An Adaptable Gaussian Neuro-Fuzzy Classifier

Minas Pertselakis, Dimitrios Frossyniotis and Andreas Stafylopatis

National Technical University of Athens
School of Electrical and Computer Engineering
9, Iroon Polytechneiou Str., Zografou, 157 80, Athens, Greece
{mper, dfros}@cslab.ntua.gr, andreas@cs.ntua.gr

Abstract. The concept of semantic and context aware intelligent systems provides a vision for the Information Society where the emphasis lays on computing applications that can sense context from the people and the environment and wrap that knowledge into adaptable behavior. In this framework the proper and automatic classification of data gathered by sensors is of major importance. Our approach describes a model that operates as a self-evaluating classifier using on-line re-clustering, addressing adequately the basic issues of modern demands. The novelty of the model lies in a flexible and efficient initialization technique that first partitions the data space utilizing Gaussian distributions and then merges clusters so as to produce an effective partitioning.

1 Introduction

The area of context-aware computing is aimed at the adaptation of the behavior of an application as a function of its current environment [1]. This environment can be characterized as a physical location, a semantic data space or a user profile. A context-aware application can sense the environment and interpret the events that occur within it, reacting accordingly. In heterogeneous environments where semantic integration is often required, the information to be accessed is heterogeneous and attribute correspondences could be fuzzy. In this framework, applications and algorithms should be able to automatically analyze vast amounts of data, extract or exchange knowledge and make decisions in a given context.

Fuzzy neural models have the ability to operate and adapt in both numeric as well as linguistic environments. In addition, they can handle fuzzy attributes and can be adaptive through learning from data. Therefore, in a context-aware computing environment, an adaptive neuro-fuzzy model should be self-evaluating and able to facilitate fast learning through data-driven knowledge embedded in its architecture. More specifically, in order to embed data-driven or expert knowledge in a neuro-fuzzy model, the usual way is to apply a clustering or partitioning method. In the clustering approach the centers of fuzzy rules are initialized as cluster vectors extracted from the input data set [2]. A learning algorithm is utilized then to fine tune the rules based on the available training data. Partitioning methods divide the input-

Work partially funded by E.C. contract No.IST-2000-26091,\"ORESTEIA:Modular hybrid artefacts with adaptive functionality".

output cross space into finer regions. Each partition is supposed to represent an *if-then* rule [3]. Both approaches present low adaptability: The number of rules that describe the physical phenomenon under examination is estimated heuristically and does not change during the learning phase.

Our approach includes a context-aware intelligent classifier model that provides self-awareness by reconfiguring its “weak” fuzzy rules when necessary. The proposed system is trained using a novel initialization procedure. This procedure combines both a partitioning method and a clustering algorithm capable of extracting fuzzy rules from a well-fined partitioned dataset in a non-heuristic way.

2 The Integrated Classifier Model

The proposed model consists of three components: the trained classifier module, the Observer and the Catalyst (Fig. 1). The trained classifier could be any of the classifiers proposed in the literature [4]. We chose to use the Subsethood-Product Fuzzy Neural Inference System (SuPFuNIS, [5]), since it utilizes Gaussian distributions and performs very well as a classifier. The Observer is an incorporated mechanism that is constantly aware about the output status, tracks down the history of the data and is responsible for Catalyst’s activation. This activation takes place when we have a great number of input patterns that produce inaccurate or too fuzzy outputs. The Observer, having located the area of effect in the input-output cross space, sends this information to the Catalyst.

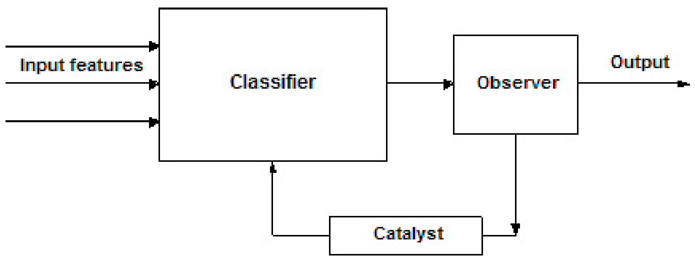

Fig. 1. The classifier model

The role of Catalyst is simple. It undertakes the task to reconstruct the rule(s) in question. How? By modifying the rule-cluster scenery in the most appropriate way. A new cluster could emerge, or an old one could be erased, a cluster could be sliced in two, or two could be merged, as indicated. The philosophy of this cluster reconfiguration is based on the general idea presented in section 3.

