Working Paper 95-14
Statistics and Econometrics Series 03
April 1995

Departamento de Estadística y Econometría
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (341) 624-9849

PROBABILISTIC AND FUZZY REASONING IN SIMPLE LEARNING CLASSIFIER SYSTEMS

Jorge Muruzábal*

Abstract

This paper is concerned with the general stimulus-response problem as addressed by a variety of simple learning classifier systems (CSs). We suggest a theoretical model from which the assessment of uncertainty emerges as primary concern. A number of representation schemes borrowing from fuzzy logic theory are reviewed, and some connections with a well-known neural architecture revisited. In pursuit of the uncertainty measuring goal, usage of explicit probability distributions in the action part of classifiers is advocated. Some ideas supporting the design of a hybrid system incorporating bayesian learning on top of the CS basic algorithm are sketched.

Key Words
Prediction, Bayesian Learning, Fuzzy Logic, Uncertainty Measuring.

*Departamento de Estadística y Econometría, Universidad Carlos III de Madrid.

1 Introduction

Learning in classifier systems (CSs) can be pursued in two broad classes of problems. One distinctive feature of these classes is the nature of the assumed relationship between consecutive input vectors (or states of the world) in the data stream, say z(t) and z(t+1). In the most general case, the distribution of z(t+1) depends on both z(t) and the particular action(s), say u(t), taken by the system upon observation of z(t). An often encountered case further assumes that z(t+1) is completely determined by z(t) and u(t). The problem faced by the CS here is essentially a *control* problem: given z(t) -- and possibly the messages posted in the internal memory or message list --, the system must learn to activate the particular u(t) that renders either direct reward or, more typically, some "profitable" state z(t+1). State z(t+1) is profitable in that it makes it possible for the system to visit a series of states leading to reward, yet this reward may be unreachable if a different u(t) is chosen. Hence, this framework brings up the issue of sensible credit apportionment, that is, the problem of promoting rules that "set the stage" but do not attain reward directly.

On the other hand, in *stimulus-response* (S-R) problems, the sequence of input vectors Z={z(1), z(2), ...} is assumed *exchangeable*, that is, the distribution of Z is the same regardless of the order in which the z(i) are arranged. Alternatively, given the knowledge of the generating probability distribution F, the components of a finite segment Z(t)={z(1), z(2), ..., z(t)} are assumed conditionally independent with identical distribution F. Thus, in contrast with the control problem, here z(t) is informative about z(t+1) only in that, together with all previous z(t-1), z(t-2), ... , it helps to mitigate the uncertainty about F. Since the message list is so much rooted on the idea of recency and this is a virtually meaningless notion in exchangeable sequences, it is not obvious how to incorporate a useful message list in this context. Also, because there are no stage-setting rules, the credit apportionment problem is much simpler than in control problems. For these reasons, CSs addressing the S-R problem are easier to analyze and usually constitute the first testbed for developing ideas.

In this paper, we focus on some basic dimensions to the general S-R problem. In this problem, input z is split into two components -- referred to as stimulus and response respectively, say z=(x,y) -- and the CS must solve a *prediction* problem: for any given x, it must learn to provide a good approximation to the associated y. For simplicity, univariate responses are assumed throughout. If the response is categorical, the formulation corresponds to a standard classification problem. If the response is continuous, the analysis is usually carried out in the context of function approximation.

Traditionally, stimuli have been represented as binary strings of fixed length. Often the stimulus vector is made up of several features or *predictors*. While continuous predictors can be encoded as substrings and concatenated together with discrete predictors, it is well-known that hiperplanes are simply not well suited to function approximation problems (the resulting representation becoming too fragmented). A number of alternative schemes borrowing from fuzzy logic theory have been proposed to deal with this situation. These schemes either use bit strings to encode membership functions or handle membership functions directly in both the condition and action part of classifiers, and some successes have been reported in problems of moderate complexity, (Valenzuela-Rendón, 1991), (Parodi and Bonelli, 1993), (Carse and Fogarty, 1994). Learning is accomplished via the standard CS mechanisms. However, an alternative architecture, also based on fuzzy reasoning but exhibiting *monotonic* learning, has proved superior to a CS in a larger classification problem, (Carpenter, Grossberg, Markuzon, Reynolds and Rosen, 1992). Moreover, the radial basis function neural model has been recently shown to bear some resemblance with the fuzzy CS approach and may bring about yet additional learning algorithms (Jang and Sun, 1993).

