

A Protocycling Methodology for Knowledge-Based Data Mining Projects

©2002 Earl Cox



Scianta Intelligence

1289 North Fordham Blvd. Suite A312
Chapel Hill, NC 27517

(919) 678-0477
www.scianta.com

A Protocycling Methodology for Data Mining



I thought everyone must know that a *short* jacket
is always worn with a silk hat at a private view in the morning

Edward VII (1841-1910)
King of the United Kingdom

Tous les jours, a tous points de vue, je vais de mieux en mieux
Every day, in every way, I am getting better and better

Emile Coue (1857-1926)
French psychologist

In *Building Intelligent Models from Data Mining and Expert Knowledge: A look at Fundamental Principles* (Sept/Oct 2001) I took up the issues associated with organizing and building intelligent, rule-based business models. That article dealt primarily with the nature of hybrid models and the way they are constructed from both discovered and subject matter expert rules. We briefly touched on a methodology supporting this development process. In this article, after a brief review of the methodology front end, I will look a little more closely at the methodology and how it directs the evolution, development and deployment of intelligent business models.

The complexity of today's business high distributed eCommerce world places a heavy burden on business analysts. Earlier modeling projects could easily address the input and output of discrete processes, running on the corporate mainframes or on early local and . The complexity in modeling CICS applications or batch applications with limited telecomputing components (such as a bank's on-line tellers running as a real-time task under IMS/VS or other communications software) made limited demands on the modeler. Early systems were often modeled with GPSS (the General Purpose Systems Simulator), SAS, or written in Fortran, PL/1, or (believe it or not!) assembly language. In today's business environment where applications are co-resident on multiple servers in peer-to-peer connections across the internet, the company's intranet, and private supply chain extranets, such linear modeling techniques are not only inadequate but almost guaranteed to fail.

Today's distributed system architects and business process modelers are faced with a much more challenging problem. Modern business systems are highly distributed with both vertical and horizontal behaviors that tend to change dynamically. Horizontal behaviors reflect the interconnected processes of applications that are serially or functionally related – applications that exchange a wide variety of transactions across globally distributed servers. Vertical behaviors reflect the capacity, performance, through-put and loading of the underlying infrastructures supporting the over-all business models. This includes such elements as application and web servers, databases, RAID (Redundant Array of Inexpensive Disks) arrays, load balancers, firewalls, and mainframes. At the same time business modelers must compensate for the underlying operating system mix – Windows, Linux, Unix, and MVS (for IBM mainframes) - found in these high distributed application segments.

Because of the complexity in modeling today's business systems, enterprise modelers have turned to fusing statistical analysis, spreadsheets, rule-base expert systems, and knowledge discovery capabilities. Knowledge discovery, or data mining, is used to find the rules of behavior for business systems by examining patterns of behavior buried in the historical records of the associated application framework.

The results of data mining is often a decision tree classification or a cluster analysis. From these trees and clusters the underling if-then-else rules of behavior can be extracted.

These approaches are *not*, however, self-sufficient techniques for building models. Models require the synergy (cooperation, collaboration, and symbiosis between human experts and other forms of knowledge). This means that the model evolution process involves a discovery phase (the data mining component) and a model construction and validate phase (the subject matter expert (or SME) component). These phases are usually iterative. They are repeated until the model outcome consistently converges on a prediction or classification that falls within a small standard error. Figure 1 shows this fusion of knowledge discovery and subject matte expertise in the model development cycle.

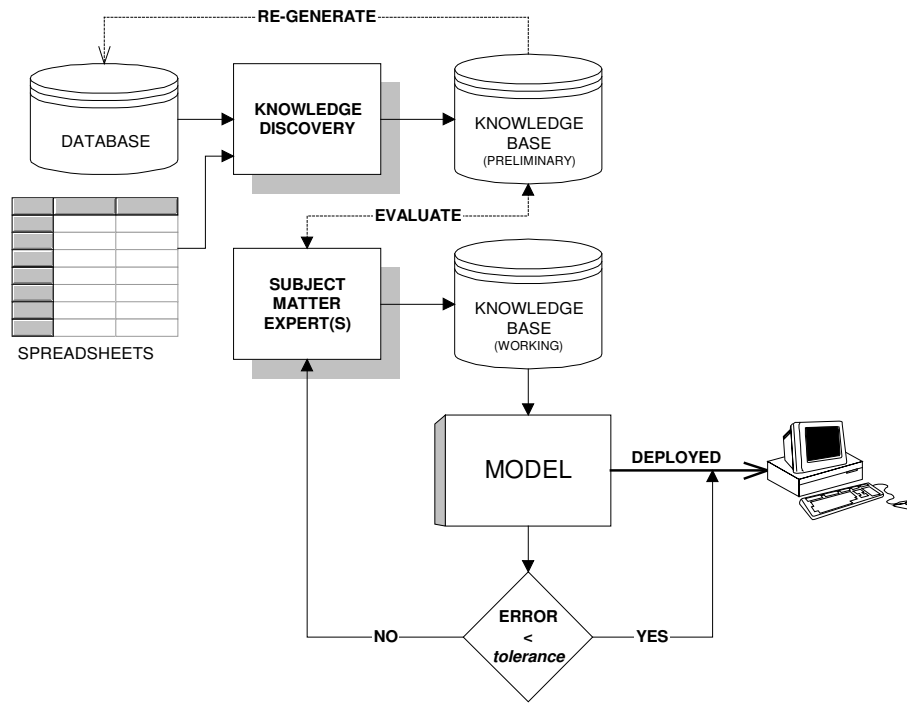


Figure 1 The Protocycling Development Cycle

Initial rule sets discovered by the data mining process are evaluated and, if needed, tuned by the subject matter expert. Rule discovery may involve many re-generation cycles as the parameters of the rule induction engine are tuned or modified to extract rules based on more focused knowledge (often these parameter changes involve adding or deleting variables or fine tuning the fuzzy sets associated with variables). The induced rules become fused with those elicited from the subject matter expert to form the model's working knowledge base. At this point, the model is executed against validation data. If the error is within acceptable tolerances, it is deployed, otherwise we start another cycle of refinements.

The Methodology

Data Mining, or, more properly, knowledge discovery, is the process of uncovering behavior patterns buried deep in large quantities of raw data. This methodology follows a rule induction technique to actually build a working process model of these behaviors. The actual model development process consists of several steps, as illustrated in Figure 2, below.

