Market modelling · 11
Data mining models - 3

Chuck Chakrapani

Pattern finding
While tree modelling techniques have been widely known and used, there are other techniques of data mining that are less well-known. Pattern finding is one such technique. It uses principles of combinatorics and induction. As a result, it is-like cluster analytic procedures-not considered to be more a heuristic than a statistical technique. (Heuristic techniques are those that work on rules defined by the user and thus do not necessarily have a solution that is unique.)

Pattern finding works on the assumption that patterns in data are hidden. They can be analysed and made explicit. This is in fact a generalized assumption whenever we use any multivariate techniques to uncover underlying relationships. While in standard analysis, we start with hypothesis about the nature of relationships, in data mining techniques such as pattern finding we depend on the technique itself to tell us what that relationship might be.

Pattern finding:
An example

Let us return to a database which we have information on our customers on a number of attributes such as age, gender. income group, number years with the institution and marital status. In short, we have a number of discrete variables. A pattern in this context would be a group of customers who share the same attributes. For instance, a pattern might be

- all females
- been a customer for over 10 years
- have children over 5 years of age
- work full-time outside the home
- are divorced or separated
- have more than 3 products offered by the institution
- is a heavy user of the company's most profitable product

If we can identify such a group, it will have tremendous marketing implications.

What is involved
While this may look like a simple problem, in reality it is not. For instance, if we have 20 attributes for each customer in our database and each of these attributes have on the average four alternatives (for example, 1. does not work outside the home; 2. works outside the home - part-time, 3. works outside the home-full-time; 4. works outside the home, but currently unemployed), just to search 7 shared characteristics out of the 20 will involve searching 20C7 x 47 = 77,520 patterns. We need to check this for each customer. So if we multiply this by our database (let us say some 30,000 records), we need to make almost *4 billion comparisons* before the pattern can be identified! What if we want to identify 5,6, and 8 shared characteristics as well? Obviously, even a highly powerful computer will find it difficult to cope with such demands. Neither can we overlook the fact that many institutions such as banks and automobile associations will have hundreds of thousands of records making pattern finding even more complex a task.

Pattern finding programs
This inevitably leads us to programs such as PattFind that are designed to eschew this 'brute force' approach to pattern finding. Such programs tend to narrow down the search by some preliminary scanning procedures. For instance, the program will first scan the database looking for probable candidate patterns. Once such candidate patterns are identified, then the program looks through the database in a more rigorous way to isolate the patterns.

Because the technique is based on heuristic principles, the initial scanning can yield different candidate patterns depending on the rules for initial scanning. Again, the user usually has the option to specify several selection criteria such as the minimum number of criteria to be satisfied, number of respondents needed before a pattern can be accepted and other rules and statistical criteria.

Software from other applications
There are many other algorithms that could solve the problem as well. One software program (Data Mariner) uses standard cluster analytic procedures combining it with rules of inductive logic to identify the patterns. The concept of pattern identification has so many applications in many areas of human endeavour that developments in this area has come from different disciplines. For example:

- *Quality control procedures* . A computer company in California used a pattern identification program (IXL) to identify a pattern in the data that resulted in a large number of disk failures.
- *Suspicious stock activities* . Net map, another pattern finding program has been used to identify unusual stock trading activities.
- *Criminal investigations* . Pattern finding software is also being developed and used to detect patterns in terms of places, time and people as they relate to criminal activities.
- *Geodemographics* . Geodemographic systems currently use some form of clustering of demographics to classify small geographic units. Pattern finding has a natural application here in identifying geographic units that exhibit similar patterns on any set of variables of interest to the marketer.

Overlapping patterns
Some of what is being done by pattern analysis techniques can probably achieved by cluster analysis if the database and the attributes are not too large. However, techniques like cluster analysis, AID, CHAID and KnowledgeSeeker divide into non-overlapping categories. A consumer may not belong to more than one cluster at any one time. Pattern finding, on the other hand, does not concern itself with mutually exclusive categories but only with the existence of a pattern. For instance, in cluster analysis a customer may be an 'Affluent Investor' or a 'Regular Saver', but not both. In pattern analysis the same customer can be both, if he or she shares the patterns exhibited by both groups. (Newer techniques such as fuzzy clustering do indicate the probability of a person belonging to more than one cluster. However, such techniques as they stand do not handle large quantities of data, neither are they readily commercially available.)