Since this model could be implemented in context-aware systems, such as health monitoring devices, or car status interfaces (see [10]), the computational error should be kept minimal and the model should be adaptive. The above model adequately addresses each of these aspects. A short description of the classifier module used is following.

2.1 The classifier module

Based on the SuPFuNIS model we implemented a subsethood product fuzzy neural classifier with the architecture shown in Figure 2 and the following characteristics:

(a) SuPFuNIS uses a tunable input fuzzifier that is responsible for fuzzification of numeric data. Numeric inputs are fuzzified using a feature-specific Gaussian spread.

(b) All information that propagates from the input layer is fuzzy. The model uses a composition mechanism employing a fuzzy mutual subsethood measure to define the activation that propagates to a rule node along a fuzzy connection.

(c) It aggregates activities at a rule node using a *fuzzy inner product*: a product of mutual subsethoods, which is different from the most common approach to use a fuzzy *min* conjunction operator.

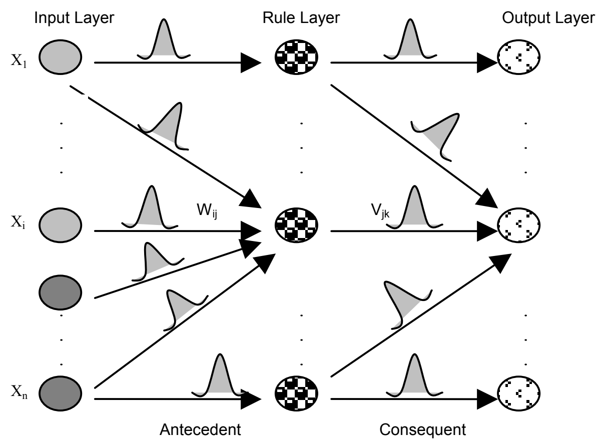

Fig. 2. Architecture of the SuPFuNIS model

Learning. Learning is incorporated into SuPFuNIS using the gradient descent method. A squared error criterion is used as a training performance parameter. The squared error $e(t)$ at iteration *t* is computed in the standard way:

$$e(t) = \frac{1}{2}\sum_{k=1}^{p}(d_k(t) - y_k(t))^2 \qquad \textbf{(1)}$$

where $d_k(t)$ is the desired output and $y_k(t)$ the defuzzified output at node *k*. The error is evaluated over all *p* outputs for a specific pattern input $\underline{x}(t)$.

The free parameters of the system, meaning both the centers w_{ij}^{c} (v_{jk}^{c}) and spreads w_{ij}^{σ} (v_{jk}^{σ}) of antecedent (consequent) connections and the spreads x_{i}^{σ} of the input features, are modified on the basis of update equations. Evaluation of partial derivatives required in weight update equations, as well as the analytic computations of the mutual subsethood, the net activation product and the defuzzified output, can be found in [5].
SuPFuNIS initialization method. In the SuPFuNIS model, the number of clusters determines the number of rules and the clustering is done in the input-output cross space. Thus, the centroids and boundaries of clusters can be applied as the values of the centers and spreads of fuzzy weights that fan in and out of a rule node. The employed clustering technique is the fuzzy *c*-means (FCM) algorithm [6] in conjunction with the Xie-Beni validity index [7] to cluster the given dataset and choose the best cluster, respectively.

3 Knowledge Extraction from Numeric Data

One of the methods to extract initial knowledge from the training data set is to cluster the data using a clustering technique. Cluster-based initialization has been known to improve the rate of learning as well as the performance of the system.

It is evident, though, that different clustering algorithms and even multiple replications of the same algorithm produce different partitioning results, due to random or parameterized initialization. For instance, SuPFuNIS employs the FCM algorithm, as described above, to partition the data set because it is simple and fast. Nevertheless, its performance depends on the initial cluster centers and, in addition, the user predefines the number of clusters. Therefore, it is necessary to run FCM several times, each time with a different number of clusters, to discover the right figure that results in the best performance of the classification system (Fig. 3).