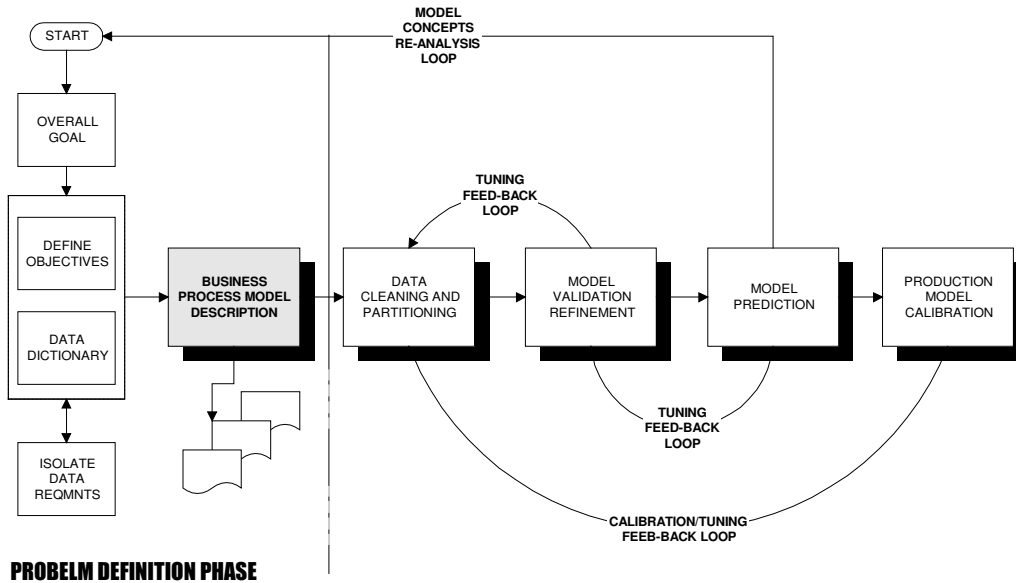


Figure 2. The Knowledge Discovery Methodology (Problem Definition Phase)

Problem Definition is a crucial first step. At this juncture we decide what we are attempting model, its basic components, and the nature and meaning of the data elements. Two critical outcomes are generated in this phase: a statement of the model Objective(s) and a complete Data Dictionary. Although this may seem elementary on the surface, the actual specification of project objectives is absolutely crucial to the success of the project. The objective statement defines what we (and the client) expected from the model, how it will be judged and evaluated (when will we know, as a not so trivial example, when the model building process is complete?), and what decisions will be made based on the model output. The objective statement also indicates the kind of model we will build (optimization, forecasting, analysis, or comparison, as an example) and the kind of knowledge discovery technique required (supervised or unsupervised machine learning)

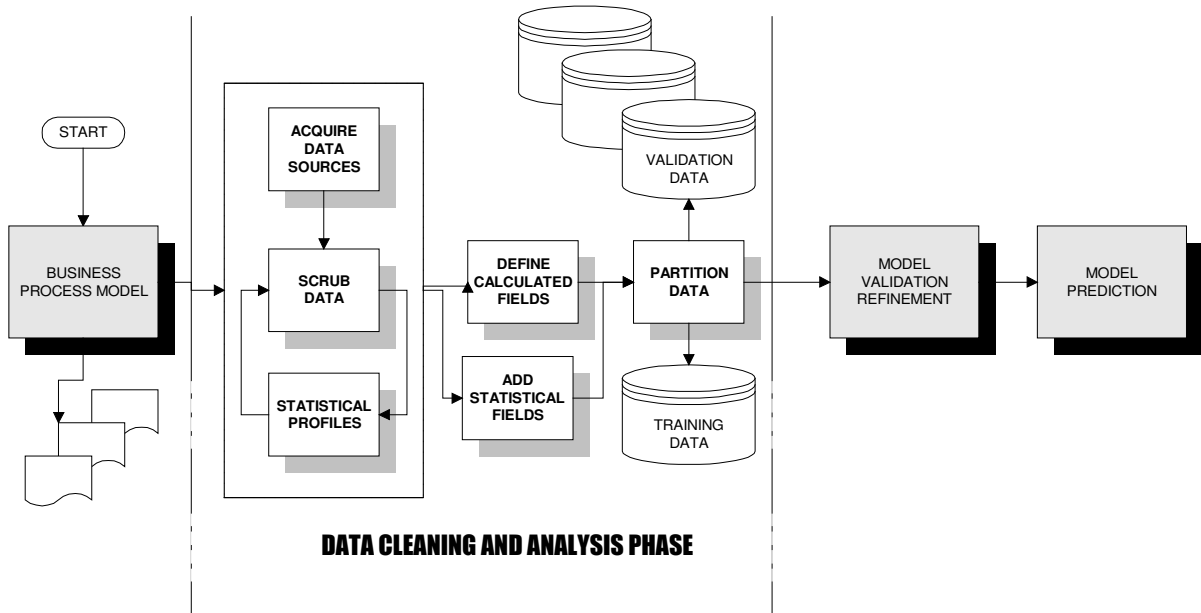


Figure 3. Data Cleaning and Analysis Phase

The next phase in the knowledge discovery process is data cleaning and analysis (see Figure 3, above). This is the most time consuming and, for many organizations, the most difficult, part of the process. Using the data dictionary, access to all the data sources must be secured. The critical elements for analysis are isolated and a process of scrubbing and purifying the data begins. Data cleaning starts with an assessment of the data properties and moves to making the complete data space as consistent and error-free as possible.

Data Type Classifications

We need to identify continuous, categorical, nominal, and ordinal data. Continuous and ordinal data must be reviewed for their potential use as fuzzy parameters. If we select one or more elements as fuzzy indicators, then the domain of the data must be decomposed into an odd number of semantic fuzzy sets by the subject matter expert or automatically by the data mining tool itself [although this automatic process tends to obscure the semantics of fuzzy parameters]. Data may also need to be normalized (divided through by the maximum, non-outlier value), rounded, or transformed in another standard value. Categorical and especially nominal data may need to be transformed into enumeration data sets (state names or sex into numbers, cities into distance metrics from a common demographic or geographic site, etc). Table 1 outlines some of the properties of these data types and the appropriate descriptive statistics.

Data Measure Scales

Type	Characteristic	Appropriate Statistics
Nominal	Unique Classifications with a restricted set of "named" values.	Mode
Categorical	Unique Classifications with a relatively unrestricted and "unnamed" values.	Mode



Ordinal	Ranking or Rating	Median or Percentiles
Interval	Known Difference Between Any Two Points	Mean, Standard Deviation
Ratio	Known Difference Between Any Two Points	Mean, Standard Deviation

Table 1 Measurement Scales

The mode is the most frequently occurring value, but mode alone does not describe the distribution of nominal values nor does mode provide insight into the correlation between frequency of nominal values and the distribution of other variable values. We often need to draw a distinction between nominal and categorical data. Generally, nominal data has a name -- such as sex, (“male” and “female”), state names, procedure codes, disease names, toxins, etc. Categorical data consists of non-continuous data elements usually without names (such as social security numbers and healthcare provider identifiers). Nominal data with hundreds or thousands of names (such as medical procedure codes) becomes *de facto* categorical data. This distinction is often very important in classification models. Models with nominal data provide relatively easy ways to partition the decision space. On the other hand, models that use categorical data are much, much harder to partition.

