Computational Intelligence Techniques for Data Analysis

Yevgeniy Bodyanskiy

Control System Research Laboratory,
Kharkiv National University of Radio Electronics
14 Lenina Av., Kharkiv, 61166, Ukraine
bodya@kture.kharkov.ua

Abstract: The paper is a survey of the computational intelligence methods and their application to the data analysis problems. Neural networks, fuzzy sets, neuro-fuzzy systems, and genetic algorithms are considered. The advantages and disadvantages of the soft computing tools as well as specific issues of their application to data processing are analyzed, and the directions for their further improvement are outlined. New clustering algorithms that can operate under substantial uncertainty and cluster overlap are proposed.

1 Introduction

The amount of information already stored in the modern databases is huge and is measured in terabytes. Virtually infinite amount of information is available to anyone anywhere through the Internet. The information needs to be summarized and structured in order to support effective decision making.

When the amount of data, dimensionality, and complexity of the relations in it are beyond human capacities, there is a need for intelligent data analysis techniques, which could discover useful knowledge from data. Data analysis is a step in the process of knowledge discovery in databases (KDD) [KJ]. This step involves the application of specific algorithms for extracting patterns (models) from data. The additional steps are data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining [MPM02].

2 Data Analysis Problems

Data analysis is the process of extraction of previously unknown, non-trivial, practically useful, and interpretable knowledge, required for decision making, from ”raw” unstructured data in large arrays or databases.

In general, the problem that is solved in data analysis consists in detection of regularities in

data of different nature. More particularly, this involves regression, prediction, classification, clustering, rule generation, summarization, dimensionality reduction, visualization, etc.

Data analysis is complicated by the following factors:

- the amount of data is huge;
- the data is heterogeneous (quantitative, qualitative, textual, etc), often with gaps;
- the nature of the dependencies, "hidden" in the data, can change in time, i.e. the data can be non-stationary; abrupt changes are possible, that must be detected (fault detection);
- the data can be distorted by unknown disturbances (stochastic, chaotic, quasiperiodic, etc.)

All this puts forward the following requirements to the data analysis tools:

- the results must be concrete and understandable;
- the tools must be user-friendly and require minimal knowledge of mathematics and programming skills;
- if the amount of data is very large and growing in time, it is advisable to use sequential processing with variable memory (in order to allow forgetting of the "old" information).

Since the data are often imprecise, their distributions and the hidden relations are unknown and very complex, there is a need for the tools that can cope with the lack of information, complexity, and imprecision. Among such methods, the computational intelligence techniques proved to be very effective.

3 Computational Intelligence Techniques

The term Computational Intelligense encompasses a number of methodologies, mainly artificial neural networks (ANNs), fuzzy sets, genetic algorithms (GAs), and their hybridizations, such as neuro-fuzzy computing [JSM97], neo-fuzzy systems [MY99, YUTK92, BKK03], wavelet-neuro systems [ICP03, ZB92, BV03a, BV03b] e.a.

3.1 Neural Networks

ANN is a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure

of the cerebral cortex of the brain [Abe96]. There are tens of artificial neural network architectures. The most popular architectures in the data analysis application are multilayer perceptrons (MLPs) [Roj96], radial basis function networks (RBFNs) [MD89] and self-organizing maps (SOMs) [Koh01].

MLPs are used for regression, approximation and classification. They usually have no more than three layers (one output layer, and two hidden layers). The efficiency of the MLPs is explained by their universal approximation properties in conjunction with relatively compact representation of the modeled system. However, the training of the multilayer networks may be very slow.

RBFNs can also be used for classification and regression. They contain only one hidden layer. The training of RBF networks is usually faster than that of the multilayer networks, but the number of neurons required can be very large with high dimension of the input space (the curse of dimensionality). However, they can perform well with reasonable number of processing units, if they are properly constructed via usual clustering approaches, such as k-means method [MD89, Mac65].

With neural networks, knowledge acquisition is done by network training. Namely, by gathering input-output data for pattern classification or function approximation and training the network using these data by the learning algorithm, the desired function is realized. The most popular learning procedure for the MLPs (and for some RBFNs) is error backpropagation [RHW86], which consists in iterative finding of the set of tunable parameters (weights and biases) that provide the minimum of error function via gradient-based nonlinear optimization. The term stands for propagating backward the differences between the desired and actual network outputs through the hidden layers for the calculation of partial derivatives of the error function with respect to the network parameters.

SOMs are used for clustering, classification, and visualization of linearly separable high dimensional data. Their learning method is called competitive learning [Koh01].

Among the other ANNs, recurrent networks should be mentioned. They are quite close in architecture to the MLPs, but contain output-to-input connections (feedbacks) with delays. The recurrent networks are effective for time series prediction.

Associative memory ANNs can be used for processing of strongly distorted information. The concept of sequential processing is most effectively implemented in the so-called Brain-State-in-a-Box neural associative memory [AR88].

The processing of large data arrays with ANNs is complicated by the slow convergence of the conventional learning algorithms. The authors [BCKO02] have proposed improved learning algorithms with high rate of convergence and filtering properties.

The quality of data processing can be further improved with non-conventional neural architectures [BKK02, BGK03a], resource allocation [BGK03b], non-standard activation functions with tunable parameters [BCKO02].

3.2 Fuzzy Systems

Fuzzy systems are based on fuzzy set theory, proposed by L. Zadeh [Zad65]. In contrast to the classical set theory, in fuzzy set theory an object may belong to several sets at the same time with certain degrees of membership, expressed as real numbers in the interval $[0, 1]$. Fuzzy sets provide means to model fuzzy values of linguistic terms, and construct linguistic (fuzzy) rule bases.

A possible application of fuzzy systems in data analysis is the induction of fuzzy rules in order to interpret underlying data linguistically [KNB99]. More specifically, fuzzy systems can be used for classification and modeling of nonlinear dependencies, when the emphasis is placed on interpretability rather than on precision.

Fuzzy rules can be determined via fuzzy clustering. In the conventional (crisp) approach to clustering it is assumed that every observation belongs to only one class. The k-means algorithm [Mac65] and the nearest-neighbor rule [Cov68] are examples of this approach. It is much more natural to assume that every observation may belong to several clusters at the same time with certain degrees of membership. This assumption is the basis of fuzzy cluster analysis [Bez81, HKK96]. At present time many fuzzy clustering algorithms are known, e.g. Bezdek's fuzzy c-means [Bez81], the Gustafson-Kessel algorithm [GK79], mountain clustering by Yager and Filev [YF94], etc.

