

Re-Discovering What You Do

A Data Mining and Rule Discovery Approach to
Business Forecasting with Adaptive,
Genetically-Tuned Fuzzy Systems Models

by
Earl Cox
©1999

Sum, Ergo Cogito!
I am, therefore I Think.
With pardons to Rene Descartes (1596-1650)

Business process modeling is often a no-win situation. Developing reliable business forecasting models requires a successful collaboration between line management and knowledge engineers. And even where the fusion of working knowledge and abstract representation is successful, its result is too often thwarted by the unpredictable dynamics of the real world: corporate objectives change, companies are bought and sold, and new products are introduced (or existing products retired). To Compound matters, corporate decision-makers and model builders are also faced with the unprecedented uncertainties and pressures of the rapidly changing nature of electronic commerce over the Internet. In fact, as Figure 1 illustrates, the high rate of change in the global world economies will continue to exert pressures on the stability and viability of even well established corporations.

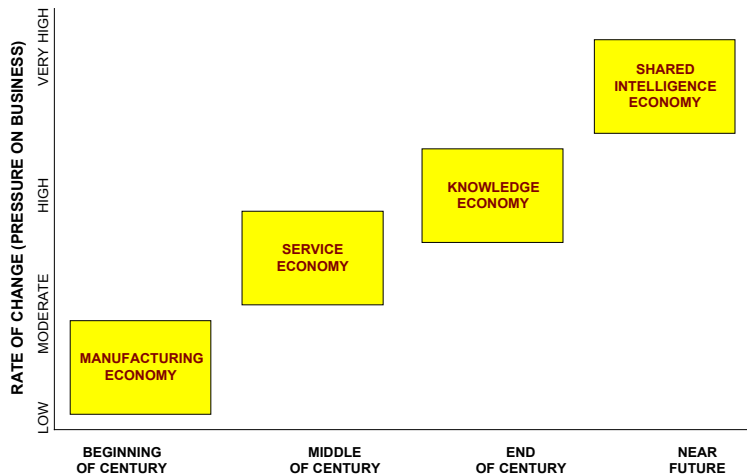


Figure 1. Rates and Types of Changes in the Global Community

As a result of these uncertainties, business-forecasting models have fallen out of favor in recent years. Instead, business planners tend to concentrate on the short term, analytical approach to business forecasting. In particular, intelligent models – known in the 1970's as Decision Support Systems and later as Expert Systems (although they use different technologies) – have been replaced by the ubiquitous spreadsheet. Yet spread sheets are no substitute for

knowledge-based models in such critical areas as risk assessment, econometric modeling, new product positioning, customer profiling, cross-marketing, sales forecasting, and impact analysis. In this article we will examine ways to make your models more responsive to changes in demographics and the economy.

A not so obvious solution to the problem of change and uncertainty is simply to incorporate these factors into the model itself. Naturally this means going beyond a statistical analysis of the data or the inclusion of certainty factors or forms of Bayesian probabilities. We must create our models so that they automatically change their internal behavior structure to accommodate changes in the outside world. One approach to this is the adaptive model – a model that alters its rules based on changes in the outside world. A powerful and robust way of building an adaptive model involves the combination of three broad technologies: Fuzzy Logic, Data Mining, and Genetic Algorithms. Fuzzy Logic provides a method for capturing the semantics or meaning of the data through a collection of fuzzy sets associated with each variable. Data Mining uses these fuzzy sets to generate an initial model of if-then rules. A Genetic algorithm creates and tests many candidate models by changing the fuzzy sets until it finds the one that performs the best.

Building A Fuzzy Business Model

The core process in creating an adaptive business process model is the generation – or discovery – of the rules. For fuzzy models, the first step involves decomposing the domain (that is, the range of values) of each variable into collections of over-lapping fuzzy sets. A fuzzy data mining tool generally does this by examining the properties of the variables and producing the initial fuzzy sets. Figure 2 illustrates the fuzzy sets for a variable representing the quarterly change in the Consumer Price Index (CPI). The gray area in the background shows the actual distribution of data in the database.

