white paper

A Discussion of Data Analysis

Prediction and Decision Techniques

August 2012

table of contents

© 2012 Fair Isaac Corporation. All rights reserved.

table of contents

© 2012 Fair Isaac Corporation. All rights reserved.

table of contents

© 2012 Fair Isaac Corporation. All rights reserved.

» Introduction

This paper serves as a reference guide to analytic tools and approaches to data analysis, text mining, predictive modeling and decision analysis commonly found in the financial services, telecommunications, direct marketing and e-commerce industries, and to the latest analytic technologies developed in academia.

The purpose is to give non-technical readers some background into some of the new and popular prediction and decision technologies, and remind more technical readers of some of the key strengths and weaknesses. No attempt is made to make direct comparisons of techniques since the features tend to be application-dependent. Nor is this paper intended to be an exhaustive or complete discussion of each technique.

A Guide for the Non-Technical Reader

At FICO we sometimes classify analytic techniques as belonging to one of four areas—exploratory data analysis, predictive modeling, optimization and decision analysis. Many of the underlying technologies described in this paper are not confined to one of these categories, and may in fact be used in multiple areas. As a result, the paper itself does not impose a classification scheme on the techniques discussed. Rather we have simply listed them in alphabetic order for ease of reference. However, for this introduction, we have listed each technique as belonging to one of the four categories described below in order to indicate its most common area of use.

Exploratory analysis (or undirected data mining) seeks to establish relationships in the data to gain insight. Within this exploration, no specific outcome is assumed. An example of this group of techniques would be cluster analysis, used to develop a strategic marketing segmentation. Other techniques in this category are factor and principal component analyses.

Predictive modeling (sometimes called directed data mining) seeks to identify and mathematically represent underlying relationships in historical data, in order to explain the data and make predictions or classifications about new data. Predictive models are frequently used as ways to summarize large quantities of data as well as to increase the value of data. In the financial services, telecommunications, direct marketing and e-commerce industries, they are commonly used as inputs to decisions. An example would be the use of logistic regression to classify prospects as good or bad credit risks. Other techniques in this category are boosting, collaborative filtering, discrete choice modeling, discriminant analysis, scorecards, log-linear models, neural networks, pattern recognition, regression, support vector machines, survival analysis and tree modeling methods. Expert systems and RFM also fit into this category, but are different in that they can be derived judgmentally without historical data.

Optimization techniques seek to efficiently and effectively search across a set of possible solutions to a problem (either constrained or unconstrained) with the goal of maximizing or minimizing a particular mathematical function. Techniques in this category are genetic algorithms, linear programming and non-linear programming. Although we do not highlight them within the sections, several of the predictive modeling and decision analysis techniques rely on optimization techniques to reach their results.

Decision analysis goes one step further. By modeling the decision itself, it allows for the optimal decision to be identified. The purpose of decision analysis is to assist decision makers in making better decisions in complex situations, usually under uncertainty. Components of decision analysis discussed in this paper include key concepts and tools, graphical decision models, multiple objective decision analysis, sequential decisions and utility theory. Since decision analysis delivers the most value when coupled with active, continuous learning from observations, the need for well-planned

© 2012 Fair Isaac Corporation. All rights reserved.

or designed data is critical in the building of a robust decision model. For this reason, it is important to point out the section on experimental design, which addresses the importance and approach to well-planned data collection.

FICO Philosophy

At FICO, our years of experience with noisy and biased data and business constraints have led us to value domain expertise and analytic experience as key components in the modeling and strategy optimization process. An analytic technique, in and of itself, works only with the empirical data provided. Often, however, there is more contextual information that should be incorporated, either through automated capture of business intelligence or by the imposition of operational constraints. Such contexts might include the source of the data; its past and future reliability; its deployment mechanism; its cost; and the potential legal, operational or customer relationship impact of using certain types of data or using certain criteria for a given decision.

FICO favors techniques that allow for the incorporation of prior knowledge beyond that provided in a particular dataset in order to create a solution of greater value. You will note that some of the strengths and weaknesses listed for each technique allude to this point. While in other publications some technologies have been criticized for being naive, the scenarios discussed are frequently describing the naive analyst.

Organizational Structure

General Description

Each section is introduced with a brief one- to two-page discussion of the technique. Since you may not be familiar with some of the terms used in this paper, a glossary is included at the back. Phrases in italics are defined in the glossary. Some of these definitions were written to clarify the terms as they are used in this paper and ignore their broader interpretation.

Applications

To place the techniques in context, we have indicated some of their most common uses. When appropriate, we have noted particular business problems to which techniques have been applied successfully.

Strengths and Weaknesses

We have included strengths and weaknesses for the techniques, where appropriate, although these are not exhaustive lists. Rarely could a weakness in one situation be a strength in another, but often a weakness (or strength) might be irrelevant for a particular application or set of data. For example, an inability to handle missing values is only a problem when there are missing values. An ability to capture interactions in data is only a positive feature where these are suspected to exist. Multivariate normality assumptions are not a problem for linear regression if the data are, in fact, multivariate normal. Other issues to consider when evaluating analytic techniques include the use of categorical and/or continuous variables, the ease of interpretation of results, the robustness of solutions, the importance of sample size, the ability to handle multiple objectives and the ability to engineer solutions.

References

Where appropriate, some additional reference material is listed as suggestion for further reading.

© 2012 Fair Isaac Corporation. All rights reserved.

Revisions and Updates

We periodically (though not frequently) update this document, deleting some topics that have become less relevant and adding additional sections. In the current version, we have added new sections on Discrete-Time Hazard Model and Time-to-Event Scorecard, Ensemble Learning, Mathematical Programming, FICO's new Xpress Optimization features, and expanded upon the Decision Trees section.

We hope you find this paper useful and welcome your feedback on future improvements.

© 2012 Fair Isaac Corporation. All rights reserved.

» Boosting

Boosting is a general class of technologies for improving the performance of an existing prediction technology. Boosting works by sequentially applying a classification algorithm to reweighted versions of the *training sample*. Boosting works particularly well in practice when you apply it to a simple (weak) form of an existing technology. For example, consider tree technology. Normally when one applies a tree to solve a complex prediction problem, a very deep and complicated tree is produced, which works reasonably well as a predictor. However, it is possible to produce a very shallow tree with, for example, eight terminal nodes. By itself, this shallow tree would not work very well as a predictor; it is called a weak learner. Boosting is a way to put many (e.g., 500) weak learners together to produce a very powerful predictor.

Let's consider the *binary outcome* problem of estimating the log *odds* of a "Good." Log odds is a function defined on the input space of *predictor variables*—call it LO(**x**). Boosting technology will produce an estimate of log odds of the general form

$$\hat{LO}(\mathbf{x}) = \sum_{r=1}^{R} \lambda_r g_r(\mathbf{x})$$

where the $g_r(\mathbf{x})$'s are a whole bunch of weak learners developed by the prediction technology that is being boosted. So if a tree is the technology being boosted, then $g_r(\mathbf{x})$ is a shallow tree. The different varieties of boosting technology amount to different methods for deriving λ_r and $g_r(\mathbf{x})$. Two boosting technologies are discussed in more detail below.

Gradient Boosting

A very general boosting technology, recently developed by Stanford University professor Jerome Friedman, is called *gradient* boosting. Analyzing the progress of a typical *optimization algorithm*, which is used to find an estimate of log odds, inspires it.

The optimization typically takes place in some parameter vector space. An iteration of the algorithm is of the form

$$\boldsymbol{\beta}_{r+1} \leftarrow \boldsymbol{\beta}_r + \lambda_r \boldsymbol{g}_r$$

where $\boldsymbol{\beta}_r$ is the current parameter vector and g_r is the gradient of some objective function with respect to the parameter vector. The scalar λ_r quantifies the amount of movement of the algorithm in the direction of the gradient. There are many strategies for choosing λ_r. After the optimization algorithm has done its job, the optimal parameter vector can be expressed as

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_1 + \sum_{r=2}^{R} \lambda_r \boldsymbol{g}_r$$

For gradient boosting, the optimization is done in log odds function space rather than a finite dimension parameter vector space. In other words, gradient boosting is a *non-parametric method*. In function space, the gradient is a function, which can be crudely approximated non-parametrically using the underlying technology. In the case of trees, a shallow tree approximates the gradient. The result is a log odds estimate of the form

$$\hat{LO}(\mathbf{x}) = LO_1(\mathbf{x}) + \sum_{r=2}^{R} \lambda_r g_r(\mathbf{x})$$

where $LO_1(\mathbf{x})$ is an initial shallow tree estimate of log odds and the $g_r(\mathbf{x})$'s are shallow tree estimates of the gradients. For gradient boosting, the amount of movement of the algorithm in the direction of the gradient falls short of the *line optimization*. This idea of falling short (shrinkage) is used in gradient boosting as a way to avoid *overfitting*.