It appears then that further evaluation of the relative merit of these representation schemes and learning algorithms is much needed, especially in connection with the type of problem being faced. For example, the analysis of large masses of data exhibiting only "weak" (or stochastic) regularities and lots of noise seems of foremost interest, if only because appropriate tools are scarce -- see (Muruzábal and Muñoz, 1994) for a simulation study. It is our working hypothesis that CSs have a great potential in this area

-- addressing pattern extraction objectives as class identification, dependency analysis or deviation detection in a highly robust way, (Matheus, Chan and Piatetsky-Shapiro, 1993) --, yet, in order to achieve the best efficiency, the basic algorithm probably needs to be enhanced. Some of such possible enhancements in the S-R context are the subject of this paper: while we focus throughout on representation issues, we will also consider briefly the cooperation with another learning procedure and the addition of a new triggered rule discovery mechanism.

The organization is as follows. Section 2 discusses the general S-R problem. Section 3 reviews some fuzzy CSs. Section 4 is concerned with probabilistic ideas. Finally, section 5 summarizes the material and proposes some directions for research.

2 The general S-R problem

All CSs carry out induction by evolving a population of predictive rules called classifiers $\{c_j\}$. All classifiers have the same basic structure: "IF Q_j, THEN R_j (WITH STRENGTH S_j)". The scalar quantity S_j summarizes the classifier's previous success. While a variety of choices are available for the Q and R substructures, the performance subsystem is always the same: a classifier is activated when an incoming x(t) "excites" condition or category Q_j, suggesting R_j as "expected response" or prediction. Beliefs R_j from activated classifiers are then combined in some way -- observing the respective strengths -- to yield the system's overall prediction R_+.

The learning system can also be described succinctly. Reinforcement proceeds at each step on the basis of the actual response y(t), with the result that those rules currently providing the best predictions tend to raise their strength. In addition, new classifiers (based on the genetic and other heuristic algorithms) replace low performance rules in the population. Such mechanisms act either periodically or in a steady-state fashion or when some triggering conditions are met.

To elaborate further into the nature of the problem, write the underlying joint distribution F=F(x,y) as the product of the marginal distribution G(x) and the conditional distribution H(y/x). Different S-R problems follow from assuming different G and H distributions as follows. In function approximation, for example, G is usually uniform and all H are deterministic, that is, the same response is observed whenever the same stimulus is presented (Wilson, 1987), (Bonelli, Parodi, Sen and Wilson, 1990), (Valenzuela-Rendón, 1991), (Parodi and Bonelli, 1993). While the various systems involved in these works solve the problems they are confronted with, the question typically remains about scale-up factors. For a discussion of the efficiency of S-R CSs in boolean function learning problems, see (Liepins and Wang, 1991).

An interesting first variation occurs when the H distributions are still degenerate but G is no longer uniform, that is, some stimuli show up less often than others. Unless special care is taken, classifiers "attending" to the atypical data will tend to be overwhelmed by those attending to the mainstream data, simply because the latter are rewarded more often and hence have a better chance to proliferate. Ultimately, of course, the best design decision will depend on the importance of predicting such atypical data correctly. In general, the issue of implicit niching in CSs is a delicate one which has begun to be rigorously approached only recently, (Horn, Goldberg and Deb, 1994).

In many applications, interest is often paid to the case where, in addition to nonuniform G, H distributions are no longer degenerate: two identical stimuli may have different responses -- see (Bonelli and Parodi, 1991) for an example in a medical diagnosis problem. We then say that the CS faces an *uncertain* problem. This uncertainty may be inherent to the problem, or it may be partially explained by one or more "responsible" predictors left out of the stimulus.

