

Data Mining with FormRules

Background

Formulators and processors collect data, by varying various possible input parameters, and seeing what effect there is on measured properties. These data are used to build up knowledge in a specific product domain, often through trial and error. In many new problems, it is not clear what input variables are important, so sometimes measurements are made on extraneous variables. However, this then makes it difficult to determine what variables actually impact on the formulation properties.

For example, a standard Multi-Layer Perceptron neural network will use the extraneous variables to fit any 'noise' in the system. And it is not always practical to use statistics, where some knowledge of the functional form (e.g. whether the relationship is linear or quadratic) must be assumed.

Neurofuzzy logic is a data mining technique that 'discovers' relationships within data, to determine what variables affect the properties. In specific implementations, it can 'weed out' the extraneous variables, and present the information as easily-understood IF...THEN rules.

To test out the capabilities of this method, embodied in the **FormRules** program, Dr Ian Robinson of INEOS Acrylics has devised a data set based on a mathematical function. This data set is designed to mimic a process application, and has six inputs $x_1 \dots x_6$. However, only x_2 , x_4 and x_6 actually contribute to the output, which is described by the functional form:

$$y = R - x_2 + x_4^2 + x_6$$

R is a random noise factor, added to simulate the effects of measuring errors.

100 data points were generated using this function, and the results were stored in a spreadsheet. The spreadsheet contained seven columns - one each for x_1 to x_6 , and one with the computed value of y. These values were input into **FormRules**, and the rules were extracted.

Determination of Critical Variables

However, the data mining technique embodied in **FormRules** develops 'parsimonious' models - the simplest ones that can account for variation in the data. This means that the important variables are extracted automatically. In the present case, as Figure 1 illustrates, **FormRules** automatically recognizes that only x_2 , x_4 and x_6 are important for this problem.

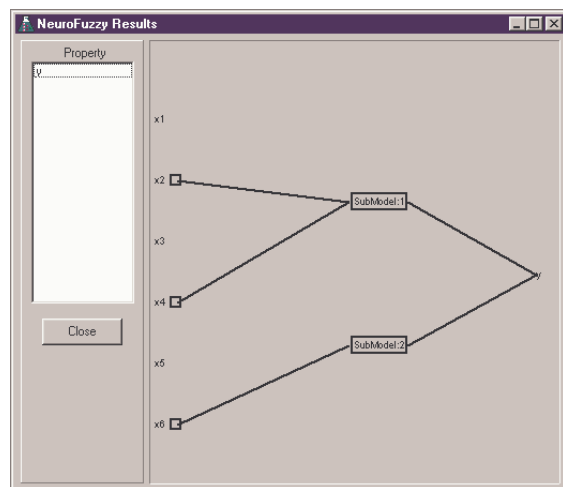
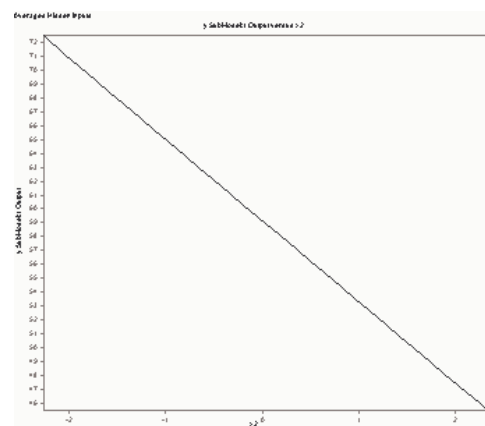


Figure 1. Two 'submodels' involve only x_2 , x_4 and x_6

Perhaps even more importantly, though, the functional form is 'mined' from the data. Figures 2 to 4 show that the software has correctly found the inverse linear behaviour for x_2 , the linear behaviour for x_6 , and the



quadratic behaviour for x_4 .
Figure 2. FormRules predicts inverse linear behaviour for x_2