© 2012 Fair Isaac Corporation. All rights reserved.

The gradient in the description above is based on some *objective function*. Different objective functions lead to different results. The objective function takes as its argument the log odds function, which is a function defined on input space. So the objective function has the form, *F(LO)*, where *LO* is a function of the form $LO(\mathbf{x})$. This is an unusual way to look at scoring technology. For most technologies, the objective function is defined on some parameter vector space.

The objective function for non-parametric logistic regression is the log likelihood function. It can be shown to be

$$F(LO) = E\left[\left(\frac{y+1}{2}\right)\bullet LO(\mathbf{x}) - \log\left\{\mathbf{1} + e^{LO(\mathbf{x})}\right\}\right]$$

where y is +1 for the Goods and –1 for the Bads. The expectation is taken over the joint distribution of *(x, y)*. The Friedman, Hastie, Tibshirani reference below explores in great detail the boosting results based on this objective function.

AdaBoost

One of the early boosting techniques is called AdaBoost. It is based on the intriguing objective function

$$F(LO) = E\left[e^{\frac{-y}{2}\bullet LO(\boldsymbol{x})}\right]$$

Why does this objective function make sense? It turns out to be a differentiable—but crude—approximation to the *misclassification cost* objective function, when the two types of misclassification costs are equal. It is also quite similar in graphical shape to the more complex log likelihood objective function. If log odds is positive, then you classify as "Good," and if log odds is negative, then you classify as "Bad." So it is desirable to have $\frac{y}{2}\bullet LO(\boldsymbol{x})$ positive as much as possible. Hence, the AdaBoost objective function is to be minimized.

Gradient boosting applied to the AdaBoost objective function yields a fascinating result. It turns out that finding the gradient is closely related to estimating log odds with respect to a revised sample-weighting scheme. Whatever weak learner is being used, it can handle any sample-weighting scheme. For each iteration of the boosting algorithm, the sample weights are adjusted according to how the last weak learner classified the observations. The sample weights for those observations, which were classified correctly, are down-weighted. The amount of down-weighting depends on the margin of the correct classification. The sample weights for those observations, which were classified incorrectly, are up-weighted. The amount of up-weighting depends on the margin of the incorrect classification. This has a certain intuitive appeal. However, this characterization of the result is due to the idiosyncrasies of the AdaBoost objective function, which has no scientific basis. For example, use of the log likelihood objective function does not lead to this novel iterative sample-weighting scheme. However, boosting with the log likelihood objective function often leads to better results than those achieved by AdaBoost.

Applications

Boosting can be used to improve the performance of any classification or predictive technology, as long as interpretability of the resulting *score formula* is not at issue. ObjectBoost, a FICO invention, borrows ideas from boosting and support vector machines to yield a powerful way to customize a score development to optimize the solution for a specific business goal.

Strengths

- Improves performance of a stand-alone modeling technology.
- If the weak learner is simple enough (e.g., stumps—a one split tree), then the resulting score is an interpretable Generalized Additive Model (GAM).
- Can easily capture non-linear, non-additive relationships in data with proper choice of weak learner.
- No data structure assumptions.
- Handles both continuous and categorical predictors.
- Competitive with the state of the art for many objective functions.

Weaknesses

- Difficult to interpret unless the weak learner is very simple.
- Score engineering is not feasible with traditional boosting technology. An exception to this is FICO's ObjectBoost technology, which has score engineering built into its fabric.

References

Friedman, Jerome; Hastie, Trevor; and Tibshirani, Robert, July 23, 1998, *Additive Logistic Regression*: a *Statistical View of Boosting*, a paper downloaded from Jerome Friedman's web page.

Friedman, Jerome, March 1999, *Stochastic Gradient Boosting*, a paper downloaded from Jerome Friedman's web page.

Freund, Y. and Schapire, R. E., *Large Margin Classification using the Perceptron Algorithm*, Machine Learning: Proceedings of the Fifteenth International Conference. Morgan Kaufmann, 1998.

» Bump Hunting and Patient Rule Induction Method (PRIM)

Given a set of observations (X,y) where X is a vector of predictors and y is a response, bump hunting refers to techniques that map out local regions of predictor space where the response is unusually large or small. This can be considered function optimization. Without loss of generality we will restrict our discussion to maximization[1].

PRIM was introduced for bump hunting in high-dimensional predictor spaces by Friedman and Fisher (1999). This technique determines regions with high response in the form of low-dimensional boxes and/or subsets of categorical values. These "boxes" are defined by conjunctions ('AND'-rules) involving a few predictors, such as:

IF 25 <= Age < 30 & Sex=' Male' & 'Has Mortgage' & No. of Dependents > 3

➔ Very high response rate to credit card offer

Benefit of a rule is characterized by two measures:

- Mean response among the observations falling in the box
- Support, which is the fraction of the observations falling in the box

We seek high response rules with statistically and economically significant data support.

Peeling

PRIM starts with all observations and iteratively restricts the ranges of informative (about response) numeric variables. Similarly it restricts value sets of informative categorical variables[2]. Restrictions are introduced gradually, thus peeling thin layers from the data along various dimensions. Any given iteration determines a peeling dimension that maximizes response after removing the peeled observations. The approach is "patient" in the sense that only a small fraction (e.g. 10%) of the current observations are peeled per iteration. This ascertains that all dimensions have multiple opportunities to be restricted. Peeling stops when the mean response no longer increases or when support of the remaining box falls below a user-specified threshold.

Figure 1 shows peeling progress for a simulated example with two ordinal predictors. Predictor values were generated from a uniform distribution over a square ranging from -3 to 3. Response values were generated by a complicated nonlinear function of the predictors, as indicated by a contour plot. Response peaks near (x1=0, x2=1.6) indicated by the red dot. The initial box comprises the entire data range. Each iteration peels off 30% of the data [3]. The first iteration peels low values from the x2-dimension along the vertical. The new box has a higher mean response and is supported by 70% of the original data. The second iteration peels low values from the x1-dimension and the resulting box is supported by 49% of the original data, further improving response. The next iteration peels off high values of x1, etc. Each iteration peels off one facet of the previous box so as to maximize the new mean. After 11 iterations a small box remains that is located at the response peak and comprises 2% of the original data.

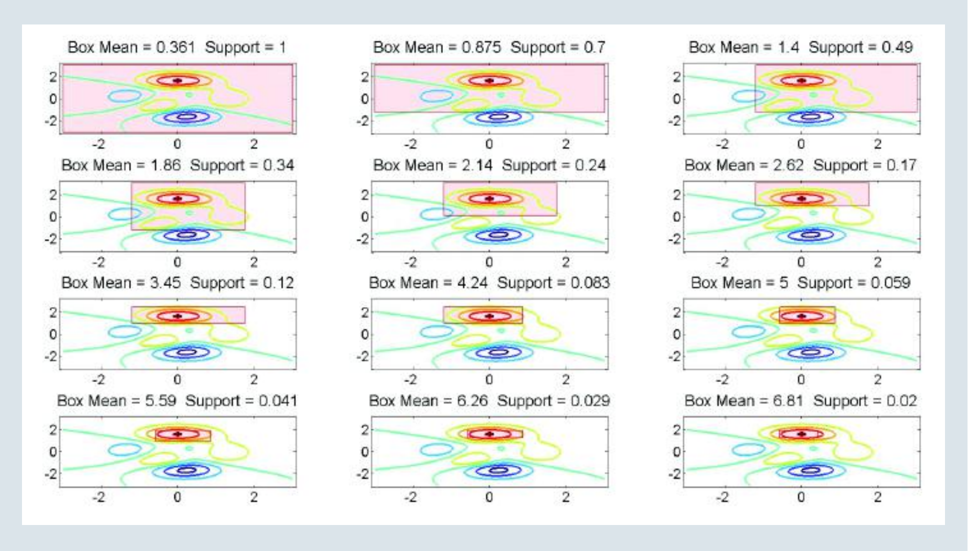
[1] For minimization we reverse the sign of y.

[2] This also accommodates variables with missing values, which can be treated as categorical values.

[3] For demonstration purposes we chose the peeling fraction higher than in real applications.

© 2012 Fair Isaac Corporation. All rights reserved.