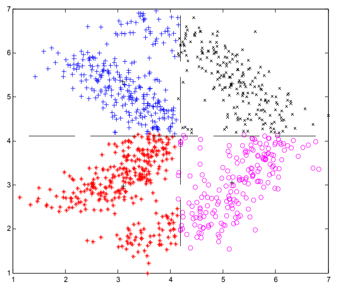

Fig. 3. Partitioning artificial data using FCM with Xie-Beni index for 4 clusters (predefined)

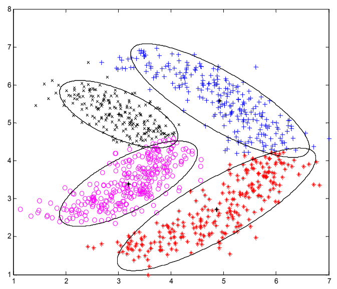

Fig. 4. Partitioning same data using Multi-Clustering Gaussian Fusion resulting in 4 clusters

To overcome these rigid restrictions we developed a multi-clustering fusion algorithm. A variant of this algorithm has been firstly introduced in [8], while, in this paper, we present a different configuration of the method, including adoption of Gaussian distributions as well as a new approach for specifying merging criteria for ellipsoid clusters (Fig. 4). According to this technique, the data are assumed to have been generated by several parameterized Gaussian distributions, so the data points are assigned to different clusters based on their posterior probabilities. In this direction, we believe that finding the optimal number of ellipsoid clusters in the data set can improve the robustness of the classification system. The proposed partitioning approach consists of two main phases: the partitioning and the fusion procedure.

3.1 Partitioning Procedure

In the Partitioning procedure, a basic clustering algorithm is applied for a number of iterations, Iter, so as to accomplish a distinct partitioning of *N* data points to a predefined number *C* of clusters. In our implementation, we incorporate the proposed methodology in the context of *k*-means as the basic clustering algorithm. The basic clustering algorithm partitions the data set in a different way for each iteration, creating a problem of deciding which cluster of one run corresponds to which in another run. This algorithm tackles this problem using the similarity between the clusters produced during successive runs. By determining the percentage of points of a cluster in the *t*-th run belonging to clusters of the *t-1*-th run, each cluster of the new run is assigned to one of the previous run, resulting in a cluster renumbering process.

After renumbering, if pattern $\vec{x}$ is assigned to cluster *q*, then a positive vote is given to cluster *q* and a negative one to all other clusters. This process defines a voting scheme, during which a voting table *VT* (of dimension $N \times C$) is updated, so that *VT(i,j)* denotes the membership degree of pattern $\vec{x}_i$ to cluster *j*, where *i=1,…,N*, and *j=1,…,C*. Using the *VT* table and the relation between the data points of one cluster with all the remaining clusters, a table *NRT* (of dimension $C \times C$) can be produced, so that *NRT(i,j)* represents the neighbourhood relation between clusters *i* and *j*.

For each cluster, a different Gaussian distribution (ellipsoid) is assigned and the specification of the Gaussian parameters is based on the expectation-minimization algorithm (EM, [9]). Thus, based on regular EM steps, fine-tuning of the parameters is carried out until convergence.

3.2 Fusion Procedure

After the parameters for each Gaussian distribution are specified, the Fusion procedure takes place and finds the optimal number of ellipsoids in the data set according to some predefined criteria. From the decomposition of the covariance matrix K_j of the *j*-th ellipsoid, we compute the angle $\phi_{kij} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ between the *k* major axis' eigenvector and the *i*-th dimension for each ellipsoid.
Given also the neighbourhood relation (table *NRT*) among ellipsoids, the fusion procedure commences with the predefined number *C* of ellipsoids and merges the ones that fulfill the following conditions:

1. Both ellipsoids are neighbour to each other,
2. The two angle vectors $(\varphi_{1i}, \varphi_{2i})$, where $i=1...(d-1)$, between the two ellipsoids satisfy one from the rules below for every i (d is the problem dimensionality):

