Juan A. Barceló

Heuristic classification and fuzzy sets. New tools for archaeological typologies

1 Introduction: from sherds to pots

Although Classification Theory has a long history in archaeology, sherd fitting has always formed an unsolved problem. Determining form from part of a vessel is limited by the fact that potters made vessels for different purposes starting with a few basic shapes. Since potters work by combining standard elements — base, bodies, rims, handles and so on — it is not always possible to infer the complete form from the fragments present in a deposit, because rims and bases of similar size and shape might actually have come from vessels of differing size and shape (cf. Montanari/Mignoni 1994; Orton *et al.* 1993). If one is trying to study pottery forms using only sherd material, then the definite absence of certain features may become as important a point to record as their presence. The usual assumption that all attributes have equal importance is wrong in that case. Therefore, we cannot describe different shapes distinguishing the individual aspects that determine relevant attributes for each aspect of the complex, because not all attributes are present in the sherd; 'relevance' cannot be computed when a part of the required information is missing.

2 The 'brittleness' problem

To classify a pot as a member of a type can be seen as a formal proof of the expression: 'pot *a* is member of Type *A*'

As logical proof we use the mechanism called logical implication. Suppose we have 5 attributes to determine the shape of Type *A* vessels. The logical implication needed to fit any sherd to the shape is:

```
IF      object i has
                attribute 1
        AND     attribute 2
        AND     attribute 3
        AND     attribute 4
        AND     attribute 5
THEN
        object i has shape Type A.
```

Let us call this rule 'proof *P*'. Archaeological descriptions (attributes) are elements of *P* because they are used in the proof. An element of proof, such as *attribute 5* (for example, ORIENTATION OF PROFILE) may have any number of instances (for example: ORIENTATION OF PROFILE = 30°, 45°, 90°, 180°, etc.). However, an element must have only one instance in each proof. When we are dealing with a fragment, and information about that element of proof is lacking, we assign a MISSING instance to that attribute.

Suppose only 2 attributes have been measured in the sherd. Following formal *modus ponens* this production rule cannot be fired; object *i* cannot be assigned to shape Type *A*. If we consider items one through five to be of equal importance, and we have to delete attribute 3 to 5 (because only attributes 1 and 2 are present in the sherd), the typology would malfunction and sherds are not classified because they do not present enough descriptive information.

This problem can be defined as the *brittleness problem*, that is, the inability of standard typologies to give a 'partial answer' in a graceful way (Sypniewski 1994). The cause of brittleness in typologies and classificatory systems is the use of an inadequate assumption about data. If we assume all necessary truths to express the idea of logical necessity are equally important to a proof, we are saying, in effect, that unless we can demonstrate all necessary truths we cannot prove what we are trying to prove. This is the problem of brittleness.

To solve the problem we can consider that *any* element of *P* can be used in the proof. We do not require all attributes but only the *necessary* elements of *P* to be present in the sherd. No reason exists why we cannot use the accidental elements of *P* in the proof, but they cannot substitute for one or more missing necessary attributes. This scenario provides a first glimpse into the definition of *importance*: Some elements of *P*, while legitimate members of *P* do not contribute to the actual proof (they are missing in the sherd). If we remove all members of *P* that are accidents or are unnecessary for *P*, we are left with P^1, which is composed of the necessary elements of *P*; all of them contribute to the proof. The theory of importance (Sypniewski 1994: 26) says that not all members of P^1 necessarily contribute to the proof process in the same way or to the same extent. The extent of that contribution is demonstrated by the importance weight of every attribute or element (E_i). Any E_i that has a larger importance weight than an E_j is more important to a particular *P* than E_j. An element of proof that is irrelevant has an importance weight of 0.0; the same value has an attribute with missing value.

It is important to realise that no item of data has an intrinsic importance weight. All weights are relative to some *P*. Also note that a particular situation may provide elements whose combined importance weights exceed 1.0. In those cases more data are available than is strictly necessary for a proof.

The degree to which an attribute contributes to prove a typological assignment is determined empirically. When we gather the data or knowledge we need for our classification, we will, as a by-product, gather information about the elements of a proof. If we introduce this material into a matrix, we will see that some bits of information fill one cell of the matrix and some bits fill more than one cell. The number of cells filled with a particular piece of data or knowledge is a rough gauge of the importance of that particular piece of data or knowledge. As a general rule, *the more often a particular piece of data or knowledge appears in our hypothetical grid or matrix, the less important it is* (Sypniewski 1994: 29). We can say that if two proofs differ only by one item of data or knowledge, then this piece of knowledge is the most important item for that proof.

Consequently, a strong importance weight is equivalent to a branch point in a decision tree.

3 Fuzzy Logic: a way to solve the problem of 'brittleness'

Starting from the idea that every sherd is a certain proportion of the whole pot it once formed part of, we can (in theory) assign a weight or importance to attributes, and compute them to obtain a class assignation. In this chapter we will study how to describe importance weights through fuzzy numbers, and how to translate classification functions as membership function to fuzzy sets (Bezdek/Pal 1992; Cox 1993; Dubois *et al.* 1994; Klir/Folger 1988; Kosko 1992; Zadeh 1965).