Another problem with the pattern finding models is the nature of the data itself. When variables are categorical in nature, pattern finding techniques look for consumers who match one another on a certain number of variables. But what if some of the attributes happen to be continuous in nature? This creates a lot of 'noise' in the data. Suppose the data includes customer satisfaction ratings. Here we are not interested in an exact match of customers in terms of their ratings. Rather, we are interested in knowing how 'close' they are to each other. While in categorical data 1 is as different from 2 or 3, in continuous variables, 1 is closer to 2 that to 3. So we need different techniques.

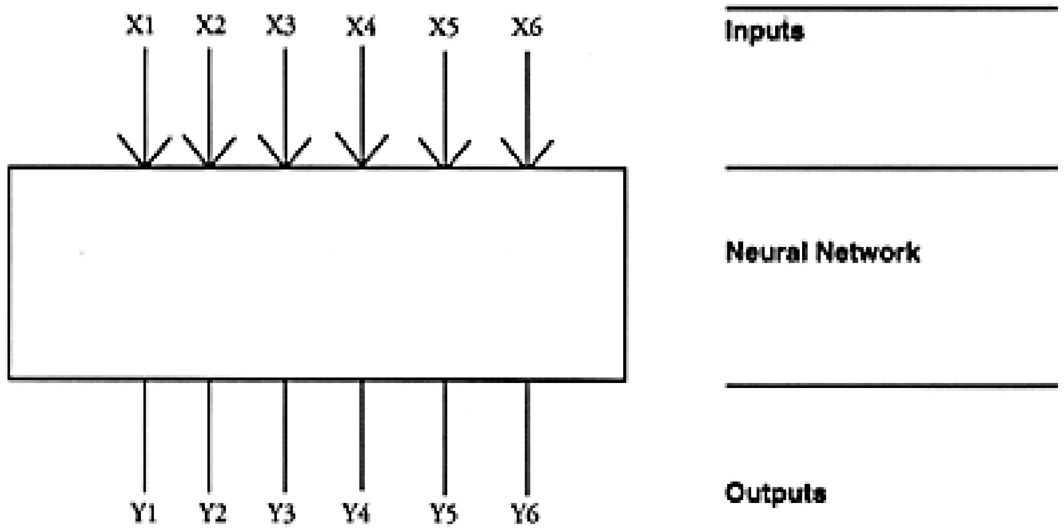
At this stage we need to return to another model we discussed earlier in this series - the Neural Networks (neural nets for short)

The neural nets
Neural nets attempt to simulate the way in which our brains (physiologically) function. In normal statistical analysis, the computer is given a specific set of instructions which are then applied to the data on hand. In neural nets, on the other hand, the instructions to the computer are such that the computer 'learns' as it goes along. It is like a child learning to approach certain things because of positive feedbacks and avoid certain things as a result of negative feedbacks.

A conceptual view of supervised neural networks

We strat with a "training set'. In the training set, we know what the input variables (X1, X2, X3, ...) and the output variables (Y1, Y2, Y3, ...) are. How minput variables such as net worth and income of a customer lead to output variables such as the purchase of various products is the problem being solved. Inside the 'box' are 'neurons' which 'learn' the connection between the input variables and output variables by assigning weights to each variable depending on how frequently an input leads to an output variable