Ordinal and Categorical data

Ordinal (ordered or ranked) data represent points in a decision space that are sequenced relative to all the other points. Ordinal data reflects serialization but does not indicate degree of separation, thus, in many cases, ordinal data can be treated as nominal data. (Of course, if we have even some slight measure of distance (either incremental or relative) then an approximation to the model behavior can often be determine through statistical measurement techniques.) We can perform non-parametric statistical analyses on ordinal data using such techniques as the Spearman rank-order correlation or the Kendall’s Tau analysis. In general, an ordinal variable is a categorical variable having an ordered relationship without a distance relation. An example is the gold, silver and bronze medals in the Olympics. We should note that there is also a related variable type called *periodic*. This is a variable for which a distance metric exists, but no order relationship. Examples here include day (or time) of the week, month or year.

There is a close relationship between ordinal and nominal data in fuzzy systems implementing linguistic variables. Linguistic variables approximate terms used in everyday English. As an example, Age, with values “Young”, “Middle-Aged” and “Old”. These values are nominal with respect to Age and also have an ordinal rank. Figure 4 illustrates this idea with the variable Age decomposed into its component fuzzy sets.

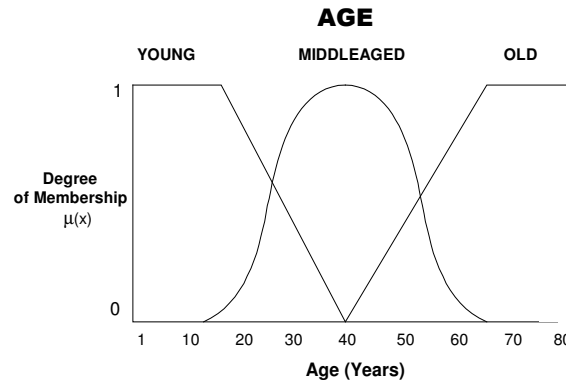


Figure 4. Ordinal Fuzzy sets for Variable Age

There are two reasons why these types of ordinal values lack a distance metric. First, they are context dependent within the model (the position is subjective) and second, there is generally no crisp boundary between neighboring classes. This final reason means that classes overlap (an age can be, to some degree “Middle-Aged” and also to some degree “Old”) so that a data element can belong to both classes simultaneously.

Discrete and Continuous Data

Model variables fall into two general classes: *discrete* and *continuous*. Nominal and Ordinal variables are discrete. These values come in “lumps”. Of course, many ordinary variables are discrete, such as age, waist size, number of moons in the solar system, number of patients treated per day, etc. Fuzzy sets, although they consist of many values (the underlying membership function) are still ordinal variables because they are referenced only by name. Continuous variables have values that span a continuum of values. Normally these variables have values that can take on any arbitrary value and have a fractional part. Examples include billing expenditures (costs), weight, height, and so forth. For a sufficiently large range of values, discrete values can often be treated as continuous.

Interval and Ratio Data

Interval and ratio variables constitute the major data types in most models. These are generally continuous variables (although variables such as Age, which are integer valued, are also interval data. In most cases, the domain or range of values for interval values integer fields allows their treatment as continuous variables). Interval means a value for which a distance metric exists. Thus the expression $X=Y-Z$ is valid for all sets of interval valued numbers. Ratio values represent (Y/Z) , often interpreted as simple percentages. Both kinds of data are found throughout large databases, representing

Scrubbing and Transformations

Large databases contain noise. Noise is an increase in entropy or disorder in the information content of the database and results from such common error conditions as:

- Missing values
- Wrong values

- Ambiguous or Conflicting field names
- Obsolete values
- Outliers

These errors propagate through databases through a lack of uniformity in field integrity controls (allowing inconsistent data encoding) and a lack of adherence to a uniform naming convention. Databases owned and managed by autonomous corporate divisions or government agencies are relatively immune from the problems caused by conflicting or ambiguous field names, however they are not immune to the more widespread and critical problem of data-resident noise.

The movement toward corporate or agency-wide data warehouses or across-the-board data mining projects often highlights and intensifies these local problems. Merging private databases (physically or logically) into corporate repositories often means a complete re-engineering of the organization's data management and information technology regime. The same is true, unfortunately, for most data mining projects. Ignoring the hurdles imposed by the politics of data and the barriers imposed by security considerations, acquiring data from multiple databases requires a significant post-modeling commitment to data scrubbing and normalization.

Resolving Anomalies, Outliers, and Noise

Data cleaning and scrubbing includes the resolution of anomalies in the data, deciding how to handle missing data, how to handle the distinction between empty and NULL data elements, how to accommodate noise in the data, how to deal with the believability of data, how to combine common types of data from different sources or from different data representations (chemical reaction data from ten years ago versus recent data where the types of tests have changed, the instrumentation has changed, the measuring criteria have changes, etc).

Recognizing and handling extreme values are critical parts of data cleaning. Extremes – sharp departures from the normal distribution of values in a population – are called outliers (since, obviously, they lay outside the normal range of values for the variable). Figure 5 shows some outlier values in a series of sales data.

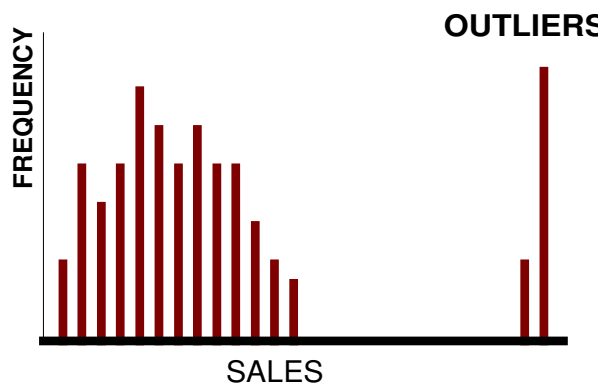


Figure 5 Outlier In a Sales Distribution

Analysts must exercise care in dealing with outliers. Any number of factor influence or perturb a population and hence introduce outlier values. An outlier can be:

- Noise. Missing and erroneous field values comprise the majority of outlier values. Outliers of this class typically indicate a data file with a high noise ratio (after all, outliers are simply the values which happen, probably as a random occurrence, to lie well outside the core range of values. If your data file has several outliers, it will almost certainly have errors in the normal part of the population's distribution.)
- New or residual values from a previous and no longer valid field domain. If your method of calculating sales performance significantly changes, then the new values may be reflected as outliers. After awhile, when enough new measurements have made their way into the database, the old values will eventually be viewed as outliers.
- An actual pattern representing some anomalous and possibly important behavior (fraud detection, as an example, focuses on the identification of anomalous behavior patterns). In fact, it is these unexplained outliers that often become the data miner's "nuggets of gold."