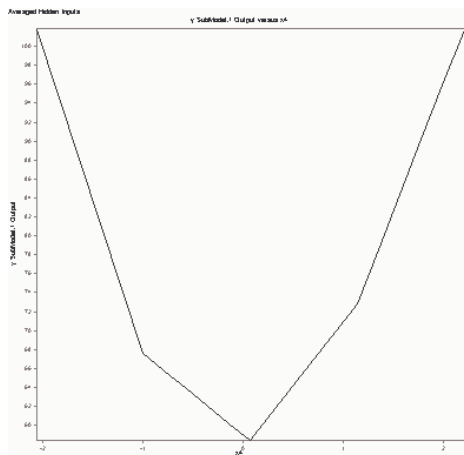


Figure 3. Predicted behaviour for x_4 approximates quadratic form

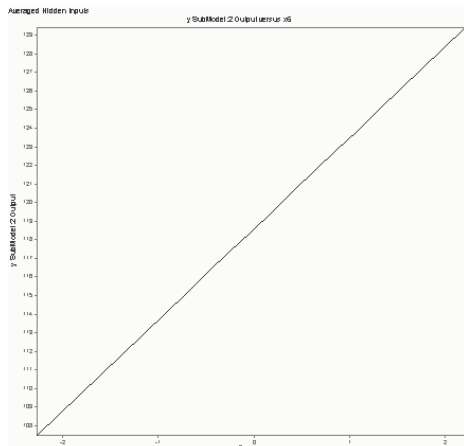


Figure 4. Linear relationship discovered for x_6

These studies were all carried out with the default options in **FormRules**.

Rules are also given directly. For submodel 2, involving x_6 only, these are shown in Table 1.

IF x_6 :LOW THEN y is LOW(0.89) or y is HIGH(0.11)
 IF x_6 :HIGH THEN y is LOW(0.60) or y is HIGH(0.40)

Table 1. The 'rules' extracted for dependence on x_6

Here, the rules say that if x_6 is low, then there is an 89% probability that y is low. If x_6 is high, then there is a 60% probability that y is low. Of course, whether y is high or low also depends on the values of x_2 and x_4 - here, we have assumed that these 'hidden' variables are the average of the values in the data set. That is why the 'rule' cannot be entirely confident that if x_6 is high, then y is high.

FormRules works by trying out various models, and creating the simplest model that is statistically significant. Various measures of statistical significance can be used - here, we used the Minimum Descriptor Length criterion.

Since this is a data mining technique, obviously the amount (and quality) of the data is crucial in determining the model. For example, let's consider this same mathematical data set, but with 200 points instead of 100. Now, we find that 3 submodels are developed, a separate one for each of x_2 , x_4 and x_6 (illustrated in Figure 5).

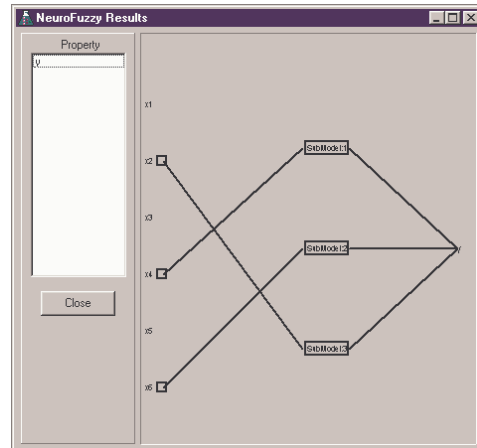


Figure 5. 3 submodels generated from 200 data points

With more data points, the model can be more sophisticated, and for x_4 the following behaviour, closer to the actual quadratic form, is observed.

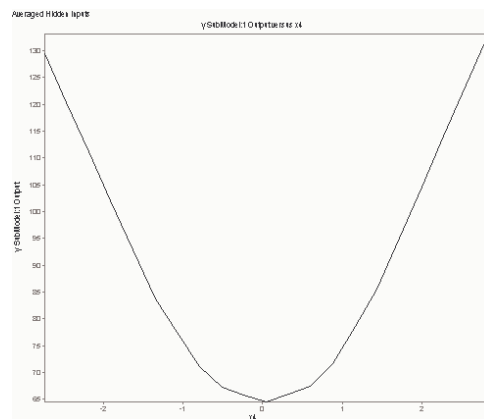


Figure 6. With 200 data points, the quadratic form of the dependence on x_4 is well reproduced

Conclusions

FormRules can determine which variables are important - even for noisy data. The correct functional dependence (linear, quadratic etc.) has been discovered automatically from the data, and (as expected) more information is obtained when more data exist.

For further information on FormRules, and applying neurofuzzy logic to your problems, contact us at the address below.

© 2002 **Intelligensys Ltd**
 All rights reserved.