On the other hand, it is often reasonable to assume that responses associated to certain stimuli present all the same stochastic behavior (some predictors may be simply irrelevant). For example, in the car insurance business, Siebes (1994) analyzes the problem of finding groups of clients that share the same probability of filing a claim in a given time period. The important point is that we are now interested in finding useful partitions in stimulus space, reproducing the training data perfectly does not make sense. Specifically, the implicit assumption is that there exist disjoint X_r (r=1,..., L) such that, whenever $x \in X_r$, $H(y/x)=H(y/X_r)$; in other words, there are only L distinct patterns of response (or *niches*) covering the entire sampling space. We sometimes refer to the partition $\{X_r\}$ as being *sufficient* in this sense. In the binary response problem just discussed, each of the $H(y/X_r)$ corresponds to a different Bernouilli parameter, say p_r. This distribution reflects in general the best prediction that can be attached to the various stimuli lying in X_r.

Uncertain problems and uncertainty measuring are of course not new in CS research. Goldberg (1990) analyzes the behavior of a simple CS when L=1, responses follow independent Bernouilli distributions with parameter p, and the system is endowed with just two classifiers (predicting 0 and 1 respectively). It turns out that the CS exhibits *probability matching* behavior, that is, the system predicts 1 about 100p% of the time. In this sense, CSs measure uncertainty *implicitly* (via the strengths). Related ideas can be found in (Holland, Holyoak, Nisbett and Thagard, 1986; section 6.2.2). In this paper, we will be concerned instead with *explicit* representations of uncertainty.

It may well be the case that not all stimuli have predictable responses, that is, some of the $H(y/X_r)$ may be relatively flat (or uninformative); if the response were binary, that would be the case when some of the p_r are 1/2. Naturally, we would like to bias the system so that the informative components are favored. Also, as mentioned earlier, some stimuli may occur rarely. If such stimuli can be safely ignored, the primary task of the CS facing an uncertain problem consists of organizing a population of classifiers such that all regions in stimulus space that have relatively high probability under G *and* exhibit exploitable regularities (in terms of relatively sharp H distributions) are simultaneously covered. Once such regions are reasonably approximated, the system must also determine as closely as possible the quantitative aspects of such regularities.

Depending on the particular representation scheme used, it may be possible to capture each regularity with a *single* classifier, but this seems too demanding in general. In other words, each of the informative X_r must be approximated by (a subset of) the available categories Q_j, and each of the associated distributions $H(y/X_r)$ must be approximated in turn by (a subset of) the available R_j. Thus, besides the obvious coupling between the Q_j and the R_j, *weak* cooperation must also occur amongst classifiers, (Horn et al., 1994). The task is of course complicated by the fact that some useful Q_j may not be reinforced appropriately because they are coupled to the wrong R_j. As discussed below, this problem may be alleviated to some extent by allowing classifiers to be modified dynamically (beyond the usual strength revision process) by the data they "filter out".

A source of theoretical complication relates to identifiability of the various (informative) H distributions. For an illustration of these complications, see eg. (Siebes, 1994). It is clear, however, that the number of niches is of no direct concern to the work of the CSs, so no special attention needs to be paid to this point until we are in a position to prove theorems about their behavior.

Many inductive systems other than CSs work on the basis of these or similar assumptions. Tree-oriented systems like CART, ID3 and their offspring, for example, build up a strict partition in essentially a top-down way, for which they require the entire data set at hand. In the next sections we consider some CS-based alternatives that attempt to approximate the "true" partition in a bottom-up way without having to record all data. We will also examine the extent to which these systems can provide useful summaries of uncertainty.

3 Fuzzy knowledge structures in S-R classifier systems

In this section we examine the fuzzy approach to CSs in the S-R problem. We do not consider learning mechanisms so much as we focus on representation issues and performance subsystems: how is information represented and how is used to provide predictions. While the following original work addresses the multivariate case, we assume $x \in R^n$ and $y \in R$ as mentioned earlier. We review first the work of Parodi and Bonelli (1993), hereafter P&B.