When outliers are forms of noise, we can compensate by using two possible forms of the trimmed mean. In practice, we replace the ordered series vector $x_1, x_2, x_3, \dots, x_i$ with weighted averages: $w_1x_1, w_2x_2, w_3x_3, \dots, w_ix_i$. Values within the series are ordered from smallest to largest. The standard trimmed mean encompass a more general statistic defined by the formula in Expression 1.

$$\begin{aligned} \bar{x}(a,b) &= \sum_{i=a}^b w_i y_{(i)} & 1 \leq a \leq b \leq N \\ w_i &= \frac{1}{(b-a+1)} & a \leq i \leq b \\ &= 0 & \text{otherwise} \end{aligned} \quad (\text{Exp.1})$$

Another averaging statistic is the *Winsored* mean. This class of mean generator is described by the equation in Expression 2.

$$\begin{aligned} \bar{x}(a,b) &= \sum_{i=a}^b w_i y_{(i)} & 1 \leq a \leq b \leq N \\ w_i &= \frac{1}{N} & a \leq i \leq b \\ &= \frac{a}{N} & i = a \\ &= \frac{(N-b+1)}{N} & i = b \\ &= 0 & \text{otherwise} \end{aligned} \quad (\text{Exp. 2})$$

We can use these formulae to trim the actual structure of a series by adjusting the values of a and b . As an example, setting $a=2$ and $b=N-1$ excludes the lowest and highest values in the series. Sliding a and b inward or outward provides a method for observing the affect of removing or including more and more outlier points.

Linear Conversion of Nonlinear Data

Nonlinear data may need to be made into linear data through logarithmic transformations. In unstructured or unsupervised data mining, you can directly include nonlinear data. However, when fitting



linear, polynomial, and growth curves you may need to convert the data or the associated expression into a linear form through such techniques as logarithmic transformation.

Data Enhancement and Virtual Fields

Another phase of scrubbing includes the enhancement of the raw data through (1) the addition of statistical indicators and (2) the addition of virtual or calculated fields. It is this last enhancement, the design of calculated fields, that often plays a critical part in the data mining process. As an example, we might define fields such as,

```
define CustSvcTime (float)
    = sum(CustCallLength)/count(CustCalls)
```

which defines the average time taken to service a customer by totally all the customer call time lengths and dividing by the total number of customer calls. This might be an interesting ratio. During data cleaning and partitioning, all the computed fields are calculated and run through the same kind of statistical analysis as the raw data fields. We need to insure that calculated fields do not carry with them unexpected relationships.

Statistical Analysis.

After data acquisition and data cleaning, all data mining projects start with statistical analysis. In fact, a broader view is needed. All model building activities must have, as an integral component of their process, a statistical analysis of the underlying data. The lack of data visualization and data structure understanding has been a critical contributor to the failure of many expert system projects. You cannot reliably divorce data knowledge from expert knowledge.

Statistical analysis begins with *data profiling*. We need to run a thorough statistical analysis on the data to discover such basic information as the compact data domain range (the minimum and maximum values for the bulk of data elements), the relative frequency of data elements (for both continuous and categorical data, within cluster buckets), mean and standard deviation of all numeric data, and the nature of outliers in the data.

Descriptive Statistics

At a minimum, data profile must include basic descriptive statistics. The first of these, shown in Expression 3, is the average or mean of the data. This is the distribution's central value.

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N} \quad (\text{Exp. 3})$$

The mean is not always a good indicator of the average of central value. For moderately small series, the mean value is highly sensitive to outliers – values that bias the measure of central tendency. As an example, consider the array of values in Table 2

Series	Mean
2, 4, 4, 5, 5, 7, 7, 7, 7, 9	5.7

Table 2 Mean of a Regular Series

The average of this series is 5.7 ($\text{sum}(x_1..x_{10})/10$). However, a single outlier distorts the mean, As an example, consider the array of values shown in Table 3,

Series	Mean
2, 4, 4, 5, 5, 107, 7, 7, 7, 9	15.7

Table 3 Mean of a Series with an Outlier

The average is now 15.7 a value highly unrepresentative of the series. As the number of values in a series increases the contribution by outliers decreases. This decrease in affect is due to the Law of Large Numbers. In some cases, the Median rather than the mean is more representative of the series. The Median is the number at the center of the series – the value x_i with an equal number of values above and below it. Finding the mean involves sorting the series. Expression 44 shows how the median is computed.

$$x_{med} = \begin{cases} \frac{x_{N+1}}{2} & \text{odd N} \\ \frac{1}{2}(x_{\frac{N}{2}} + x_{\frac{N}{2}+1}) & \text{even N} \end{cases} \quad (\text{Exp. 4})$$

Turning back to Table 3, the series with an obvious outlier, Table 4 shows the calculation of the median value. Although slightly above the average when we excluded the outlier, the median is still more representative of the series than the average.

Series	Median
2, 4, 4, 5, 5,7 , 7, 7, 9, 107	6.0

Table 4 Median of a Series with an Outlier

Another fundamental statistic is the variance. While the mean characterizes the distribution's central value, the variance characterizes its diffusion or width. This tells how tightly the data is packed or clustered around the mean. A conventional measure of width is shown in Expression 5. This is called the standard deviation.

$$\sigma = \sqrt{\sum_{i=1}^N \frac{(x_i - \bar{x})^2}{N}} \quad (\text{Exp. 5})$$

Two populations with the same mean can have very different standard deviations. Figure 6, as an example, shows two bell shaped distributions with exactly the same mean. However each distribution has a different compactness relative to the mean.

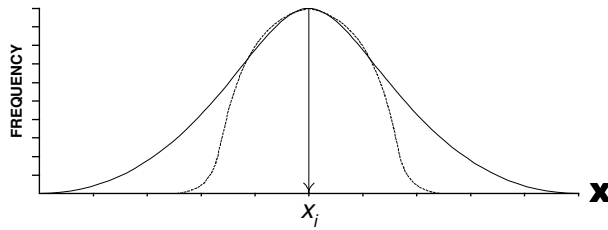


Figure 6. Distributions with Different Standard Deviations

In addition to mean and standard deviation, basic statistical profiling should also investigate two other important population characteristics: the skew and the kurtosis. Skew indicates the degree of asymmetry in the distribution relative to the mean. Expression 6 shows how the skew for a distribution is calculated.

$$skew = \frac{1}{N} \sum_{i=1}^N \left[\frac{x_i - x}{\sigma} \right]^3 \quad (\text{Exp. 6})$$

Skew is expressed as a positive or negative number (without units of measure). A positive skew value indicates that the asymmetry extends out toward the positive values along the X-axis. This is also called a right-hand skew. A negative skew value indicates that the asymmetry extends back toward the negatives values along the X-axis. This is also called a left-hand skew. Figure 7 shows two curves with left and right handed skew.