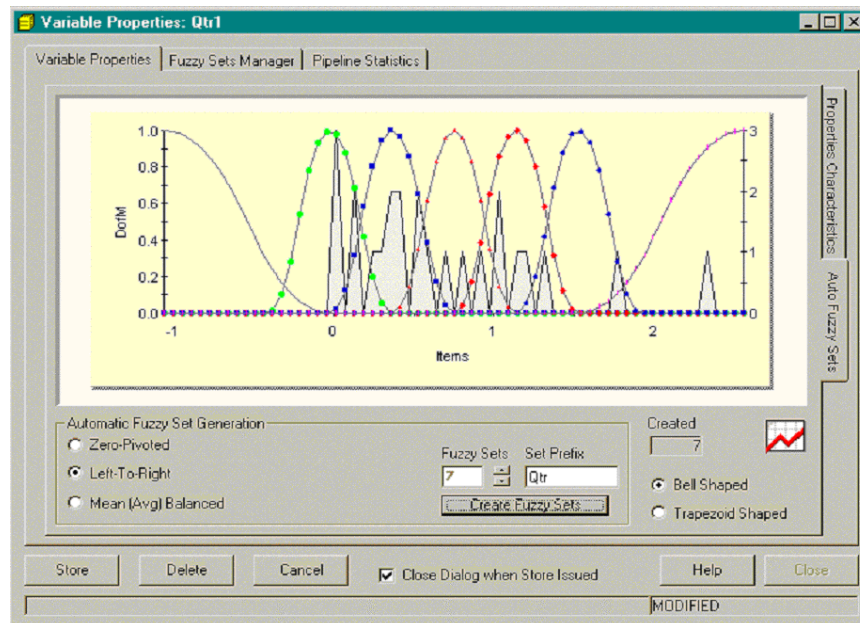


Figure 2. Fuzzy Sets for a Model Variable (Quarterly CPI Changes)

Each fuzzy set provides a handle on a specific space within the variable's range. As you can see in Figure 2, fuzzy sets overlap, representing the natural mixing of concepts – as the idea of a Low change in CPI decreases (fuzzy set in red at left), the corresponding idea of a moderate change in CPI increases (fuzzy set in blue, third from left). Thus, a data value can be in both fuzzy sets at the same time but to different degrees. A rule induction or discovery system fits data to the fuzzy sets and produces several different rules for the same set of data. Each rule has a different degree of evidence or certainty that it describes a behavior pattern in the data.

By changing the number of fuzzy sets, their shape, or their overlap we can change the final model. This ability to tune a model by modifying its fuzzy sets is the key to building self-measuring and adaptive model. Figure 3 illustrates the basic model generation process of generating and testing rules.

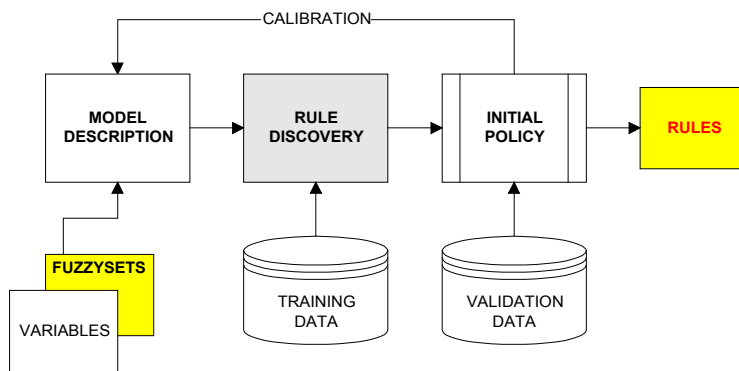


Figure 3. A Core Model Evolved from Rule Discovery

A rule discovery facility takes a training set and finds the behavior patterns buried in the data. These behaviors are expressed in the form of *if-then* rules, tying together a set of independent variables with the outcome or dependent variable. The generated rules produce a working model or business policy. This policy is run against a validation file to measure how well it predicts the outcome variable's value. Figure 4 illustrates the results of running the generated model against the validation data file. The red line is the actual CPI change and the amber line is the change predicted by the model.

