

FormRules Demonstrator

1. Introduction

Seven examples are illustrated in this **FormRules** Demonstrator. The Demonstrator is based on **FormRules** version 4.

The first four examples have continuous numeric property values for which a model will be developed using the ASMOD Fuzzy Logic algorithm. Two of these examples use literature data. These relate to pharmaceuticals - a tablet formulation (Tablet), and a controlled release tablet formulation (referred to as Invitro). The third and fourth examples use data provided by Dr Ian Robinson of Lucite (formerly Ineos Acrylics, so the data sets are called Ineos40 and Ineos100 respectively), and they represent a typical process problem. The notes below (Section 7 to 9) attempt to explain what the formulator is trying to achieve in each of these cases, so that you can try to relate these examples to your own formulation problem.

The last three examples have classified (text) property values. The model is developed using a modified version of the ID3 Decision Tree algorithm including many of the C4.5 extensions to generate a set of rules derived from the decision tree. The notes below (Section 10 to 12) briefly explain these data sets.

Although on-line Help is available throughout the Demonstrator, the following guide gives a few more tips and pointers about the specific examples given in the Demonstrator, including things which you might want to try for yourself. For more information, though, or for full explanations, please check out the Help.

2. Loading the Examples

When you launch the Demonstrator, you will see an Introduction screen with buttons that include **Create New Task** and **Open Existing Task**. These are not operational in the demo version. In the full version of **FormRules**, they allow you to proceed directly to the task that you want to undertake.

The **New**, **Open**, **Save**, **Save As** and **Recent Tasks** option on the **Task** menu are also not operational in the demo version. Where features are not available in this Demonstrator you will be given the "Feature not enabled in DEMO version" message.

To load an example in the Demonstrator, select **Task | Load Demo** (the first menu item under the Task pull-down menu) and choose the example you want to look at from the picklist provided. You will then be taken immediately to the Set Field Types screen for that example, with the appropriate choices of *Ingredient*, *Processing Condition* and *Property* already made. In the first instance, it is probably sensible to leave the selections as they are. However, subsequently you might want to see what happens if you leave out some of the variables, using the *Not Used* option.

If you want to go back to look at the original data (though you won't be able to change it) then select **Model | Enter/Edit Data**, or click on the **Previous** button. You can not edit this data in the Demonstrator.

When you click on **Next** at any stage, you will be moved forward automatically to the next step. So, clicking on **Next** on the Set Field Types screen will take you to the Data Analysis screen.

3. Data Analysis

This step is frequently omitted by many users, who simply click on the **Next** screen to move on to Training screen. However, it does offer an opportunity to examine the data prior to model development, should you wish to do so.

The Data Analysis screen looks a lot like the Enter/Edit Data screen. However, here it is not possible to edit the data – even in the full program, if you try to replace the information in one of the spreadsheet cells, you will see it is not possible.

You can see from the buttons across the top of the Data Analysis screen that there are various options which can be used to analyse the data. First, let's look at the Graph capability. Press the **Graph** button to get the Data Graph Options window. Pick one of the items for the x-axis, and another for the y-axis, and press **OK**. The Data Graph Options window will also give you additional options, which you might want to explore. You cannot plot classified data using this option.

To see what happens if you want to look at Statistics, select (that is, highlight) all the data. To do this, hold the mouse down while you sweep across the grey fields at the top of each column, from A to H. Then, press the **Statistics** button. You will see that a new sheet has been generated (denoted by a new tab, labelled Statistics#1, at the top of the spreadsheet) that contains the statistical data. Statistics can only be obtained for columns containing numeric data.

The **Analyze** button will get the Analysis Options window. The Analyze option is optional; you do not need to perform this step. It is used to check data for outliers using either statistical criteria (specifying the number of standard deviations from the mean) or else by specifying ranges, to identify outliers.

The **Preprocess** button opens the Preprocess Options screen. Here you can repair or compress your data using hierarchical clustering, use Principal Feature Analysis to analyze your data for input correlations, rebalance or compress your data.

If the data is modified using either of these options then it can be restored to its original form using the **Restore** button.

Click on the **Next** button when you are ready to move on to develop the models, a process called Training.

