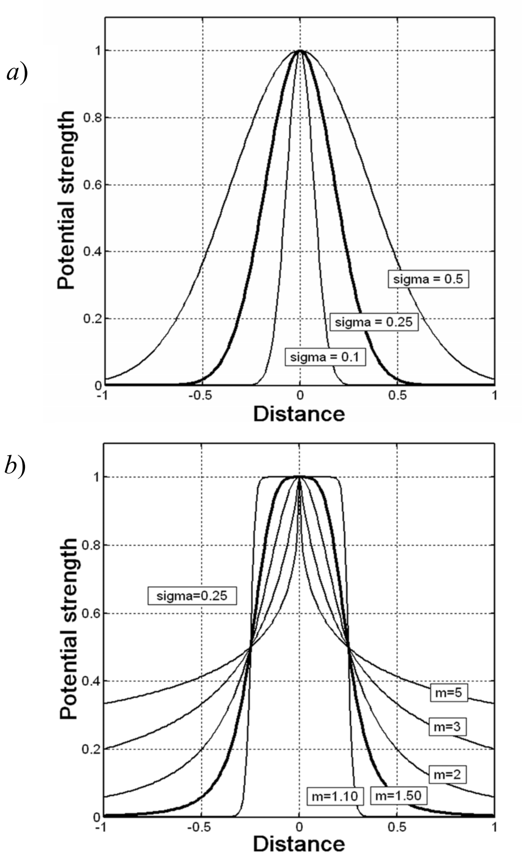


Figure 4. Gaussian (a) and generalized bell (b) potential functions for various values of shaping parameters.

Consequently, one can adopt the point with maximum intensity of the potential field as a center of the first cluster. Following the original idea of Mountain method, the second cluster may be found if the influence of the first cluster on the potential field is removed and after that another point, having maximum potential is selected within remaining potential field. For that purposes it is necessary to construct field suppressor function in the form:

$$\mathbf{\Phi}^R = \mathbf{\Phi} \circ S \qquad (17)$$

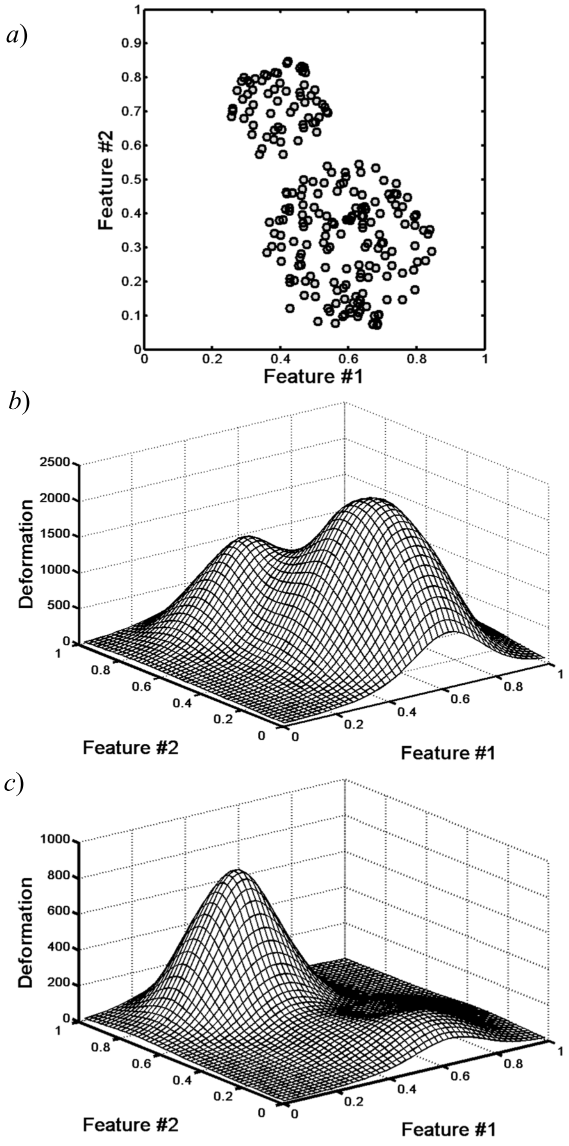
such that:

$$S = S\left[d^2(v_1,x)\right], \text{ and } \Phi^R(v_1) = 0 \ . \qquad (18)$$

where S is the suppressor function and v_I is the center of the first identified cluster. Again, a candidate for suppressor function may be radial basis function, P_G or P_B and in that case, following the relation (17) and restriction (18), suppressor may be formulated as:

$$S = 1 - P\left[d^2(v_1,x)\right], \qquad (19)$$

The procedure may be further repeated until the last cluster center is identified. Figure 5 shows an example of the whole process on the synthetically generated pattern set. It is easy to recognize how the pattern set deforms two-dimensional feature space, and how this deformation follows the cluster centers (they can be visually identified). It is quite logical that maximum deformation belongs to the most representative pattern.



Pattern vector set: 220 randomly generated pattern vectors concentrated within 2 well separated classes.
Cluster #1: center: (0.6046, 0.3804); pattern vectors: 155
Cluster #2: center: (0.3650, 0.7522); pattern vectors: 65

Figure 5. An example of two well separated pattern classes (top) and two iterations of proposed continual potential field clustering algorithm.