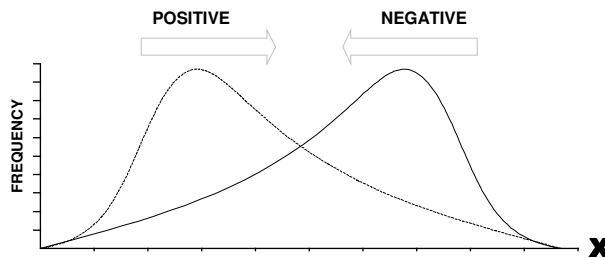


Figure 7. Positive and Negative Skew

Kurtosis is a measurement of distribution curve shape. Kurtosis reflects the relative degree to which a curve is flat or pointed or rounded. Expression 6.7 shows how this quantity is calculated.

$$kurtosis = \left\{ \frac{1}{N} \sum_{i=1}^N \left[\frac{x_i - x}{\sigma} \right]^4 \right\} - 3 \quad (\text{Exp. 7})$$

The measure is relative to the normal distribution curve (hence the “-3” which makes the value zero for normal distributions). Figure 8 shows a variety of curves each with a different kurtosis.

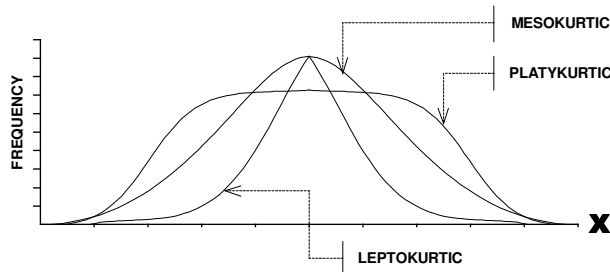


Figure 8. Curves with Various Degrees of Kurtosis

A curve with positive kurtosis is pointed and is called **leptokurtic** (from the Greek, *lepto*, meaning sharp). A curve with negative kurtosis is flattened and is called **platykurtic** (from the Greek, *platus*, meaning fat). Normal curves have zero kurtosis and are called **mesokurtic** (from the Greek, *meso*, middle).

Another population measure is the mode. This is the most frequently occurring value (or for a probability distribution function, the value of x_i where it assumes its maximum value). Recognizing multi-model distributions is very important in building fuzzy models, since the existence of several models often determines the center points for underlying fuzzy sets. Modes are easily recognized by visually displaying the population's data distribution. Modes are also important properties for categorical and nominal data.

Mean of Frequency Distributions

Data mining often involves tables of stratified data, essentially frequency tables. These form the basis for revealing patterns in multi-dimensional data sets. Data profile almost always involves generating frequency counts of candidate model variables. Finding the mean and standard deviation of frequency data is an important analytical process. Table 5 shows the number of clients visiting a client's e-business web site per minute.

		Customer Visits Per Minute (x00)									
		1	2	3	4	5	6	7	8	9	10
Site 1		9	32	26	19	24	39	17	11	8	0
Site 2		18	21	23	36	41	22	15	13	5	1

Table 5 Frequency of Web Site Visits

The data in table 5 compresses two collections of 190 access tracking counts into a frequency table. Computing the mean of the visits requires re-computing all 190 readings. Expression 8 provides a simplified formula.

$$\bar{x}_f = \frac{\sum_{i=1}^N f_i c_i}{\sum_{i=1}^N f_i} \quad (\text{Exp. 8})$$

where N is the number of categories or classes (ten [10] in this example), f_i is the frequency in the i^{th} category, and c_i is the response value of the i^{th} category. Using this approach, Expression 9 calculates the sample variance for a frequency table.

$$s^2 = \frac{\sum_{i=1}^N f_i (c_i - \bar{x})^2}{k - 1} \quad (\text{Exp. 9})$$

where $k = f_1 + f_2 + \dots + f_N$ and is the sample size. Analyzing the average and the variances for frequency tables tells the knowledge engineer much about the information content in these tables. Many data mining projects in managed health care, pharmaceuticals, and retailing actually use frequency tables as their “raw” data.

Correlations and Hidden Variables

We must also decide whether there are any explicit, implicit, suspected, or underlying correlations in the data (correlated data will directly affect the behavior discovery process used to generate model rules.) In many cases you will need to run least squares regression or polynomial (nonlinear) regression against your data to determine whether or not a correlation exists and the degree to which variables X and Y are correlated (of course, you should also be aware that an apparent relationship between X and Y may, in fact, be caused by another variable Z .)

Data profiling not only provides an insight into the data (identifying outliers, as an example), but can raise important warnings about inconsistencies and definitional problems that might destroy or restrict the usefulness of the data analysis. Data profiling is not a stand-alone statistical enterprise, but part of the data cleaning, aggregation, stratification, and consolidation process that analysts wade through during the initial discovery and design phase of a model.

Factor Analysis

Factor analysis is closely related to statistical analysis and involves reducing the number of input variables to a model. There are several approaches to this, one of the most common is Principal Component Analysis (PCA). Basically, principal component analysis identifies a k -dimensional subspace of the m -dimensional input space that appears most significant, and then projects the data into this decision space. In performing this transformation, the number of input variables is reduced from m to k . PCA generates orthogonal vectors that maximize the variance in the model. Through this technique we can reduce the dimensional complexity of model (dimensional explosion is a problem in nearly all real world models).

Another factor analysis technique removes (all but one) highly correlated variables. These variables are recognized through standard statistical correlation tests (such as the t -test) or through visualization of the data with conventional scatter graphics. Other factor analysis techniques for eliminating unimportant input variables include ranking variable importance using a step-wise linear regression (selecting the top n variables); stochastic selection, and forward and backward feature selection. In the forward approach we start with p one-input models and select the best. We then create $p-1$ two input models (combined with the best one input model). This is repeated until you either reach the desired number of variables or until the entropy of the model stabilizes (that is, until adding an additional variable does not increase the amount of predictive information above a specified threshold point). Backward

feature selection starts with p models each containing a different and unique mixture of variables. The best models are selected as variables are removed and intermixed. Both forward and backward factor analysis are generally implemented using genetic algorithms.