The source information for all the mentioned algorithms is the data set of N n-dimensional feature vectors $X = \{x_1, x_2, \ldots, x_N\}$, $x_k \in X$, $k = 1, 2, \ldots, N$. The output of the algorithms is the separation of the source data into m clusters with some degree of membership $w_{j,k}$ of the k-th feature vector to the j-th cluster. Here $w_{j,k} \in [0, 1]$ is the degree of membership of the vector x_k to the j-th cluster. The result of the clustering is assumed to be a $N \times m$ matrix $W = \{w_{k,j}\}$, referred to as fuzzy partition matrix.

When the elements of the matrix W can be regarded as the probabilities of the hypotheses of data vectors membership to certain clusters, the clustering are referred to as the probabilistic clustering algorithms [Bez81, GK79, GG89]. The most important disadvantage of the probabilistic approach is that they demand that the sum of memberships for each data vector be unity. To overcome this limitation, the so-called possibilistic methods of fuzzy clustering were proposed [KK93b].

If the dataset that needs to be processed is very large, processing it in batch mode may be very slow or impossible at all. The processing of large datasets can be accelerated by applying recursive fuzzy clustering algorithms [BKS02, BGKK02]. These algorithms are characterized by low computational complexity and high rate of convergence.

The fuzzy rules, obtained via the fuzzy clustering process, can be further improved by the application of the data-driven learning neuro-fuzzy techniques. The respective algorithms will be considered below.

3.3 Hybrid Neuro-Fuzzy Approaches

The strengths and weaknesses of ANNs and fuzzy systems are complementary. ANNs can learn complex mappings from presented input-output data. But since ANNs represent the "black box" ideology in system modeling, the decision obtained using the ANNs are difficult to interpret, and the incorporation of a priori information into a neural net is also difficult. At the same time, fuzzy systems can implement complex input-output mappings (in the form of rule bases) based on human experience and available a priori information. Because fuzzy systems operate on linguistic variables and rules, they are easily interpretable. But fuzzy systems do not possess learning capabilities.

Neuro-fuzzy systems [JSM97, BKS01, BGK03b] have been an increasingly popular technique of soft computing during recent years. They are based on modified fuzzy inference mechanisms, adapted for the use of learning procedures similar to those of ANNs. And from fuzzy systems they inherit the linguistic interpretability and ease of incorporation of a priori information.

The most widely used method of learning in such systems is the error back-propagation [RHW86], based on the gradient descent. Along with its simplicity and satisfactory performance in solving many problems, it has some essential drawbacks, such as slow convergence, sensitivity to noise, and dependence of performance on the heuristically selected learning rate.

We have proposed efficient learning algorithms with higher rate of convergence without significant increase of computational load [BKS01].

Hybrid neuro-fuzzy computing is not yet a finally well-formed approach, though it has already proved to be effective in solving many problems, such as nonlinear time series prediction and classification with strongly overlapping data clusters. A number of important issues, such as the selection of appropriate architecture and membership functions, improvement of the convergence rate of the known gradient-based learning algorithms, as well as the acceleration of the processing of very large data sets, still have to be investigated.

In our opinion, the most promising neuro-fuzzy approaches are those based on non-conventional neurons (neo fuzzy architectures [BKK03]), learning probabilistic neuro-fuzzy networks [BGK03b], fast learning algorithms [OBK03], wavelet-neural architectures [ICP03, ZB92, BV03a, BV03b].

In the works [BKO04, BK04a, BK04b] a neuro-fuzzy architecture based on the Kolmogorov's superposition theorem was proposed. This architecture provides better quality of approximation than the conventional artificial neural networks and fuzzy inference systems.

3.4 Genetic Algorithms

GAs are a subclass of evolutionary algorithms - optimization methods, inspired by the biological evolution. GAs were first proposed and analyzed by John Holland [Hol75]. There are three features which distinguish GAs, as first proposed by Holland, from other evolutionary algorithms: the representation used - bitstrings; the method of selection - proportional selection; and the primary method of producing variations - crossover. Of these three features, however, it is the emphasis placed on crossover which makes GAs distinctive [BFM97].

One of the most important advantages of GAs consists in the possibility of their application to the training of neural networks with non-analytical criteria having multiple local extrema. GAs are also effectively used for feature (input) selection, fuzzy rule base generation. In addition, the implementation of GAs is quite simple.

However, because of slow convergence of the classical GAs [Hol75], it is often advisable to consider some alternative approaches, such as "bee family", "islands model", and the other based on the ideas of elite selection.

Other alternatives are the random search with learning, complex-method, derivative-free multicriterion optimization [BR04].

In our opinion, it is the "mathematization" of GAs that can give them a new impulse for development.

4 Data Analysis with Adaptive Fuzzy Clustering Algorithms

Clustering and data classification are key problems of data analysis, and solving these problems is important for effective knowledge accumulation on the basis of observational analysis. In general, cluster analysis is the algorithmic basis of data classification by means of separation of the available data into a number of classes (clusters). In the traditional (crisp) approach it is assumed that every observation belongs to only one class.

It is much more natural to assume that every observation may belong to several clusters at the same time with certain degrees of membership. This assumption is the basis of fuzzy cluster analysis [Bez81, HKK96].

4.1 Batch Fuzzy Clustering Algorithms Based on Objective Function

The objective function based algorithms [Bez81] that have become widely used are designed to solve the clustering problem via the optimization of a certain predetermined clustering criterion, and are, in our opinion, the best-grounded from the mathematical point of view.

For pre-standardized feature vectors (the standardization is performed component-wise so

that all the feature vectors belong to the unit hypercube $[0,1]^n$), the objective function is

$$E(w_{k,j}, c_j) = \sum_{k=1}^{N} \sum_{j=1}^{m} w_{k,j}^{\beta} d^2(x_k, c_j) \tag{1}$$

subject to constraints

$$\sum_{j=1}^{m} w_{k,j} = 1,\ k = 1, \ldots, N, \tag{2}$$

$$0 < \sum_{j=1}^{m} w_{k,j} < N,\ j = 1, \ldots, m, \tag{3}$$

Here c_j is the prototype (center) of the j-th cluster, β is a non-negative parameter, referred to as "fuzzifier" (usually $\beta = 2$), $d^2(x_k, c_j)$ is the distance between x_k and c_j in the Euclidean metrics.

Note that since the elements of the fuzzy partition matrix W can be regarded as the probabilities of the hypotheses of data vectors membership to certain clusters, the procedures generated from (1) subject to constraints (2), (3) are referred to as the "probabilistic clustering algorithms".

