Hybrid systems III: Neuro-fuzzy classifiers

Conventional approaches of pattern classification involve clustering training samples and associating clusters to given categories.

The complexity and limitations of previous mechanisms are largely due to the lacking of an effective way of defining the boundaries among clusters. This problem becomes more intractable when the number of features used for classification increases.

On the contrary, fuzzy classification assumes the boundary between two neighboring classes as a continuous, overlapping area within which an object has partial membership in each class.

This viewpoint not only reflects the reality of many applications in which categories have fuzzy boundaries, but also provides a simple representation of the potentially complex partition of the feature space.

In brief, we use fuzzy IF-THEN rules to describe a classifier.

Assume that K patterns $x_p = (x_{p1}, \dots, x_{pn})$, $p = 1, \dots, K$ are given from two classes, where x_p is an n-dimensional crisp vector.

Typical fuzzy classification rules for $n = 2$ are like

If x_{p1} is *small* and x_{p2} is *very large* then $x_p = (x_{p1}, x_{p2})$ belongs to C_1
If x_{p1} is *large* and x_{p2} is *very small* then $x_p = (x_{p1}, x_{p2})$ belongs to C_2

where x_{p1} and x_{p2} are the features of pattern (or object) p, *small* and *very large* are linguistic terms characterized by appropriate membership functions.

The *firing level* of a rule

$$\Re_i : \text{If } x_{p1} \text{ is } A_i \text{ and } x_{p2} \text{ is } B_i$$
$$\text{then } x_p = (x_{p1}, x_{p2}) \text{ belongs to } C_i$$

with respect to a given object x_p is interpreted as the degree of belogness of x_p to C_i.

This firing level, denoted by α_i, is usually determined as

$$\alpha_i = \min\{A_i(x_{p1}), A_2(x_{p2})\}.$$

As such, a fuzzy rule gives a meaningful expression of the qualitative aspects of *human recognition*.

Based on the result of pattern matching between rule antecedents and input signals, a number of fuzzy rules are triggered in parallel with various values of firing strength.

Individually invoked actions are considered together with a combination logic.

Furthermore, we want the system to have *learning ability* of updating and fine-tuning itself based on newly coming information.

The task of *fuzzy classification* is to generate an appropriate fuzzy partition of the feature space.

In this context the word *appropriate* means that the number of misclassified patterns is very small or zero.

Then the rule base should be optimized by deleting rules which are not used.

Consider a two-class classification problem shown in Figure 1.

Suppose that the fuzzy partition for each input feature consists of three linguistic terms

$$\{small, medium, big\}$$

which are represented by triangular membership functions.

Both initial fuzzy partitions in Figure 1 satisfy 0.5-completeness for each input variable, and a pattern x_p is classified into Class j if there exists at least one rule for Class j in the rule base whose firing strength with respect to x_p is bigger or equal to 0.5.

So a rule is created by finding for a given input pattern x_p the combination of fuzzy sets, where each yields the highest degree of membership for the respective input feature.

If this combination is not identical to the antecedents of an already existing rule then a new rule is created.

However, it can occur that if the fuzzy partition is not set up correctly, or if the number of linguistic terms for the input features is not large enough, then some patterns will be missclassified.