Clustering process is performed using generalized bell potential function, (σ=0.15, *m*=1.5) and generalized bell suppressor function, (σ=0.5, *m*=1.5). Generated initial potential field and corresponding feature space deformation Fig. 5*b*. Following the deformation of feature space the first cluster center is assigned to the maximum deformation point, i.e., v_1=(0.6046, 0.3804) (belongs to the larger cluster). Remaining potential field after first potential suppression is performed and corresponding deformation of feature space is shown in Fig. 5*c*. Now, applying the same procedure, it is possible to identify the second cluster center, i.e., v_2=(0.3650, 0.7522) (belongs to the smaller cluster). Since the pattern vectors are generated randomly, identified cluster centers well correlate with centers used in generation process of pattern vectors, i.e. (0.6, 0.3) and (0.4, 0.7).

In order make proposed algorithm suitable for real-time application, continuous model have to be turned into the discrete one, considering the existing pattern data only. For that purpose, a new matrix function is formulated:

$$\mathbf{\Phi} = \begin{bmatrix} P\left(d^2(x_1,x_1)\right) & \cdots & P\left(d^2(x_1,x_n)\right) \\ \vdots & \ddots & \vdots \\ P\left(d^2(x_n,x_1)\right) & \cdots & P\left(d^2(x_n,x_n)\right) \end{bmatrix}, \ \Phi \in R^{n\text{x}n}. \quad (20)$$

Since the distance function d is symmetric (3), and potential function P is real-valued (13), matrix Φ is symmetric and positive-definite. For convenience, this matrix may be expressed as a column vector containing the sum of each row:

$$\mathbf{\Phi} = \left[\sum_{k=1}^{n} P\left(d^2(x_1,x_k)\right) \cdots \sum_{k=1}^{n} P\left(d^2(x_n,x_k)\right) \right]^T, \ \Phi \in R^{n\text{x}1} \quad (21)$$

First cluster center v_1 is then the pattern vector that satisfies:

$$v_1 = x_k \rightarrow \sum_{i=1}^{n} P\left(d^2(v_1,x_i)\right) = \Phi_{v1} = \max \mathbf{\Phi} \ . \quad (22)$$

Continuous suppressor function (19) is also modified and transformed into vector form:

$$\mathbf{S} = \left[1 - P\left(d^2(v_i,x_1)\right) \ \cdots \ 1 - P\left(d^2(v_i,x_n)\right) \right],$$

$$i = 1,...,c, \ \mathrm{S} \in R^{1\text{x}n} \quad (23)$$

where c is a number of clusters that has to be identified or that has been identified. Then, remaining potential field will be calculated transforming general relation (17) into vector dot product:

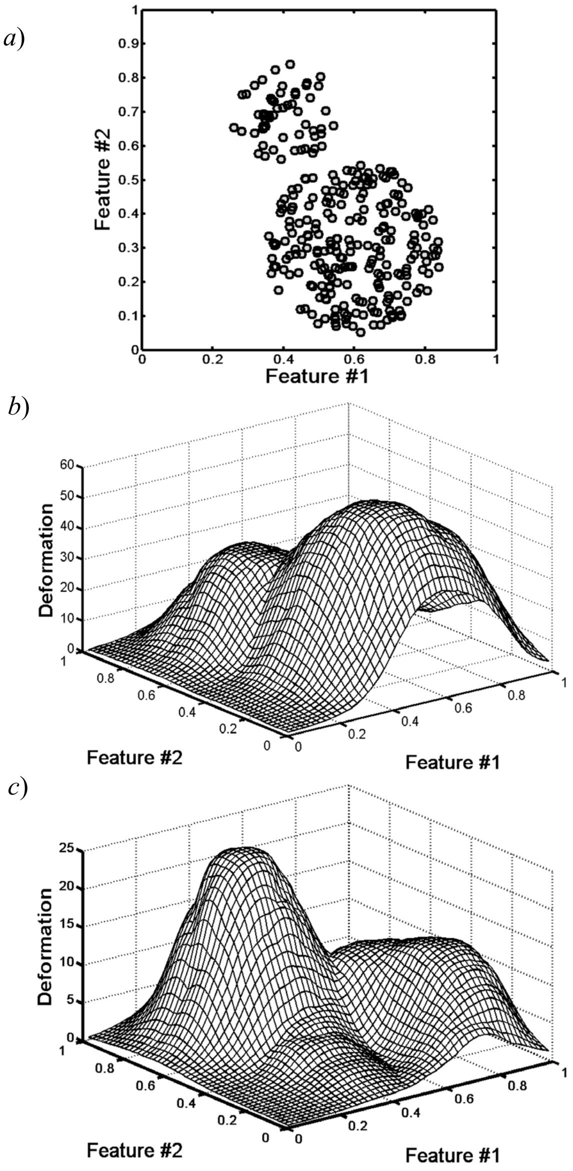
$$\mathbf{\Phi}^R = \mathbf{\Phi} \cdot \mathbf{S} \ . \quad (24)$$

Second cluster center v_2 is the pattern vector that satisfies:

$$\max \mathbf{\Phi}_k^R = \Phi_{v2} \rightarrow v_2 = x_k \ . \quad (25)$$

The procedure has to be repeated until cluster center v_c is identified or appropriate objective function is satisfied.

