Pattern recognition applied to Rock Art

Diego Diaz
Universidad de Buenos Aires
Nicolás Vila 470 21 C
Buenos Aires, Argentina 1421
Phone: +54-11-4381-5933 - E-mail: didibart@hotmail.com

Damián Castro
Universidad de Buenos Aires
17525 NE 40th Street Appt D320
Redmond, WA 98052, USA
Phone: +1-425-8618679 - E-mail: damianc@microsoft.com

Abstract: This paper was made in the context of a project that also develops a DA (database application), aimed at storing a large part of Argentinean rock art sites data.

This database is an analytical approach to that subject, and it is the second stage of the mentioned project, that began with a bibliographical and geographic data store. That software is currently in use and is part of the PROINDARA project that is supported by the University of Buenos Aires. The goal of this second stage, NA (neural application), is to add classification and tools to the database. Pattern recognition is among the capabilities that we are adding at the analysis stage.

Our conclusion is that, bearing in mind that the quality of our working pattern populations is very important, while we are storing images in the databases, we will also have to improve our aim to achieve good initial classifications.

Key words: Neural, Rock Art, Argentina, Database, Application

The database application

This presentation is an overview of our current works in Rock Art within the University of Buenos Aires PROINDARA project. We are working in two areas: Database Application DA and Neural Application NA. The first is concerned with the organization of the data and the second is concerned with meaning of that data.

The aim of the project is then, to provide informatic solutions that could fit the needs of the rock art researcher. The first stage was to develop a database system inside Database Application DA to load geographic information about rock art sites. This stage was called "descriptive database" and was focused on the political division of the world with tables like countries, states, counties, etc. to help the researcher to locate the rock art remains quickly and in situ, and to acquire the geographical information of the site. DA also has tables that can store coordinates to easily export this data to a GIS. This descriptive database also contains tables with information about authors and articles, that the user can link with the sites that are referred to in these articles.

At a second stage we are developing another database system of DA to load analytical information about the image. This stage is called "analytical database" and is focused on the description of the set of elements, in a formal and technical way. The contextual information is related to the images' position in the archaeological site. To analyze this data we use traditional queries (written in SQL language) and some tools taken from the IA like neural networks and data mining.

From an epistemological view, the computer can help anthropologists to do their jobs in two ways: the analytical and the synthetic. Firstly we consider the amazing computer ability to calculate in all of its aspects (mathematical and logical) and secondly, we consider the relatively new ability to make exploratory analyses and suggest new hypotheses. As regards the analytical aspect, our DA is not restricted to aristhotelical logic. And this is not trivial since very few people know about these other kinds of logics, even in the academic world. These logics are an output of the logic revolution of Frege, Russell, Whitehead, etc. at the beginning of the XX century. It was demonstrated, in Prof. Reynoso's work (tesis inédita), the capability of PROLOG to write his own rules in a metalanguage sense, or similarly, the ability of PROLOG to implement any kind of logic (like modal, deontic, fuzzy, etc.); and this was the key to prove the versatility of these logics in the adecuation to different kinds of anthropological problems. In the synthetic aspect we include tools like neural networks, as our NA, and simulation models among others. This tools deals with empirical aspects of the investigation, generating hypothesis or testing in virtual time different kinds of simulations. A great tip of the synthetic issue is the fact that the actual output is more important than the process which generates this output; in the sense that, for example, in simulation models, the rules may always be the same, but the output has a huge range of possibilities with minimal changes in the input data. The epistemological literature distinguished two stages in the scientific work: the discovery context and the justification context. The first is related with the process where the hypotheses arises and with the moment where the researcher interacts with the empirical data; the second is related with the validation of this empirical data and with the consequences that these facts bring to the theory that rules the research. The parallel between analytical aspect and justification context and synthetic aspect and discovery context is almost obvious. The inference we extract is that there are some informatic tools to help the scientific in the justification context and there are some others tools to help them in the discovery context. This is just an idea and should not be considered as the final say in the matter. This idea could bring a bit of order in the scientific work, but all of us know that reality is much more complex than the idea behind it.