P&B propose the following fuzzy CS for learning a functional relationship between x and y. Each classifier c_j consists of n+1 membership functions (mapping R into the closed unit interval). The first n, say m_{ji}, i=1,2, ..., n, form the condition part or fuzzy category Q_j; the (n+1)th function, m_{jo} say, constitutes the action part or fuzzy prediction R_j, j=1,...,J. All m_{ji} share the same basic shape, so they are determined by their respective centers $c_{ji} \in R^n$ and widths $w_{ji}>0$, i=0,1,...,n. They can have, for example, the familiar gaussian form

$$m_{ji}(v)=\exp\left\{-\frac{(v-c_{ji})^2}{w_{ji}^2}\right\};$$

P&B actually use symmetric triangular functions. An input vector $x=(x_1,...,x_n)$ "excites" classifier j to the extent given by the fuzzy AND operator applied to its condition part; P&B use

$$e_j=\min_{i=1,\dots,n} m_{ji}(x_i);$$

the alternative

$$e_j^*=\prod_{i=1,\dots,n} m_{ji}(x_i);$$

is sometimes considered. Each classifier produces an output function o_j defined as $o_j = e_j\, m_{jo}$; the system's overall output function is computed as the weighted average of such functions

$$o=\sum_j S_j o_j.$$

A point prediction is obtained by simply taking the centroid of o (other mechanisms are possible as well, see below).

Jang and Sun (1993) note that, under certain conditions, fuzzy inference systems like P&B's fuzzy CS are functionally equivalent to a simple radial basis function (RBF) network (Poggio and Girosi, 1989), namely, a network consisting of J receptive neurons, all of which receive input x and feed forward to a single output neuron. The output of receptive neuron j is computed according to a multivariate gaussian

$$\varepsilon_j = \exp\left\{-\|x-\chi_j\|^2/\omega_j^2\right\} = \exp\left\{-(x-\chi_j)' D(\omega_j)^{-1}(x-\chi_j)\right\},$$

where χ_j and ω_j are the neuron's parameters, and $D(\omega_j)$ is the diagonal matrix with entries the squares of the ω_j. In a simple implementation, the network's output is computed as the weighted sum

$$o = \sum_j \varepsilon_j f_j,$$

where f_j is the (adaptive) function value associated with each receptive neuron. Parameters χ_j and ω_j are typically fixed beforehand using some clustering algorithm; simultaneous learning of the χ_j and the f_j is addressed more rarely.

P&B's fuzzy CS is then functionally equivalent to the simple RBF network provided (i) gaussian membership functions are used in the antecedent of classifiers; (ii) the excitation function is taken to be e*; (iii) all w_{ji} ($i=1,\ldots,n$) are the same for each j; and (iv) $m_{jo}(y)=c_{jo}$ for all y and for all j; (of course, all strengths S_j should be fixed and equal to 1). As noted by Jang and Sun, this equivalence makes it possible to exchange terminology, results or even learning algorithms between the two paradigms; we can talk freely, for example, of receptive fields when referring to the classifiers' antecedents.

Let us briefly comment on these conditions next. The first two clearly refer to choices to be made; criteria to guide these choices seem presently unclear. P&B suggest that gaussian membership functions may yield smoother fits to continuous mappings. Other authors consider yet different membership functions, see eg. (Valenzuela-Rendón, 1991). It is likely that performance is relatively robust under these options, although a systematic study may be in order. Expressing a preference between e and e* does not seem easy either. This choice can perhaps be related to the amount of noise anticipated in the data: function e* might seem more resistant to outliers in that a single "misplaced" coordinate in an input vector -- otherwise close to some receptive field center -- has the same effect on a classifier than a really distant vector.

The last two conditions do extend in principle the scope of the RBF network. Generalizing (iii) seems useful in that allows modelling of differently shaped receptive fields. In a similar fashion, replacing $D(\omega j)$ by an arbitrary positive definite matrix in ε_j goes one step further to allow for arbitrary correlation among input coordinates, yet it will be computationally prohibitive unless n is small. It is well known in cluster analysis that decorrelating the entire set of input vectors before learning is not helpful in general, so P&B's choice seems to strike a reasonable balance between complexity and flexibility. In their paper, however, they argue that the centers are the really decisive entities and keep the widths fixed, thereby moving closer to the generalized RBF approach discussed in (Poggio and Girosi, 1989; section 5.2). This simplification makes extension (iv) perhaps hardest to justify.