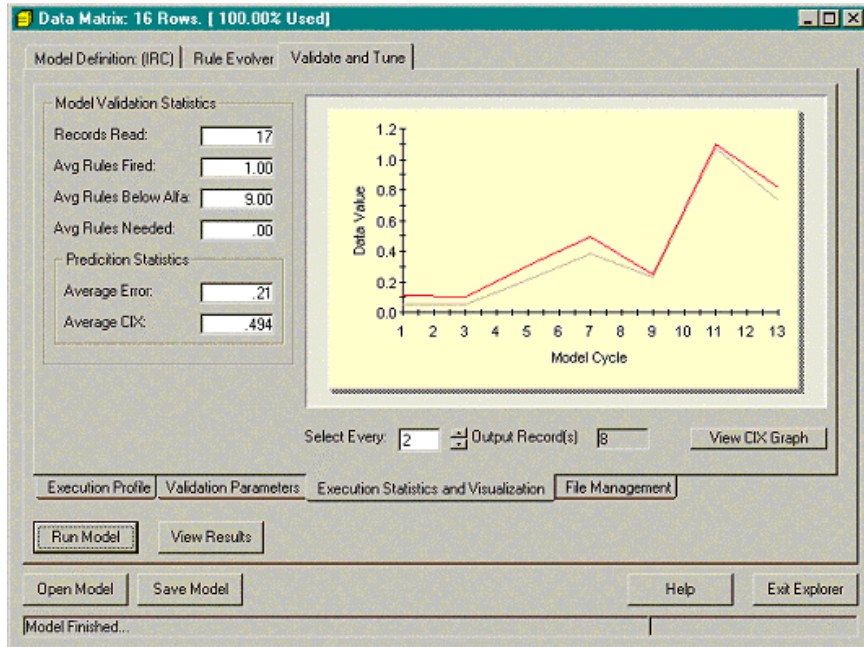


Figure 4. Validating a Generated Fuzzy Model

If there is a significant error in the prediction, the model description is changed and a new model is produced. Predictive error is generally measured as the average of the sum of the squared distanced between the actual outcome value and the predicted outcome value (although other measures of error are also used.)

Optimizing the Model Architecture

If the model lacks sufficient predictive power (the standard error is too high) the knowledge engineer or business analysts must tune the model by changing how the rules are generated. This generally involves changing, for one or more variables, the number of fuzzy sets, the shape of the fuzzy sets, the degree of overlap, and sometimes the range or domain of the variable (in order, for instance, to “pull in” data that is clustered at the edge of the domain). The number of permutations can be very large. A better approach is to use a genetic algorithm to explore a large number of possible configurations and find the one that has the best predictive power. Such a genetic algorithm can automatically tune the generated business policy. Figure 5 illustrates the flow of control in a genetically tuned fuzzy model.