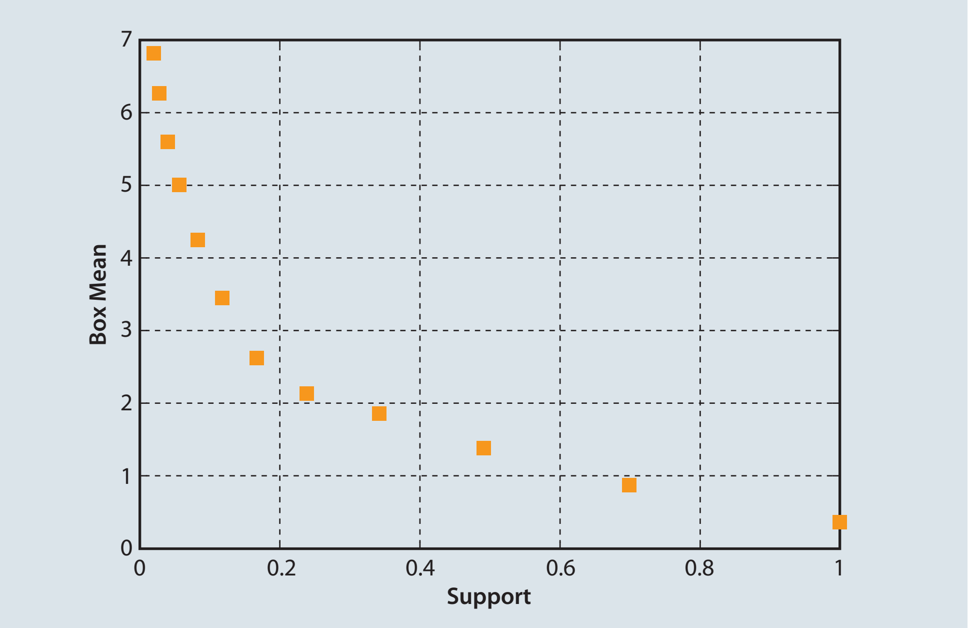
FIGURE 1: PEELING PROGRESS



Increasing response contours are colored from blue to red.

Associated with the iterations is a peeling trajectory that illustrates the tradeoff between support and response. Users may consider intermediate rules created during the peeling progress that strike any desired balance between response and support.

FIGURE 2: PEELING TRAJECTORY



The rightmost marker indicates the initial data set. A sequence of 11 peeling iterations generates performances from right to left.

Pasting and Rule Simplification

Rules created by peeling can sometimes be further improved by subsequent "pasting" and rule simplification. There it is tested whether enlargements (by iteratively adding thin layers) or removal of certain peeling restrictions can maintain a high response mean while simplifying the rule and increasing support. Due to the greedy nature of the peeling process, and in particular when predictors are highly correlated and peeling decisions compromised by noise in the data, it may be possible to relax or remove redundant or insignificant restrictions, resulting in simpler and more powerful rules.

Increasing Coverage

Rule discovery can be applied sequentially to generate a rule sequence k = 1, …, K whose union covers wider, flexible regions of the predictor space. Initially a working data set is defined comprising all observations. The k'th iteration consists of generating rule k from the working data at iteration k followed by removing observations conforming to rule k from the working data. This generates a new working data set for iteration k+1, etc. Iterations stop when the union of rules covers a desired fraction of the population or when the mean response from all the rules falls below a user-defined threshold.

Applications

The techniques described are applied to an increasing range of problems spanning from marketing to medical and materials research. For business analysts, this technique may be interesting to complement prediction and classification analyses, as it offers a very different approach to discover and to describe homogenous sub-populations with extreme behaviors. Not only does this help with customer understanding but it can also facilitate targeting and motivate customer treatments.

Strengths

- Gives users the ability to inform rule discovery.
- Handles numeric and categorical variables, also with missing values.
- Makes minimal assumptions on the data.
- Supervised: unlike clustering, can target for desired responses.
- Less greedy and more forgiving than methods that partition the data (e.g., CART).
- Rules are often simpler and stronger than rules defined by top leaves of CART.

Weaknesses

- Rules defining boxes may have many conditions making them less interpretable.
- A single rule discovers a local optimum, could get stuck in a sub-optimal region.
- Unions of rules covering wider regions can be more tedious to interpret.

References

Friedman, J.H. & Fisher, N.I. (1999). *Bump-hunting for high dimensional data, Statistics and Computing,* 9, 123–143.

» Causal Modeling

Causal modeling refers to techniques for inferring from data the effects of actions on outcomes. It is distinct from regression analysis which has no notion of causality. Causal models are of particular interest for financial services, retail, marketing and health care applications.

Given a set of observations (X,a,y) where X is a vector of account-level random variables (covariates), a is an action or treatment variable and y is an outcome or response, causal models generate estimates or predictions of the form:

$$\hat{y} = f(X,a) \qquad \text{(Action-effect model)}$$

or

$$\widehat{dy} = \hat{y}(a = A1 \mid X) - \hat{y}(a = A2 \mid X) = g(X) \qquad \text{(Uplift model)}$$

Action-effect models predict outcomes under multiple treatment scenarios based on X and a. Uplift or incremental response models predict differences in outcomes between two treatment alternatives based on X. This addresses important questions for decision makers:

- Does a treatment affect the outcome?
- Does a treatment affect the outcome differently for different customer types?
- Can we predict sensitivities of customers with respect to treatments or rank order customers according to these sensitivities?

Well developed causal models are valuable for decision making by objectively informing the most profitable treatments for customers. Action-effect models represent an important empirical ingredient of decision models where effects of alternative treatments on business outcomes are explored. This is frequently referred to as what-if analysis. Such models are increasingly used for optimizing customer treatments across financial, retail and health care applications. Uplift models are useful for marketing applications to rank order customers according to the *incremental* effects of offers on outcomes, such as sales.

Developing a causal model can be analytically challenging for several reasons:

- Effects of actions on outcomes can be subtle when compared to stronger correlations that may exist between covariates and outcomes.
- Actions can be strongly correlated with covariates.
- Unobserved variables simultaneously associated with treatments and outcomes can generate spurious correlations that will bias causal effect estimates.

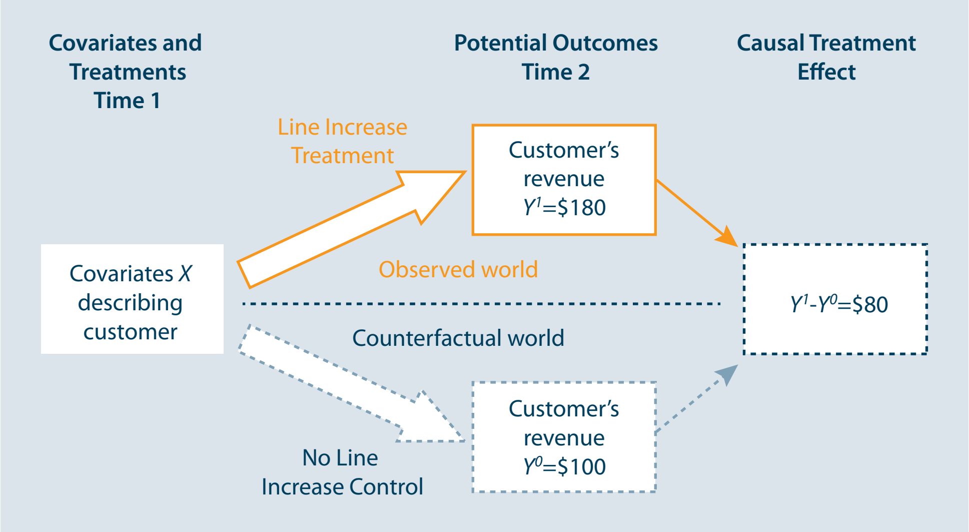
These challenges can lead to difficulties in identifying and estimating treatment effects, especially with non-experimental (business-as-usual) data and when important variables are omitted from the analysis. Naïve application of regression analysis or other data mining techniques can obscure potential difficulties in causal inference, which may lead to opaque analysis results and models that provide wrong answers.

These issues are addressed by the potential outcomes framework applied first by Rubin to the study of causation [Holland (1986)]. This framework defines causal effects, clarifies data conditions under which they can be estimated, and motivates transparent and robust causal modeling approaches.

Rubin Causal Model

The simplest formulation, as depicted in Figure 3, concerns dichotomous treatments, in the following referred to as "control" and "test"[4]. Applied to a credit line increase decision, the control might be no increase and the test may be an increase by a certain amount. Covariates *X* are measured prior to the treatment. For each customer, we posit a pair of potential outcomes Y^0, Y^1 under the control and test, respectively. A customer can only receive one treatment, so only one potential outcome is observed. The unobserved potential outcome is called a counterfactual. The customer-level causal effect is defined as the difference between his/her potential outcomes $Y^0 – Y^1$. The difference is unobservable as Y^0 and Y^1 are never observed together.