Fuzzy logic deals with uncertainty. It holds that all things are matters of degree. It measures the degree to which an event occurs, not whether it occurs. Mathematically fuzziness means multivaluedness or multivalence and stems from the Heisenberg position-momentum uncertainty principle in quantum mechanics. Multivalued fuzziness corresponds to degrees of indeterminacy or ambiguity, partial occurrence of events or relations. In 1965 Lofti Zadeh introduced the concept of *fuzzy set*, as a way to represent the logical nature of categories. Fuzzy sets are constituted by elements, however those elements are not crisp instances of the categories but elements that belong only *to a certain degree*. The essence of fuzzy logic is then the notion of fuzzy membership as a continuous value measuring the elementhood or degree to which element *x* belongs to set *A*.

We can translate logical implications (proof of classificatory assignments) using fuzzy production rules, where the output of the rules is a fuzzy set, whose members are the elements of the proof. Each element, as a member of a fuzzy set, has a fuzzy membership value or importance weight. For instance,

IF	object *i* 's PROFILE is concave	(0.875)
	object *i* 's RIM has shape *B*	(0.358)
	object *i* 's MAX. DIAMETER is on top of the pot	(0.47)
THEN		
	object *i* has shape *Type A*	

The values in the rule's antecedent are *fuzzy*, because they belong to a fuzzy set. This value is not the confidence we have in that information, but the importance this element of a proof has in type *A*'s logical implication. To evaluate these rules, fuzzy logic software computes the degree to which each rule's situation applies. The rule is active to the degree that its IF part is true; this in turn determines the degree to which each THEN part applies. Since multiple rules can be active simultaneously, all of the active rules are combined to create the final result. At each cycle, the full set of logical implications is scanned to see which fires. A rule or logical implication will fire when its condition made up of a (fuzzy) logical combination of its antecedents, results in a non zero value. Each rule therefore samples its inputs and calculates the truth value of its condition from the individual importance weight of each input. In this way, the fuzzy membership function of each element acts as a kind of restriction or constraint on the classification process.

Let us imagine that *P*, a proof for a classificatory assignment, is a set. Then *P* = {attribute 1, attribute 2, attribute 3, attribute 4, attribute 5}, where each attribute or descriptive feature are the elements of proof needed to prove *P* (for example, to prove *Type A*). We can assume that *P* is a fuzzy set, and consequently, each element has a membership value. Given the fact that *P* is fuzzy, the membership value for each element is a continuous number between 0 and 1, meaning the importance weight of that attribute in the logical implication described by *P*. In this case, fuzziness is only a general methodology to compute the sum of partial implications. I do not think that archaeological types have to be intrinsically *fuzzy*, but the sherd fitting process will only be computed if that type is described in a fuzzy way: if we do not know how an instance relates with its type, the relationship remains fuzzy. Inferences made using incidental associations ('always' in archaeological classification) are inherently uncertain. And some associations are 'less' uncertain than others.

Fuzzy logic permits ambiguous instances to be included in a fuzzy set through a membership value. The degree of membership is given by the membership function, which has a value between 0 and 1. The interpretations is that 0 means no membership (or that the instance is certainly not in the set) and 1 denotes complete membership (or that the instance is certainly in the set), and a value in between denotes a partial or uncertain membership. Fuzzy logic thus overcomes a major weakness of crisp sets: they do not have an arbitrarily established boundary separating members from non members.

Fuzzy systems directly encode structured knowledge but in a numerical framework, where each rule stands for an input-output transformation, where inputs are the antecedent of fuzzy rules, and outputs are their consequent. In our case, inputs are the descriptive features we can measure on sherds, and outputs are an assignation of the sherd to an artefact or class of artefacts. Most fuzzy systems represent inputs and outputs as membership functions whose interactions are the bases for rules. The fuzzy input and desired output ranges are based on fuzzy set values and used to create a matrix called *fuzzy associative memory*. When actual input values enter the system, the entire memory fires at once, producing multiple outputs. Each input's membership in the fuzzy input sets must be calculated — this is called the truth value or importance weight. The information from all inputs is then applied to the rule base, which results, for each system output, in several fuzzy outputs. Since system inputs have multiple fuzzy values and each can be involved in the triggering of multiple rules, since each rule can have several fuzzy input values for its antecedents and each rule also can produce several outputs, and since each output itself has multiple fuzzy values, this process becomes quite complex.

A *Fuzzy Cognitive Map* (FCM) is a special type of *fuzzy associative memory* where the variable concepts are represented by nodes, which can also be called *conceptual states*, and the interactions by the edges, or *causal events*. Consequently, FCMs model the world as a collection of classes and causal relations between classes. Each node is a fuzzy set (fig. 1). In our case, logical implication between different elements of a proof is represented by fuzzy causal flows. The fuzzy cognitive map tries to represent the way a scientist thinks, because the nodes (concepts) affect each other, either directly or indirectly and either positively or negatively (Kosko 1986, 1992; McNeill/Thro 1994; Taber 1991).