Introducing a Lagrange function

$$\begin{aligned} L(w_{k,j}, c_j, \lambda_k) &= \sum_{k=1}^{N} \sum_{j=1}^{m} w_{k,j}^{\beta} d^2(x_k, c_j) + \sum_{k=1}^{N} \lambda_k \Big(\sum_{j=1}^{m} w_{k,j} - 1 \Big) = \\ &= \sum_{k=1}^{N} \Big(\sum_{j=1}^{m} w_{k,j}^{\beta} d^2(x_k, c_j) + \lambda_k \Big(\sum_{j=1}^{m} w_{k,j} - 1 \Big) \Big) \end{aligned} \tag{4}$$

(here λ_k is an undetermined Lagrange multiplier) and solving the following system of Kuhn-Tucker equations

$$\begin{cases} \dfrac{\partial L(w_{k,j}, c_j, \lambda_k)}{\partial w_{k,j}} = 0, \\ \nabla L(w_{k,j}, c_j, \lambda_k) = 0, \\ \dfrac{\partial L(w_{k,j}, c_j, \lambda_k)}{\partial \lambda_k} = 0, \end{cases} \tag{5}$$

the desired solution can be readily obtained as

$$w_{k,j} = \frac{(d^2(x_k, c_j))^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m} (d^2(x_k, c_j))^{\frac{1}{1-\beta}}}, \tag{6}$$

$$c_j = \frac{\sum_{k=1}^{N} w_{k,j}^{\beta} x_k}{\sum_{k=1}^{N} w_{k,j}^{\beta}}, \tag{7}$$

$$\lambda_k = -\Big(\sum_{l=1}^{m} (\beta d^2(x_k, c_l))^{\frac{1}{1-\beta}} \Big)^{1-\beta}. \tag{8}$$

The equations (6)-(8) generate a wide range of clustering procedures. Choosing $\beta = 2$ and adopting the Euclidean distance $d^2(x_k, c_j) = \|x_k - c_j\|^2$, we obtain the simple and efficient Bezdek's fuzzy c-means algorithm [Bez81]:

$$w_{k,j} = \frac{\|x_k - c_j\|^{-2}}{\sum_{l=1}^{m} \|x_k - c_j\|^{-2}}, \tag{9}$$

$$c_j = \frac{\sum_{k=1}^{N} w_{k,j}^{2} x_k}{\sum_{k=1}^{N} w_{k,j}^{2}}, \tag{10}$$

$$\lambda_k = -\sum_{l=1}^{m} \Big(\frac{\|x_k - c_l\|^{-2}}{2} \Big)^{-1}. \tag{11}$$

The Gustafson-Kessel [GK79], Gath-Geva [GG89], and a number of other procedures also belong to the probabilistic clustering methods. The most important disadvantage of the probabilistic approach is the constraints (2) [KKT97]. In the simplest case of two clusters ($m = 2$), it can be readily seen that an observation x_k, equally belonging to both clusters, and an observation x_p, not belonging to either, have equal degrees of membership $w_{k,1} = w_{k,2} = w_{p,1} = w_{p,2} = 0.5$. Naturally, this circumstance, significantly reducing the classification accuracy, led to the development of the possibilistic approaches to fuzzy clustering [KK93b, KK93a].

In the possibilistic algorithms, the criterion is

$$E(w_{k,j}, c_j) = \sum_{k=1}^{N} \sum_{j=1}^{m} w_{k,j}^{\beta} d^2(x_k, c_j) + \sum_{j=1}^{m} \mu_j \sum_{k=1}^{N} (1 - w_{k,j})^{\beta}, \tag{12}$$

where the scalar parameter $\mu_j > 0$ determines the distance at which the degree of membership equals 0.5, i.e. if $d^2(x_k, c_j) = \mu_j$, then $w_{k,j} = 0.5$.

Minimization of (12) with respect to $w_{k,j}$, c_j, and μ_j gives an obvious solution

$$w_{k,j} = \Big(1 + \Big(\frac{d^2(x_k, c_j)}{\mu_j} \Big)^{\frac{1}{\beta-1}} \Big)^{-1}, \tag{13}$$

$$c_j = \frac{\sum_{k=1}^{N} w_{k,j}^{\beta} x_k}{\sum_{k=1}^{N} w_{k,j}^{\beta}}, \tag{14}$$

$$\mu_j = \frac{\sum_{k=1}^{N} w_{k,j}^{\beta} d^2(x_k, c_j)}{\sum_{k=1}^{N} w_{k,j}^{\beta}}. \tag{15}$$

It can be readily seen that the possibilistic and probabilistic algorithms are very much alike, and transform into one another with the substitution of the equation (13) for the equation (6), and vice versa. The common drawback of the algorithm considered is their computational complexity, and the unsuitability for the online (real-time) operation.

The operation of the algorithm (6)-(8) begins with the setting of initial (usually random) partition matrix W^0. On the basis of its values, the initial set of prototypes c_j^0 is computed, which is then used to compute a new matrix W_1. Then in batch mode $c_j^1, W^2, \ldots, W^t, c_j^t, W^{t+1}$, etc are computed until the difference $\|W^{t+1} - W^t\|$ becomes less than some pre-specified threshold ε. In that way all the available data set is processed repeatedly.

The solution obtained with the help of the probabilistic algorithm is recommended as the starting point for the possibilistic procedure (13)-(15) [KK93a, KK97]. The distance parameters μ_j are initialized according to (15) from the results of a probabilistic algorithm, and then remain fixed [KKT97, KK97] during the clustering using the equations (13), (14).

4.2 Recursive Fuzzy Clustering Algorithms

In some practical problems, such as [XWI05] speech processing, text and web mining, medical and technical diagnosis, stock market forecasting, robot sensor analysis, etc. the data are coming sequentially, often in real time, that is why the application of recursive clustering procedures is advisable.

Analyzing the equation (6), it can be noticed that the following local modification of the Lagrange function (4) can be used for the re-computing of the degrees of membership $w_{k,j}$:

$$L_k(w_{k,j}, c_j, \lambda_k) = \sum_{j=1}^{m} w_{k,j}^{\beta} d^2(x_k, c_j) + \lambda_k \Big(\sum_{j=1}^{m} w_{k,j} - 1 \Big). \tag{16}$$

Optimization of (16) using the Arrow-Hurwitz-Uzawa procedure gives the following algorithm:

$$w_{k,j} = \frac{(d^2(x_k, c_j))^{\frac{1}{\beta-1}}}{\sum_{l=1}^{m} d^2(x_k, c_{k,l})^{\frac{1}{\beta-1}}}, \tag{17}$$

$$\begin{aligned} c_{k+1,j} &= c_{k,j} - \eta_k \nabla_{c_j} L_k(w_{k,j}, c_{k,j}, \lambda_k) = \\ &= c_{k,j} - \eta_k w_{k,j}^{\beta} d(x_{k+1}, c_{k,j}) \nabla_{c_j} d(x_{k+1}, c_{k,j}), \end{aligned} \tag{18}$$

where η_k is the learning rate parameter, $c_{k,j}$ is the prototype of the j-th cluster computed on the data set of k observations.