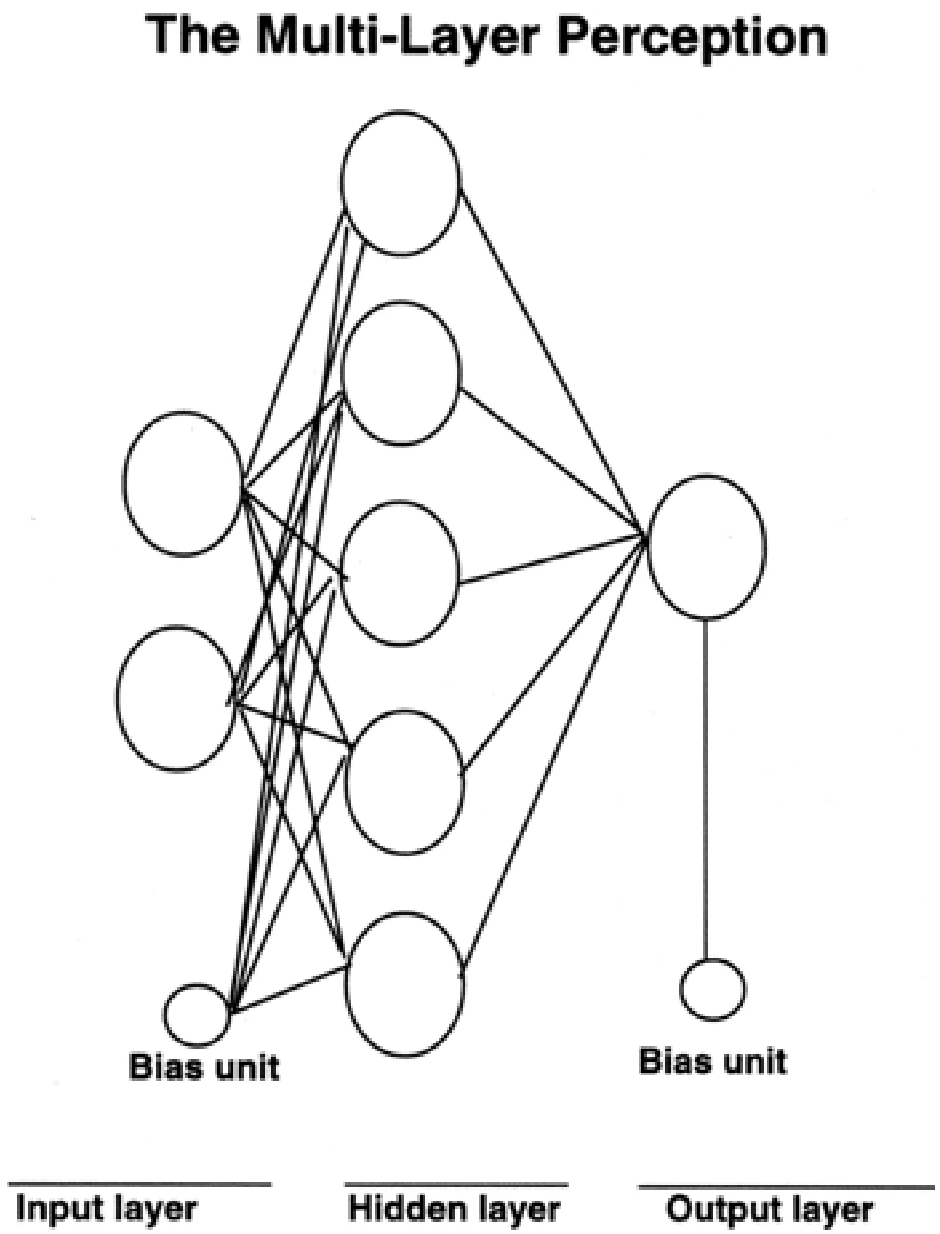
Once the neural net 'learns' the connection between the input and output variables, the model is tested using a test set - a set of data which was not used to 'teach' the neurons the connections between the input set and the output set.



Supervised neural nets
In supervised neural nets, we start with a training set. A training set is a set of known inputs and known outputs. For instance the known inputs could be the different demographic characteristics and business dealings of a customer. The known outputs are the purchasing behaviour of the same customers. The network is then 'trained' to model the outputs from the inputs of the training set.` This is known as 'supervised training'.

The above process involves a collection of processing units known as 'neurons' connected together to form the 'network'. Training these neurons involves modifying the strengths of associations. This is like iteratively modifying the 'weights' attached to different variables. Once the neurons learn the connections, the model is tested using so the connection between the input and output variables can be assessed as accurately as possible.

Once the neurons learn the connection between the input and the output variables has been established, the systems has to be tested. This is done with the use of a with the help of a 'test set' (a set of data previously not used for training purposes). This procedure is akin to the use of 'hold-out sample' in traditional data analysis.



The above example consists of three layers of perception. Every neuron in each layer is connected to every neuron in the next layer. This is the common pattern in MLP models, which are becoming increasingly popular in commercial applications.

Multi-Layer Perception
There are many other types of neural nets. One techniques that is particular popular for commercial applications is known as the Multi-Layer Perception or MLP.

The chart on this page shows a three-layered example. The input layer consists of just two input values. The hidden layer consists of 5 values while the output contain a single value. Note that 'neurons' in each layer is connected to every neurons of the next layer. In addition, there is also the 'bias unit'. (One can quickly see the parallel between /bias unit' in MLP and the 'error term' in multiple regression analysis.

This type of (MLP) model is substantially different from not only the supervised neural nets but also other models such as Radial Basis Function (RBF). Yet all neural net models share the same underlying idea of learning - connections between two variables strengthening and weakening based on empirical data - much like the way we form judgements about on the basis of our experience. The difference is that neural nets learn in a much more systematic fashion.

We will discuss the MLP and other models in the next issue of Imprints.

Dr. Chuck Chakrapani of Standard Research Systems is a Toronto-based consultant, author and seminar leader. He works internationally.