Discrete algorithm that is stated above is illustrated in Fig. 6 using the same example as that one which is used in case of continuous potential field calculation (actually there are slight differences that comes as consequence of random generator used in creation of pattern data). Clustering process is performed using the same parameters for potential and suppressor function as in previous case. Generated initial potential field and corresponding feature space deformation is shown in Fig. 6*b*. Maximum deformation of the feature space was occurred for the pattern x_k = (0.5469, 0.2424) and consequently, this pattern is promoted to be the first cluster center. Remaining potential field after first potential suppression is performed is shown in Fig. 6c. Applying the same procedure, the second cluster center, v_2 = (0.3827, 0.7103) is identified.



Pattern vector set: 288 randomly generated pattern vectors concentrated within 2 well separated classes:

Cluster #1: center: (0.5469, 0.2424); pattern vectors: 232
Cluster #2: center: (0.3827, 0.7103); pattern vectors: 56

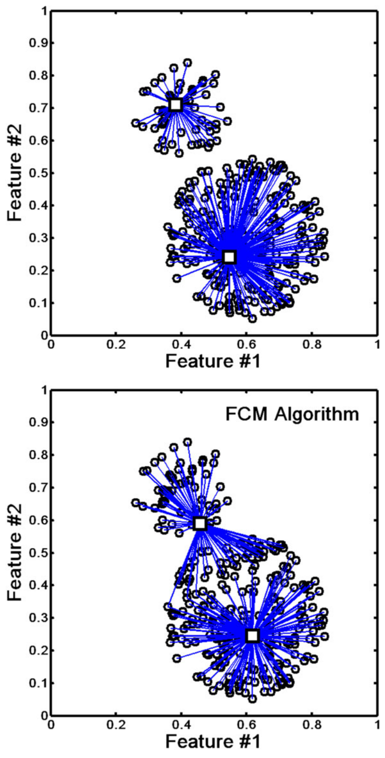
Figure 6. An example of two well separated pattern classes (top) and two iterations of proposed discrete potential field clustering algorithm (note that the pattern data slightly defers from those one used in example shown in Fig. 5).

Fuzzy partition matrix U defined by (4) will be generated using FCM approach (9), but with some modifications that follow properties of given pattern data set X. As it can be seen from examples given in Fig. 5 and Fig. 6, total potential of pattern vector x_k that is selected to be a prototype v_j for class j well coincide with the actual size of that class, i.e., it is clear that 'larger' cluster generates higher potential field (and larger deformation of feature space), while the ‘smaller’ cluster generates the lower intensity potential field (and smaller deformation of the feature space). This relation is introduced in (9) and on that way was obtained the new equation for calculation membership degree for each pattern vector:

$$\mu_{ki} = \left(\frac{\left\| x_k - v_i \right\|}{\Phi_{vi}} \right)^{-2/(w-1)} \sum_{j=1}^{c} \left(\frac{\left\| x_k - v_j \right\|}{\Phi_{vj}} \right)^{2/(w-1)} , \qquad (26)$$

$$i = 1, \ldots, c, \quad k = 1, \ldots, n.$$

where Φ_{vk} is the intensity of accumulated potential by prototype v_k defined by (22) and (25), and w is the fuzzifier as in original fuzzy c-means algorithm. Since equation (26) preserves constraint (1) matrix U will be partition matrix of the pattern set X. Effectives of proposed partitioning equation (26) is shown in Fig. 7, where is given partition of the pattern set used as example in Fig. 6. This example is typical case where fuzzy c-means algorithm shows its tendency that portion of the larger cluster is apt to be drawn into the small one, when large and small clusters are close. Shown are the cluster centers connected with lines to their respective member patterns. Note, that all patterns are correctly classified by proposed partitioning algorithm (Fig. 7). Partition of the same pattern set generated by fuzzy c-means algorithm is given in Fig. 7 below, where above-mentioned tendency is evident and lot of patterns is misclassified. Proposed partitioning algorithm is able to control the size of cluster in accordance to the real distribution of patterns.



Cluster #1 (larger cluster):
center: (0.5469, 0.2424)
center_FCM: (0.6170, 0.2450)
Cluster #2 (smaller cluster):
center: (0.3827, 0.7103)
center_FCM: (0.4580, 0.5900)

Figure 7. Partition of pattern set given in Fig. 6 - shown are the cluster centers connected with lines to their respective member patterns (cluster flower). Partition generated by proposed algorithm (above) and generated by fuzzy *c*-means algorithm (below).

The proposed pattern recognition algorithm is essentially one-pass algorithm, although within its structure there are some parts that are repeating itself. This repetition is governed by the number of pattern classes that are exist in a given pattern set, in this way it may be addressed to one of the major problems with any classification algorithm, namely, the need to know the number of classes. Instead of using a priori number of fuzzy clusters c, the partitioning may be optimized by introduction additional criteria that correlate with the structure underlying the pattern set. This problem is discussed in literature and several validity measures my be found in [7], [20], [10], [2], [3], [6], [5], [16] and [4], but no single measure was proven as general one. Common problem for all of them is monotonicity of validity function. As a general rule it should be emphasized that multi criteria approach is the one who possess sufficient robustness to cover various cases that may occur in practice.