FIGURE 3: POTENTIAL OUTCOMES AND CAUSAL EFFECT FOR A LENDING PROBLEM


Covariates and Treatments Time 1
Potential Outcomes Time 2
Causal Treatment Effect
Line Increase Treatment
Customer's revenue Y^1=$180
Covariates X describing customer
Observed world
Counterfactual world
Y^1-Y^0=$80
Customer's revenue Y^0=$100
No Line Increase Control

Under certain conditions on the data known as *unconfoundedness* and *common support* [Rosenbaum, 1983] it is possible to estimate causal effects:

- *Unconfoundedness*, sometimes called *conditional independence* or *selection on observables*, requires that potential outcomes and treatments are conditionally independent given *X*. From a practical perspective, this means that all the covariates that influence the treatment and that are possibly associated with the outcome should be included in *X*.
- *Common support*, sometimes called *overlap*, requires (in its global form) that at each value of *X*, there is a nonzero probability of receiving each treatment; i.e., for every customer we find similar customers who received different treatments.

Warranting Data Conditions

Experimental design is the gold standard to fulfill these conditions. Random treatment assignment, possibly stratified by treatment probabilities depending on *X*, guarantees unconfoundedness. Global common support is ascertained if probabilities for the treatment alternatives remain nonzero for all *X*. Regression analysis or ANOVA can be used to model the effects of actions or treatment factors on outcomes.

[4] Extensions to more than two treatment alternatives are possible [Fahner, 2011a].

© 2012 Fair Isaac Corporation. All rights reserved.

Yet many business applications generate highly constrained experiments where certain treatment probabilities for certain customer types are reduced to zero, so there will not be global common support. However, local support regions may still be created, which produces some information for a partial estimation of causal effects.

While experimental data is highly desirable, modelers often need to rely on business-as-usual data generated by historic treatment policies with limited or no experimentation. For rule-based treatment policies, customers who received different treatments differ from each other; this is called operating point bias or selection bias. The analytic challenge is how to deal with this bias in a transparent and robust way. Sometimes it is possible to develop causal models from business-as-usual data, but only if the selection bias is small. Careful data collection and the following steps for the vetting of data conditions is necessary. Otherwise, the validity of causal inferences cannot be guaranteed.

- The process that generated the historic data should be well understood. All the covariates that influenced treatment choice and that are possibly associated with outcomes should be collected to bolster the unconfoundedness condition. Otherwise, there could be hidden selection bias and spurious correlations could bias causal effect estimates (omitted variables bias).
- The common support situation should be analyzed and overt selection bias understood and controlled for.

Matching on the Propensity Score

A good analytic approach for analyzing common support, deciding whether the data is rich enough for causal inference, and capable of removing overt selection bias, is called *matching on the propensity score*.

Propensity scores model customer-level treatment probabilities. Restricting our discussion to dichotomous treatments, a propensity score can be developed by estimating the probability of receiving one treatment over the other[5], depending on X, via logistic regression:

$$\Pr(a = A1) = h(X) \qquad \text{(Propensity Score)}$$

After the propensity score is developed, the score distributions conditioned on each treatment group are inspected (Figure 4). Ideally, the conditional distributions would sit on top of each other, indicating global common support. However, in many applications, customers in different treatment groups tend to be different from each other.

Completely separated distributions indicate severe selection bias and total lack of common support. Estimation of causal effects is then impossible without making strong, untestable functional assumptions for extrapolating functional relationships from one treatment group into the other. Deep domain expertise is highly recommended to guide these extrapolations (score engineering), while blind reliance on model-based extrapolation puts the credibility of data-driven models into question.

In more fortunate situations, the distributions will partially overlap (*local common support*). This happens when limited experiments are conducted or when historic treatment selection shows idiosyncratic variations (natural experiments[6]). If there are enough counts in the common support,

[5] Propensity scoring extensions are possible for applications that require more than two treatments.

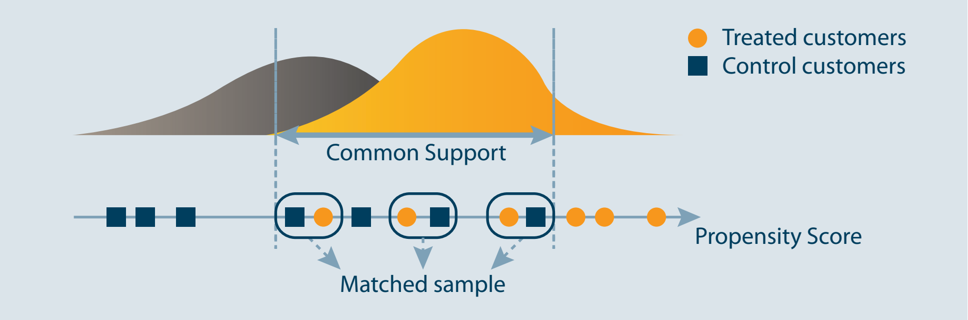
[6] Natural experiments are beneficial for common support but raise the specter of hidden selection bias. One has to judge carefully whether treatment variations have a truly idiosyncratic component, or whether they might be associated with potential outcomes, possibly violating the unconfoundedness condition.

© 2012 Fair Isaac Corporation. All rights reserved.

it is possible to infer causal effects for customers in the overlap region in a data-driven way, without relying on strong extrapolation assumptions.

If sufficient overlap is found, a matched sample can be generated by matching on the propensity score. A simple scheme, pair-wise matching, works as follows: for each customer in the common support region, find a "twin" customer who has a similar propensity score, but who received the other treatment. The resulting matched pairs constitute a matched sample, which is a subset of the original observation set.

FIGURE 4: PARTIAL OVERLAP OF SCORE DISTRIBUTIONS AND MATCHING ON THE PROPENSITY SCORE



Matching is very powerful as a precursor for estimating treatment effects. It can be shown that in the matched sample, "test" and "control" observations have the same distribution for all *X* available as candidate predictors for the propensity score. So the matched sample is similar to a sample that would have been obtained by assigning treatments randomly (quasi-experimental design). Common support in the matched sample exists globally and selection bias is removed.

Action-Effect and Uplift Model Development

Because of their preferable data conditions, causal models such as action-effect models can be developed by applying regression analysis to matched samples[7]. However, it is important to note that results will be based on the matched samples and not necessarily representative for the entire population. Moreover, matched sampling reduces sample size. Hence, while selection bias is removed, variance of the estimates can become enlarged, and provide more uncertainty in the models. Simplified models and estimation techniques such as penalized regression that reduce variance of the estimates are useful in this context.

An interesting application concerns the development of uplift models deriving from results of pair-wise matching. For each matched pair, the outcomes between the matched "test" and "control" twins are differentiated. This creates a new random variable *dy* associated with each *X*. The *dy* are then regressed on *X* to obtain the uplift model. The method regresses nonparametric estimates of treatment effects on the covariates. Intuitively, the nonparametric estimates are noisy because the compared customers are not identical twins. We only compare customers who are similar on their propensity score. Importantly, while two customers with very similar propensity scores could differ

[7] Extensions of propensity score-based matching methods exist that generate matched data for developing action-effect models for multiple treatment levels [Fahner, 2011a].

© 2012 Fair Isaac Corporation. All rights reserved.

substantially from each other on several covariates, it is a property of the propensity score that these covariate differences are not systematic[8], therefore this method will not lead to biased results [Rubin, 2006], [Fahner, 2011b].

Implications for Test-and-Learn

The theory of causal modeling from empirical data underlines the business value of testing. Smart experimental designs create rich common support while limiting testing in a safe and cost-effective way. "Boundary-hugging test designs" were proposed for maximizing common support at controlled costs for testing [Fahner, 2011c]. Where testing is given its proper due, organizations benefit from more informative data, better causal models and faster learning about their customers, which will help them to determine more profitable customer treatments based on empirical evidence.

Strengths

- More robust analysis approaches result in better causal models.
- More transparent analysis approaches clarify whether a data set is informative about treatment effects or not.
- Less likely to be misguided than naïve application of regression analysis, which does not warn the user about lack of common support.
- Common support considerations can inform the need and design of experiments.

Weaknesses

- Size of matched sample may end up too small to develop desired models.

References

Holland, P. (1986). Statistics and Causal Inference, *Journal of the American Statistical Association, Vol. 81, No. 396*, 945-960.

Fahner, G. (2011a). Estimating Causal Effects of Credit Decisions, *International Journal of Forecasting*. Article in press. (Feb., 2011).