Data Cleaning and Model Training Files

The outcome from this phase is a set of data files representing the cleanest possible data. This data will be used to create the actual model. A final step in the data cleaning and analysis phase is the creation of the rule induction data sets. This consists of two data repositories: the training set and the validation set. Generally we have a single training set (consisting of a very large amount of data) and perhaps fifteen to twenty validation data sets. The training set is used to evolve the model and, hopefully, contains all the patterns we want to discover. Validation sets are used to reduce the error in the model. This brings us to the next phase.

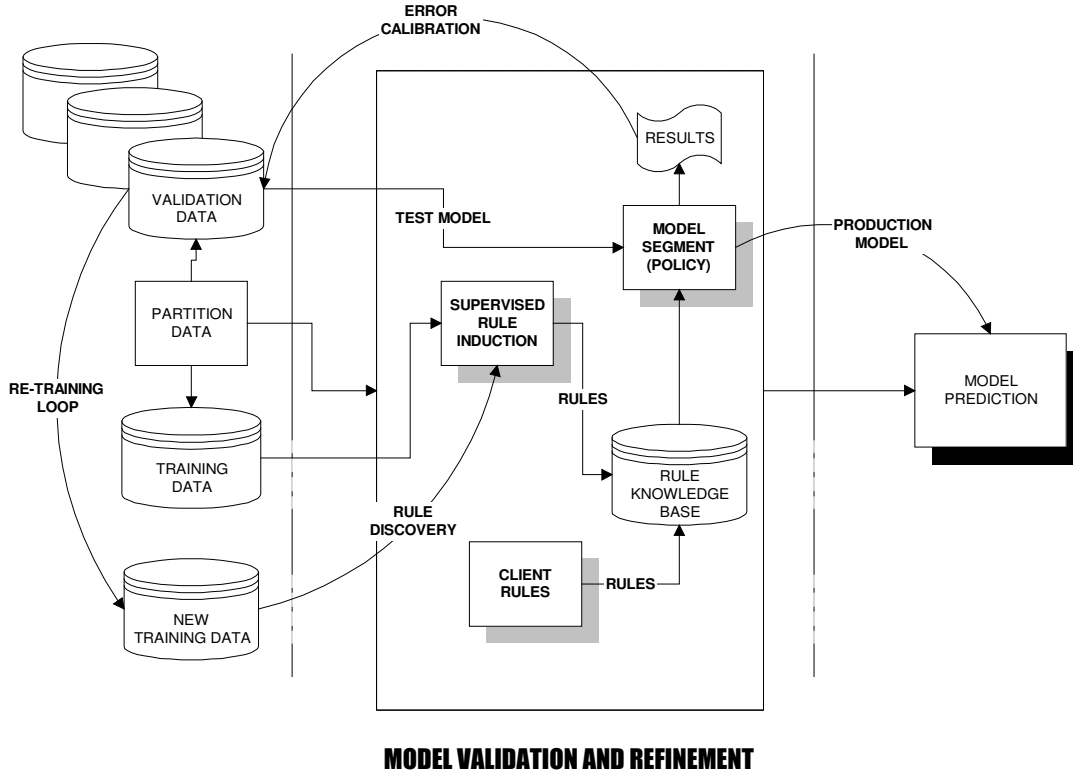


Figure 9. Model Validation and Refinement

The Model Validation and Refinement phase (Figure 9) actually generates the initial computer model of the business process. The model evolution procedure uses an advanced supervised data mining technique to discover behavior relationships in the data. A supervised technique attempts to find the rules that map the changes in a set of independent variables to one or more dependent variables. We can visualize this as a functional mapping between the R^n -dimensional surface created by the variables,

$$V_{d(t)} = f(v_{i(t)}^1, v_{i(t)}^2, v_{i(t)}^3, \dots, v_{i(t)}^n) \quad (\text{Exp. 10})$$

where “t” represents some time varying (lead or lag) relationship. This the behavior (state or value) of dependent variable V at time “t” is a function of the independent variables at various times. The complete process also selects arbitrary clusters of independent variables to discover deeper inelastic (static) and elastic or periodic relationships between the independent variables,

$$v_{id(t)} = f((v_{i(t)}^1, [v_{i(t)}^2], v_{i(t)}^3, \dots, (v_{i(t)}^n \dots))) \quad (\text{Exp. 11})$$

Naturally, of course, one or more independent variables may not be related to the behavior of the current target dependent variable and the rule induction process automatically adjusts for this situation. From this supervised rule induction mechanism a complete knowledge base of if-then-else rules is created. This knowledge base contains both induced rules (learned from the training set) as well as client rules, added by the subject matter experts. We now run the first validation set against the generated model and examine the output. Figure 10 details this process.

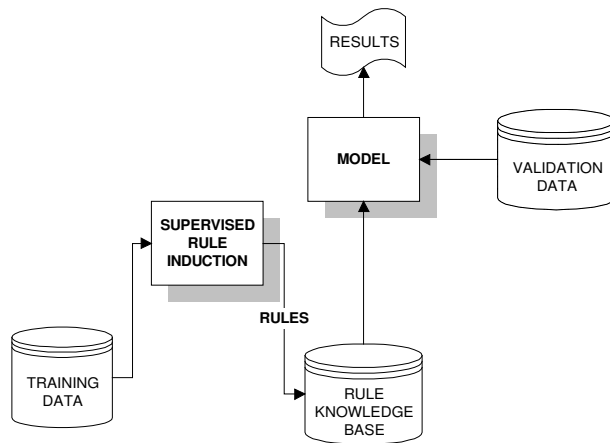


Figure 10. Validating the Model (detail)

A validation set contains a known value for the objective function. We are interested in how well the model predicts this known objective value. If the model does a good job of predicting the dependent variable for each validation set, then we have discovered the underlying system behaviors (that is, we have trained our model well.) If the model does not predict the objective function well, we need to re-train the model. Retraining involves operations such as,

- Increasing the size of the training file to include a greater variety of patterns. This is usually the first re-training approach since it yields the greatest opportunity to improve model performance without changing the actual properties of the model.
- Re-define the model by including additional variables, reducing the number of variables, or consolidating variables. This is often done by some form of factor analysis or automatic variable induction process. Two relatively simple ways to reduce variables or increase variables is through include a step-wise multiple regression (observing the affects of additional variables on the regression equation) and correlation analysis (observing which variables have a high degree of auto-correlation).

- Change the properties of one of more variables. This often involves changes to such characteristics as the variables' explicit domain (permissible range of values), the number, overlap, or shape of the underlying fuzzy sets, data type, or data organization (from continuous to categorical, as an example).
- Re-specify the rule generation and consolidation parameters in the rule induction process. Changes to the model controls often eliminates weakly contributing rules or, conversely, brings in rules that directly address outlier and ambient noise issues.