The rock art area provides a fertile field for computer jobs. Between image management and geographic information the range is very wide. And in all of them the researcher can use a lot of tools in the epistemological sense. From an analytical perspective we develop a geographic and a bibliographic database system as a part of DA. Based on a relational model (model that has a wide spread in the scientific and commercial worlds) we design the different tables and the different relations between them to answere queries for the geographical location of the site and for bibliographic information related with the sites. The central unit in this model is the Sites table (Fig. 1).

The table stores the following information: Country, State, County, Site, Absolute Height, Localization (this is a memo field where stored data about certain references to locate the site like "to the west of Lechuza River", etc, and in future releases we expect to implement full text queries on it), Longitude, Latitude, Kind of Site (i.e., cavern, wall, exposed, etc.) and Image (this field relates with another table that stores the n images of the site that the researcher could access). We don't consider the ecological and spatial context in our original design by the great offer of the GIS available today so we focus on product portability. The universal measures we use, like Longitude and Latitude, the versatility of the product base (MSAccess '97) to import and export data in almost all kind of file format (including the most power relational database engine like SQLServer, Oracle, Informix, etc.) and the fact that we store the image reference, not the image itself in the database, (allowing the researcher to access the image without opening the database) garantees the application portability.

The Articles table (Fig. 2) stores the following information: Article, Author (this table relates with the AuthorsXArticles, a table that covers the large amount of relationships between Articles and Authors), Site (as the relationship between Sites and Articles are many as well, we have another table to cover it: ArticlesXSites), Article Title, Publication Title, Collection (i.e. if it is a journal), Edition, Edition Place, Edition Year and a Topographic Index (that help to locate in the physical library of the I.C.A. (Instituto de Ciencias Antroplógicas de la Universidad de Buenos Aires) the required Article). At this time the system stores just the name of the Article, not the article itself. In future releases we expect to load the article itself and let the archaeologist implement full text queries. To gain this objective we must export our product base (MSAccess '97) to a most powerful engine like SQLServer which in its last release has introduced this new feature.

The second stage of the DA, the "analytical database" (named in this way because it stores formal and technical information about the draws and not for its epistemological connotations) is focused in the Element; a single unit in the draw, i.e.: in a herd of Llamas, a single Llama. Another important table is the Element Set, i.e.: the herd of Llama, that keeps a foreign key relationship with Element and provides a range from 1 to n elements in the element set.

The Element table (Fig. 3) stores the following information: Site, Sector, Topographic Unit, Element Set, Element, Relative Position (with some site references like a crevice, panel, etc.), Sizes (of the element), Module, Attitude, Expression and Image. We have another set of tables related to Element where we load data about formal and technical attributes of each element. These tables are: General Formal Type, Master Formal Type, Detail Formal Type, Formal Type by Element, General Technique Type, Master Technique Type, Detail Technique Type, Technique Type by Element, where the user can create up to 3 levels of attributes of each type.

The Element Set table stores the following information: Site, Sector, Topographic Unit, Element Set, Relative Position, Image and Articulation (This field relates this table with ArticulationXElementSet where we keep the relationship between elements in the Element Set with regard to any kind of Articulation i.e.: opposite, rotation, etc.).

From a synthetic point of view we expect to implement in the future, data mining techniques to DA, to give the researcher a new source of hypothesis. This relatively new technique is part of a wider area known as Knowledge Discovery in Databases (KDD), that involves data warehousing and artificial learning among others. These tools focus on the evolution of data along the time or another continuous variable, extracting hidden knowledge from databases. These knowledge banks could be retrieved with standard techniques like those written in ANSI SQL, but it would be time-consuming to implement it with the standard techniques. The information extracted from database brings to archaeologists new configurations that can suggest new hypotheses.

The neural application

One of the sections of the project is the implementation of the neural application (NA) to make classifications of the image data from the different sites. This module is the first implementation of the synthetic methods mentioned earlier. Even though NA is partly dependent on the DA, in the sense that the images will primarily be located there, the interoperation procedures are not yet developed. At this experimental stage, we are feeding the neural application externally, mostly to simplify the testing process.

The NA has two main sections. On one hand, the problem section, which is in charge of administrating the pattern storage.

On the other hand the Neural Wizard that creates and trains the neural network. But there is an important interaction between them because depending on the type of network the user chooses, the information stored in the database will be interpreted differently by the program.