A related framework is proposed by Valenzuela-Rendón (1991). As P&B's fuzzy CS, his system also combines ideas from fuzzy logic and classifier systems. The main difference is that the set of membership functions that classifiers can use for each individual input (or output) coordinate is restricted to the set of arbitrary fuzzy OR functions (defined by simply replacing "min" with "max" in the right hand side of e_j above) that can be constructed from a finite "basis" of M prespecified membership functions (whose peaks are equispaced along the range of that coordinate). For example, when M=5, condition (10001) represents a bimodal function yielding high excitation when the associated input coordinate presents either very low or very high values. This is a potentially interesting possibility not available in P&B's system. Unfortunately, Valenzuela-Rendón does not discuss whether his fuzzy CS does indeed use bimodal conditions on input variables when it would be economical to do so -- he tests out, for example, the symmetric map $y=4(x-.5)^2$.

As regards the output, each classifier posts a (fuzzy) message identical to the string in its consequent part. Each of such messages carries along an activity level which depends on both the excitation value (computed via the e function as discussed above) and the strength of the classifier that posted it. The system's output is produced by first transforming the previous set of messages into a set of equivalent *minimal* messages (ie., strings that contain a single "1"), whose activity levels are obtained by adding up the activity levels of the original messages that "contain" them. A fuzzy OR operation is then performed on the set of basic membership functions multiplied by their associated activity levels. *Crisp* output is obtained as the centroid of the resulting function.

As an aside, Furuhashi, Nakaoka and Uchikawa (1994) build on Valenzuela-Rendón's representation but only allow minimal conditions for each input coordinate. They tackle a control problem in a novel way by connecting several fuzzy CS in series, that is, the output of one system becomes (part of) the input to the next system; only the last system is directly involved with decision making. They study the behavior of the joint system when information is transferred from CS to CS via both fuzzy messages and crisp values. Their results suggest that defuzzification is required to achieve learning.

Note that, in contrast with the standard CS representation, neither of the above systems allows for wildcard characters in the syntax of their classifiers' receptive fields. However, were the widths of membership functions allow to evolve, wildcards could be implicitly implemented via diverging widths.

4 The role of probability distributions in uncertain S-R problems

In section 2 we argued that uncertain S-R problems differ markedly from the more common function approximation ones in that there is no necessarily a single best point prediction and we are interested instead in accurate summaries of uncertainty. In section 3 we discussed some proposals incorporating fuzzy logic ideas into the S-R CS architecture. These proposals treat the condition and action part of classifiers in exactly the same way, a decision for which no supporting arguments are provided. In our view, fuzzy logic seems quite useful in the antecedent part of classifiers: it allows for sensible handling of continuous data, introducing an easily interpretable notion of partial matching. In contrast, fuzzy sets may not be the best choice at the right side of the arrow. An obvious question refers to the precise meaning of membership functions when looked at as predictive statements. More specifically, what quantitative sense can we make out of their widths with regard to prediction? At least in uncertain problems, probability distributions may have an advantage over fuzzy predictions on the following grounds:

- They constitute a readily interpretable measure of uncertainty from which a variety of conclusions can be extracted. For example, in clinical trials we are not only interested in the average survival time but also in the probability of survival beyond certain point. Similarly, we can construct predictive intervals with given coverage probability reflecting faithfully the underlying tail behavior. Because membership functions are not easily interpretable in probabilistic terms, it is not clear how to draw similar inferences from fuzzy predictions.

- There exists a vast catalogue of probabilistic results and techniques that can be brought into play to our advantage. For example, we will discuss below how coherent, on-line updating of beliefs can be accommodated as an additional learning mechanism in CSs.

We briefly review now a prototypical system, called PASS, incorporating probability distributions R_j in classifiers, (Muruzábal, 1993). Because this system has been so far primarily concerned with boolean stimuli, it sticks to the traditional schema-based representation of receptive fields, which means that classifiers either are or are not excited by any given input. In addition, a subset of *winning* classifiers is selected among those excited -- an idea emanating from the original interest in *default hierarchies*, (Riolo, 1989). With this exception, the performance subsystem is basically the same as above: the system's overall predictive distribution R_+ is obtained as a mixture of the selected distributions R_j weighted by their respective strengths.