Total variance of fuzzy partition of pattern set may be a good candidate for validity criteria, since it was used as an objective function in fuzzy c-means algorithm (6). Minimum total variance means that identified partition is optimal. Dividing total variance by size of pattern set, in [4] *compactness* of the fuzzy partition was introduced:

$$Comp = \frac{1}{|X|} \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^{w} d(x_k\text{-}v_i)^{\,2}, \qquad w > 1 \qquad (27)$$

while in [16] the same relation is interpreted as *within-fuzzy cluster-fluctuations* (the only difference is that the total sum of variances is not divided by the size of pattern set). This criterion is generally governed by idea that the similar patterns should be as close as possible, so the partition that has more compact clusters will be better then the other one having lower compactness. However, compactness alone has no sufficient potential to completely describe fuzzy partition of pattern set. In order to solve this problem, one more criterion has to be added - the criterion that is able to measure dissimilarity between clusters.

It is clear that the partition having compact but too close clusters (too close cluster centers) is not a good partition. Potential answer to this problem may be to introduce another criterion, this one that controls the distance between the clusters. This leads to the criterion named *separation* [4]:

$$Sep = \min_{i \neq k} \sum_{i=1}^{c} d(v_k\text{-}v_i)^{\,2} \qquad (28)$$

where the squared distance between clusters are suggested to be a feature that represents quality of the partition.

An alternative approach is presented in [16]. As appropriate feature that can control distance between clusters is introduced a sum of so called *between-fuzzy cluster-fluctuations*, i.e., the pattern fluctuation between clusters:

$$Fluct=\sum_{i=1}^{c}\sum_{k=1}^{n}(\mu_{ik})^{w}\left\|v_i-x_s\right\|^{2},\quad x_s=\frac{1}{|X|}\sum_{k=1}^{n}(x_k)\,. \qquad (29)$$

Above given criteria, i.e., those one that consider compactness and those one that consider separation of the clusters, are opposite in general, so their combination obviously leads to the composite criterion that do not possess monotonicity, although its constituents are monotone functions. This composite criterion may be local minimum of the function:

$$Val_S(U,V)=Comp/Sep\,, \qquad (30)$$

or:

$$Val_N(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{n}(\mu_{ik})^{w}\left\{\left\|x_k-v_i\right\|^2-\left\|v_i-x_s\right\|^2\right\} \qquad (31)$$

where in the both cases optimal partition is that one which has minimal *Val*(*U*, *V*). For the example shown in Fig 6. the following is obtained: Comp = [NaN 0.0527 0.0373], Sep = [NaN 0.5761 0.1662], that leads to: *Val*_1 = [NaN 0.0916 0.2242] and accordingly, optimal number of fuzzy clusters is: Opt_Clust_No = 2. The same is confirmed by the second criterion (31).

The proposed pattern recognition algorithm may be summarized by the algorithm that follows:

```
Step I     Given an input pattern set $X$; normalize input pattern
           Set all initial parameters ($\sigma$, $m$, $w$)
           Calculate
           $\mathbf{\Phi}=\left[\sum_{k=1}^{n}P\left(d^2(x_1,x_k)\right)\ \cdots\ \sum_{k=1}^{n}P\left(d^2(x_n,x_k)\right)\right]^T$
           Select the first cluster center using
           $v_1=x_k\rightarrow\sum_{i=1}^{n}P\left(d^2(v_1,x_i)\right)=\Phi_{v1}=\max\mathbf{\Phi}$
Step II Reduce potential field by $\mathbf{\Phi}_1^R=\mathbf{\Phi}\cdot\mathbf{S}_1$
           Identify the second cluster center using
           $\max\mathbf{\Phi}_k^R=\Phi_{v2}\rightarrow v_2=x_k$,
           Set validity criterion $Val_{(1)}=[\mathrm{NaN}]$, set $l = 2$
           Calculate partition matrix $U^{(l)}$ using Eq. (26)
           Calculate validity criterion $Val_{(l)} = Val(U, V)$
     WHILE ($Val_{(l)} < Val_{(l-1)}$)
Step III   Reduce potential field by $\mathbf{\Phi}_l^R=\mathbf{\Phi}_{l-1}^R\cdot\mathbf{S}_l$
           Identify the next cluster center using
           $\max\mathbf{\Phi}_{l1}^R=\Phi_{v(l+1)}\rightarrow v_{l+1}=x_k$
           Calculate validity criterion $Val_{(l)} = Val(U, V)$
     END WHILE
```

Poposed Feature Space Deformation (FSD) Algorithm

3. EXPERIMENTAL VERIFICATION

In this section presented are three examples that illustrate proposed pattern recognition algorithm. In all of the following examples, the parameter values used in the experiments are provided together with appropriate plots.

A. Example I: Butterfly

This is the well known example widely used as a benchmark test for various clustering algorithms. The pattern set consists of 15 points such as shown in Figure 8. The pattern vectors create in 2-dimensional feature space visually symmetric figure having two dense regions and one pattern that bridge those regions. The identified clusters should be symmetric and the pattern 'between' should have equal membership degree in both clusters. The same is identified by proposed clustering algorithm. Identified cluster centers as well as partition matrix well correlate with those one identified by FCM clustering algorithm.