The procedure (17), (18) is quite close in structure to the fuzzy competitive learning algorithm of Chung-Lee [CL94], and when $\beta = 2$ the procedure coincides with the Park-Dagher gradient based fuzzy c-means clustering algorithm [PD84]:

$$w_{k,j} = \frac{\|x_k - c_{k,j}\|^{-2}}{\sum_{l=1}^{m} \|x_k - c_{k,l}\|^{-2}}, \tag{19}$$

$$c_{k+1,j} = c_{k,j} - \eta_k w_{k,j}^2 (x_{k+1} - c_{k,j}). \tag{20}$$

Within the possibilistic approach, it is possible to introduce a local criterion

$$E_k(w_{k,j}, c_j) = \sum_{j=1}^{m} w_{k,j}^{\beta} d^2(x_k, c_j) + \sum_{j=1}^{m} \mu_j (1 - w_{k,j})^{\beta}, \tag{21}$$

and to optimize it using the equations

$$w_{k,j} = \Big(1 + \Big(\frac{d^2(x_k, c_{k,j})}{\mu_j}\Big)^{\frac{1}{\beta-1}}\Big)^{-1}, \tag{22}$$

$$c_{k+1,j} = c_{k,j} - \eta_k w_{k,j}^{\beta} d(x_{k+1}, c_{k,j}) \nabla_{c_j} d(x_{k+1}, c_{k,j}), \tag{23}$$

where the distance parameters μ_j can be initialized according to (15). In this case, N in the equation (15) will be the length of the available data set, used for the initialization. In the quadratic case, the algorithm (22), (23) transforms into a quite simple procedure

$$w_{k,j} = \frac{\mu_j}{\mu_j + \|x_k - c_{k,j}\|^2}, \tag{24}$$

$$c_{k+1,j} = c_{k,j} - \eta_k w_{k,j}^2 (x_{k+1} - c_{k,j}), \tag{25}$$

where the distance parameters μ_j can be initialized from the results of probabilistic clustering (e.g. through the fuzzy c-means algorithm (9), (10)) according to the following equation:

$$\mu_j = \frac{\sum_{k=1}^{N} w_{k,j}^2 \|x_k - c_j\|^2}{\sum_{k=1}^{N} w_{k,j}^2}. \tag{26}$$

The considered recursive algorithms can be used in both the batch mode for the repeated processing of the same data set, and in the online mode with the number of observation k being the current discrete time $k = 1, 2, \ldots, N, N+1, \ldots$. In this case, the algorithm sequentially processes the incoming observations, adapting the degrees of membership and cluster prototypes to the newer data.

We have tested the performance of proposed algorithm (24), (25) in the problems of data classification, and compared the results with those obtained using the fuzzy c-means algorithm (9), (10), batch possibilistic clustering algorithm (13)-(15), and the Park-Dagher recursive clustering algorithm (19), (20). For testing purposes, three well-known data sets were used: the Wine data, the Iris data, and the Thyroid data from the UCI repository [UCI]. The Iris data set contains the descriptions of 150 instances of iris flowers, distributed equally into 3 species (setosa, versicolour, virginica). The flowers are described by 4 attributes (sepal length, sepal width, petal length, petal width). The Wine data set contains 178 results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the 3 types of wines (there are 59 instances of the first type, 71 of the second type, and 48 of the third type). So the first data set has 4 features, and 3 classes, and the second data set has 13 features, and 3 classes as well. The Thyroid data set contains 215 results of medical tests with five parameters, divided into 3 classes. The problem of classification is to relate each presented combination of features to a certain class. For the sake of simplicity, we assume here that the number of clusters is equal to the number of classes.

To build a classifier, we apply the above described clustering algorithms. In the data set X, which is the input to the clustering algorithms, each instance $x_k \in X$, $k = 1, 2, \ldots, N$ consists of its feature vector (4-dimensional or 13-dimensional respectively), and one of the following adjoined 3 dimensional class target vectors $(0, 0, 1)^T$, $(0, 1, 0)^T$, or $(1, 0, 0)^T$. Thus, the clustering is performed in the input-output space [Abe96]. This is done to improve the classification accuracy. The centers of the resulting clusters in the input space represent the prototypes in the feature space, and the adjoined class target coordinates in the output space represent the corresponding class labels.

The membership of an object to a certain cluster during the process of classification is calculated according to equations (9) or (25), depending on the type of the clustering algorithm that was used. In the classification, the vectors x_k correspond to the feature vectors, and the adjoined class target coordinates in the prototype vectors c_j are discarded. The cluster to which the given object belongs with maximum degree of membership determines the class of that object.

We carried out two experiments. In **the first experiment**, we compared the performance of the clustering algorithms in the problem of classification when instances of all the available classes were present in the data set used for clustering, i.e. the number of classes was known a priori and equal to 3. The data sets were divided into the training and testing sets with 70% and 30% of data respectively. For better performance of the recursive clustering algorithms, the data sets were randomly shuffled.

The training sets were used for the initialization of the classifier through fuzzy clustering, and the testing sets were used for the comparison of the classification accuracy. We used the learning rates $\eta = 0.01$ in the recursive procedures (19), (20) and (24), (25), and the "fuzzifier" parameter $\beta = 2$ in the batch possibilistic clustering procedure (13)-(15). Both possibilstic procedures (batch and recursive) were initialized from the results of probabilistic clustering through the fuzzy c-means algorithm.

We performed 10 iterations for the batch clustering procedures, and 10 runs over the training data for the recursive clustering procedures. The experiment was repeated 50 times, and then average results were calculated. The results are given in Table 1. They represent the percentage of the incorrectly classified objects from the testing data set.

Table 1: Classification with known number of classes (error rate on the testing data)

Data	**Fuzzy c-means**	**Batch possibilistic**	**Park-Dagher**	**Recursive possibilistic**
Iris	7.4 %	7.6 %	6.9 %	7.8 %
Wine	3.8 %	4.4 %	3.9 %	4.3 %

The results of the first experiment in Table 2 are quite close for all the clustering algorithms that were tested. The recursive possibilistic clustering algorithm (24), (25) showed results similar to those of the batch possibilistic clustering algorithm (13)-(15). In **the second experiment**, we included only the instances of 2 out of 3 available classes into the training sets. Thus, the number of classes in the training sets was less than that in the testing sets. The clustering procedures were used to create 2 clusters, corresponding to the classes of the objects in the training sets. The objects of the third unknown class were introduced in the testing sets. The results of the second experiment are shown in Table 2.