The logical structure of an FCM allows each state (or node) to have any value between 1 and -1:

- +1 meaning that the originating or causing state results in a complete increase in the target or affected state;

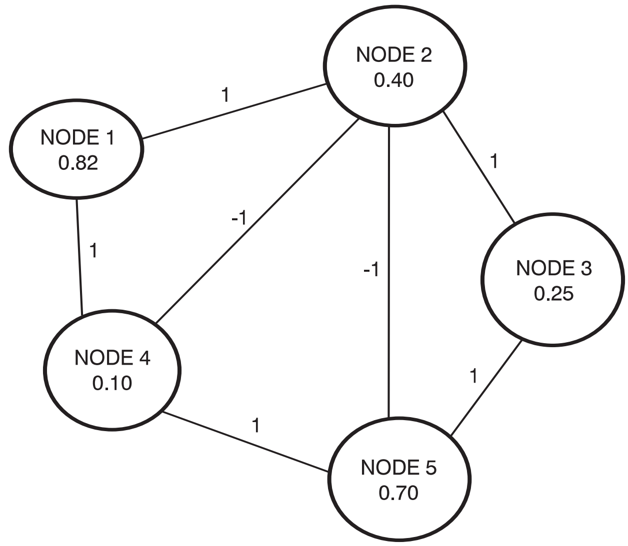

Figure 1. A Fuzzy Cognitive Map.

- -1 meaning that the causing state results in a complete decrease in the affected state;
- 0 meaning that the causing state does not change the affected state.

The number is the degree of causation and ranges from a negative one through zero to a positive one. Zero means no causal effect. Negative importance weights are used to say that some proof element instantiation tends to disprove or reduce the likelihood of a proof. Disproofs can be active or passive. To be an active disproof, the instantiation of some element of proof E_i must have an importance weight that is a negative number. Therefore, the system will subtract effectively the value of its importance weight from the current proof value V. A passive disproof, on the other hand, is simply a proof element that is not available (MISSING), and because it has not been observed, it is never added to V.

Disproofs can be calculated using a formula for fuzzy entropy:

$$\frac{\text{degree of overlap between every pair of outputs}}{\text{degree of underlap between every pair of outputs}}$$

Overlap is the result of logical intersection between types, whereas the *underlap* can be defined as the union between them (Kosko 1992; McNeill/Thro 1994).

As in a neural network, each state or node is 'squashed' through an activation function. In other words, each state value is a modification from the previous value during each forward step of the dynamic map. Each state's value is the result of taking all the event weights pointing into the state, multiplying each by the causing state's value, and adding up all the results of these multiplications. The results are then squashed so that the result is between 0 and 1 (0 and 100%). This multiply- and sum-process is a linear operation; that is, the new activation value for a fuzzy set (output node) is a weighted sum of all membership values for that set. If the unit's input is less than some threshold level (0.00 in our case), then the new activation value is equal to that unit's minimum activation range (also 0.00). Otherwise, if the inputs are positive (greater than the threshold 0.00), then the new activation value is set equal to the inputs.

FCM nodes act as binary neurones in a neural net. They sum, weight and threshold the causal energy that flows into them through the fuzzy causal edges. The states in an FCM are *state machines*, that is, they receive some input from somewhere (other units in the network), use it, change and 'export' a value. Given the fact that these states are linked in a graph, each one receiving unique inputs from other states, changing as a result, and affecting some other states. Time is a component of this architecture, because dynamic action continues as long as one state is able to effect a change in another one. This function (or *gain*) determines the high and low values of a cycle and can affect the map's operation. The higher the gain is, the more exaggerated the cycle.

Before activation all elements for all the possible proofs in the system are zero because none of them is active. You begin with a static diagram of the system. It shows the assumptions of the model. Then you set up an initial condition and perform iterated vector-matrix multiplication until a limit cycle is found. The limit cycle shows how the system behaves. In other words, vector matrix multiplication changes the state to something else. What we get as a result is a classification assignment.

4 PYGMALION: using Fuzzy Logic to classify Phoenician pottery

PYGMALION is the code name for a joint project, currently under way at the Universitat Pompeu Fabra Dept. of Humanities and the Universitat Autònoma de Barcelona Dept. of Prehistory. The goal is to create a computer system able to classify Phoenician pottery (800-550 BC), and to derive chronologies, production characteristics and exchange networks from descriptive features of archaeological material. PYGMALION release 0.1 is a prototype version to study the logical properties of the full-scale Expert System (PYGMALION release 1.0). This prototype is a Fuzzy Cognitive Map acting as a pattern recognition machine for pottery sherds.

The process of recognising a pattern is the classification of a sample into one or more predefined categories. If the pattern is successfully associated with a previously known type, the pattern is said to be recognised. At the end, the system should provide a confidence estimate in the classification; for example, the system is 75% confident that this sherd is part of a Type *A* pot and 25% confident that it is a type *B*. This confidence estimate is a measure of the degree to which the pattern-recognition system believes that the pattern data belongs to the specified class. To carry out this task, PYGMALION is implemented as a graph with evaluated nodes and evaluated arcs that represent relational structures among types. The aim is to decide whether the reality represented by a sherd qualitative description matches prior knowledge about the whole pot incorporated into the graphical model.