Experimental data:

Pattern vector set:
15 pattern vectors
data_x =[0.2, 0.2, 0.2, 0.3, 0.3, 0.3, 0.4, 0.5, 0.6, 0.7, 0.7, 0.7, 0.8, 0.8, 0.8];
data_y =[0.3, 0.5, 0.7, 0.4, 0.5, 0.6, 0.5, 0.5, 0.5, 0.4, 0.5, 0.6, 0.3, 0.5, 0.7];

Potential function type and parameters:
potential field → Generalized bell (σ = 0.25, m = 1.5)
suppressor → Generalized bell (σ = 0.5, m = 1.5)

Val_S = [NaN 0.1860 0.2361] → *c_opt* = 2

Cluster #1:	center:	[0.3000	0.5000]
	center_FCM:	[0.2850	0.5000]
Cluster #2:	center:	[0.7000	0.5000]
	center_FCM:	[0.7150	0.5000]

B. Example II: Modified butterfly

Es an extension of the previous example here is considered modified butterfly pattern set where simmetricity is no longer preserved since one of the wings is scaled by a factor 0.5 (Figure 9). In this way two clusters close to each other and with the same number of patterns but with different density are generated in respect to the original butterfly set.

Experimental data:

Pattern vector set:
15 pattern vectors
data_x = [0.2, 0.2, 0.2, 0.3, 0.3, 0.3, 0.4, 0.5, 0.55, 0.6, 0.6, 0.6, 0.65, 0.65, 0.65];
data_y = [0.3, 0.5, 0.7, 0.4, 0.5, 0.6, 0.5, 0.5, 0.5, 0.45, 0.5, 0.55, 0.4, 0.5, 0.6];

Potential function type and parameters:
potential field → Generalized bell (σ = 0.25, m = 1.5)
suppressor → Generalized bell (σ = 0.5, m = 1.5)

Val_S = [NaN 0.1724 0.1771] → *c_opt* = 2

Cluster #1:	center:	[0.6000	0.5000]
	center_FCM:	[0.5940	0.5000]
Cluster #2:	center:	[0.3000	0.5000]
	center_FCM:	[0.2680	0.5000]

C. Example III: Pattern set from Fig. 6 contaminated with 20% noise patterns

In order to test the sensitivity of proposed recognition algorithm to noise, the pattern set that was used in example shown in Fig. 6 is contaminated with 20% noise patterns (Figure 10). Despite to such a high noise contamination the proposed algorithm behaves quite stable producing no misclassifications and exactly the same cluster centers are identifyed as in case of noncontaminated pattern set (compare the results shown in Fig. 6). At the opposite side FCM algorithm reacts clearly to the changes, producing different clustters then in original case. Although the same number of clusters is identified, cluster centers are not in the same location and the number of misclassified patterns is enlarged. Proposed algorithm is robust to outliers and shows clear potential to classify such patterns in the correct way.

Experimental data:

Pattern vector set: 358 randomly generated patterns vectors concentrated within 2 well separated classes:

Cluster #1: center: (0.6, 0.3); radius: 0.25; pattern vectors: 227
Cluster #2: center: (0.4, 0.7); radius: 0.15; pattern vectors: 73
+ 58 randomly generated noise pattern over the whole feature space

Potential function type and parameters:
potential field → Generalized bell (σ = 0.15, *m* = 1.5)
suppressor → Generalized bell (σ = 0.5, *m* = 1.5)

Val_S = [NaN 0.1134 0.2572]→ *c_opt* = 2

Cluster #1:	center:	[0.6616, 0.2687]
	center_FCM:	[0.6230, 0.2580]
Cluster #2:	center:	[0.3900, 0.7034]
	center_FCM:	[0.4170, 0.6380]