The problem section has a tree-like view to represent the hierarchy where the problem itself is represented by the root, with the predefined classes on the second level and the samples for each of these on the third level. When the program, after the training, identifies a pattern as belonging to a class, it puts it under the corresponding class node. There are slight changes in this behavior when the Kohhonen Learning vector is applied: all the patterns are reclassified and may change their affiliation to a class state. In fact, in future versions, we are thinking of using an "Unknown" system class to store the data before the Kohonen learning vector. The screen shot in figure 4 shows what the tree looks like.

Because the actual images are stored through DA, in NA only the binary black and white representation exists, and is stored as an array of bytes. It is expected to develop a color version in the future, though it represents an important trade off regarding the memory needed to represent color. However, each point in the image is represented by two bytes, so theoretically 65536 different values might be allocated. Hence, a color scale of this quantity at least, will be supported without losing performance or without needing additional memory.

The neural section of NA was designed as a wizard that guides the user through the processes of creating and training the network.

The first step is when the user decides what the name will be, to identify the set of classes and patterns in the database-referred above as the "problem"- and to select a physical place on the disk where the file will be stored. In this step, you can also select how big the pattern matrix will be.

The second step is to decide the type of the neural network. The alternatives are Back Propagation, Learning Vector and Probabilistic. Back Propagation and Probabilistic are both used to classify new patterns when a predefined set of classes with their sample patterns are already known. Learning Vector (the Knohonen network) is used when the classes are not known. In those cases the network classifies the patterns using a class quantity as a goal. It is expected that similar patterns will belong to the same class and the network may classify the pattern in less classes than the goal. It will never find more classes than that.

The options apply to the types. The compression network is used to reduce the sizes of the input matrix before the training begins. This feature was proposed by Sejnowski et al. in 1991. When Vanilla is selected, a normal back propagation algorithm is used. Momentum option applies a change to the algorithm: new weight changes of neuron connection in each separate training step do not totally affect the final weights. They will affect some percentage instead. In this way, we can prevent the "oscillation" of the network, where the class of last trained pattern is always the winner.

Bidimensional neighborhood applies to Learning Vector type and Normalized to the probabilistic. (Fig. 5)

The following step is the network topology (Fig. 6). Here, we must decide the quantity of layers and neurons. For the Learning Vector there are no internal layers and the output layer neurons is the maximum quantity of classes to classify. In the Probabilistic there is only one internal layer and is equal to the quantity of patterns. The Back Propagation network permits decide freely the internal layers and their neurons.

Finally, we have to decide the training parameters: how many cycles, how much error is allowed, the learning rate, etc. A cycle means a propagation through all the layers. The error is how far is the actual output from the correct one (of course, this is only valid for supervised learning)

The first experiments

We are working with one of the rock art sites included in the DA. The site and its analysis were presented in "Congreso Internacional de Arte Rupestre", Tarija, Bolivia September of 2000, by María Isabel Hernandez Llosas et al.

There is one concept we extract from the Methodology section of this paper in order to organize the training experiments of the neural network: element. This means that we are isolating the paintings from the context and we are not using the special units, even though they are relevant to the taxonomies that might be created after the training. But it would add an overhead to input matrix and we choose, for this very first approach, work only at the lowest level.

We trained our network using three classes of elements: abstract forms, zoomorphic and anthropomorphic. Figure 7 shows some patterns we used.

We used the Back Propagation method. We added a set of three patterns per class and trained the network with different combinations of cycles and learning rates.

One thing that happened from the beginning was the oscillation. The last class of the last pattern trained was always the winner. To avoid this situation, we decrease the learning rate. This means more training time. In the unique case we successfully recognize a pattern of the network, it took almost 24 hours of training.

Of course this was in made in a desktop machine and not in a two or more processors server, but we think that we can optimize this process in some way. There is no other alternative, in fact, because considering the increase in time when a pattern is added, a minimal quantity of them, like 50, may take weeks to be trained.

Future perspectives

There are several tasks that we have to make, besides the optimization of our code and the improvement of the application capabilities. The more important is the writing of our own neural procedures, optimized for this specific kind of problem. Even though the user interface and the database code were developed by our team, we used an external code library, which implemented the neural network algorithm. So it would give us more versatility if we can optimize the neural code for this specific problem.