A) φ_{1i} and φ_{2i} have the same sign, and $abs\ (\varphi_{1i}-\varphi_{2i})<70^o$, or **(2)**

B) φ_{1i} and φ_{2i} have different sign, and **(3)**
$(180^o - [abs\ (\varphi_{1i}) + abs\ (\varphi_{2i})] < 20^o$ or $[abs\ (\varphi_{1i}) + abs\ (\varphi_{2i})] < 20^o)$

The next step is to merge these ellipsoids into one and to reconfigure the voting table accordingly, by adding the votes of the second ellipsoid to the first one. The fusion procedure is running iteratively until the algorithm fails to find any pair of ellipsoids to merge.

4 Experimental Results

This section presents a comparative experimental evaluation of the proposed approach using different initialization techniques. Results concern the effectiveness of the classification module rather than the integrated classifier model as described in Section 2. A systematic assessment of the latter based on an involved experimental setup is ongoing [10]. In the following, the case of random initialization will be referred to as *Random-SuPFuNIS,* whereas the classifier resulting from using the FCM partitioning method combined with the Xie-Beni index will be referred to as *FCM-SuPFuNIS*. Similarly, using the proposed multi-clustering gaussian fusion algorithm for partitioning and for initializing the classifier will be referred to as *Multi-Fusion-SuPFuNIS*.

Two benchmark data sets were used to demonstrate the performance of the *Multi-Fusion-SuPFuNIS*: the Clouds and Pima Indians data sets. The Clouds artificial data from the ELENA project, ftp://ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases, are two-dimensional produced by three different Gaussian distributions. There are 5000 samples in the data set belonging to three clusters, which are relatively highly overlapped (see Fig.5). The Pima Indians set contains 8-dimensional data and can be obtained from the UCI data set repository ftp://ftp.ics.edu/pub/machine-learning-databases. It is based on personal data from 768 Pima Indians obtained by the National Institute of Diabetes and Digestive and Kidney Diseases. For each data set and for each number of rules, five experiments were performed with random splits of the data into training and test sets of fixed size, from which *mean* classification accuracy was calculated. More specifically, for the Clouds we used 2500 for training and 2500 for testing, while for the Pima Indians we used 500 for training and 268 for testing respectively. The results of our experiments are shown in Tables 1 and 2. Both number of clusters (rules) produced by each partitioning method and test accuracy of the respective classifier system are shown. Note that, for the *Random-SuPFuNIS* and for the *FCM-SuPFuNIS*, we give several results according to the number of rules, contrary to the *Multi-Fusion-SuPFuNIS* where the number of rules is computed in advance. For all the experimental cases considered we used twenty training epochs for the classifier.

Table 1. Experimental results for the Clouds data set

Number of Rules	*3*	*5*	*8*	*10*	*15*	*20*
Random-SuPFuNIS	78.58%	79.07%	72.3%	79.62%	85.96%	87.44%
FCM-SuPFuNIS	75.72%	88.40%	88.30%	88.90%	88.10%	88.70%
Multi-Fusion-SuPFuNIS	Number of Rules produced = 5 Generalization accuracy = 89.36%					

Table 2. Experimental results for the Pima Indians data set

Number of Rules	*3*	*5*
Random-SuPFuNIS	75.8%	76.63%
FCM-SuPFuNIS	76.2%	76.13%
Multi-Fusion-SuPFuNIS	Number of Rules produced = 4 Generalization accuracy = 79.48%	

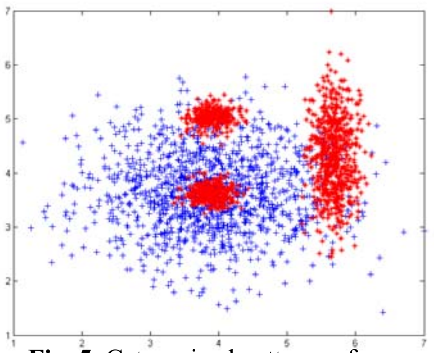
Fig. 5. Categorized patterns of clouds dataset