Fahner, G. (2011b). Predicting Coupon Effects on Consumer Buying Behavior in Absence of a Control Group, *Edinburgh Scoring Conference Proceedings*. (Aug., 2011).

Fahner, G. (2011c). Causal Modeling-Based Approach for Testing and Improving Credit Decisions Over Time, *Edinburgh Scoring Conference Proceedings*. (Aug., 2011).

Rosenbaum, P.R. and Rubin, D.B. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects, *Biometrika, Vol. 70, No. 1*. (Apr., 1983), pp. 41-55.

Rubin, D. B. and Waterman, R.P. (2006). Estimating Causal Effects of Marketing Interventions Using Propensity Score Methodology, *Statistical Science, Vol. 21, No. 2*, (2006), pp. 206–222.

[8] If they were systematic, then X would reveal information about the treatment on top of the propensity score, and hence the propensity score could (and should) be improved.

» Cluster Analysis

Cluster analysis encompasses a group of data exploration techniques that is used to search for a structure of "natural" groupings of *multidimensional* objects or observations[9] according to their degree of similarity or distance. Clustering is distinct from classification methods. Classification pertains to a known number of groups (classes), and the operational objective is to assign new observations to one of these groups. Cluster analysis is a more primitive technique in that no assumptions are usually made concerning the number of groups or the group structure. The objective of cluster analysis is to discover natural groupings of the observations, such that observations in a given cluster tend to be similar in some sense to other observations in the same cluster and dissimilar to observations in other clusters.

For example, groupings may be defined based on students' scores across different aptitude tests. Students with high verbal ability might tend to cluster into one group while students with high artistic ability might tend to cluster into another group. How close should test scores be before students are grouped into the same cluster is a question of degree of within-cluster similarity and the total number of clusters desired.

In practice, the most-often-used clustering techniques are hierarchical clustering and non-hierarchical (disjoint) clustering. The hierarchical clustering is very much a "tree-like" procedure. It proceeds by either a series of successive mergers (often called agglomerative hierarchical methods) or a series of successive divisions (called divisive hierarchical methods). Agglomerative hierarchical methods start with the individual sample points as N clusters of one observation each. The number of clusters is then reduced to N-1 by merging two of the points into a single group based on some measure of similarity between them, applying some optimality criterion. The process is continued until the desired number of groups remains. Divisive hierarchical methods work in the opposite direction by starting with one group of N observations and dividing this group into smaller groups based on some measure of distance[10] between them until some optimality criterion is satisfied.

The non-hierarchical clustering techniques are designed to group observations into K disjoint clusters so that the total within group distance is minimized. The number of clusters, K, may either be specified in advance or determined as part of the clustering procedure. The K-means method is a popular non-hierarchical procedure. It first assigns the observations into K initial clusters. In each iteration, the distances between an observation's location (based on the observation's values for the variables that define the clusters) and cluster centers are calculated. The observation remains in the same cluster or is assigned to a different cluster according to the distances. The cluster centers (means) are then updated. The iterations continue until no more reassignments take place. Non-hierarchical/disjoint clustering techniques can handle a much larger data set than hierarchical techniques.

The cluster analysis is often preceded by principal component and/or factor analysis for *dimension reduction*. Even when dimension reduction techniques are used, the selection of variables used to define the clusters often poses a difficult combinatorial problem. In many cases this may be effectively addressed by optimization procedures such as genetic algorithms.

[9] Objects in this case can be either observations OR variables. This paper deals predominantly with identifying clusters of observations (cases). Clustering of variables is related to Principal Component Analysis and Factor Analysis. See section on Factor Analysis and Principal Component Analysis for brief description of those topics.

[10] In defining similarity/dissimilarity, various distance functions can be used. A commonly used distance is Euclidean distance.

© 2012 Fair Isaac Corporation. All rights reserved.

Applications

Cluster models resulting from cluster analysis have traditionally been used quite extensively in marketing applications to help characterize groups of similar consumers. The ability to better understand these groups can lead to more effective messaging and new product development efforts. More recent extensions to traditional clustering techniques have focused on finding clusters related to a specific objective, such as finding groups that are responsive to various marketing channels and messages. This is accomplished by using conditional distributions across auxiliary variables (such as response) to help drive the selection of cluster drivers. This hybrid approach helps ensure the clusters will consist of multi-dimensional groupings that are useful with respect to specific objectives.

Clustering can be used to enhance the performance of scoring models in one of three basic ways:

- By including indicators of cluster membership as simple predictors in the scoring model.
- By including interaction terms involving cluster membership in the scoring model.
- By building separate scoring models for each cluster.
- The approach that should be used depends on the degree to which the drivers of the measure being modeled differ in each cluster.

Strengths

- Does not assume any statistical distribution within data.
- Does not require performance measure. Provides actionable information where no performance outcome exists.
- Results can be presented in a way that is intuitive and easy to explain and understand.

Weaknesses

- Most of the clustering techniques are sensitive to outliers and the results could vary substantially for different initial seeds[11].
- Solutions are not unique and are extremely sensitive to algorithm parameter choices.
- Does not handle categorical data (nominal or ordinal) without preprocessing.
- Results are reliant on the subjective interpretation of "similarity" between observations.

References

Johnson, R. A. and Wichern, D. W. (1982), *Applied Multivariate Statistical Analysis*, Englewood Cliffs: Prentice-Hall, Inc.

Everitt, B. (1974, 1980), *Cluster Analysis, 2nd ed.,* New York: Halsted Press.

Kaufman, L. and Rousseeuw, P. J. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis,* New York: Wiley & Sons, Inc.

Sikkonen, J. and Kaski, S. (2002), *Clustering Based on Conditional Distributions in an Auxiliary Space,* Neural Computation Vol. 14 Number 4.

[11] Seeds refer to the choice of starting locations for the clusters which the analysis proceeds to iteratively adjust.

© 2012 Fair Isaac Corporation. All rights reserved.

» Collaborative Filtering

Collaborative filtering is a technology utilized primarily to predict individuals' preferences. The concept of collaborative filtering has its origin in information filtering, which guides a reader's choice by filtering a large amount of information and obtaining preferences collaboratively based on preferences shared by like readers.

Collaborative filtering works by first sifting through an individual's preferences or purchase history to find a group of individuals, or a 'neighborhood', with similar preferences or purchase histories, and then predicting what else the individual will like, based on the collective preferences or purchase histories of other individuals in the neighborhood. The predicted preferences can then be used to make product or service recommendations to the individual. Once a database of preferences, via surveys or transaction history, is accumulated, the following general steps can be applied to obtain a recommendation for an individual who supplies his/her preferences:

- Using a measure of similarity, individuals with similar past preferences are identified.
- A weighted average of the preferences for that neighborhood of individuals is calculated.
- The resulting preference function is used to make recommendations to the individual.

The individual's preferences are then added to the existing database. As the database grows, so does the time to compute the recommendation.

The basic premise of collaborative filtering is that people with similar tastes tend to like similar type of items. In order to work well, collaborative filtering requires a fairly representative sample of individuals, and a rich record of preferences or purchase histories from them. For example, if no like individuals can be found in the database, the recommendation will either not be provided or will be based on individuals with potentially very dissimilar tastes. In addition, collaborative filtering is most successful in the recommendation of products and services where the decision to purchase is largely driven by qualitative personal "taste" or preference. In the business environment where only a small number of complex products (such as financial products) are offered or products are purchased infrequently (e.g., durable goods such as computers or cars), the collaborative filtering approach may not be the most appropriate or effective. Incorporating other factors, such as product pricing, brand, and features directly into the preference prediction may lead to a more appropriate prediction and recommendation.

A variety of alternative technologies has been used effectively for 'preference' prediction and recommendation. These include:

- Segment or cluster popularity-based approach, which finds useful rules (e.g., When people buy JAVA books they also buy XML books 70 percent of the time) on pre-defined customer segments and/or product group.
- A hybrid of collaborative filtering and the segment or cluster popularity-based approach.
- Discrete Choice Modeling.
- Conjoint analysis.

Applications

Collaborative filtering technology has been applied widely in the last decade to the internet marketing efforts of consumer products that involve a large number of "taste"-oriented products and large customer bases. Product examples include movies, music, books, food, wine, clothing, art, news, web pages and so on. In the e-tail environment, it's often used to suggest products based on the purchases of customers with a similar purchasing pattern.

© 2012 Fair Isaac Corporation. All rights reserved.

Collaborative filtering has also been applied to call centers, such that catalogue-based businesses can increase sales by cross-selling items based on an individual's profile, for email based marketing campaigns, and internally for knowledge management such as document recommendation.