Table 2: Classification with one unknown class (error rate on the testing data)

Data, unknown class	**Fuzzy c-means**	**Batch possibilistic**	**Park-Dagher**	**Recursive possibilistic**
Iris, class 3	33.3 %	14.0 %	33.3 %	15.3 %
Iris, class 1	38.0 %	6.0 %	38.0 %	6.7 %
Wine, class 3	29.0 %	19.6 %	29.0 %	19.6 %
Wine, class 1	35.9 %	23.6 %	35.9 %	23.4 %
Thyroid, class 2	18.1 %	20.9 %	18.6 %	11.6 %
Thyroid, class 3	15.4 %	17.2 %	15.4 %	5.6 %

As an indication of the unknown class, we used the threshold of 0.2 for the sum of degrees of membership. Naturally, the classifiers based on probabilistic clustering were practically unable to distinguish the objects of the unknown class from the objects of other two classes, because the probabilistic clustering procedures rely on the unity sum of degrees of membership. This is not the case with the possibilistic clustering procedures, which showed considerably better performance in the second experiment.

5 Robust Adaptive Fuzzy Clustering Algorithms

The approaches mentioned in the preceding section are capable of efficient data clustering when the clusters are overlapping, but only with the assumption that the clusters are compact, i.e. they do not have abrupt (anomalous) outliers. Whereas real datasets usually contain up to 20% of outliers [BL78, Rey78, Hub81], the assumption of cluster compactness may sometimes become inadequate.

Thus, the problem of cluster analysis of data with heavy-tailed distributions has received more and more attention in recent years. Various modifications of clustering methods mentioned above are proposed in the papers [Loo99, Loo01, TSMI97, HK00, GK04] and designed to process data containing outliers.

5.1 Robust probabilistic fuzzy clustering algorithm

As it was noted above, probabilistic fuzzy-clustering approach belongs to a class of the objective function based algorithms [Bez81] that are designed to solve the clustering problem via the optimization of a certain predetermined clustering criterion, and are, in our opinion, the best-grounded from the mathematical point of view.

For pre-standardized feature vectors, the objective function is

$$E^R(w_{k,j}, c_j) = \sum_{k=1}^{N}\sum_{j=1}^{m} w_{k,j}^{\beta} D(x_k, c_j) \tag{27}$$

subject to constraints

$$\sum_{j=1}^{m} w_{k,j} = 1,\ k = 1, \ldots, N, \tag{28}$$

$$0 < \sum_{k=1}^{N} w_{k,j} \leq N,\ j = 1, \ldots, m. \tag{29}$$

Here $D(x_k, c_j)$ is the distance between x_k and c_j in the adopted arbitrary metrics.

The distance function $D(x_k, c_j)$ is usually assumed to be the Minkowski L^p metrics [Pau81]

$$D(x_k, c_j) = \Big(\sum_{i=1}^{m} |x_i(k) - c_{ji}|^p \Big)^{\frac{1}{p}}, p \geq 1 \tag{30}$$

where $x_{k,j}$, c_{ji} are i-th components of $(n \times 1)$-vectors x_k, c_j respectively.

The estimates connected with the quadratic objective functions, considered in the preceding section, are optimal when the processed data belong to the class of distributions with bounded variance. The most important representative of this class is the Gaussian distribution. Varying the parameter p, we can improve the robustness property of the clustering procedures. However, the estimation quality is determined by the distribution of data. Indeed, the estimates corresponding to $p = 1$ are optimal for the Laplacian distribution, but obtaining them requires a lot of computations.

The approximate normal distributions are a reasonable compromise [Tsy84]. They are a mixture of a Gaussian density and a distribution of an arbitrary density, contaminating the Gaussian distribution with outliers. The optimal objective function in this case is the quadratic-linear, the linear part appearing as the distance from the minimum increases.

One of the most important approximate normal distribution density is the function [Tsy84]

$$p(x_i, c_i) = \mathrm{Se}(c_i, s_i) = \frac{1}{2s_i}\ \mathrm{sech}^2 \frac{x_i - c_i}{s_i}, \tag{31}$$

where c_i and s_i are the parameters that define the center and width of the distribution respectively. This function resembles the Gaussian in the vicinity of the center, but differs with its heavy tails. The distribution (31) is connected with the objective function [HW77, Wel77]

$$f_i(x_i, c_i) = \beta_i \ln \cosh \frac{x_i - c_i}{\beta_i}, \tag{32}$$

where the parameter β_i determines the steepness of this function, which is close to the quadratic one in the vicinity of the minimum, and tends to the linear one as the distance from the minimum increases.

It is interesting to note that the derivative of this function is

$$f_i'(x_i) = \varphi(x_i) = \tanh \frac{x_i}{\beta_i}, \tag{33}$$

which is, in fact, a standard activation function of an artificial neuron [CU93].

Using the following construct as the metrics

$$D^R(x_k, c_j) = \sum_{i=1}^{n} f_i(x_i(k) - c_{ji}) = \sum_{i=1}^{n} \beta_i \ln \cosh \frac{x_{k,i} - c_{ji}}{\beta_i}, \tag{34}$$

we can introduce an objective function for robust clustering

$$E^R(w_{k,j}, c_j) = \sum_{k=1}^{N} \sum_{j=1}^{m} w_{k,j}^{\beta} D^R(x_k, c_j) = \sum_{k=1}^{N} \sum_{j=1}^{m} w_{k,j}^{\beta} \sum_{i=1}^{n} \beta_i \ln \cosh \frac{x_{k,i} - c_{ji}}{\beta_i}, \tag{35}$$

and a corresponding Lagrange function

$$L(w_{k,j}, c_j, \lambda_k) = \sum_{k=1}^{N}\sum_{j=1}^{m} w_{k,j}^{\beta} \sum_{i=1}^{n} \beta_i \ln\cosh\frac{x_{k,i} - c_{ji}}{\beta_i} + \sum_{k=1}^{N} \lambda_k \Big(\sum_{j=1}^{m} w_{k,j} - 1 \Big), \quad (36)$$

where λ_k is an undetermined Lagrange multiplier that guarantees the fulfillment of the constraints (28), (29). The saddle point of the Lagrange function (36) could be found solving the following system of Kuhn-Tucker equations

$$\begin{cases} \dfrac{\partial L(w_{k,j}, c_j, \lambda_k)}{\partial w_{k,j}} = 0, \\ \dfrac{\partial L(w_{k,j}, c_j, \lambda_k)}{\partial \lambda_k} = 0, \\ \nabla_{c_j} L(w_{k,j}, c_j, \lambda_k) = 0. \end{cases} \quad (37)$$