Another important task is to implement genetic algorithms to search for the optimal network for a specific set of patterns. The genetic algorithm might be very powerful for this kind of optimization and may considerably reduce the learning time to successfully produce recognitions and classifications.

References

Hernández Llosa, María Isabel. 2000. "Los grabados de campo morado", Congreso Internacional de Arte Rupestre", Tarija, Bolivia

Hertz, John. 1991.*Introduction to the theory of neural computation*, Redwood City: Addison Wesley.

Figures

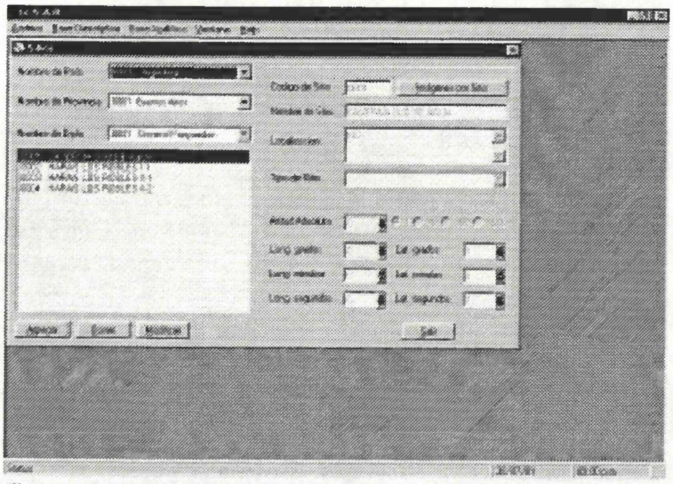

Figure 1. Sites table

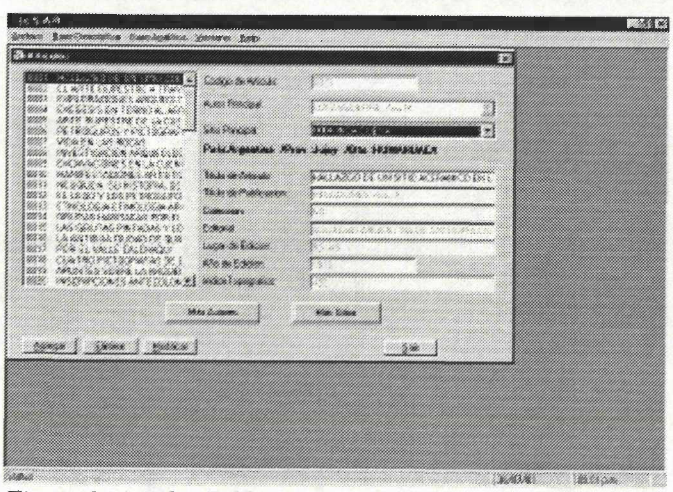

Figure 2. Articles Table

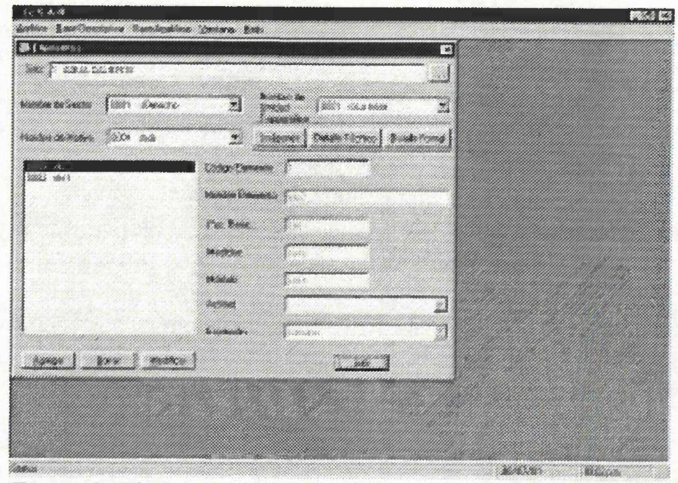

Figure 3. Element Table

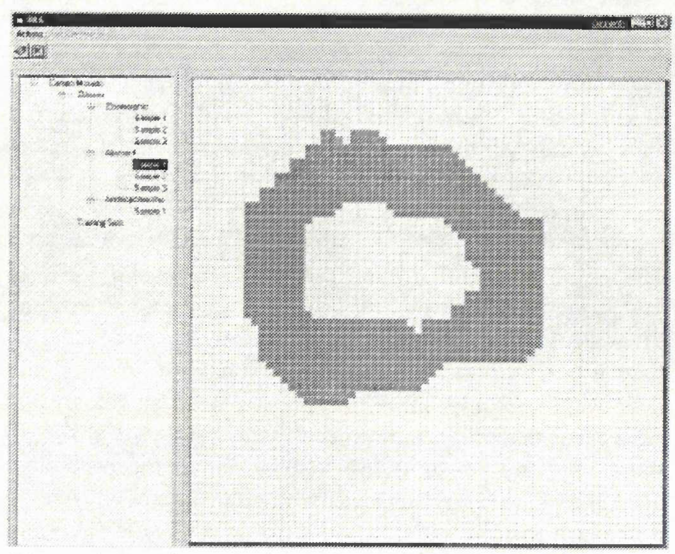

Figure 4. Problem window, where classes and patterns are defined

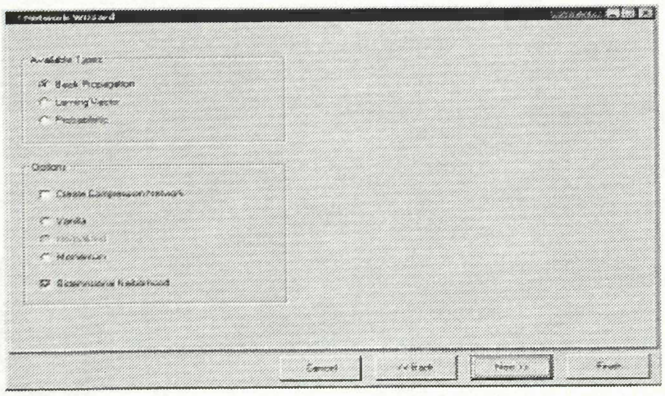

Figure 5. Second Step: neural network type

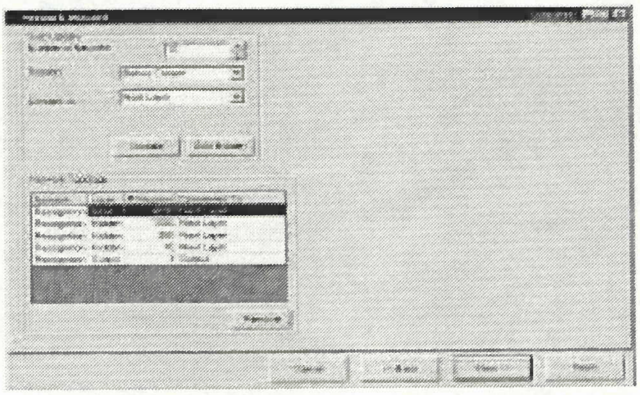

Figure 6. Neural network topology window

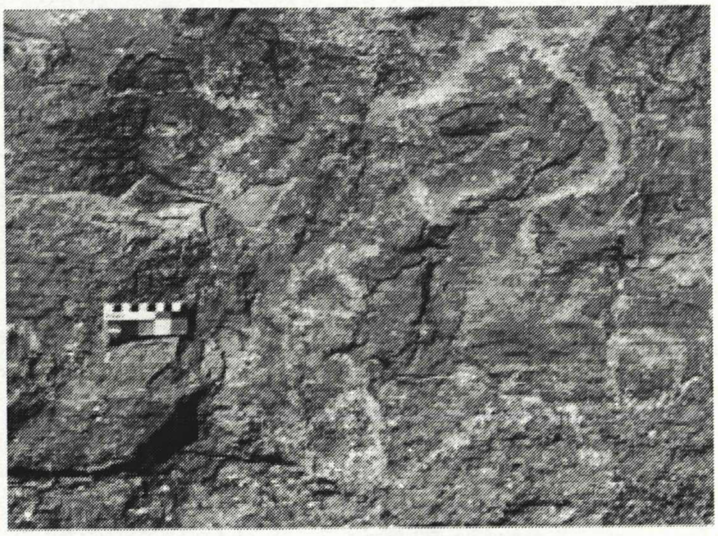

Figure 7. Compo Morado photograph (detail)