How are such distributions represented? Reported experiments involve discrete distributions spreading mass over a "small", connected subset of the response range. For example, if 10 bits were used to depict this range, a prediction would be encoded as two structures, namely, a binary vector of length 10 with at most k *contiguous* ones (selecting k equispaced subintervals as the *support* of the distribution), and a set of k associated probabilities (k≤4, say, is a system parameter). This representation is related to Valenzuela-Rendón's approach discussed earlier, yet the contiguity restriction implies in effect that each classifier can only describe a unimodal distribution. Unimodal distributions mean no constrain in principle because they might be combined to yield bimodal predictions when needed. However, Muruzábal points out that the required speciation and cooperation phenomena do not occur frequently in practice, so more general families of distributions may be needed.

Another peculiarity of this system is the ability to dynamically modify individual classifiers as data are processed. Following the spirit of the original CS architecture, most systems rely on strength updating and rule recombination as sole learning mechanisms. It is sometimes argued, however, that CSs may greatly benefit from a more active processing of the data stream. In PASS, two related, non-standard procedures have been implemented so far. They are briefly described as follows.

The first procedure updates the set of probabilities comprising a given R_j. At the outset, each of these distributions is initialized as the uniform distribution over its support (whose location is itself initialized at random and remains fixed throughout). Any incoming response -- whose stimulus excited the classifier -- may or may not fall on this support. When it does, the probability of the particular subinterval containing the response is incremented slightly. In the long run, the resulting set of probabilities reflects the (truncated) distribution of the responses witnessed by the classifier. Hence, a useful receptive field Q_j enjoys some time to have its prediction corrected. This effect would be more important if the support were allowed to evolve or were equal to the entire response range (see below).

The second procedure enables an individual classifier to temporarily remember data which proved it wrong (that is, the stimulus satisfied the schema, but the response fell outside the support). This form of mid-term memory is implemented as a (small) buffer attached to each classifier. When the buffer is filled up, a special-purpose operator is triggered to scan its contents in search of additional regularities. Were these detected, new classifiers depicting them would be injected into the population -- see (Muruzábal, 1993) for details. In any case, the list is emptied after each call. While this operator signifies an additional burden on the system, it seems very helpful in preliminary experiments. For example, using 6-bit precision Gray code for stimuli, PASS can solve (a stochastic version of) the y=x problem with 40 classifiers in about 2,000 trials; this compares very favorably with the results reported in (Valenzuela-Rendón, 1991) and (Parodi and Bonelli, 1993).

Departing slightly from Muruzábal's work, but continuing along the same line, assume now that the conditional distributions $H(y/X_r)$ can be approximated by (mixtures of) densities of some fixed parametric form $\Phi(y/\theta)$, with θ in some Θ. When viewed as a function of θ, $\Phi(y/\theta)$ is called the likelihood function; typically, the support of Φ will be the entire response range. Once "correct" approximating schemata are found, the system faces the problem of inferring the best parameter choice, say θ_j, in each case. The multinomial family is an immediate generalization of Muruzábal's discrete case, although we will use a continuous family for illustration. As originally suggested by Lane (1992), the idea of updating the classifier's predictive distribution can be implemented in a fully coherent way by adopting a bayesian approach. To do this, we only need to specify a prior distribution for each θ_j, say $\pi_j(\theta)$, and set up the equation describing the transition from prior to posterior, namely

$$\pi_j(\theta / y) \propto \Phi(y / \theta)\pi_j(\theta).$$

This posterior summarizes our uncertainty about θ_j after each datum is received. The process is repeated *iteratively* -- the posterior after the ith response playing the role of prior for the next response --, and *in parallel*, but of course classifiers are exposed to different responses. The distribution R_j after data $D_j=\{y_1, y_2, \dots, y_d\}$ have been processed is given by

$$R_j(y)=\int \Phi(y / \theta)\, \pi_j(\theta / D_j) d\theta.$$

It is acknowledged that these calculations may be too demanding in general. The usefulness of the approach relies on the existence of suitable choices for likelihood and prior from which closed-form formulae can be derived and "hard-wired" into the system. For example, if we assume Φ to be a gaussian distribution with mean μ and variance τ, and if the usual (non-informative) prior $\pi(\theta)=\pi(\mu,\tau) \propto 1/\tau$ is chosen, then the posterior given D_j is