Solving the first and the second equation of the system (37) leads to the well-known result

$$\begin{cases} w_{k,j} = \dfrac{(D^R(x_k, c_j))^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m} (D^R(x_k, c_l))^{\frac{1}{1-\beta}}}, \\ \lambda_k = -\Big(\sum_{l=1}^{m} (\beta D^R(x_k, c_l))^{\frac{1}{1-\beta}} \Big)^{1-\beta}, \end{cases} \quad (38)$$

but the third one

$$\nabla_{c_j} L(w_{k,j}, c_j, \lambda_k) = \sum_{k=1}^{N} w_j^{\beta} \nabla_{c_j} D^R(x_k, c_j) = 0 \quad (39)$$

obviously has no analytical solution. The solution of (39) could be computed with use of a local modification of the Lagrange function [Loo99] and the recursive fuzzy clustering algorithms [BKS02]. Furthermore, searching the saddle point of the local Lagrange function

$$L_k(w_{k,j}, c_j, \lambda_k) = \sum_{j=1}^{m} w_{k,j}^{\beta} D^R(x_k, c_j) + \lambda_k \Big(\sum_{j=1}^{m} w_{k,j} - 1 \Big) \quad (40)$$

using the Arrow-Hurwitz-Uzawa procedure gives the following algorithm:

$$\begin{cases} w_{k,j} = \dfrac{(D^R(x_k, c_j))^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m} (D^R(x_k, c_l))^{\frac{1}{1-\beta}}}, \\ c_{k+1,ji} = c_{k,ji} - \eta_k \dfrac{\partial L_k(w_{k,j}, c_j, \lambda_k)}{\partial c_{ji}} = c_{k,ji} + \eta_k w_{k,j}^{\beta} \tanh \dfrac{x_{k,i} - c_{k,ji}}{\beta_i}, \end{cases} \quad (41)$$

where $c_{k,ji}$ is the i-th component of the j-th prototype vector calculated at the k-th step.

In spite of the low computational complexity of (41), it has a drawback due to the constraint (28) which is common to all the probabilistic fuzzy clustering algorithms.

5.2 Robust possibilistic fuzzy clustering algorithm

In the possibilistic clustering algorithms, the criterion is

$$E^R(w_{k,j}, c_j, \mu_j) = \sum_{k=1}^{N}\sum_{j=1}^{m} w_{k,j}^{\beta} D(x_k, c_j) = \sum_{j=1}^{m} \mu_j \sum_{k=1}^{N} (1 - w_{k,j})^{\beta}. \tag{42}$$

Minimization of (42) with respect to $w_{k,j}$, c_j, and μ_j leads to the following system of equations

$$\begin{cases} \dfrac{\partial E^R(w_{k,j}, c_j, \mu_j)}{\partial w_{k,j}} = 0, \\ \dfrac{\partial E^R(w_{k,j}, c_j, \mu_j)}{\partial \lambda_k} = 0, \\ \nabla_{c_j} E^R(w_{k,j}, c_j, \mu_j) = 0. \end{cases} \tag{43}$$

While the solution of the first two equations of the system (43) leads to the well-known result

$$\begin{cases} w_{k,j} = \left(1 + \left(\dfrac{D^R(x_k, c_j)}{\mu_j}\right)^{\frac{1}{\beta-1}}\right)^{-1}, \\ \mu_j = \dfrac{\sum_{k=1}^{N} w_{k,j}^{\beta} D^R(x_k, c_j)}{\sum_{k=1}^{N} w_{k,j}^{\beta}}, \end{cases} \tag{44}$$

the third one

$$\nabla_{c_j} E^R(w_{k,j}, c_j, \mu_j) = \sum_{k=1}^{N} w_j^{\beta} \nabla_{c_j} D^R(x_k, c_j) = 0 \tag{45}$$

completely coincides with (39) with all the negative consequences when the metrics (34) is used.

Introducing a local modification of (42)

$$E_k^R(w_{k,j}, c_j, \mu_j) = \sum_{j=1}^{m} w_{k,j}^{\beta} D^R(x_k, c_j) + \sum_{j=1}^{m} \mu_j (1 - w_{k,j})^{\beta} =$$
$$= \sum_{j=1}^{m} w_{k,j}^{\beta} \sum_{i=1}^{n} \beta_i \ln \cosh \frac{x_{k,i} - c_{ji}}{\beta_i} + \sum_{j=1}^{m} \mu_j (1 - w_{k,j})^{\beta} \qquad (46)$$

and optimizing it using the Arrow-Hurwitz-Uzawa procedure, we obtain:

$$\begin{cases} w_{k,j} = \left(1 + \left(\dfrac{D^R(x_k, c_j)}{\mu_j}\right)^{\frac{1}{\beta-1}}\right)^{-1}, \\ c_{k+1,ji} = c_{k,ji} - \eta_k \dfrac{\partial E_k(w_{k,j}, c_j, \mu_j)}{\partial c_{ji}} = c_{k,ji} + \eta_k w_{k,j}^{\beta} \tanh \dfrac{x_{k,i} - c_{k,ji}}{\beta_i}, \end{cases} \qquad (47)$$

where the distance parameters $\mu_{k,j}$ could be considered with respect to the second equation of the system (44), where only k observations should used instead of the data set of size N.

Note that the last equations of systems (41) and (47) are completely identical and are determined only by the adopted metrics. This circumstance allows us to use any distance metrics suitable for each particular case that will define only the tuning procedures for the prototypes, whereas the equation for the calculation of weight remains the same.

Considered robust recursive methods could be used in the multi-pass batch mode as well as in the on-line mode. In the latter case, the observation number k will be the discrete time index.

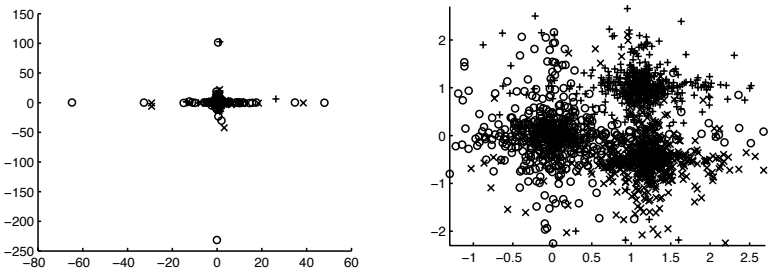
We used the proposed algorithms in the problem of data classification on a specially generated artificial data set containing three two-dimensional data clusters with samples labeled as 'o','x', and '+' (see Fig. 1). Each cluster of the dataset is distributed according to the heavy-tailed Laplacian density

$$p(x_i) = \sigma(1 + (x_i - c)^2)^{-1}, \qquad (48)$$

where σ and c are the width and expectation respectively.

The data set contains 9000 samples (3000 for each cluster), divided into the training (7200 samples) and checking (1800 samples) sets.

For each of the compared algorithms, the procedure was as follows. First, the training set was clustered using the respective algorithm and the prototypes of the clusters were found. Then, the training and checking sets were classified according to the results of clustering. The membership of a sample to a certain cluster during the process of classification was calculated according to the equations (9), (41), or (47) depending on the type of the clustering algorithm that was used. The cluster to which the given sample belongs with maximum degree of membership determines the class of that sample.



(a) (b)

Figure 1: Fragment of the data set: (a)Complete data set with outliers; (b)Central part of the data set

Classification and training were performed in on-line mode assuming $\beta = 2$, $\beta_1 = \beta_2 = \beta_3 = 1$, $\eta_k = 0.01$. The results are shown in Table3.