4. Training the Model

The Training screen has options for you to set up Training parameters, or to work in an 'interactive' mode that lets you change parameters for different properties. However, quite a bit of effort has been made to ensure that the default parameters will develop reasonably good models, provided the data points are reliable and accurate. So, you can simply press the **Train** button in the first instance.

Training involves ASMOD neurofuzzy logic when the properties take continuous numerical values, and Decision Trees when the properties are in different discrete 'classes'. The Training display area shows the progress of the model development.

When training a ASMOD Fuzzy Logic model **FormRules** tries out many possible models to determine which best fit the data you have provided, using the current Model Selection Criteria. (MS) Error displays the mean-squared error for an ASMOD model and displays the fraction of incorrect predictions for a decision tree model. This shows how well the current model fits the training data. Although the System Structure gives information about the model, this can be obtained more easily by looking at the **Results**. For a neurofuzzy ASMOD model, the final column displays the **R²** value of the training data as a percentage. The **R²** value normally lies between 0 to 100% and tells you how well the model has fit to the training data. The higher the **R²** value the better the fit. For a decision tree model, the percentage (%) of correct predictions is displayed (**R²** value cannot be calculated for classified output values).

Once the model has trained, you will get a small window advising you that 'Training is complete!'. Click on **OK** to dismiss this window. You will see that now the **Options**, **Spreadsheets** and **Graphical** buttons, which were inactive prior to training, are now available. These are described in the sections below.

If you want to investigate the effect of changing the parameters, for example changing the Model Selection Criterion for ASMOD models, you can **Abort** this training session, and press the **Parameters** button to access the Model Training Parameters screen.

4. Graphical Display of Models

Pressing the **Graphical** button will give a graphical display of the models. For ASMOD models, built when a property has continuous numeric values, the graphical structure shows the submodels used to model the property and the inputs to each submodel. For decision tree models this provides a visual display of which inputs are used in the rules.

You can pick which model you want to examine from the list at the left of the Neurofuzzy Results window.

For an ASMOD model if there are lines connected to a *Submodel* box, this indicates that the input contributes to the model for that property. If more than one input connects to the same Submodel box, then this indicates that there is an interaction between those inputs. The submodel that makes the largest positive and largest negative (or smallest positive) contribution to the overall model is shown in purple, so that you can see where the major contribution lies. If you click on a Submodel box, a small window will pop up, allowing you to Show Rules or to produce an Output Plot. If you click on the small square next to an input name, then you will see the fuzzy sets that are used as the basis set for the model.

Close the graphical display window when you are finished looking at it.

5. Spreadsheet Results

Pressing the **Spreadsheets** button on the Training window will give the Training Results window, which displays a number of tabbed spreadsheets. The one which displays on top gives a summary of all the *Rules* that are generated from the models. The format in which the rules are displayed will depend on whether the model was developed using ASMOD or a Decision Tree.

For ASMOD models, the rules are presented in the form IF (antecedent) THEN (consequent) where the antecedent refers to one or more of the inputs, and the consequent is either LOW or HIGH. Following the rule, there is a numerical value in brackets. These values are 'confidence levels' that say how low or how high the trends from this submodel are. For example, LOW (1.00) means that the contribution from this submodel is at its lowest value. LOW (0.8) would be higher, since it would have a 20% membership in the HIGH set, and the value would lie 1/5 along the line connecting the LOW and HIGH values. The rules may be displayed in colour. The rule given in blue makes the largest positive contribution to the overall model. The rule given in red makes the largest negative contribution to the overall model. (Note that if no contributions are negative, then the rule that makes the smallest positive contribution will be shown in red.). The **Options** button accessed from the Training screen gives you various options of how the models and rules are displayed. In the first instance, it is perhaps best not to change these.

If Decision Trees are used, a list of rules is displayed. Following each rule is a value that indicates the Confidence of the rule.

In addition to the *Rules* sheet, the next most useful sheet is the one given by the *Model Statistics* tab. For ASMOD models this shows the ANOVA (Analysis of Variance) statistics that show how well the models fit the training data. As mentioned above the **R²** value gives an indication of how reliable a model you have.

Decision Tree statistics analyze the performance of the rules as described in the Training Results.

You might also want to look at the *Training Data* tab, since this recaps the training data but also gives new columns with the values predicted by the model.