4.1 Describing shape

Defining the shape of an object can prove to be very difficult. Pottery shape is influenced by a large number of factors. The decisions made by the potter, the tools and materials available and his/her skill in manipulating them all contribute to the finished product. While many practical shape description methods exist, there is no generally accepted methodology of shape description. The principal disadvantage of most pottery shape description systems is that they cannot be applied to the sherd material which forms the majority of the pottery recovered from archaeological sites (see amongst others Kampffmeyer *et al.* 1988; Orton *et al*. 1993; Rice 1987).

We have designed a new 'qualitative' descriptive framework, based on modern theory of robot vision (Biederman 1987; Saund 1992; Sonka *et al*. 1993).

Representation of visual shape can be formulated to employ knowledge about the geometric structures common with specific shape domains. We seek representations making explicit *many* geometric properties and spatial relationships at many levels of abstraction. Therefore, the problem of visual shape representation is to determine what information about objects' shapes should be made explicit in order to classify sherds as parts of whole pots. Knowledge about the pottery making process can be built into a shape representation in the form of a descriptive vocabulary making explicit the important spatial events and geometrical relationships comprising an object's shape.

The decomposition approach is based on the idea that shape recognition is a hierarchical process. Shape *primitives* are defined at the lower level, primitives being the simplest elements which form the region. Then, an object's shape will be analysed largely in terms of the spatial arrangement of labelled chunks or fragments of shape. A decomposition of the contour, for instance, uses its structural properties, and a syntactic graph description is the result. This graph is constructed at the higher level–nodes result from primitives, arc describes the mutual primitive relations. Particular shape fragments are labelled by individual *shape tokens* instantiated in the appropriate type. Each token is tagged with the characteristics (location, orientation and size) of the archaeological item it denotes. That is to say, a shape is described simply in terms of *primitive-edge* tokens placed along the bounding contour at the finest scale.

We are working with a contour-based object description method which uses as input information the properties of object boundaries. The contour or border of an object is the set of pixels within the region that have one or more neighbours outside that object. In other words, the contour or profile is the set of points at the limit of the object. We are dealing with partial segmentation looking for non-disjoint subregions in the contour. That is, the existing border is divided into separate regions that are homogeneous with respect to a chosen property. *Curvature* is that property. As profiles are a continuous series of pixels, curvature can be defined as the rate of change of slope. The curvature scalar descriptor (or boundary straightness) finds the ratio between the total number of boundary pixels (length) and the number of boundary pixels where the boundary direction changes significantly. The smaller the number of direction changes, the straighter the boundary. Contour primitives are delimited by the gradient of the image function that is computed as the difference between pixels in some neighbourhood. The evaluation algorithm (not fully implemented in PYGMALION 0.1) is based on the detection of angles between line segments positioned by boundary pixels in both directions (fig. 2).

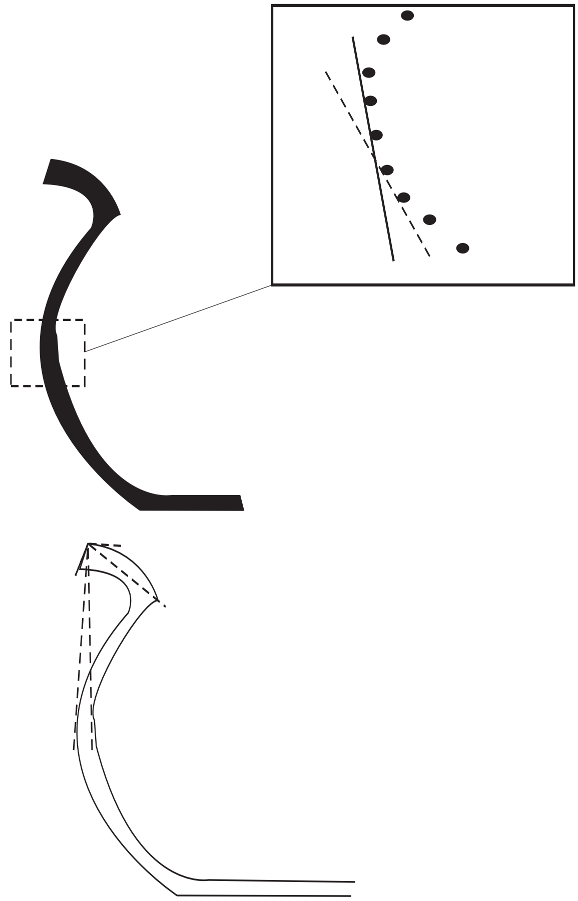

Figure 2. Describing the curvature of a contour.

Consequently, we are representing a boundary using segments with specified properties. If the segment type is known for all segments, the boundary can be described as a chain of segment types. The problem lies in determining the location of boundary vertices. Boundary vertices can be detected as boundary points with a significant change of boundary direction using the curvature (boundary straightness) criterion.