$$\pi_j(\theta / D_j) \propto \tau^{-d/2-1} \exp\left\{-\frac{1}{2\tau}\left[SS_y + d(\bar{y}-\mu)^2\right]\right\},$$

where $SS_y = \sum_{i=1}^{d}(y_i - \bar{y})^2$, and $\bar{y} = \frac{1}{d}\sum_{i=1}^{d} y_i$. Upon integration, the predictive distribution is well-known to be of t type, so the system's overall prediction R_+ would be a weighted mixture of t distributions.

In practice, once the likelihood has been decided upon, the analysis will often be based on its *conjugate* family, that is, a family such that, if the prior is a member thereof, so is the posterior (and simple updating formulae exist for the hyperparameters). In the example above, it is easy to show that the conjugate family is precisely the gaussian-inverted-gamma family whose basic form is given by $\pi_j(\theta/D_j)$ above. Of course, the initial prior is eventually overwhelmed by the data, so the particular member chosen to get the process started is in no way crucial. When an increasingly sharp posterior obtains as d increases, we can alternatively consider predictive distributions of the form $R_j(y) = \Phi(y / \hat{\theta})$, where $\hat{\theta}$ is, for example, the posterior mean.

Note the *local* character of the present bayesian mechanism. It is in contrast to other proposals which use bayesian ideas to *globally* organize the knowledge base (the minimum description length principle, for example, admits an interpretation of this sort). Aside from computational complexity issues, global organization ideas may seem too far away from the CS spirit.

Finally, note also that the previously discussed idea of remembering a few data pairs can also be incorporated into this formulation. For each excited classifier, define the *score* of any given response y* as the value $R_j(y^*)$. Then, an observation would be deemed an exception (worth remembering) whenever its score fell below certain threshold. Incidentally, $R_j(y^*)$ should prove a useful quantity in the assignment of individual classifier reward.

5 Summary and concluding remarks

Learning in uncertain S-R problems may be accelerated by letting data "mold" classifiers while the standard learning procedures act on the population. This is an attempt to increase the power of CSs by providing classifiers with additional processing abilities. It is not intended to downplay the importance of triggered rule discovery in CSs (Booker, 1989); in fact, both sets of ideas are expected to coexist and benefit each other.

We have reviewed some fuzzy CSs and discussed their connection with the RBF neural model. The learning task faced by fuzzy CSs is harder than that typically taken up in the neural setting. This suggests that some molding process of the envisaged sort may be crucial to achieve successful learning in a reasonable time span. It is proposed that probability distributions can be treated in general in a way akin to fuzzy sets in the consequent part of classifiers, providing also a potentially more useful interpretation. In addition, to the extent that bayesian updating can be blended in at a reasonable computational cost, probability distributions allow for an interesting implementation of the suggested refinement. An empirical test of these ideas is called for.

We conclude with some extensions and open questions. Probability distributions and local bayesian learning can also be considered in principle when the response is multivariate, although the increase in memory and computational load should be monitored carefully . It would be interesting to determine whether certain dependencies among response components can be learned probabilistically, for the current form of the fuzzy representation presents a number of limitations in the multivariate case, cf. (Parodi and Bonelli, 1993). Once again, the tradeoff between accuracy and computational effort needs to be analyzed in depth.

Tough competition to the CS-based family of inductive systems is coming also from next door!: some recent advances in genetic programming (Gathercole and Ross, 1994) stress the fact that CSs currently lack the ability to create new predictors and, therefore, may be handicapped in problems involving relatively complex patterns of interaction among predictors. Future CSs should perhaps try to make their job easier by incorporating operators that propose tentative transformations. Can we do any better than the random strategy found in genetic programming?

Finally, how should we proceed when we have predictors of both categorical and continuous type? Can we paste boolean and fuzzy representations in a graceful way?

Acknowledgements

Financial support from DGICYT grant PB92-0246 (Spain) and CICYT grant TIC93-0702-C02-02 (Spain) is gratefully acknowledged.