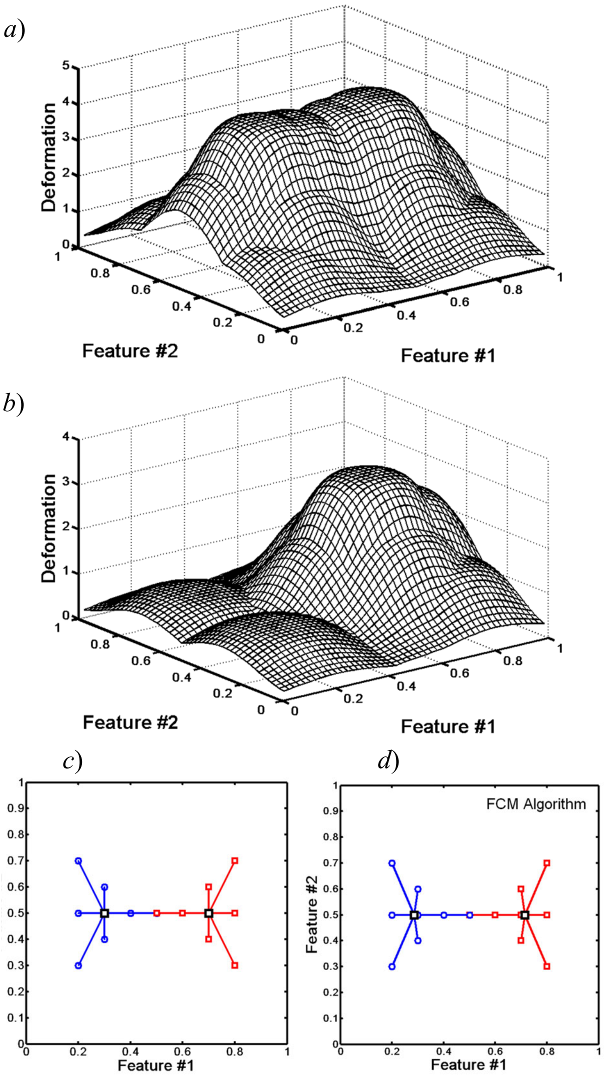


Figure 8. Partition of 'butterfly' pattern set; a) Initial deformation of the feature space; b) Suppressed deformation of the feature space after identification of the first cluster; c) Cluster centers connected with lines to their respective member patterns (cluster flower) for proposed and FCM algorithm (note that in both cases middle pattern has membership degree 0.5 and consequently belongs to both identified clusters).

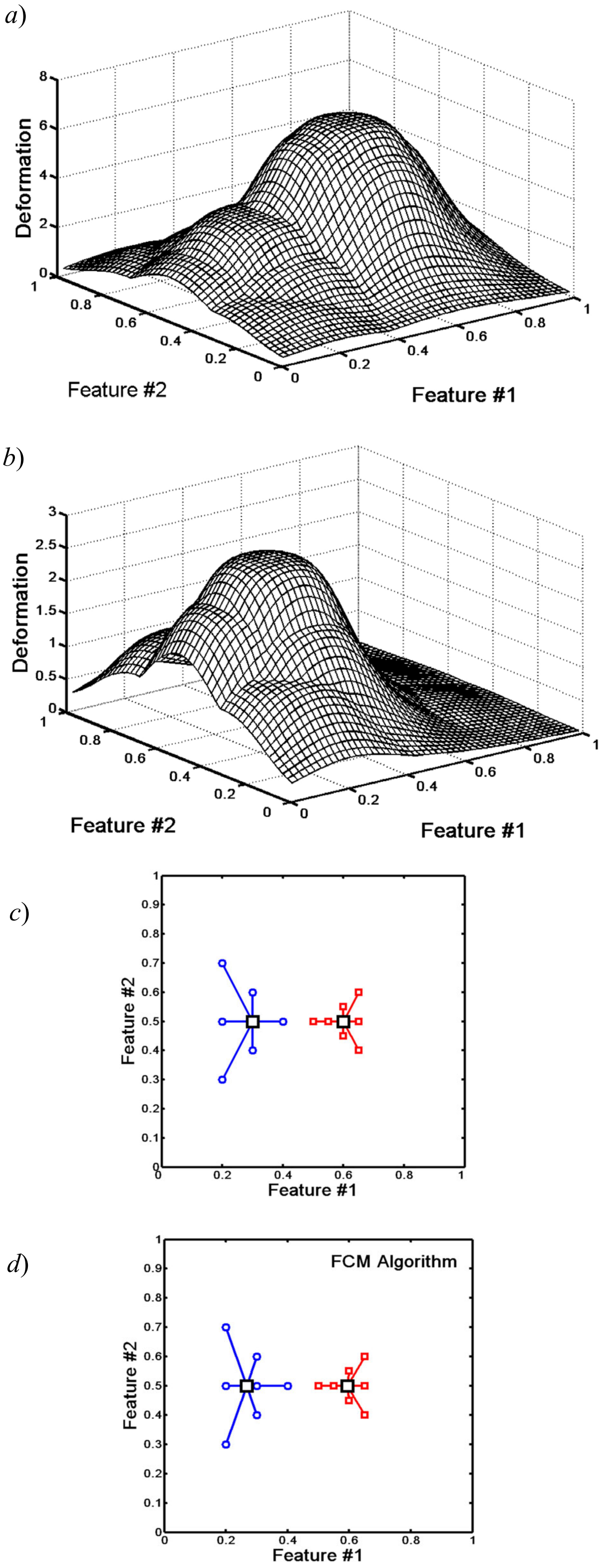


Figure 9. Partition of asymmetric 'butterfly' pattern set; a) Initial deformation of the feature space; b) Suppressed potential field after identification of the first cluster and corresponding deformation of the feature space; c) Cluster centers connected with lines to their respective member patterns (cluster flower) for proposed and FCM algorithm.

