HyperLogic

TECHNICAL NOTES

A series of informal papers on technical topics for customers and friends of HyperLogic Corporation

Number 6
Revision 3

# Questions and Answers about HyperLogic's Rule Maker

This technical note answers some common questions about HyperLogic's Rule Maker add-on for its CubiCalc line of fuzzy tools.

## What is Rule Maker?

Rule Maker is a tool for automatically building fuzzy systems from data you supply. A transparently integrated add-on, it puts a new menu command in the Project menu of the CubiCalc interactive shell. You can use it with CubiCalc, CubiCalc RTC or CubiCard, but not with CubiQuick.

Projects created by Rule Maker are just like those you create yourself. After automatically building the fuzzy rule base, you can later change it or use CubiCalc's built-in features for tuning and analysis. You can also add rules to an existing project or replace the rules already there. It's easy to experiment since Rule Maker doesn't affect other parts of project — like your plant simulation or your plots.

You set up a Rule Maker run with the dialog shown in Figure 1. You need to provide a data file and identify which fields in the file correspond to the input and output variables. Rule Maker can use variables you've already defined or it can create them for you.

Each data record is a sample of the underlying input-output relationship. Using one of several methods, Rule Maker analyzes the data to decide how to create the rules and fuzzy sets.

## What are Rule Maker's methods?

Rule Maker provides several methods, grouped generally into three classes: Slice Interpolation, Correlation Analysis, and the Sparse Data techniques.

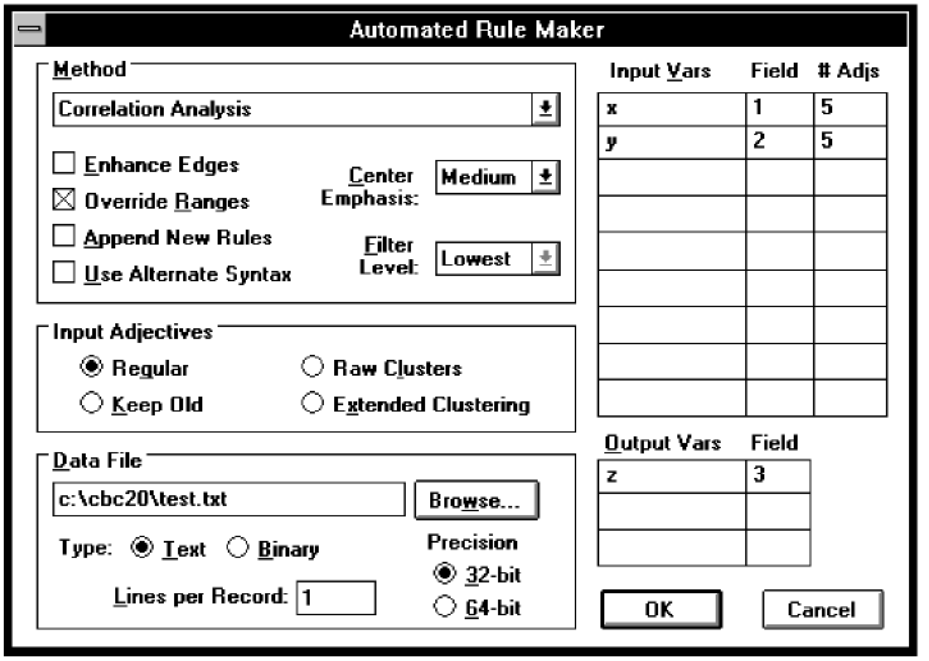
*Correlation Analysis* combines statistical and neural methods. More flexible than purely neural techniques, it offers several choices for handling the input fuzzy sets or "adjectives." You can start with arbitrarily-shaped input fuzzy sets that you've already created and just have it make the rules and the output fuzzy sets. Or you can have it create the fuzzy sets by finding "clusters" in the data. You can even have it create regularly spaced input fuzzy sets; this is like making them yourself with CubiCalc's adjective editor but it's a little more convenient.

*Slice Interpolation* creates a small number of rules by using complex shapes instead of triangles for the fuzzy sets of one variable. With this method, Rule Maker tries to reproduce the function along each slice; the natural interpolation capability of fuzzy systems produces intermediate values between the slices. The number of rules is twice the number of slices you choose. The method is most useful for  systems with two inputs.

The two sparse methods, *Data Point Rules* and *Sparse Interpolation*, are essentially the same but provide different approaches to handling situations for which no rules can be found.  With both you provide explicit information about the rules to create.  This is useful if you are using another tool to develop rules or if you have only a very small data set.  It is also helpful for building the first cut at a system: You can use your favorite editor to create a text file containing examples of the input-output relationship you want.