Table 3: Classification results

Algorithm	**Classification error rate**	
	Training	**Checking**
Bezdek's fuzzy c-means	17.1 % (1229 samples)	16.6 % (299 samples)
Probabilistic robust clustering (41)	15.6 % (1127 samples)	15.6 % (281 samples)
Possibilistic robust clustering (47)	15.2 % (1099 samples)	14.6 % (263 samples)

The drawback of fuzzy clustering methods based on the quadratic objective function could be visually shown by plotting the obtained prototypes over the data set. From Fig. 2 it could be easily seen that the cluster centers (prototypes) obtained using Bezdek's fuzzy c-means algorithm are displaced from the visual cluster centers due to the heavy-tailed distribution density of observations, in contrast to the robust objective function-based methods (41) and (47) which found the cluster prototypes more accurately. This is confirmed by lower classification error rates as shown in Table 3.

6 Conclusion

Computational Inteligence is a powerful methodology for a wide range of data analysis problems. The constantly growing number of successful applications of these techniques in the field of data analysis confirms the versatility of this approach. Examples are financial forecasting [TT], pattern identification in gene expression data [WW00, BPDB01], industrial applications [Dat], etc.

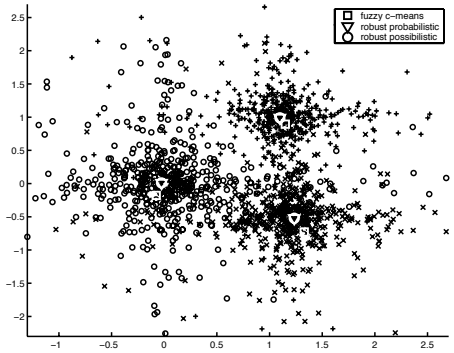


Figure 2: Cluster prototypes layout obtained by different learning algorithms

At the same time, the problems that arise in information processing, complicate the use of the existing algorithms, and demand the development of new tools.

References

[Abe96] S. Abe. *Neural Networks and Fuzzy Systems*. Kluwer Academic Publishers, Boston, 1996.

[AR88] J.A. Anderson and E. Rosenfeld. *Neurocomputing: Foundatons of Research*. The MIT Press, Cambridge, MA, 1988.

[BCKO02] Ye. Bodyanskiy, O. Chaplanov, V. Kolodyazhniy, and P. Otto. Adaptive quadratic radial basis function network for time series forecasting. In *Proc. 10th East-West Fuzzy Colloquim, Zittau, Germany*, pages 164–172, 2002.

[Bez81] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[BFM97] T. Bäck, D. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computing*. IOP Publishing Ltd. and Oxford University Press, 1997.

[BGK03a] Ye. Bodyanskiy, Ye. Gorshkov, and V. Kolodyazhniy. Combined neural network for non-linear system modeling. In *Proc. Int. Conf. on Computational Intelligence for Modeling, Control and Automation CIMCA'2003, Vienna, Austria*, pages 692–703, 2003.

[BGK03b] Ye. Bodyanskiy, Ye. Gorshkov, and V. Kolodyazhniy. Resource-allocating probabilistic neuro-fuzzy network. In *Proc. 3rd Int. Conf. of European Union Society for Fuzzy Logic and Technology (EUSFLAT '2003), Zittau, Germany*, pages 392–395, 2003.

[BGKK02] Ye. Bodyanskiy, Ye. Gorshkov, I. Kokshenev, and V. Kolodyazhniy. On adaptive fuzzy clustering algorithm. *Adaptive Systems of Automatic Control*, 5:107–117, 2002. (In Russian).

[BK04a] Ye. Bodyanskiy and V. Kolodyazhniy. Fuzzy Kolmogorov's network. In *Proc. 8th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES 2004), Wellington, New Zealand, September 20-25, Part II*, pages 764–771, 2004.

[BK04b] Ye. Bodyanskiy and V. Kolodyazhniy. Fuzzy neural network with Kolmogorov's structure. In *Proc. 11th East-West Fuzzy Colloquium, Zittau, Germany*, pages 139–146, 2004.

[BKK02] Ye. Bodyanskiy, V. Kolodyazhniy, and N. Kulishova. Generalized forecasting Sigma-Pi neural network. In P. Sincak et al., editor, *Frontiers in Artificial Intelligence. IOS Press, Amsterdam*, pages 29–33, 2002.

[BKK03] Ye. Bodyanskiy, I. Kokshenev, and V. Kolodyazhniy. An adaptive learning algorithm for a neo fuzzy neuron. In *Proc. 3rd Int. Conf. of European Union Society for Fuzzy Logic and Technology (EUSFLAT '2003), Zittau, Germany*, pages 375–379, 2003.

[BKO04] Ye. Bodyanskiy, V. Kolodyazhniy, and P. Otto. Universal approximator employing neo-fuzzy neurons. In *Proc. 8th Fuzzy Days, Dortmund, Germany, Sep. 29 Oct. 1*, 2004. (CD-ROM).

[BKS01] Ye. Bodyanskiy, V. Kolodyazhniy, and A. Stephan. An adaptive learning algorithm for a neuro-fuzzy network. *Computational Intelligence. Theory and Applications, Lecture Notes in Computer Science*, 2206:68–75, 2001.

[BKS02] Ye. Bodyanskiy, V. Kolodyazhniy, and A. Stephan. Recursive fuzzy clustering algorithms. In *Proc. 10th East-West Fuzzy Colloquim, Zittau, Germany*, pages 276–283, 2002.

[BL78] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley and Sons, Chichester-New York-Brisbane-Toronto, 1978.

[BPDB01] S. Bicciato, M. Pandin, G. Didone, and C. Bello. Analysis of an associative memory neural network for pattern identification in gene expression data. In *Proc. BIOKDD01, Workshop on Data Mining in Bioinformatics*, 2001.

[BR04] Ye.V. Bodyanskiy and O.G. Rudenko. *Artificial Neural Networks: Architectures, Learning, Applications*. TELETEKH, Kharkiv, 2004. (In Russian).

[BV03a] Ye. Bodyanskiy and O. Vynokurova. Adaptive wavelet-neural predictor. *Problemy bioniki*, 58:10–17, 2003. (In Russian).

[BV03b] Ye. Bodyanskiy and O. Vynokurova. Artificial wavelet-neural networks learning in nonstationary stochastic signals processing. *Radioelectronika i informatica*, 1:85–89, 2003. (In Russian).

[CL94] F.L. Chung and T. Lee. Fuzzy competitive learning. *Neural Networks.*, 7:539–552, 1994.

[Cov68] T.M. Cover. Estimates by the nearest-neighbor rule. *IEEE Trans. on Information Theory*, 14:50–55, 1968.