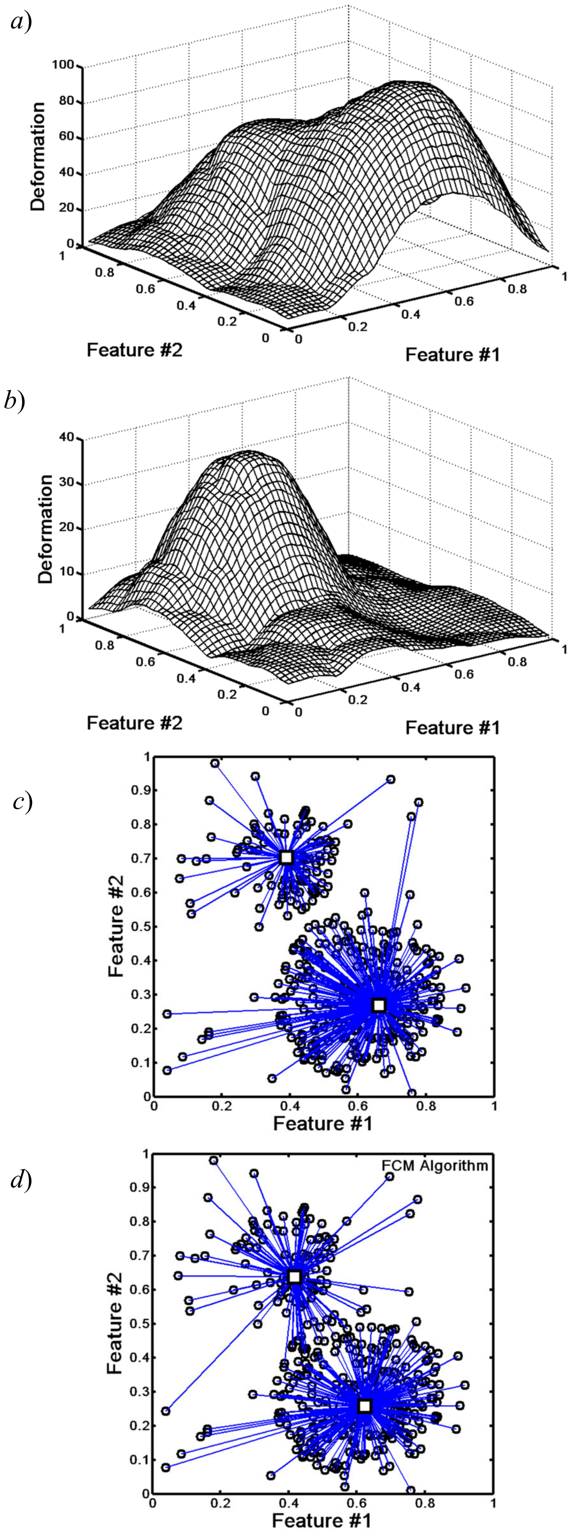


Figure 10. Partition of data set used in experiment shown in Fig. 6 contaminated with 20% of noise patterns; a) Initial potential field and corresponding deformation of the feature space; b) Suppressed potential field after identification of the first cluster and corresponding deformation of the feature space; c) Flower diagrams for proposed and FCM algorithm.

4. CONCLUSION AND DISCUSSION

In this article a general discussion about the problem of classifying physical objects using mathematical pattern recognition is presented. Two specific stages in mathematical pattern recognition are identified, i.e., the transduction stage and the classification stage, as well as that the above mentioned stages are not independent each other. Classification is also present in transduction stage since feature extraction, as a part of transduction stage, is also classification (classification of features in order to generate appropriate feature set, necessary for any classification of physical objects). Pattern recognition is hierarchically organized task, where classification lies in its background as a meta-problem.

Mathematical pattern recognition, based on precise transparent algorithms is often mathematically intractable, making them intrinsically unsuitable for real-world application. Translating the Zadeh’s principle of incompatibility into the pattern recognition frame leads to conclusion that with increasing of system complexity, our ability to formulate transparent and yet effective algorithm for recognition of patterns diminishes, until precision and heuristic search become almost mutually exclusive, giving in this way a room for soft computing to prevails conventional approach.

Further, several iterative and non-iterative approaches to fuzzy clustering are reviewed, giving a special attention to their suitability for real-time application. Furthermore, non-iterative algorithms were discussed, i.e., mountain algorithm and subtractive clustering algorithm that are based on using potential functions to discover prototype pattern vectors dispersed in a given feature space. It is observed that these algorithms has remarkable potential for real-time application but, there are some drawbacks which has to be removed:

- First, cluster generation process is governed by a priori given set of parameters, and no relation with pattern data set is established at all. According to that, there are no inherent abilities of the clustering process to adjusts itself to the specific properties of the pattern set that has to be clustered.
- Second, generation of partition matrix is not well related to the given pattern set in term as it was made in various iterative clustering algorithms.

In order to overcome these drawbacks, SCM algorithm is modified and new non-iterative algorithm is generated, following the basic idea of original SCM algorithm. This algorithm is based on idea that each pattern deforms feature space and further, that the most deformed areas will coincide to the most dense areas of pattern population, i.e., the highest deformation of the feature space will be related to the most representative patterns – prototypes. This idea is formulated by appropriate set of analytical relations. Fuzzy partition matrix is generated using modified FCM approach. Size of cluster is related to the accumulated potential of the prototype and this information is entered in FCM relation for calculation of membership degrees. This modification provides partitioning process with ability to follows properties of pattern data set. Furthermore, the problem of cluster validity, which is especially important for the proposed algorithm, because it is related with algorithm ending condition is analyzed. Instead of using a priori number of fuzzy clusters, additional criteria that correlate with the structure underlying the pattern set are introduced. In order to show potential of proposed algorithm, a number of experiments are provided, showing that algorithm possess better characteristics than FCM in term of robustness to outliers and tendency to combine or intersect close clusters different in size.