When an ASMOD model has been developed it can be useful to select **Graph**, and plot the actual vs predicted values for a property. Select one of the actual values for the x axis, and the corresponding predicted value for the y axis, and make sure that Show Linear Regression Fit Line is switched on (i.e. there is a tick in the tick-box). This will give you a line, with the corresponding slope and intercept. Obviously in a perfect world, this would be a straight line with slope = 1 and intercept = 0. In real life, the amount of scatter in the plot gives a good idea of how accurately the model fits the initial data.

6. Consult Mode

Consulting is not one of the main features of **FormRules**, but has been included for those users who do not have **INForm**. The Consult screen allows you to fill out ingredient values, and calculate Predicted properties, i.e. perform a 'what if' trial to see how making a change to inputs will affect the outputs. There are also facilities for retrieving existing data records.

First, let's look at what happens when you press the **View Data** button. You will see that this opens the View Data window, with the *Complete Data* tab spreadsheet showing all the values that were used in training. (The *User Data* tab can be used to make predictions using data which was not used in developing the model - this is not possible with the Demonstrator.) Press the **Predict All** button at the top of this screen, and you will see that a **Report** screen with two tabs is created. One tab shows the Model Statistics and the other gives the Predicted Complete Data with new columns added for the predicted values and the errors for each of the properties.

On the **View Data** screen select a line - say, *Record#11*. You can select this line by clicking on the grey cell marked 12 (for the 12th line) in the leftmost column. Then, press the **To Consult->** button, followed by the **Close** button. The data from Record#11 will now be filled out in the Given columns on the main Consult screen.

If you now press the **Predict** button on the Consult screen, you will see that the *Predicted* values in the Properties side of the screen are calculated. This is the prediction from the model, for these values, and compares with the experimental value (which was filled out in the *Actual* column for the Properties, when we pressed To Consult-> from the Complete Data screen).

Let's now predict a new 'formulation' that was not part of our original data set. Change one of the Ingredient values, making sure that it is within the range of the existing experimental values. (You will be warned, if it is not.) Press the **Predict** button. You should see that the *Predicted* Property value will change to give the value that the model would expect if this formulation would be made.

Finally, let's look at the retrieval function- called Best Match. Fill out some values in the Ingredients side of the screen, and press the **Best Match** button at the right of the screen. The Best Match screen appears. Click the Match to **Ingredients** button which will find the record which best matches the *Target* ingredient values and enter the values in the *Found* columns of the Ingredients and Properties. The record has been retrieved is shown in the bottom left of the window.

This gives a brief introduction to the Consult capabilities. For more information on Consult and other aspects FormRules, see the Manual or explore the online help. But remember - **FormRules** is developing relatively

simple models to fit to your data, and so using the models for prediction may not be ideal. **INForm** might be a preferred choice here.

7. Tablet Formulation

This example is of interest because many of the input variables are either/or in nature (e.g. Diluent is either Lactose or Datab). Therefore, they are assigned to discrete, rather than fuzzy, input sets.

The Tablet example is taken from Kesavan and Peck (Proc. 14th Pharm Tech Conference, Barcelona, 1995). Chapter 6 of Intelligent Software for Product Formulation by Rowe and Roberts (Taylor and Francis, 1998) also uses this as its illustrative example. Briefly, this is a tablet formulation consisting of:

- anhydrous caffeine (40% w/w) as a model active
- dicalcium phosphate dihydrate (Datab) or lactose (44.5-47.5% w/w) as a filler
- polyvinylpyrrolidone (PVP) (2.0 -5.0% w/w) as a binder
- corn starch (10% w/w) as a disintegrant
- magnesium stearate (0.5% w/w) as a lubricant

Two types of granulation equipment - fluidized bed and high shear mixing - are used, and the binder is added either dry, or as a solution. The amount of caffeine and the percentage of cornstarch were held constant, so the five variables were:

- Diluent (Datab or lactose)
- Diluent%
- PVP%
- Binder Addition (wet or dry)
- Granulation Equipment (Fluidized Bed or High Shear Mixer)

Properties measured included tablet hardness, tablet friability, tablet thickness and disintegration time. The aim of many formulators is to make hard tablets (which will be robust and will not break up while you are carrying the bottle around in your pocket, for example) that also disintegrate quickly (so that the drug can get to work right away). Friability has some relationship with hardness, since typically hard tablets are not very friable. Thickness, for the purpose of our study, is pretty unimportant.