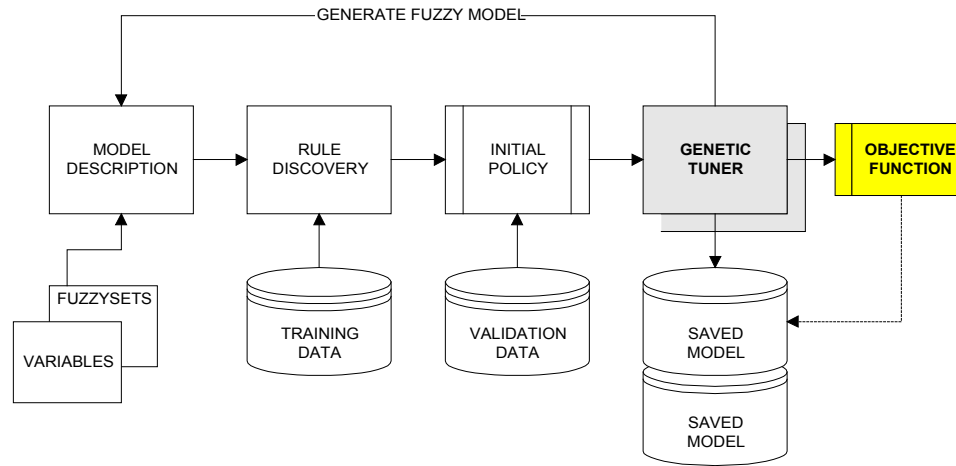


Figure 5. Genetic Tuning of a Fuzzy Business Model

This is a multi-objective genetic algorithm. Each possible fuzzy model configuration (the description) is represented as a chromosome – a string of bits corresponding to features of the model. The chromosome’s bits are partitioned into genomes (short strings of bits found in same place on each chromosome) representing the number of fuzzy sets, the overlap, the shape (trapezoid or bell-shaped), and the width of the variable’s domain measured as plus or minus a small percentage of the range. Initially a genetic algorithm forms a population of several dozen configurations by creating chromosomes with random bit values. For each chromosome, the genetic algorithm creates a new model description, evolves a set of rules, runs the validation-process and measures the standard error (which is the objective function or goal). This is called a generation of the genetic algorithm. Models with small standard errors (those with good chromosomes) are retained and used in the next generation. The remainder of the population for the next generation is replaced by combining these good chromosomes to produce new offspring. After a number of generations the population, by selecting fitter and fitter chromosomes, moves toward a configuration that has the best organization. At the end, the model with a standard error smaller than any previous model is selected as the current optimum. Figure 6 shows a genetic tuning process at work for the first five generations.

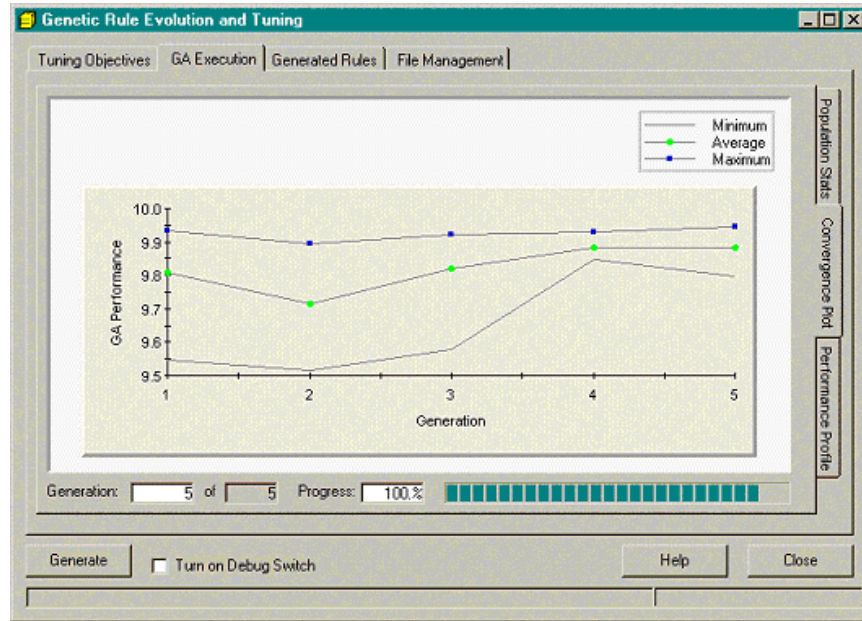


Figure 6. Genetically Tuning the Econometric Model
(Shown for the First Five Generations)

The best models from each generation are preserved and used to populate the next generation through the standard mechanisms of cross-over and mutation (in Figure 6, the maximum value actually represents the smallest standard error). After many generations, a finely tuned fuzzy business model or policy is produced.