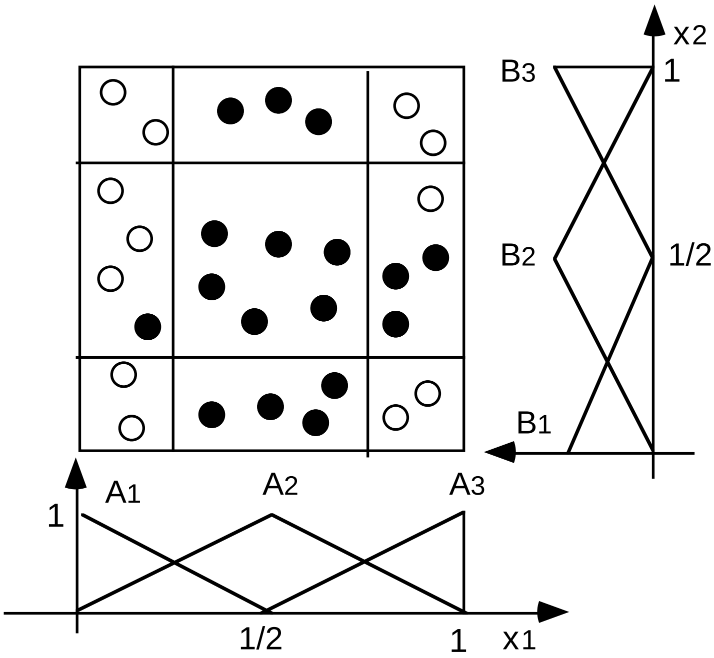

Figure 1 Initial fuzzy partition with 9 fuzzy subspaces and 2 misclassified patterns. Closed and open circles represent the given pattens from Class 1 and Class 2, respectively.

The following 9 rules can be generated from the initial fuzzy partitions shown in Figure 1:

$\Re_1$: If x_1 is *small* and x_2 is *big* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_2$: If x_1 is *small* and x_2 is *medium* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_3$: If x_1 is *small* and x_2 is *small* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_4$: If x_1 is *big* and x_2 is *small* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_5$: If x_1 is *big* and x_2 is *big* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_6$: If x_1 is *medium* and x_2 is *small* then $x_p = (x_1, x_2)$ belongs to Class C_2
$\Re_7$: If x_1 is *medium* and x_2 is *medium* then $x_p = (x_1, x_2)$ belongs to Class C_2
$\Re_8$: If x_1 is *medium* and x_2 is *big* then $x_p = (x_1, x_2)$ belongs to Class C_2
$\Re_9$: If x_1 is *big* and x_2 is *medium* then $x_p = (x_1, x_2)$ belongs to Class C_2

where we have used the linguistic terms *small* for A_1 and B_1, *medium* for A_2 and B_2, and *big* for A_3 and B_3.

However, the same rate of error can be reached by noticing that if ”x_1 is medium” then the pattern (x_1, x_2) belongs to Class 2, independently from the value of x_2, i.e. the following 7 rules provides the same classification result

$\Re_1$: If x_1 is *small* and x_2 is *big* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_2$: If x_1 is *small* and x_2 is *medium* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_3$: If x_1 is *small* and x_2 is *small* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_4$: If x_1 is *big* and x_2 is *small* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_5$: If x_1 is *big* and x_2 is *big* then $x = (x_1, x_2)$ belongs to Class C_1
$\Re_6$: If x_1 is *medium* then $x_p = (x_1, x_2)$ belongs to Class C_2
$\Re_7$: If x_1 is *big* and x_2 is *medium* then $x_p = (x_1, x_2)$ belongs to Class C_2

As an other example, Let us consider a two-class classification problem.

In Figure 2 closed and open rectangulars represent the given from Class 1 and Class 2, respectively.

If one tries to classify all the given patterns by fuzzy rules based on a simple fuzzy grid, a fine fuzzy partition and ($6 \times 6 = 36$) rules are required.

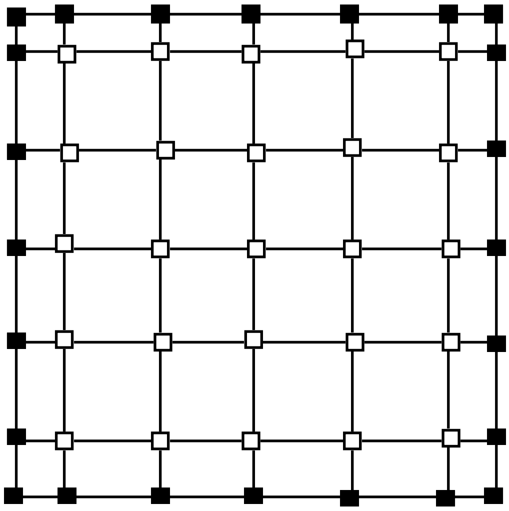

Figure 2 A two-dimensional classification problem.