Strengths

- Intuitive, easy to comprehend and implement.
- No data structure assumptions.

Weaknesses

- Requires a large sample to make meaningful recommendations.
- Erroneous recommendations result when close neighbors don't exist.
- Direct insights into the drivers of the exhibited preferences are difficult to derive.
- Does not directly use product or item content information and customer profile or behavior information for making recommendations.

As database size increases, the recommendation computation becomes computationally more intensive.

References

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994), *Grouplens: An open architecture for collaborative filtering of netnews*. In Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work, 175-186, New York. ACM.

Breese J. S., Heckerman D. and Kadie C. (1998), *Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Proceedings* 14th Conference on Uncertainty in Artificial Intelligence, Madison WI: Morgan Kauffman.

Mobasher, B., Colley, R., and Srivastava, J. (2000), *Automatic personalization based on web usage mining*. Communication of the ACM, Vol. 43, No. 8.

Kitts, B., Freed, D., and Vrieze, M. (2000), *Cross-sell: A Fast Promotion-Tunable Customer-item Recommendation Method Based on Conditionally Independent Probabilities*. In proceedings of KDD 2000 conference, 437-446, Boston MA, USA.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000), *Analysis of Recommendation Algorithms for E-Commerce*. In Proceedings of the 2nd ACM E-Commerce Conference (EC'00). Oct., 2000.

Fader, P. and Hardie, B. (1996), *Modeling Consumer Choice Among SKUs*. Journal of Marketing Research, 33, 442-452.

Guadagni, P. M. and Little, John D. C. (1998), *When and What to Buy: A Nested Logit Model of Coffee Purchase*. Journal of Forecasting 17, 303-324.

© 2012 Fair Isaac Corporation. All rights reserved.

» Decision Analysis: Key Concepts and Tools

Decision analysis refers to the broad quantitative field, overlapping operations research and statistics, that deals with modeling, optimizing and analyzing decisions made by individuals, groups and organizations.

The purpose of decision analysis is to assist decision makers in making better decisions in complex situations, usually under uncertainty. The quality of the decisions is measured by their expected consequences and the stated preferences of the decision maker(s). The decision analytic framework helps the decision maker think systematically about his or her objectives and preferences, the structure and uncertainty in the problem, and model quantitatively these and other important aspects of the problem and their interrelationships.

All complex decision problems include the following main elements:

- Decisions.
- Uncertain events.
- Consequences.
- Objectives and preferences.

A decision, D_i, $i = 1, 2,...$ refers to a point in time when the decision maker has to choose one alternative, d_i, out of a domain of available alternatives, that could be discrete (e.g., extend or don't extend a credit offer) or continuous (e.g., the range of credit line assignments from $500 to $50,000), or a combination of the two. What separates one decision, D_1, from another, D_2, is the difference in the information available to the decision maker before each decision is made. The information corresponding to a decision is the set of all observations available to the decision maker prior to making that decision.

Uncertain events, X_j, $j = 1, 2,...$, typically occur interspaced between subsequent decisions. If an uncertainty is realized before a decision is made, its outcome, x_j, will typically be observed by the decision maker before that decision is made. For example, an applicant's income and other credit application information, while uncertain at the time a credit offer decision, D_1, is made, will have been realized and observed before the subsequent credit line decision, say D_2, is made. Still, at the time D_2, is made, there is a number of unresolved uncertainties, like the true credit worthiness of the applicant and his or her future use of the credit and repayment patterns. Together with the decisions already made, such remaining uncertainties determine the consequences to the decision maker, and therefore need to be estimated up front in the form of probability distributions or predictions.

The consequences, V_k, $k = 1, 2,...$, to the decision maker are the results of, and determined by, the alternatives chosen at all decision points D_i, and the outcomes of the uncertain events X_j[12]. They are themselves uncertain at the time all decisions need to be made and are closely related to the objectives of the decision maker. For example, the cost of a marketing campaign, and its size, revenue and loss to a portfolio, are all consequences.

The decision maker's objectives, M, in solving a decision problem are the quantities he or she cares about, including their preferred direction. Maximizing a portfolio's size is an example of an objective; minimizing a portfolio's loss is an example of another.

[12] That is, each V_k is a function of all decisions made $\mathbf{D} \equiv \{D_i, i = 1, 2,...\}$ and all realized events $\mathbf{X} \equiv \{X_j, j = 1, 2,...\}$: $V_k = V_k(\mathbf{D}, \mathbf{X})$, $k = 1, 2,...$

© 2012 Fair Isaac Corporation. All rights reserved.

Rarely in realistic decision problems is there a single objective. When there are two or more objectives, they typically conflict, in the sense that some strategy is optimal (performs best) with respect to one objective, while a different strategy is optimal with respect to another. Analysis of such decision problems is discussed in the section "Multiple-Objective Decision Analysis."

Main Tools of Decision Analysis

Important methodologies and tools in decision analysis include:

- Graphical models, which are very important in modeling the decision problem and evaluating results. Two such graphical tools—influence diagrams and decision trees—are discussed in the section "Graphical Decision Models."
- Bayesian inference or learning, which is the fundamental learning mechanism in decision models. It is essential in decision situations that involve two or more decisions, made at different points in time, which are closely related in affecting chance events, each other, consequences and objectives. Refer to the section on "Analysis of Sequential Decisions" for a discussion of these notions.
- The expected value of (perfect or partial) information, which measures the value of information about sources of uncertainty in the problem in terms of the decision maker's objectives.
- Constrained optimization, which allows the inclusion of constraints on objective and/or decision domains.
- Quantitative risk analysis tools, which allow one to quantify and assess a decision problem's total uncertain outcomes. These tools include:
    - Predictive and decision modeling techniques, which establish the mathematical relationships between inputs, decisions and outcomes.
    - Sensitivity analysis, which, through an iterative process, facilitates the building of a requisite decision model[13] and allows testing of its robustness.
    - A stress-testing methodology which allows for simulation and testing of decision model assumptions and relationships, helping to capture the effect of exogenous factors to the model, including how the future may be different than the past.

Applications

Decision analysis has been widely used for medical diagnosis and treatment, bidding, negotiations and litigation. A non-exhaustive list of other applications in business and government includes:

Business:

- Airline and hotel yield management
- Oil exploration
- Quality assurance and control
- Crop protection
- Credit and loan portfolio management
- New product development
- New venture launching

[13] A requisite decision model is a model that contains only and all that is essential in optimally solving the decision problem.

Government:

- Emergency management
- Environmental risk management
- Choice of new energy sources
- Research and development programs

References

Bedford, T., Cooke, R. (2001), *Probabilistic Risk Analysis: Foundations and Methods*, Cambridge: Cambridge University Press.

Clemen, R. T., (1996), *Making Hard Decisions: an Introduction to Decision Analysis*, Duxbury Press, 2nd Ed.

Lindley, D. V. (1985), *Making Decisions*, Wiley.

Hammond, J. S., Keeney, R. L., and Raiffa, H. (1999), *Smart Choices: A Practical Guide to Making Better Decisions*, HBS Press.

Marshall, K. T. and Oliver, R. M. (1995), *Decision Making and Forecasting*, McGraw-Hill.

Raiffa, H. (1968), Decision Analysis, Addison-Wesley.

Saltelli, A., Chan, K., Scott, E. M. (2000), *Sensitivity Analysis*, Chichester: John Wiley & Sons.

Vose, D. (2000), *Risk Analysis: A Quantitative Guide*, Chichester: John Wiley & Sons.

© 2012 Fair Isaac Corporation. All rights reserved.

» Discrete Choice Modeling

Discrete choice modeling may be thought of as a generalized case of logistic regression modeling that evolved from *conjoint analysis*. In logistic regression, the *dependent variable* is *categorical* and takes on one of two outcomes. For example, a credit card account holder may default or not default. Discrete choice modeling extends this modeling structure to the case where the dependent variable takes on two or more discrete values, which are unordered. An example of an unordered dependent variable would be a consumer's choice of a credit card, an installment loan or a line of credit to finance the purchase of a durable good. This dependent variable is unordered because there is no underlying ranking to the discrete choices.

The goal of discrete choice analysis is typically to determine the key characteristics that explain why a consumer makes a specific choice from a set of available products. By understanding the key trade-offs, marketing decisions can be made to maximize sales, profitability or market penetration.