Figure 11 illustrates the re-training cycle along with the parameter tuning phase of the rule induction process. In practice, of course, models are re-trained in one way at a time.

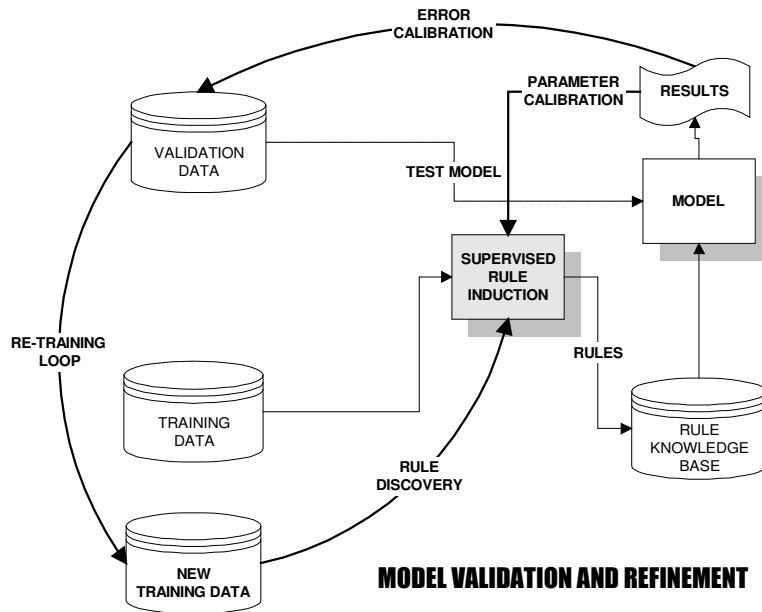


Figure 11. The Validation Tuning Cycle (detail)

This is done in one of two ways: we add the current validation set to the training set or (if possible) we create a new training set (perhaps by removing one of validation files and using it as a training file). Creating a new training set is always preferable. Figure 12 shows the details of the re-training process as a close-in loop.

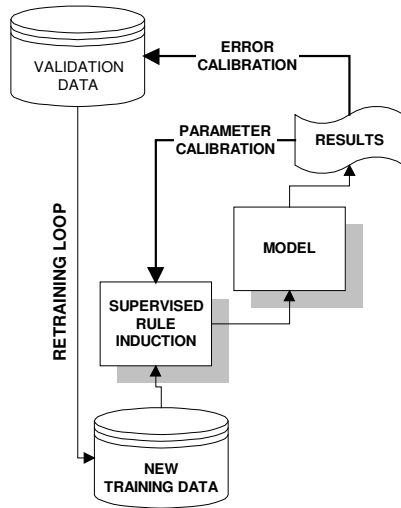


Figure 12. The Re-Training Cycle (detail)

Care must be exercised in training a fuzzy rule-based model. Failure to include a sufficiently deep and broad training file will leave the rule-based with pattern voids. These voids or gaps in the complete set of patterns reduces the accuracy of the model. On the other hand, too much data will over-train the model – generating rules that map almost 1:1 to each existing pattern. This makes the predictive or classification function of the model very brittle.

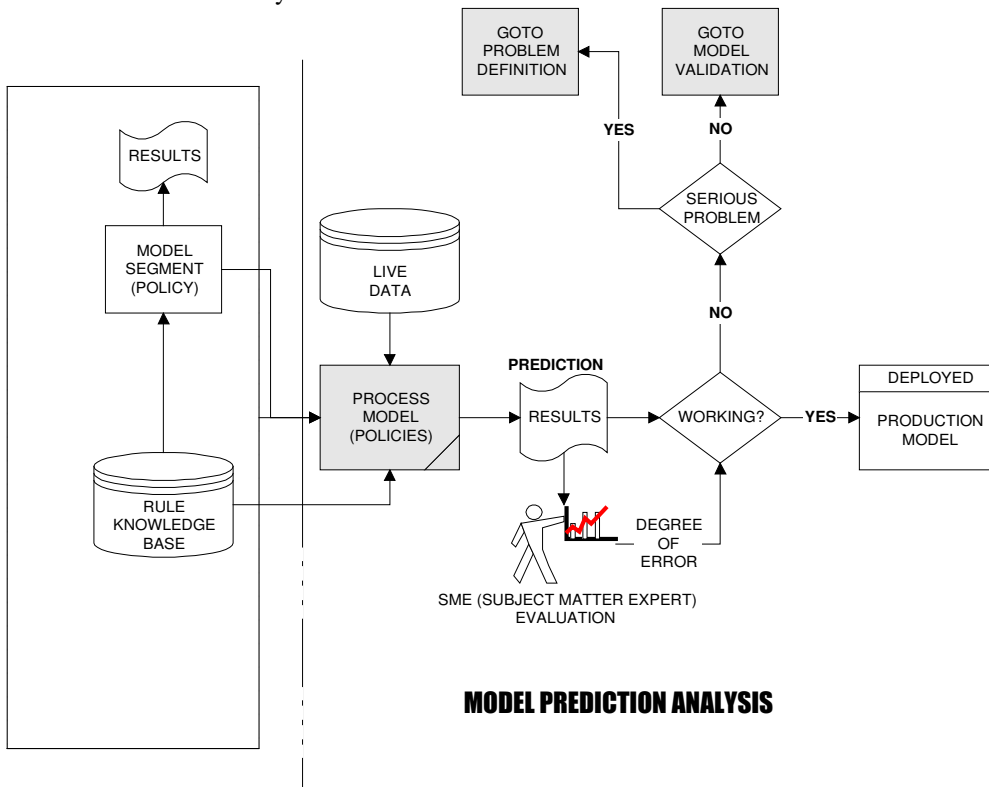


Figure 13. The Model Prediction Phase

Once a model has been created and validated with test data, we move to the prediction analysis phase (see Figure 13). In this phase the model is actually tuned and calibrated by the cadre of subject matter experts using live data (that is, data without known objective function values). The purpose of a model is forecasting; the prediction of future objective function states given a set of known independent variable states. We now feed the model a small set of “live” data and observe its ability to predict outcomes. Since we do not know the correct answer, the subject matter experts (SME) vote on the correctness of the solution. Figure 14 shows, in more detail, the process of evaluating the quality of a model by comparing its predictions with the judgement of the experts.