References

Bonelli, P. and Parodi, A. (1991). An efficient classifier system and its experimental comparison with two representative learning methods on three medical domains. In R. K. Belew and L. B. Booker (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kauffman, CA.

Bonelli, P., Parodi, A., Sen, S. and Wilson, S. W. (1990). NEWBOOLE: A fast GBML system. Proceedings of the Seventh International Conference on Machine Learning, Morgan Kauffman, San Mateo, CA.

Booker, L. B. (1989). Triggered rule discovery in classifier systems. In J. D. Schaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kauffman, CA, pp. 265-274.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H. and Rosen, D. B. (1992). FUZZY ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Networks, Vol. 3 (5), pp. 698-713.

Carse, B. and Fogarty, T. C. (1994). A fuzzy classifier system using the Pittsburgh approach. In Y. Davidor, H.-P. Schwefel and R. Manner (Eds.), Parallel Problem Soving from Nature III, Springer-Verlag Lecture Notes in Computer Science, Vol. 866, pp. 260-269.

Furuhashi, T., Nakaoka, K. and Uchikawa, Y. (1994). Suppression of excessive fuzziness using multiple fuzzy classifier systems. Proceedings of FUZZ-IEEE'94, World Congress on Computational Intelligence, pp. 411-414.

Gathercole, C. and Ross, P. (1994) Dynamic training subset selection for supervised learning in genetic programming. In Y. Davidor, H.-P. Schwefel and R. Manner (Eds.) Parallel Problem Soving from Nature III, Springer-Verlag Lecture Notes in Computer Science, Vol. 866, pp. 312-321.

Goldberg, D. E. (1990). Probability matching, the magnitude of reinforcement, and classifier system bidding. Machine Learning, Vol. 5 (4), pp. 407-425.

Holland, J. H., Holyoak, K. J., Nisbett, R. E. and Thagard, P. R. (1986). Induction: processes of inference, learning and discovery. MIT Press, Cambridge, MA.

Horn, J., Goldberg, D. E. and Deb, K. (1994). Implicit niching in a learning classifier system: nature's way. Evolutionary Computation, Vol. 2 (1), pp. 37-66.

Lane, D. A. (1992). Personal communication.

Liepins, G. E. and Wang, L. A. (1991) Classifier system learning of boolean concepts. In R. K. Belew and L. B. Booker (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kauffman, CA.

Jang, J.-S. R. and Sun, C.-T. (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. IEEE Transactions on Neural Networks, Vol. 4 (1), pp. 156-159.

Matheus, C. J., Chan, P. K. and Piatetsky-Shapiro, G. (1993). Systems for knowledge discovery in databases. IEEE Transactions in Knowledge and Data Engineering, Vol. 5 (6), (Special issue on Learning and Discovery in Knowledge-Based Databases).

Muruzábal, J. (1993). PASS: a simple classifier system for data analysis. Tech. Rep. 93-20, Statistics and Econometrics Dept., University Carlos III, Madrid, Spain.

Muruzábal, J. and Muñoz, A. (1994). Diffuse pattern learning with fuzzy ARTMAP and PASS. In Y. Davidor, H.-P. Schwefel and R. Manner (Eds.), Parallel Problem Soving from Nature III, Springer-Verlag Lecture Notes in Computer Science, Vol. 866, pp. 376-385.

Parodi, A. and Bonelli, P. (1993). A new approach to fuzzy classifier systems. In S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kauffman, CA.

Poggio, T. and Girosi, F. (1989). A theory of networks for approximation and learning. A. I. Memo 1140, M. I. T., Boston, MA.

Riolo, R. L. (1989). The emergence of default hierarchies in learning classifier systems. In J. D. Schaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kauffman, CA.

Siebes, A. (1994). Homogeneous discoveries contain no surprises: inferring risk-profiles from large databases. Report CS-R9430, CWI, Amsterdam.

Valenzuela-Rendón, M. (1991). The fuzzy classifier system: a classifier system for continuously varying variables. In R. K. Belew and L. B. Booker (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kauffman, CA.

Wilson, S. W. (1987). Classifier Systems and the Animat Problem. Machine Learning, Vol. 2., pp. 199-228.