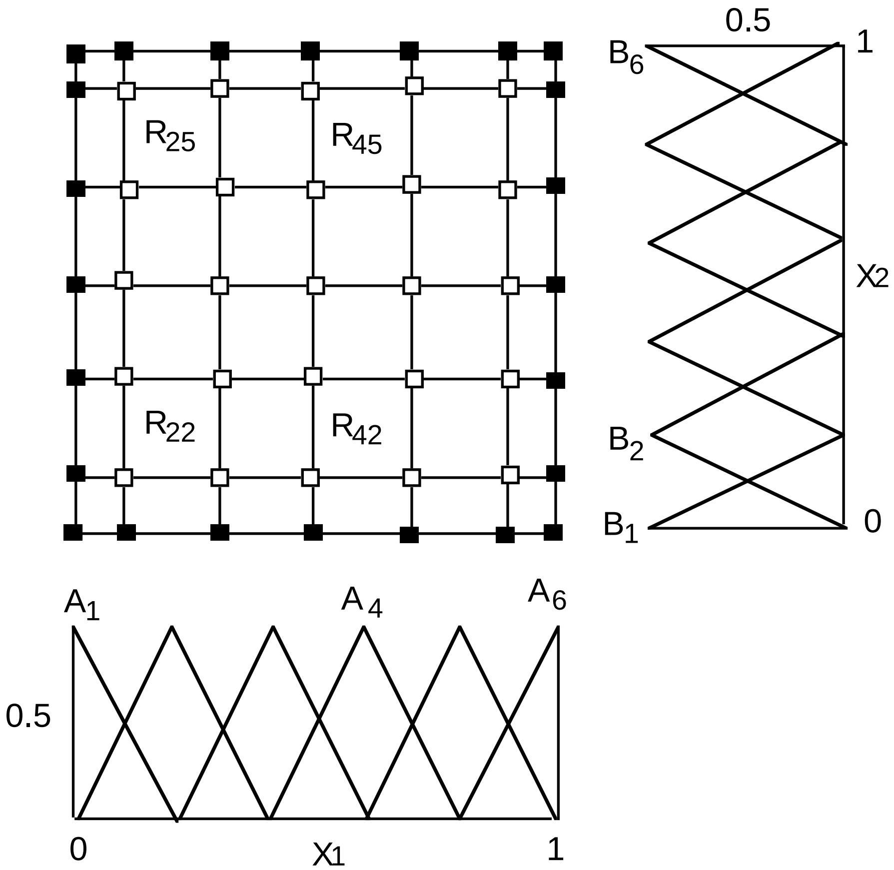

Figure 3 Fuzzy partition with 36 fuzzy subspaces.

However, it is easy to see that the patterns from Figure 3 may be correctly classified by the following five fuzzy IF-THEN rules

$\Re_1$: If x_1 is *very small* then Class 1,
$\Re_2$: If x_1 is *very large* then Class 1,
$\Re_3$: If x_2 is *very small* then Class 1,
$\Re_4$: If x_2 is *very large* then Class 1,
$\Re_5$: If x_1 is *not very small* and x_1 is *not very large* and x_2 is *not very small* and x_2 is *not very large* then Class 2

Sun and Jang propose an adaptive-network-based fuzzy classifier to solve fuzzy classification problems.

Figure 4 demonstrates this classifier architecture with two input variables x_1 and x_2.

The training data are categorized by two classes C_1 and C_2. Each input is represented by two linguistic terms, thus we have four rules.