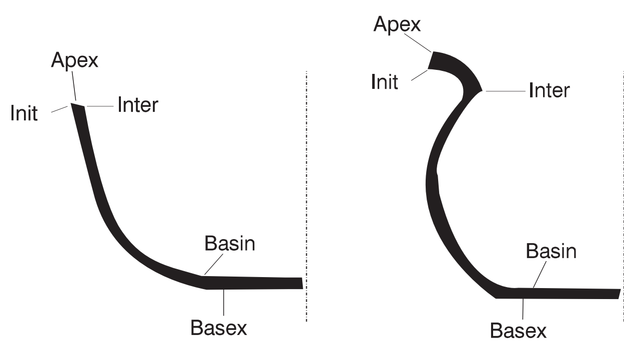

Figure 3. Main points for dividing contour into segment.

Once segmented, contour parts can be described qualitatively. Our approach is based upon the psychological theory by I. Biederman (Biederman 1987). He proposes to use only four qualitative features in describing objects. We have translated his ideas into the following components:

- CONTOUR: straight or curved
- CURVATURE: convex or concave
- COMPLEXITY: number of contour primitives
- ORIENTATION: in a 8-neighbourhood area.

The prior knowledge we have about the contour of a pot allows us to know the starting point (INIT) and the ending point (BASEX) of the border (fig. 3). The process is then a decomposition of the external *and* the internal profile in its primitive curves. We begin describing the *body* of all whole pots we know by looking at the right and the left profile and determining symmetry or asymmetry. The *exterior profile* and *interior profile* are described by detecting the number of 'curvatures'; they can be continuous or discontinuous. *Exterior and interior discontinuity* is defined by counting the number of profile primitives after detecting more than one 'curvature'. We consider also the *shape of exterior and interior profile* (straight, concave or convex) if there is only one curve; or the *exterior and interior profile primitives shape* if discontinuity is present. The *location* (the place where curvatures have been measured) of all profile primitives is also a very useful attribute: between rim and body, at the centre of the body, at the centre of the rim, etc. The *maximum diameter location* (in the upper part of the pot, at the centre, at the rim etc.) helps to distinguish some types; and the same is true for some different descriptions of *orientation: external profile orientation, internal profile orientation, rim orientation*. Finally, *rim shape* (geometric form from APEX to INTER) is evaluated. All orientations are calculated according to an 8-neighbourhood window (fig. 4).

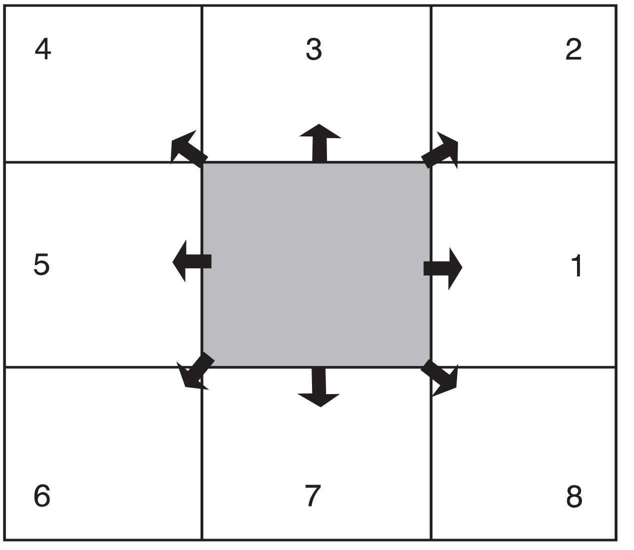

Figure 4. A schema to fix orientations of contour segments.

4.2 BUILDING A *FUZZY COGNITIVE MAP*

Any object, even with non-regular shape, can be represented by a collection of its topological components. Topological data structures describe the pot as a set of elements and their relations. These relations are represented using a Fuzzy Cognitive Map, containing the object structure. The elementary properties of syntactically described objects are called primitives; these primitives represent parts of contours with a specific shape. After each primitive has been assigned a symbol, relations between primitives in the object are described, and a relational structure results. However, given the indeterminacy of PYGMALION inputs (incomplete pots) we have decided not to use arcs representing binary relations such as *adjacent to, to the left of, above,* etc., but a fuzzy cognitive map where arcs represent the importance weight of primitives (nodes).

The actual version of our program is a continuous-state model, because every node may have any value between 0 and 1. A negative weight (between -1 and 0) means the element is a disproof for some particular type. This value is less than the unit's threshold, consequently, the goal of negative weights is only to deactivate units previously activated.

PYGMALION 0.1 contains 54 units or nodes. 36 of these are input nodes and represent qualitative information introduced by the user; 18 nodes represent the answer of the system, or the outputs of the classification (fig. 5). Input nodes are connected among themselves using negative weights. That is to say, there are relationships between elements of different fuzzy sets. There is only a single membership link between every element (attribute) and the fuzzy set (type) it belongs to. Negative links also connect fuzzy set units (output nodes) among them, because some sherds cannot be part of two very different shapes. Negative links among output units represent the degree of overlapping *allowed* by the classificatory system. For instance, there is no degree of overlapping between a carinated bowl and an amphora; however the degree of overlapping between two different kinds of plates can be very high. In the prototype presented here, all output units are linked by the same negative weight: the maximum activation level for a unit (1.00) divided between the number of competing units.