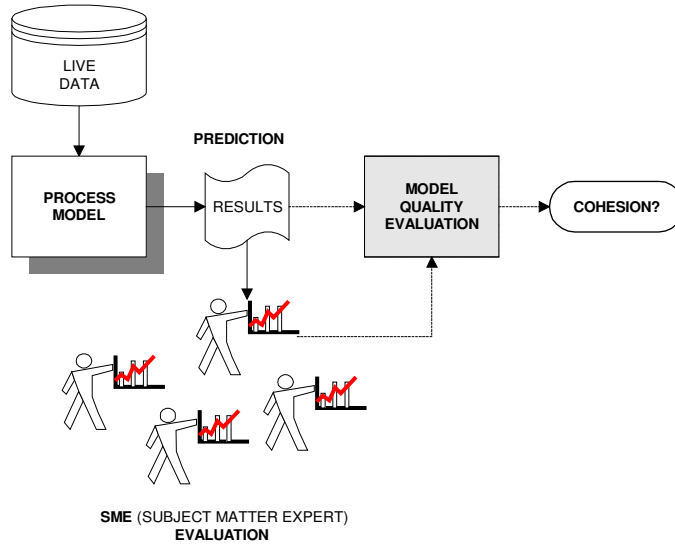


Figure 14 Model Quality Evaluation (detail)

A model performs well when the preponderance of SME votes, weighted by the proficiency of the expert, confirm the predicted answer. Expression 12 provides one way of judging the cohesion of expert judgments among peers of various competencies. The value F_j measures the goodness of fit for the j^{th} execution of the model.

$$F_j = e_m - \frac{\sum_{i=1}^N (e_{p_i} \times w_{p_i})}{\sum_{i=1}^N w_{p_i}} \leq \epsilon \quad (\text{Exp12})$$

where e_m is the model’s estimate (prediction) of the outcome variable; e_p is the subject matter or peer estimate; w_p is the competency weight associated with the peer and ϵ is some arbitrary error tolerance. When we run the model through K test cycles, the average goodness of fit, shown in Expression 13, is used to measure the model’s overall predictive capabilities.

$$\bar{F} = \frac{\sum_{j=1}^K F_j}{K} \quad (\text{Exp. 13})$$

When the average goodness of fit is within tolerances, the mode is deployed into field test use. If the model is not working well, the corrective action depends on the severity of the problem. For less severe problems, you need to go back and re-validate the model. For very severe problems, you may need to go back to the beginning of the model definition and re-evaluate the objectives of the project or the available data elements.

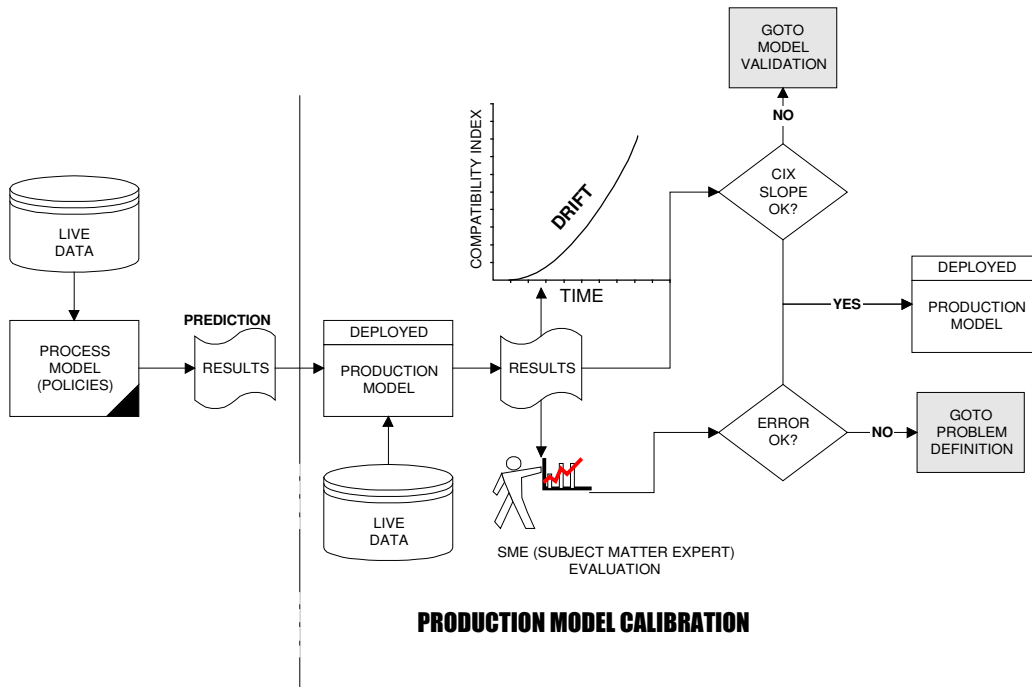


Figure 15. Calibrating the Production Model

Once a model is placed into production (and we call a model that has been in production for less than ninety days as field test simulation), we must monitor it for its predictive robustness (see Figure 15). A model can become less and less robust when the outside world changes. These changes usually do not happen quickly, but occur over a long period of time. Model calibration is done by a combination of synergistic methods: through an evaluation of the statistical compatibility index associated with the model's own execution and a judgment by the SME cadre of the model's predictive power. When the slope of the compatibility index stream tends significantly up or down, the model must be re-trained. If, however, the slope is within tolerances [.03 to .88], but the model is not producing correct answers according to the majority of the experts, then you may need to go back and redefine the nature of the model and the underlying support data.

In Conclusion

A Methodology is no substitute for common sense, the use the right tools, staffing with the right people, and a clear understanding of a data mining project's objectives and constraints. Nor is a methodology going to insure that you don't make mistakes or provide inexperienced analysts and managers with the requisite insights and courage to bring a project to a successful conclusion (assuming that a successful conclusion is both possible and desirable.) A methodology will, however, provide the project team with a roadmap that guides them in the design and evolution of the project. This roadmap is not cast in concrete. It must be adapted to corporate (or agency) cultures and augmented by the available technologies and the constraints of management.

References

- Booch, G., *Object-Oriented Analysis and Design With Applications*, 2nd Ed., New York, Addison-Wesley Publishing (1994)
- Booch, G., *Object Solutions: Managing the Object-Oriented Project*, New York, Addison-Wesley Publishing (1996)
- Freedman, D., Pisani, R. et alia, *Statistics* 2nd Ed, New York, W.W. Norton & Company (1991)
- McPherson, G., *Statistics in Scientific Investigation: Its Basis, Application, and Interpretation*, New York, Springer-Verlag (1990)
- Press, W., Teukolsky, S., et alia, *Numerical Recipes in C* 2nd Ed., New York, Cambridge University Press (1992)

For more information or to schedule a presentation call (919) 678-0477 or visit www.scianta.com



©2004 Scianta Intelligence, LLC
AR-PA-016