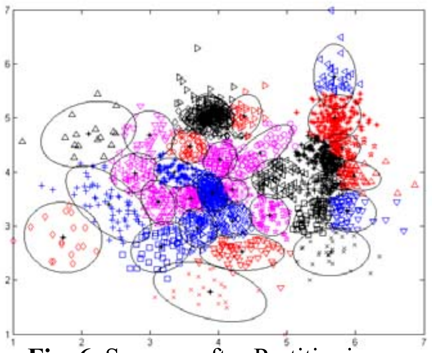
Fig. 6. Scenery after Partitioning Procedure

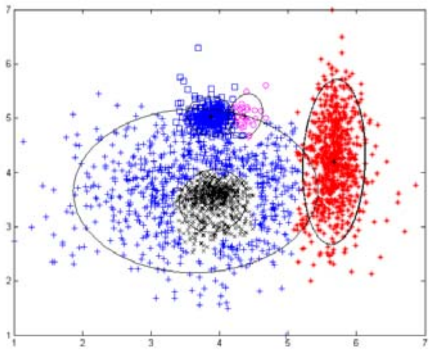
Fig. 7. Scenery after Fusion Process resulting in five clusters

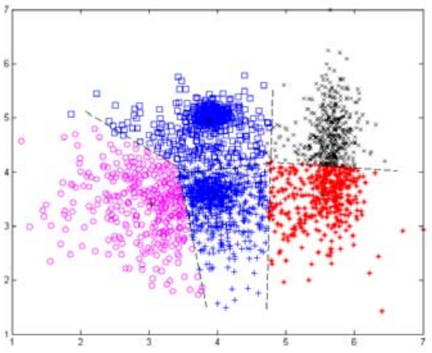
Fig. 8. Best Partitioning result of FCM – Xie-Beni index for the same number of clusters

Figures 6 and 7 show the application of the proposed approach to Clouds data at the end of each procedure, respectively, whereas Fig. 8 presents the best partitioning obtained by FCM – Xie-Beni combination. It is clear that the multi-clustering gaussian fusion method yields a much more accurate partitioning result with respect to the actual dataset structure.

5 Conclusions

In this paper we propose a classifier model that employs a novel multi-clustering initialization procedure. The major strengths of this intelligent model are its economy of parameters, fast learning and the ability to learn and adequately perform even in heterogeneous environments, where significant changes of the sensed data characteristics can be manifested. Ongoing and future work includes experimental evaluation of the integrated model using real-data test sets in non-stationary environments.

References

1. Abowd, G.: Software engineering issues for ubiquitous computing. Proceedings of the International Conference on Software Engineering, pp. 5 - 84, (1999).
2. Nauck, D., Kruse, R.: A neuro-fuzzy method to learn fuzzy classification rules from data. Fuzzy Sets and Systems, Vol.89, pp. 277-288, (1997).
3. Lin, Y., Cunningham, G. A. III, Coggeshall, S. V.: Using fuzzy partition to create fuzzy systems from input–output data and set the initial weights in a fuzzy neural network. IEEE Trans. Fuzzy Systems, Vol. 5, pp. 614–621, (1997).
4. Kovacs, T.: Learning Classifier Systems Resources. CSRP-00-19, (2000) [http://citeseer.nj.nec.com/article/kovacs02learning.html]
5. Paul, S., Kumar, S.: Subsethood-Product Fuzzy Neural Inference System (SuPFuNIS). IEEE Trans. On Neural Networks, Vol. 13, No. 3, pp. 578-599, (2002)
6. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, (1981).
7. Xie, X., Beni, G.: Validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Machine Learning, Vol. 3, pp. 841–846, (1991).
8. Frossyniotis, D., Pertselakis, M., Stafylopatis, A.: A Multi-Clustering Fusion Algorithm. Proceedings of the Second Hellenic Conference on Artificial Intelligence (SETN2002), LNAI 2308, pages 225-236, Springer-Verlag, Thessaloniki, (2002).
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Roy. Statist. Soc. B, 39: 1-38, (1977).
10. Modular Hybrid Artefacts with Adaptive Functionality (ORESTEIA), IST Project, 2001-2003. [http://www.image.ntua.gr/oresteia/index.html]