Causal energy flows synchronously between elements and sets (fig. 6). That means that there is no control of rigid timing signals. Instead, each element (node) in PYGMALION sends fuzzy membership values and importance weights as it is ready (as there is input information for it). As long as the element and the type are set up to send and recognise the right combination, the membership message will get through. In general, the signal being sent from one node to another is equal to the activation value of the first node multiplied by the weight from the first to the second. The FCM performs these computations every time the network is *cycled*. When we cycle the network, we give each input node some information from an archaeological description. Given the fact that we are processing sherds, not all inputs are activated. The aim is to obtain a degree of activation on the output nodes, even though input activation is incomplete. Negative links and asynchronous updating help in this process. The program stops when all activations have been distributed around the network.

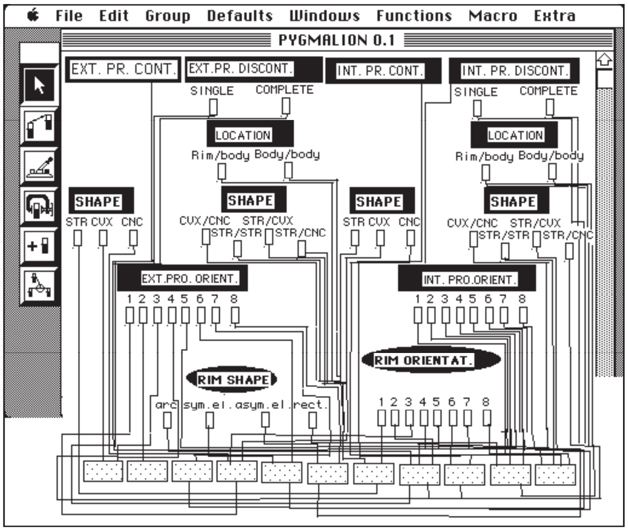

Figure 5. An ideal representation of PYGMALION Cognitive Graph.

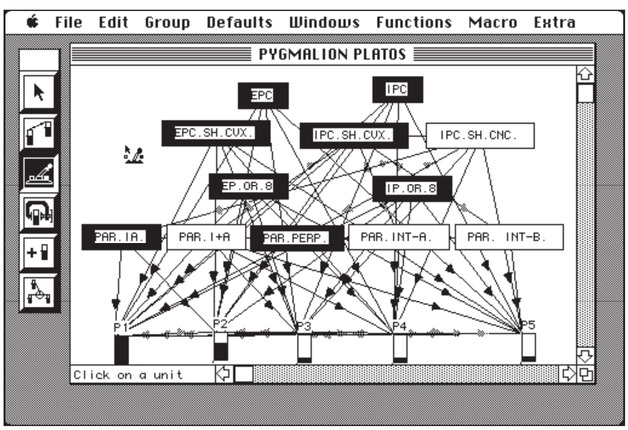

Figure 6. A subset of PYGMALION 0.1 Fuzzy Cognitive Map.

4.3 Determining importance weights

The importance weights assigned to a particular item or data or knowledge are, in a sense, always relative. These numbers have to be *fuzzy* because not all elements of a proof will have a fixed importance weight. Therefore, the importance weight of all nodes in the FCM must be determined in relation to the elements of other proof sets. For instance it is more important to know the shape of a rim when distinguishing between a bowl and a plate than if we were distinguishing between a bowl and an amphora.

The importance weight of any item of data or knowledge has been calculated from empirical evidence and its environment. By environment, we mean the number and nature of the proofs possible in the specific domain of discourse in which the data are being used. To determine the weight of an item of data, we must determine whether it does not depend on any other item of data. However, in any collection of data patterns, some individual data items will appear in more than one data pattern. This enables us to say that one data pattern is similar or dissimilar to another depending on how many individual data they have in common. The more they share, the more similar they are and vice versa.

Then, their importance weight can by calculated by the ratio of their non-occurrence in all fuzzy sets.

- if an element *attribute* x_i of the proof set *Type A* completely proves *Type A* without condition, then *attribute* x_i has an importance weight of 1.0; otherwise *attribute* x_i has an importance weight less than 1.0 under all circumstances.
- for every *attribute* x_i that is shared by all conflicting proof sets *Type N* , *attribute* x_i has an importance weight of 0.0; otherwise *attribute* x_i has an importance weight greater than 0.0 under all circumstances.

Weights have floating-point values such as 0.5, 0.2, 0.9 according to their importance weights in a proof. Those values have been calculated dividing the number of types with that feature between all types in the classification. We are doing some experiments with more complex measures of 'importance', such as *entropy*. Table 1 shows a subset of importance weights between the elements of the proof (descriptive features) and axioms to be proved (Types). The first row shows the effect of attribute 1 on all types. The second row shows the effect of the second attribute on the types. The matrix is square, since we have a place for the effect of each attribute on all types.