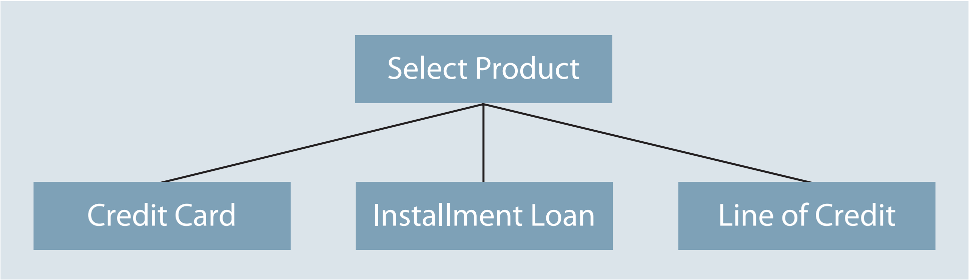
To model these preferences we rely on the concept of utility theory. A utility function (which we may write mathematically as U(Product) maps a consumer's preferences to some numeric value such that if the consumer prefers product A to product B, U(A) is greater than U(B). We can also define a utility function not as a function of the product in its entirety, but more usefully as a function of the underlying product features. We can also include variables describing the consumer in the model. For example, a utility function for a financial product may look like the following:

$$U\,(\textit{financial product}) = \\ a^* \textit{Revolving} + b^* \textit{APR} + c^* \textit{creditline} + d^* \textit{Revolving}^* \textit{consumer\_buys\_new\_car}$$

where "revolving" and "consumer_buys_new_car" are dummy variables, taking a value of 1 for yes or 0 for no. The terms a, b, c, and d are weighting terms to be estimated. In this example, we would expect *a* to be positive (since revolving products are more flexible than non-revolving products), *b* to be negative (since APR is the "price" of the financial product), *c* to be positive (since people prefer more credit to less), and *d* to be negative (since a consumer needing to finance a new car is not likely to finance it with a revolving product). We model the consumer's choice by calculating the consumer's utility for each product and selecting the product with the highest utility.

Great care must be taken when structuring the utility function to be estimated. For example, the simplest structure might be as follows:

FIGURE 5

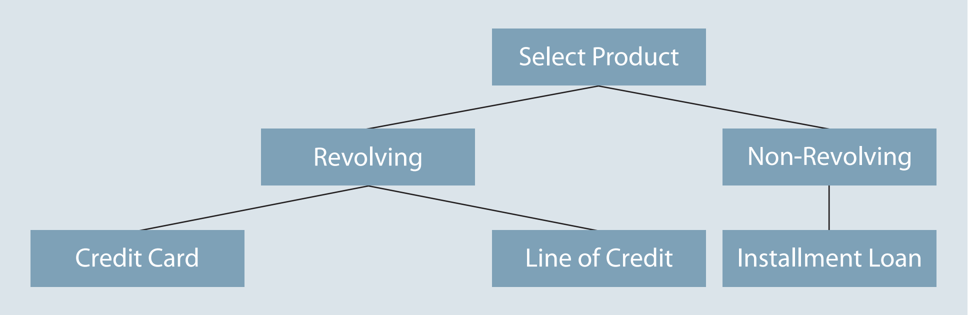


In the above case, the consumer considers all product features at once before deciding on a choice. However, if the typical consumer makes decisions in a staged manner, first considering certain features and rejecting products that don't contain those required features, and then applying the remaining product features to further narrow the n choices, a model developed using the structure in the figure on previous page will typically not validate well.

© 2012 Fair Isaac Corporation. All rights reserved.

An alternative decision structure might look as follows:

FIGURE 6



The preceding model is known as a nested model, because the final choice of product is "nested" within the prior choice of a revolving or non-revolving product.

Data used to develop the discrete choice model should contain the relevant attributes of the choice alternatives (e.g., the product features and customer demographics, and the choice itself). Depending on the uses of the model, it may also be desirable to include attributes of the decision made. In applications such as travel demand forecasting, discrete choice models have been developed and proved useful with unplanned data. However, in marketing research, where data planning (typically through experimental design) is critical to obtaining good estimates of the effects, these data are collected from a sample of consumers participating in a test. The test participants are asked to choose their favorite alternative. The resulting data are very effective in determining important parameters, such as brand-price *interactions* and how they differ for different consumers.

Applications

Discrete choice modeling is mostly used for the modeling of consumer choices of products, but can be extended to any type of unordered outcome. Examples include modeling which debt obligation a risky customer defaults on and which set of product features to offer a consumer.

Strengths

- Can have mixed continuous and categorical predictor variables.
- Results are already on probability scale.
- Can handle the case of a consumer making decisions multiple times.
- Good to use where the consumer cannot be asked directly which attributes he considers most important.

Weaknesses

- Model validation is sensitive to segmentation and decision structure.
- For categorical predictor variables that are converted to dummy variables, there is currently no effective mechanism for engineering the model.

© 2012 Fair Isaac Corporation. All rights reserved.

References

Ben-Akiva, M. E., Lerman, S. R. (1985), *Discrete Choice Analysis*, MIT Press.

Louviere, J. J., Hensher, D. A., Swait, J. D. (2000), *Stated Choice Methods: Analysis and Application*, Cambridge University Press.

Maddala. G. S., (1999), *Limited Dependent and Qualitative Variables in Econometrics*, Cambridge University Press.

© 2012 Fair Isaac Corporation. All rights reserved.

» Discriminant Analysis

The goal in discriminant analysis is usually two-fold:

- Segment or separate individuals into two or more previously defined groups.
- Classify a new individual into one of the groups.

A rule or "discriminant function" is developed based on measurements (variables) associated with each of a sample of individuals from two or more populations. As in *regression*, the general approach is to construct, in some optimal way, a *linear combination* of measurements or predictor variables which will best distinguish (discriminate) between the groups. The model is in the form of multiple formulæ, each corresponding to one group[14]. A new individual can then be assigned or classified into the correct population based on the highest value of the linear combinations (scores) from among the discriminant functions for that particular individual.

The predictor variables can be predetermined by the analyst or can be selected using *stepwise* discriminant analysis. Stepwise discriminant analysis operates in principle like stepwise regression; variables are included in the model sequentially until no further improvement (within the stop criterion limits) in discrimination is gained.

Applications

Often used in marketing (e.g., to distinguish purchasers of a new product from non-purchasers, to identify low/medium/high response groups). Also used for developing credit risk models.

Strengths

- Can separate and classify individuals into multiple groups.
- The idea of scoring an individual and use of a cutoff is inherent in this methodology. Hence it can be easily perceived as the "right tool" for credit scoring.
- Can model multiple outcomes.

Weaknesses

- Assumes that the predictor variables are distributed as multivariate normal (having a combined distribution that is normal in multiple dimensions—this results in some elegant simplifications on which discriminant analysis relies). This assumption is usually violated in our typical scoring applications. Although the technique is somewhat robust with respect to minor violations of the assumption, serious violations will often result in unreliable estimates.
- If stepwise discriminant analysis is used, the problems associated with variable selection procedures are present. The "best" subset selected for a given data set may perform poorly in future samples.
- When some or all of the independent variables are very highly correlated (i.e., a situation often termed multicollinearity), the procedure could select an unreasonable set of variables as optimal. In fact, in situations of multicollinearity, estimates of regression coefficients from sample to sample fluctuate markedly.

References

Everitt, B. S. and Dunn, G. (1992), *Applied Multivariate Data Analysis*, New York: Oxford University Press.

Goldstein, M., Dillon, W. R. (1978), *Discrete Discriminant Analysis*, New York: John Wiley & Sons.

[14] For binary prediction, the formulæ degenerate to a single formula since probability of membership in one of two mutually exclusive groups also reveals the probability of membership in the other group.

» Ensemble Modeling

Ensemble modeling refers to techniques where the predictions of a group of base models are combined to generate more accurate composite predictions. Ensemble modeling involves two main activities:

- Constructing an ensemble of base learners from training data
- Combining predictions of ensemble members into a composite prediction

Ensemble learning is modular in the sense that many types of base learners can be used and there are multiple ways to generate composite predictions. However, not all combinations of ensemble construction and base models are useful. Here, we will discuss three major variants of ensemble modeling and we will provide insights into their function:

- Bagging [Breiman 1996]
  - Create many base models, each using a different bootstrap sample from the training data, and average their predictions. The name "bagging" derives from "bootstrap aggregation." Chose base model from a highly flexible model family, such as deep regression trees.
- Boosting [Freund & Schapire 1996, Friedman, Hastie & Tibshirani 1998], see also section on "Boosting"
  - Here, the base models are trained iteratively, focusing on previously hard-to-predict observations. A subsequent base model attempts to improve upon the prediction errors of the preceding models, thus "boosting" performance. The predictions are calculated as a weighted sum of base model predictions. Chose base model from a simple, less flexible model family, such as shallow regression trees.
- Segmentation, see also section on "Recursive Scorecard Segmentation"
  - Segment the population into more homogenous sub-segments such that base learners perform well for their dedicated sub-segments. Chose base model from a model family of intermediate flexibility, such as additive regression models or scorecards.