The future work will be addressed to establish connection of a priori defined parameters in potential and suppressor function, as well as to further refine partitioning equations to establish better correlation with pattern set at hand. Furthermore, the future work will be addressed to the modification of proposed algorithm in terms of their capability to classify patterns that are not present in advance, but arriving into classifying machine sequentially in a time.

REFERENCES

[1] Bezdek, J.C., “Numerical taxonomy with fuzzy sets”, *J. Math. Biol.* , Vol.1, pp:.57-71, (1974).

[2] Bezdek, J.C., “Cluster validity with fuzzy sets”, *J. Cybernetics,* Vol.3, No. 3 , pp.: 58-73,(1974).

[3] Bezdek, J.C., "*Pattern Recognition with Fuzzy Objective Function Algorithms*", Plenum Press, USA, (1981).

[4] Chen, M.-S. and Wang, S.-W., “Fuzzy Clustering for Parameter Identification of Fuzzy Membership Functions”, Proc. on Int. Conf. IIZUKA’96, Fukoka, Iizuka, JAPAN, pp.:287-390, (1996).

[5] Chiu, S., “Fuzzy model identification based on cluster estimation”, J. of Intelligent and Fuzzy Systems, Vol. 2 No. 3, pp: 267-278 (1994).

[6] Dave, R.N. and Krishnapuram, R., “Robust clustering, methods: a unified view”, *IEEE on Trans. Fuzzy Sys.*, Vol.5, No.2., pp.:270-293, (1997).

[7] Dubois. D. and Prade, H., *Fuzzy Sets and Systems – Theory and Applications*, Academic press, USA, (1980).

[8] Dunn, J.C., “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters”, *J. Cybernetics*, Vol.3, No. 3 , pp.: 32-57, (1974).

[9] Gaines, B.R., “Sequential fuzzy system identification”, Proc. of the IEEE Conf. on Decision Control, New Orleans, USA, pp.: 1309-1314, (1977).

[10] Geva, A.B., Hierarchical Unsupervised Fuzzy Clustering, *IEEE Trans. on Fuzzy Systems*, Vol. 7, No. 6, pp.: 723-733, (1999).

[11] Jain, A.K. and Dubes, R.C., *Algorthms for Clustering Data*, Engelwood Cliffts, (1988).

[12] Kosko, B., *Neural networks for signal processing*, Prentice Hall Inc., New Jersey, USA, (1991).

[13] Krishnapuram, R., and Keller, J.M., “A Possibilistic Approach to Clustering”, *IEEE Trans. on Fuzzy Sys.*, Vol.1, No.2., pp.:98-110, (1993).

[14] Krishnapuram, R., and Keller, J.M., “The Possibilistic *C*-means Algorithm: Insights and Recommendation”, *IEEE Trans. on Fuzzy Sys.*, Vol.4, No.3., pp.:385-393, (1996).

[15] Li, R.-P. and Mukaidono, A., “A Maximum Entropy Approach to Fuzzy Clustering”, *Proc. of the 4th IEEE Int. Conf. on Fuzzy Sys.*, Yokohama, Japan, pp.:2227-2232, (1995).

[16] Nakanishi, H., Turksen, I.B. and Sugeno, M., "A Review and Comparison of Six Reasoning Methods", *Fuzzy sets and systems* (57). pp: 257-294 (1993).

[17] Ruspini E.H., “A new approach to clustering”, *Inf. Control*, No. 15, pp.: 22-32, (1969).

[18] Yager, R.R., and Filev, D.P., “Approximate clustering via the mountain method”, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.24, No.8, pp.:1279-1284, (1994).

[19] Zadeh, L.A., “Outline of a new approach to the analysis of complex systems and pecision processes”, *IEEE Trans. on Syst., Man, and Cybernetics,* No 3, pp: 28-44 (1973).

[20] Zimmermann, H.-J., *Fuzzy Set Theory and its Applications*, Kluver Academic Publishers, USA, (1991).

КОНЦЕПТ НОВЕ МЕТОДЕ ФАЗИ ПРЕПОЗНАВАЊА ОБЛИКА ПРИМЕНОМ ДЕФОРМАЦИЈЕ ПРОСТОРА ОСОБЕНОСТИ

Петар Б. Петровић

Способност препознавања облика је једно од најзначајнијих својстава која карактеришу интелигентно понашање биолошких или вештачких система. Математичко препознавање облика представља формалну основу за решавање овог задатка применом прецизно форумулисаних алгоритама, који су у највећем делу базирни на конвенционалној математици. Код комплексних система овакав приступ показује значајне недостатке, првенствено због захтева за обимним израчунавањима и недовољне робусности. Алгоритми који су базирани на 'soft computing' методама представљају добру алтернативу, отварајући простор за развој ефикасних алгоритама за примену у реалном времену, полазећи од чињенице да значење садржаја информација носи већу вредност у односу на прецизност. У овом раду излаже се модификација и проширење 'Subrtactive Clustering' методе, која се показала ефикасном у обради масивних скупова облика у реалном времену. Нови приступ који је базиран првенствено на повезивању параметара алгоритма са информационим садржајем присутним у скупу облика који се обрађује, даје додатне степене слободе и омогућава да процес препознавања буде вођен подацима који се обрађују. Предложени алгоритам је верификован великим бројем симулационих експеримената, од којих су неки наведени у овом раду.