Table 2 shows negative weights or disproofs among elements. They have been calculated from the degree of overlapping among fuzzy sets. For instance, a continuous convex shaped internal contour appears sometimes in pots with coincident parametric points; however, it is impossible to see a continuous convex shaped internal contour with the INTER parametric point below the INIT parametric point. Coincidences have been tabulated as a 0.00, and discrepancies as a -1.

The Fuzzy Cognitive map topology can be described using a set of specific variables (see figs 7a, b).

4.4 Using the Fuzzy cognitive map to identify incomplete pots

PYGMALION 0.1 is being used to validate a typology for Phoenician open forms. We have worked with a data set of nearly 200 whole pots from Phoenician sites in the southern Iberian Peninsula (mostly Toscanos, Trayamar, Almuñecar, Cerro del Villar). Validation is carried out by comparing the

Table 1. Fuzzy membership values between elements and fuzzy sets.
Abbreviations: E.- external; I.- Internal; P.- Profile; C.- Continuous; SH.- Shape; STR.- Straight; CVX.- Convex; CNC.- Concave; ORIENT.- Orientation; PAR.- Parametric Points. IA.- Inter = Apex. I+A.- Inter ≠ Apex. PERP.- perpendicular to Init; INT-A.- Inter above Init; INT-B.- Inter below Init.

	P1	P2	P3	P4	P5
EPC	.2	.2	.2	.2	.2
IPC	.2	.2	.2	.2	.2
EPC.SH.CVX	.2	.2	.2	.2	.2
IPC.SH.CVX	.5	.5	0	0	0
IPC.SH.CNC	0	0	.3	.3	.3
EP.OR.8	.2	.2	.2	.2	.2
IP.OR.8	2	.2	.2	.2	.2
PAR. IA	.5	.5	0	0	0
par. I+A	0	0	.3	.3	.3
PAR. PERP.	1	0	0	0	0
PAR. INIT-A	0	0	1	1	0
PAR. INIT-B	0	0	0	0	1

Table 2. Negative Weights among elements (descriptive features).
Abbreviations: E.- external; I.- Internal; P.- Profile; C.- Continuous; SH.- Shape; STR.- Straight; CVX.- Convex; CNC.- Concave; ORIENT.- Orientation; PAR.- Parametric Points. IA.- Inter = Apex. I+A.- Inter ≠ Apex. PERP.- perpendicular to Init; INT-A.- Inter above Init; INT-B.- Inter below Init.

	IPC.SH.CVX	IPC.SH.CNC	PAR.IA.	PAR.I+A	PAR.PERP.	PAR.INT-A	PAR.INT-B
IPC.SH.CVX	0	-1	0	-1	0	-1	-1
IPC.SH.CNC	-1	0	-1	0	-1	0	0
PAR. IA	0	-1	0	-1	0	-1	-1
PAR. I+A	-1	0	-1	0	-1	0	0
PAR.PERP.	0	-1	0	-1	0	-1	-1
PAR.INT-A	-1	0	-1	0	-1	0	-1
PAR.INT-B	-1	0	-1	0	-1	-1	0

classificatory assignments made by the program to assignments made by experienced archaeologists.

Once we confirm the quality of the *answers* proposed by our *automatic archaeologist*, we will begin introducing descriptions for incomplete pots. PYGMALION 0.1 then computes partial membership functions and proposes a fuzzy assignment. Of course, *natural archaeologists* have to use these assignments and decide what has more sense, that a sherd be 55% of Form 3 or 45% of Form 13.

5 The concept of heuristic classification

When using fuzzy logic tools to build classification systems, we have proceeded through identifiable phases of data abstraction, heuristic mapping onto a hierarchy of pre-enumerated solutions, and refinement within this hierarchy. We have obtained a classification, but with the important twist of relating concepts in different classification hierarchies by non-hierarchical, uncertain inferences. This combination of reasoning has been called *heuristic classification* (Clancey 1985). The heuristic classification model builds on the idea that categorisation is not based on purely essential features, but rather is primarily based on *heuristic*, non-hierarchical, *but direct* associations between concepts.

Heuristic classification is a method of computation, not a kind of problem-to-be solved. In other words, it is a way to solve an archaeological problem (sherd fitting to form) and not a new philosophy about archaeological classification. We must not confuse what gets selected at the end of the FCM — what constitutes a solution — with the method for computing the solution. A common misconception is that there is a kind of problem called a ‘classification problem’. Heuristic classification as defined by W. Clancey (1985) is a *description of how a particular problem is solved by a particular problem-solver*. If the problem solver has a priori knowledge of solutions and can relate them to the problem description by data abstraction, heuristic association, and refinement, then the problem can be solved by classification.