This example is of interest because many of the input variables are either/or in nature (e.g. Diluent is either Lactose or Datab).

8. In Vitro Release

The In vitro example looks at how formulating a tablet with a specific release profile. Models that relate in vitro profiles to in vivo release exist, so that getting the correct in vitro profile is a key step in finding the correct formulation.

The data here are taken from information provided by A I Ware Inc., and were determined by Y Chen and his colleagues. They published their study in the Journal of Controlled Release 59 33-41 (1999), although the data are not actually given in the paper. Chen and colleagues used 10 different formulation variables

- Amount of Polymer A in each tablet
- Amount of Polymer B in each tablet
- Amount of Dextrose
- Amount of Lubricant

- Tablet Weight
- Drug/(Polymer+Drug) ratio
- Polymer A/Polymer B ratio
- Tablet hardness
- Particle size
- % Moisture

Several of these variables are dependent on other variables – e.g. Tablet Weight depends on the amount of other ingredients added (since the amount of drug was a constant 9.6 g), and obviously the Polymer A/Polymer B ratio depends directly on the amounts of Polymer A and Polymer B. We chose to leave these dependent variables out of the model -- you can see that by noting that, when you load the data set, they are set as Not Used. If you want to try to reproduce the Chen et al paper, you might want to set them to be Ingredients.

The outputs are measured releases at different times. Clearly there is error in the measurements, since some of the in vitro results suggest a release of over 100%.

One of the interesting things about this example is that it finds that Polymer A dominates the short-term release, while Polymer B is important for the long-term release. This is characteristic of two-polymer release systems, but what is significant here is that FormRules has discovered this relationship directly from the data, with no input of this knowledge from the formulator.

9. 'Ineos' examples

These examples are taken from a data set that has been devised by Dr Ian Robinson of Lucite, Wilton, UK, to represent a typical case where the researcher might measure more parameters than necessary. This data set has six different inputs $x_1 \dots x_6$, which represent input plant conditions. There is one output, y , which has the mathematical expression

$$y = 100 - 5x_2 + 10x_4^2 + 5x_6 + 5R$$

R represents some random noise on the data. As can be seen, only x_2 , x_4 and x_6 contribute to the expression; x_1 , x_3 and x_5 (although 'measured') have no effect. x_2 , x_4 and x_6 are generated at random, so the results are not necessarily perfectly uniformly distributed.

The purpose of this example is to see if FormRules can discover the important relationships even when there are unnecessary input variables, and noise within the data. Two data sets are provided, one with 40 data records and the other with 100. Exploring both should show that more complicated models and rules are developed when more training data points are available.

10. Iris flower data set

The *iris* data set was introduced by Sir Ronald Aylmer Fisher (1936) and has become a typical test case for many classification techniques in machine learning. The data set consists of 50 samples from each of three species of Iris flowers (*Iris setosa*, *Iris virginica* and *Iris versicolor*). The length and the width of sepal and petal (in centimeters) is measured for each sample. Based on the combination of the four features, Fisher developed a linear discriminant model to distinguish the species from each other.

11. Olive oil data set

The *oil* data set contains analytical data from 572 Italian olive oils produced in nine different regions of Italy. For each sample the normalized concentrations of eight fatty acids are given. The data was collected by Prof. Michele Forina, University of Genova, Italy.

The data is from a paper by Forina, Armanino, Lanteri, Tiscornia (1983) Classification of Olive Oils from their Fatty Acid Composition, in Martens and Russwurm (ed) Food Research and Data Analysis.

12. Wine data set

The *wine* data set consists of the chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

Forina, M. et al, PARVUS - An Extendible Package for Data Exploration, Classification and Correlation.
Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy.

13. And in Conclusion

Remember, there is on-line help available throughout the Demo, to show you what the different buttons are for.

We hope that you have found this **FormRules** Demonstrator useful. Please refer any queries (or provide any feedback) to

Intelligensys Ltd
Springboard Business Centre
Ellerbeck Way
Stokesley, North Yorkshire TS9 5JZ, UK
e-mail: postmaster@intelligensys.co.uk

© Intelligensys Ltd 2011