## What can I expect?

Many of the reasons for using fuzzy systems stem from the development process. The good match between human cognition and fuzzy computational structures



Method: Correlation Analysis

☐ Enhance Edges
☒ Override Ranges
☐ Append New Rules
☐ Use Alternate Syntax

Center Emphasis: Medium

Filter Level: Lowest

Input Adjectives: ● Regular ○ Raw Clusters ○ Keep Old ○ Extended Clustering

Data File: c:\cbc20\test.txt

Type: ● Text ○ Binary

Precision: ● 32-bit ○ 64-bit

Lines per Record: 1

| Input Vars | Field | # Adjs |
|---|---|---|
| x | 1 | 5 |
| y | 2 | 5 |

| Output Vars | Field |
|---|---|
| z | 3 |

*Figure 1. Rule Maker Dialog*

gives acceptable results faster. "Acceptable" means fast enough, small enough, and close enough. The head start you get from Rule Maker can speed up the process even more.

However, an automated tool lacks the judgment of a human expert to know what's important and what isn't. It must assume that the data you provide represents the results you want. It also assumes you want an answer for every possible input. The result is that automatic rule builders can create a lot of rules.

You should not expect to distill a handful of rules from a data set involving many variables. Usually you get a rule for every combination of input fuzzy sets. A four-input system with five fuzzy sets describing each input gives $5^4$ or 625 rules.

If data is present to describe the input-output relationship in a region, Rule Maker assumes the region is important enough to warrant a rule. But you can exert a lot of control over the process by creating your own fuzzy sets before running Rule Maker. You can also use the clustering techniques, which assume that more data in a region means more emphasis should be placed there.

### What about the data?

One thing that makes it difficult to create a fuzzy system from data is that you don't have a formula for the function your data represents and you don't really know what the function is. Your data samples might be noisy or inconsistent. They might not cover the entire input domain.

For example, if you've collected data by observing a human operator controlling a process, you probably just want the operator's "perfect" runs. Otherwise, you risk creating a fuzzy system than consistently makes certain errors only to recover from them.

Because of these issues and because the rule making techniques are essentially statistical, the system that Rule Maker creates is simply not the "last word." You should always budget some time for testing and manual tuning.

On the other hand, sometimes results can be pretty good. Figures 2 and 3 show a two-input function and the results of a fuzzy rule base created by Rule Maker.

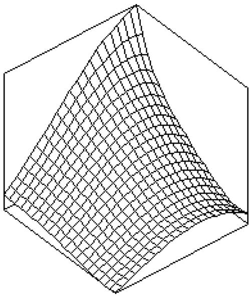


*Figure 2. Original Function*

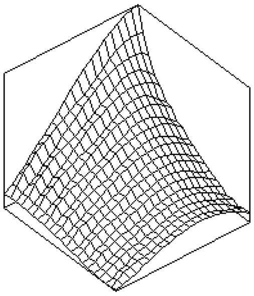


*Figure 3. Fuzzy System Output*

### How do I choose a method?

Correlation Analysis is always a good starting point when you have a lot of data. It can work quite well even with small data sets, especially when they are free of noise. In either case, the data should be a good representative sample of the entire input-output relationship.

Correlation Analysis has clustering methods for locating adjectives. If your sampling procedure collects more data in regions of more importance, select one of these so that the method will place rules in those areas.

Slice Interpolation is particularly good for two-input functions with many local minima and maxima. Reproducing this type of function with triangular fuzzy sets tends to take many rules. Slice Interpolation can lead to a much smaller system.

Try Slice Interpolation when the input-output relationship is complex but you need a small rule base. Remember that some of the fuzzy sets are more complex than simple triangles, so they can take longer to evaluate or use more memory.

Correlation Analysis often gives better accuracy than Slice Interpolation. However, this can come at the cost of a much larger rule set. Rule Maker can build both kinds of systems, so it's easy to try both methods. Then use the Compile dialog in CubiCalc RTC to estimate the memory usage.

The sparse methods are most useful for data sets you create by hand. With these methods, you should always start with a small collection of points. Each data point can create a rule.

The Data Point Rules method uses triangular fuzzy sets exclusively. Sparse Interpolation is similar, but may decide to extend fuzzy sets to ensure that a rule covers every situation.

### When should I use Rule Maker?

In many practical problems, the desired action has no "linguistic model" that you can use to build a fuzzy system. Instead, you have a mountain of data. For these situations, an automated system building tool is necessary.

Even if you have a good idea of the proper behavior, Rule Maker can speed up the development effort by quickly creating a starting point or by giving you a reference system for comparison.

Finally, if you've never used fuzzy rules before, Rule Maker can show you how to do it.

In any case, Rule Maker is an essential companion for work with modern fuzzy systems.