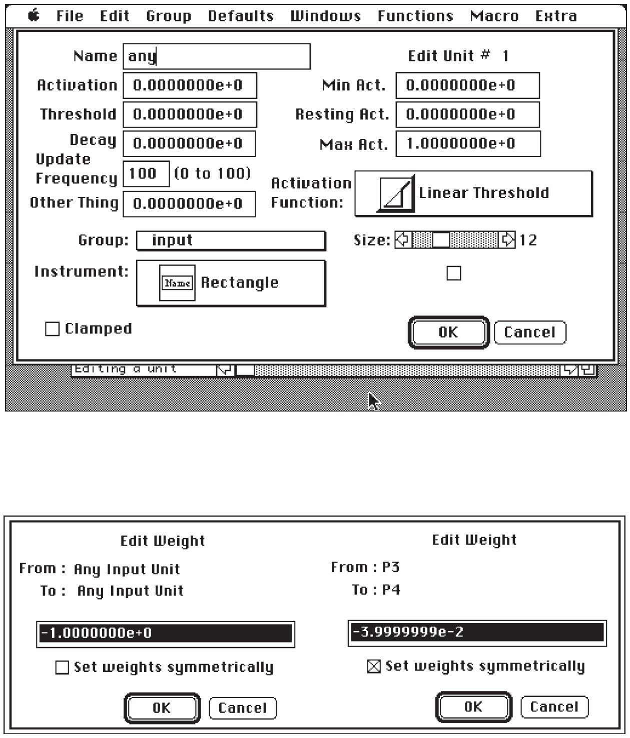

Figure 7. a. An FCM node's topology. b. Two different FCM weights' topology.

Often problems of classification are not amenable to solution by heuristic classifications because possible final states cannot be practically enumerated, exhaustively learned or for some reason a previously used solution is just not acceptable; solutions must be constructed rather than selected. However, even when solutions are constructed, classification might play a role.

In this paper we have described what an expert system does by describing it in terms of inference-structure diagrams (Fuzzy Cognitive Maps). This demonstrates that it is highly advantageous to describe systems in terms of their configuration, *structurally*, providing dimensions for comparison. A structural map of systems reveals similar relations among components, even though the components and/or their attributes may differ.

Acknowledgements

I would like to thank M.E. Aubet, P. Báscones, E. Curià, A. Delgado, A. Fernandez, M. Párraga and A. Ruiz for their work on the Guadalhorce Project, and for suggestions and encouragement. Although the work on a General Tipometry for Phoenician Pottery is a collective effort by all members of the Guadalhorce Project, none is responsible for any mistake I may have committed in this paper.

This research has been supported by the Ministerio Español de Educación y Ciencia (DGYCIT, PB93-0397).

references

Bezdek, J.C., S.K. Pal — 1992 — *Fuzzy systems for Pattern Recognition*. Piscataway (NJ): IEE Press.

Biederman, I. — 1987 — Recognition by component: A Theory of Human Image Understanding, *Psychological Review* 94 (2), 115-147.

Clancey, W.J. — 1985 — Heuristic Classification, *Artificial Intelligence* 27, 289-350.

Cox, E. — 1993 — *The Fuzzy Systems Handbook. A practitioner's guide to building, using and maintaining fuzzy systems*. Cambridge (MA): Academic Press.

Dubois, D., H. Prade, D. Smets — 1994 — Partial truth is not uncertainty. Fuzzy logic versus possibilistic logic, *IEE Expert. Intelligent Systems and their applications* 9 (4), 15-19.

Kampffmeyer, U., P. Zamperoni, W.R. Teegen, I. Graça 1988 *Untersuchungen zur rechnergestutzen Klassifikation der Form von Keramik*. Frankfurt: Peter Lang Verlag.

Klir, G.J., T.A. Folger 1988 *Fuzzy Sets, Uncertainty and Information*. Englewood Cliffs (NJ): Prentice Hall.

Kosko, B. 1986 Fuzzy Cognitive Maps, *International Journal of Man-Machine Studies* 24, 65-75.

1992 *Neural Networks and Fuzzy Systems. A dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs (NJ): Prentice Hall.

McNeill, F.M., E. Thro 1994 *Fuzzy Logic. A practical approach*. Boston: Academic Press.

Montanari, A., S. Mignoni 1994 Notes on the bias of dissimilarity indices for incomplete datasets. The case of archaeological classification, *Questió. Quaderns d'Estadística i Investigació Operativa* 18 (1), 39-49.

Orton, C., P. Tyers, E.A. Vince 1993 *Pottery in Archaeology*. Cambridge: Cambridge University Press.

Rice, P. 1987 *Pottery Analysis. A source book*. Chicago: The University of Chicago Press.

Saund, E. 1992 Putting knowledge into a visual shape representation, *Artificial Intelligence* 54, 71-119.

Sonka, M., V. Hlavac, R. Boyle 1993 *Image Processing, Analysis and Machine Vision*. London: Chapman and Hall.

Sypniewski, B.P. 1994 The importance of being data, *AI-Expert* 9 (11), 23-31.

Taber, R. 1991 Knowledge processing with Fuzzy Cognitive Maps, *Expert Systems with Applications* 2 (1), 83-87.

Zadeh, L.A. 1965 Fuzzy Sets, *Information and Control* 8, 338-353.

Juan A. Barceló
Area de Prehistoria
Facultat de Lletres
Universitat Autonoma de Barcelona
08193 Bellaterra
Spain
e-mail: ilphd@blues.uab.es