[CU93] A. Chichocki and R. Unbehauen. *Neural Networks for Optimization and Signal Processing*. Teubner, Stuttgart, 1993.

[Dat] Data mining for water industry applications. http://www.edie.net/Library/Features/WTJ9848.html.

[GG89] I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence.*, 11:773–781., 1989.

[GK79] E.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE CDC, San Diego, California*, pages 761–766, 1979.

[GK04] O. Georgieva and F. Klawonn. A clustering algorithm for identification of single clusters in large data sets. In *Proc. 11th East-West Fuzzy Colloquium. HS Zittau-Görlitz*, pages 118–125, 2004.

[HK00] F. Höppner and F. Klawonn. Fuzzy clustering of sampled functions. In *Proc. 19th Int. Conf. of the North American Fuzzy Information Processing Society (NAFIPS), Atlanta, USA*, pages 251–255, 2000.

[HKK96] F. Höppner, F. Klawonn, and R. Kruse. *Fuzzy-Klusteranalyse. Verfahren für die Bilderkennung, Klassifikation und Datenanalyse*. Vieweg, Braunschweig, 1996.

[Hol75] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[Hub81] P.J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[HW77] P.W. Holland and R.E. Welsh. Robust regression using iteratively re-weighted least squares. *Comm. Statist. Theory and Methods*, pages 813–827, 1977.

[ICP03] S. Iyengar, E Cho, and V. Phoha. *Foundation of Wavelet Neural Networks and Application*. Chapman Hall, CRC Press LLC, Florida, 2003.

[JSM97] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing - A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, Upper Saddle, NJ, 1997.

[KJ] V. Kumar and M. Joshi. Tutorial on high-performance data mining. *Univ. of Minnesota, Minneapolis, USA*. http://www.cs.umn.edu/ mjoshi/myw.html.

[KK93a] R. Krishnapuram and J. Keller. Fuzzy and possibilistic clustering methods for computer vision. *IEEE Trans. on Fuzzy Systems*, 1:98–110, 1993.

[KK93b] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Trans. on Fuzzy Systems*, 1:98–110, 1993.

[KK97] F. Klawonn and R. Kruse. Constructing a fuzzy controller from data. *Fuzzy Sets and Systems.*, 85:177–193, 1997.

[KKT97] F. Klawonn, R. Kruse, and H. Timm. Fuzzy Shell Cluster Analysis. *Learning, Networks and Statistics.*, pages 105–120, 1997.

[KNB99] R. Kruse, D. Nauck, and C. Borgelt. Data mining with fuzzy methods: status and perspectives. In *Proc. 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99, Aachen, Germany), CDROM. Verlag Mainz, Aachen, Germany*, 1999.

[Koh01] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, London, 2001.

[Loo99] C.G. Looney. A fuzzy clustering and fuzzy merging algorithm. Technical Report, CS-UNR-101-1999., 1999. http://sherry.ifi.unizh.ch/looney99fuzzy.html.

[Loo01] C.G. Looney. A fuzzy classifier with ellipsoidal Epanechnikovs. Technical Report, Computer Science Department, University of Nevada, Reno, NV, 2001. http://sherry.ifi.unizh.ch/looney01fuzzy.html.

[Mac65] J. MacQueen. On convergence of k-means and partitions with minimum average variance. *Ann. Math. Statist*, 36:1084, 1965.

[MD89] J. Moody and C. Darken. Fast-learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[MPM02] S. Mitra, S.K. Pal, and P. Mitra. Data mining in soft computing framework: a survey. *IEEE Trans. on Neural Networks*, 13:3–14, 2002.

[MY99] T. Miki and T. Yamakawa. Analog implementation of neo-fuzzy neuron and its on-board learning. *Computational Intelligence and Application*, pages 144–149, 1999.

[OBK03] P. Otto, Ye. Bodyanskiy, and V. Kolodyazhniy. A new learning algorithm for a forecasting neuro-fuzzy network. *Integrated Computer-Aided Engineering*, 10:399–409, 2003.

[Pau81] L.F. Pau. *Failure Diagnosis and Performance Monitoring*. Marcel Dekker Inc., NY, 1981.

[PD84] D.C. Park and I. Dagher. Gradient based fuzzy c-means (GBFCM) algorithm. In *Proc. IEEE Int. Conf. on Neural Networks*, pages 1626–1631, 1984.

[Rey78] W.J.J. Rey. *Robust Statistical Methods. Lecture Notes in Mathematics*. Springer-Verlag, Berlin-Heidelberg-New York, 1978.

[RHW86] D.E. Rumelhart, G.R. Hinton, and R.J. Williams. Learning internal representation by error propagation. *Parallel Distributed Processing*, 1:318–364, 1986.

[Roj96] R. Rojas. *Neural Networks. A Systematic Introduction*. Springer-Verlag, Berlin, 1996.

[TSMI97] K. Tsuda, S. Senda, M. Minoh, and K. Ikeda. Sequential fuzzy cluster extraction and its robustness against noise. *Systems and Computers in Japan*, 28:10–17, 1997.

[Tsy84] Ya.Z. Tsypkin. *Foundations of Information Theory of Identification*. Nauka, Moscow, 1984. (In Russian).

[TT] D.L. Toulson and S.P. Toulson. Intra-day trading of the FTSE-100 futures contract using neural networks with wavelet encodings. http://www.if5.com/papers/Intra-day_Trading_of_the_FTSE-100_Futures.pdf.

[UCI] The UCI Repository Of Machine Learning Databases and Domain Theories. http://www.ics.uci.edu/ mlearn/MLRepository.html.

[Wel77] R.E. Welsh. Nonlinear statistical data analysis. In *Proc. Comp. Sci. and Statist. Tenth Ann. Symp. Interface. Nat'l Bur. Stds. Gaithersburg, MD*, pages 77–86, 1977.

[WW00] P.J. Woolf and Y. Wang. A fuzzy logic approach to analyzing gene expression data. *Physiol.Genomics*, 3:9–15, 2000.

[XWI05] R. Xu and D. Wunsch II. Survey of clustering algorithms. *IEEE Trans. on Neural Networks*, 16:645–678, 2005.

[YF94] R.R. Yager and D.P. Filev. Approximate clustering via the mountain method. *IEEE Trans. on Syst., Man and Cybern.*, 24:1279–1284, 1994.

[YUTK92] T. Yamakawa, E. Uchino, Miki T., and H. Kusanagi. A neo fuzzy neuron and its application to system identification and prediction of the system behavior. In *Proc. 2-nd Int. Conf. on Fuzzy Logic and Neural Networks "IIZUKA-92", Iizuka, Japan*, pages 477–483, 1992.

[Zad65] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[ZB92] Q. Zhang and A. Benveniste. Wavelet networks. *IEEE Trans. on Neural Networks*, 3:889–898, 1992.