A Self-Measuring Model Based on Evidence

A unique and powerful feature of a fuzzy rule base is its ability to measure the degree of evidence in a prediction. In a rule such as the following which measures over-all economic health,

```
If Qrtly_CpiChange is Low and Personal_Income is Increasing
Then Economic_Health is Improved.
```

(where *Low*, *Increasing*, and *Improved* are fuzzy sets) the degree to which a data element from each variable is a member of the associated fuzzy set indicates the strength of the rule. This is a measure of how much evidence exists that *Economic_Health is Improved*. When all the fuzzy rules have been executed, the final solution variable has an associated degree of evidence (called the Compatibility Index). This index will be in the range [0,1] inclusive. A compatibility measure that is too close to [1] or too close to [0] indicates a model that has a significant mismatch between the fuzzy set descriptions and the model data. That is, most of the model data consistently falls near the edges of the fuzzy sets. Knowing this fact provides a means of measuring the performance of a model by tracking the regression line of the compatibility index values across time. If the line begins to slope significantly up or down (see Figure 7) then we know that the model must be re-trained.

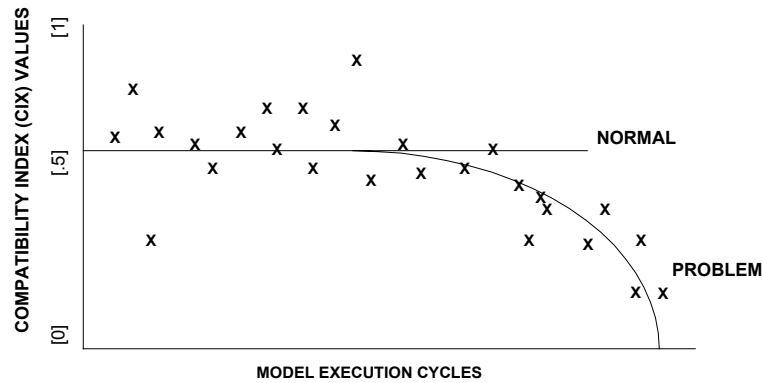


Figure 7. Measuring the Performance of a Fuzzy Model

Naturally, the compatibility index might be close to one or zero for an individual execution of a model. It is the long term statistical trend that is important – essentially as expressed in the positive or negative slope value. We can easily use a moving average of the slope change to determine when a new model must be re-generated (and also flag the current model as defective – as a precaution to users who must then subscribe to or download the new, updated model.) What we are doing here is actually quite simple. We save the amount of evidence for the outcome in each run of the model. We then perform a simple sum of squares linear regression on these values to see if the regression line has a steady movement up or down over a large number of model executions. This is an important factor – does the line have a real up or down trend? When this trend becomes pronounced (say more than 25% of the initial average compatibility index) it is time to re-build (adapt) our model to the new data.

And So In Conclusion...

Using the evidence-based measurement of performance, an adaptive model automatically invokes the rule induction and genetic tuning facilities to re-generate a new business process policy. This feed-back loop – produce model, test performance, update model when performance is below an acceptable threshold – keeps our business forecasting models synchronized with changes in the outside world. In effect, then, we have embedded a mechanism for handling uncertainty and change in our modeling methodologies.

Thus, we have examined a tightly coupled modeling approach: a data mining engine to discover business process rules, a genetic tuner to find the best performing architecture, and an evidence-based analyzer to detect a discontinuity between the model and the outside world. Linked together they provide a highly adaptive way of creating and using business models. The use of these models will give business planners, knowledge engineers, and systems analysts a new generation of modeling tools. Further, the predictions and forecasts from such models will be more reliable over a much longer period of time. Adaptive modeling should do much to bring decision support and expert systems back into the business mainstream.