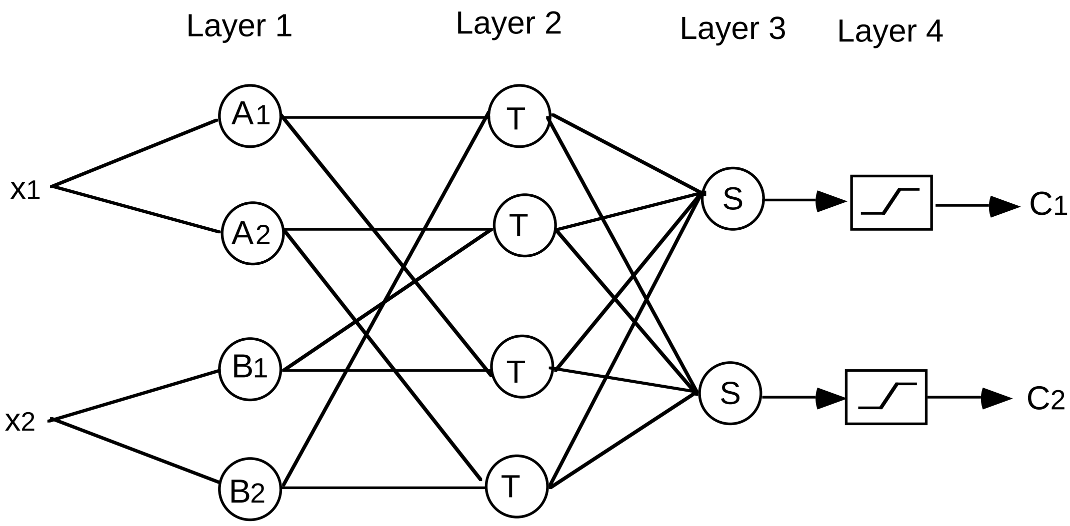

Figure 4 An adaptive-network-based fuzzy classifier.

- **Layer 1** The output of the node is the degree to which the given input satisfies the linguistic label associated to this node. Usually, we choose bell-shaped membership functions

$$A_i(u) = \exp\left[-\frac{1}{2}\left(\frac{u - a_{i1}}{b_{i1}}\right)^2\right],$$

to represent the linguistic terms, where

$$\{a_{i1}, a_{i2}, b_{i1}, b_{i2}\},$$

is the *parameter set*.

As the values of these parameters change, the bell-shaped functions vary accordingly, thus exhibiting various forms of membership functions on linguistic labels A_i and B_i.

In fact, any continuous, such as trapezoidal and triangular-shaped membership functions, are also quantified candidates for node functions in this layer.

The initial values of the parameters are set in such a way that the membership functions along each axis satisfy ϵ-completeness, normality and convexity.

The parameters are then tuned with a descent-type method.

- **Layer 2** Each node generates a signal corresponing to the conjuctive combination of individual degrees of match.

All nodes in this layer is labeled by T, because we can choose any t-norm for modeling the logical *and* operator.

The nodes of this layer are called *rule nodes*.

We take the linear combination of the firing strengths of the rules at *Layer 3* and apply a sigmoidal function at *Layer 4* to calculate the degree of belonging to a certain class.

If we are given the training set

$$\{(x^k, y^k),\ k = 1, \dots, K\}$$

where x^k refers to the k-th input pattern and

$$y^k = \begin{cases} (1,0)^T & \text{if } x^k \text{ belongs to Class 1} \\ (0,1)^T & \text{if } x^k \text{ belongs to Class 2} \end{cases}$$

then the parameters of the hybrid neural net (which determine the shape of the membership functions of the premises) can be learned by descent-type methods.

This architecture and learning procedure is called ANFIS (adaptive-network-based fuzzy inference system) by Jang.

The error function for pattern k can be defined by

$$E_k = \frac{1}{2}\Big[(o_1^k - y_1^k)^2 + (o_2^k - y_2^k)^2\Big]$$

where y^k is the desired output and o^k is the computed output by the hybrid neural net.