Comparing Approaches from a Statistical Perspective

To improve predictive accuracy, each approach follows a different strategy.

Bagging benefits from the capacity of deep regression trees with many leaf nodes to represent a large family of complicated functions closely by virtue of capturing high-order nonlinearities and interactions. There is practically no bias in the base model. However, the dilemma of deep trees is that their predictions have high variance ("sampling errors"). For this reason, a single deep tree may not produce accurate predictions. When a single tree is built traditionally, variance is mitigated to by stopping tree growth early and/or by pruning techniques. In contrast, bagging aims to reduce variance by averaging over predictions from many deep trees. To succeed with this, the errors of different trees must be somewhat uncorrelated. This is achieved by training deep trees on different bootstrap samples. It helps that the base models are "unstable" in the sense that small changes to the training data tend to generate diverse trees. This instability de-correlates errors across the different base models, leading to rapid variance reduction, as predictions from an increasing number of trees are averaged. Other unstable base learners, such as regression with stepwise selection, may also be used profitably in combination with bagging. Bagging does not work well for stable, biased base learners, such as small and fixed regression formulas or shallow trees.

Boosting follows a different strategy. Base learners are simple, e.g. shallow trees with few leaf nodes. While predictions from a single base learner tend to be biased and less accurate, a longer sequence of base learners generated by boosting iteratively expands the degrees of freedom of the approximation function, thereby reducing approximation error. The ability to control for the complexity of the base model offers fascinating insights: When using a stump tree (root node split into two leaf nodes) as base model, boosting remains constrained to additive, albeit arbitrary nonlinear, approximation functions. If slightly deeper trees are allowed, boosting can capture first order or higher order interactions. The order of interactions that can be represented is thus judiciously controlled by the complexity of the base learner (e.g., number of leaf nodes allowed). This is useful for testing for interactions.

Segmentation works in yet another way. Scorecards are highly interpretable base models, as they are often preferred for financial services, insurance or marketing applications. They can also be engineered to counteract data limitations. Technically, scorecards capture nonlinear, additive dependencies between predictors and outcome, similar to boosting with stump trees. It has long been known that segmented scorecards can sometimes predict better than a single scorecard. In statistical terms, the reason is the possible presence of interactions, which a single scorecard without explicit coding of interaction terms cannot capture. In contrast, a segmented scorecard ensemble can capture interactions between the variables used to segment the population and predictors used in the segment-specific scorecards. If significant interactions are present, then a segmented scorecard ensemble can not only lead to more accurate predictions, but can also generate more palatable scorecards that are more attuned to the specific information available in each segment.

Applications

In the academic literature, ensemble models are most often applied to classification tasks. However, ensemble predictions can also be used as rank-ordering scores or interpreted as regression functions, making them potentially useful for a wide range of tasks across financial service and marketing applications where account or customer behavior needs to be predicted. We are investigating new applications of ensemble modeling for scoring and causal modeling applications and we see good potential in various areas.

Strengths

- Superior predictive performance
- Makes minimal assumptions on the data
- Requires minimum prior knowledge
- Ability to test for interactions
- Some approaches can lead to more palatable models

Weaknesses

- Some approaches lead to difficult to interpret models
- Some approaches don't allow for (or don't make easy) the imputing of prior knowledge or business constraints on the models
- Model training can be computationally demanding
- Some approaches lead to models that are difficult to deploy

References

L. Breiman, "Bagging predictors," *Machine Learning*, Vol. 24, 1996.

Y. Freund & R. Schapire, "Experiments with a new boosting algorithm," Proc. 13th Nat'l Conf. Machine Learning, 1996.

J. Friedman, T. Hastie, T. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting", July 23, 1998.

© 2012 Fair Isaac Corporation. All rights reserved.

» Experimental Design

Experimental Design is a mature and extremely successful science dating back to the pioneering work of R. A. Fisher in the 1930s. It hosts a body of techniques for generating efficient experiments or tests such that the resulting data will yield precise analysis results with low systematic or personal bias. Typical goals of such an analysis include:

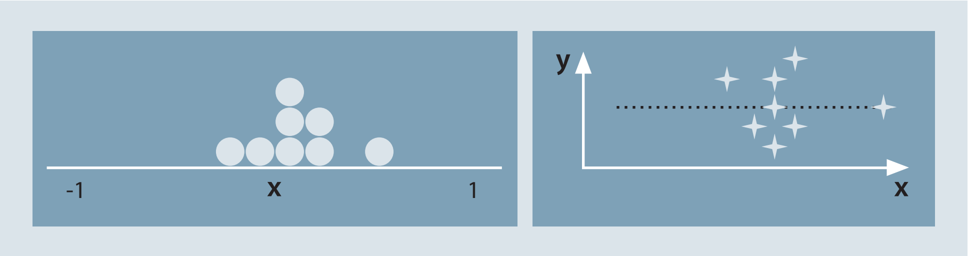
- Understanding the causes of variation in the measured outcome of a complex system or process.
- Prediction of how the outcome would change if certain control factors or operating conditions were changed.
- Optimization of control factor settings to achieve or approach a desired outcome.

Experimental design takes into account the scope of analysis that is envisioned for the experimental data. To design the experiment, the researcher requires a general idea of the predictive model structure and plausible ranges for the control factors where predictions and optimization are to be attempted. Then data are collected in the most efficient manner to provide sufficient coverage throughout the operating region of interest, such that the model will yield accurate predictions and optimization results. Prior experience and theoretical insight into a problem help with the task to design the best experiment.

In the experimental design paradigm, predictions of the dependent variable are attained from *regression* models. In many financial or marketing applications, observational data have been acquired as a result of running the business "as usual," without giving consideration to the scope of a desired analysis. In these cases, regression techniques are applied to fit the observational data, and to predict new outcomes under similar operating conditions. This analysis of unplanned data is limited when compared to experimental design, because extrapolation for novel control factor settings or operating conditions is notoriously unreliable.

The concept of experimentation can be illustrated with a simple example. Consider data samples with x-values (control factor settings) within the range [–0.8, 0.8], and associated noisy y-values (outcomes). If only the expected outcome ŷ matters, any arrangement of control factor settings is equally good. The unplanned data on the left hand side of Figure 7 provides the estimate of average outcome displayed on the right hand side. However, if the researcher is interested in analyzing the relationship between the factor value and the expected outcome, she should give some consideration to what would constitute an adequate regression model to solve her problem, and select the experimental design accordingly.

FIGURE 7



The researcher may have theoretical insights and/or practical experience with a similar problem, pointing to the fact that a linear relationship will model the relationship well enough for all practical purposes. If the goal is to fit a linear relationship (ŷ = a + b * x), then a better experimental design is

© 2012 Fair Isaac Corporation. All rights reserved.

the one illustrated in Figure 8. It places the test points at the extremes of the allowed range, thereby achieving the most accurate determination of the slope parameter b (note however that the design given by Figure 8 will not allow for a check of the possible lack of fit against a nonlinear relationship).

If the goal is to develop a model with a quadratic ($\hat{y} = a + b*x + c*x^2$) or cubic relationship ($\hat{y} = a + b*x + c*x^2 + d*x^3$), then the experimental designs in Figures 7 and 8 respectively can be used.

FIGURE 8: EXPERIMENTAL DATA FOR FITTING STRAIGHT LINE

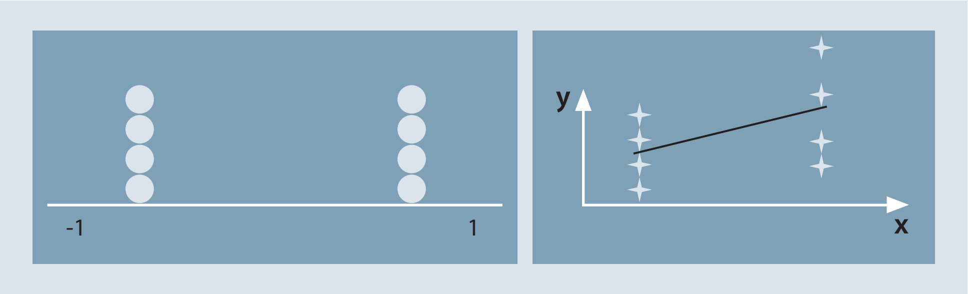


FIGURE 9: EXPERIMENTAL DATA FOR FITTING QUADRATIC

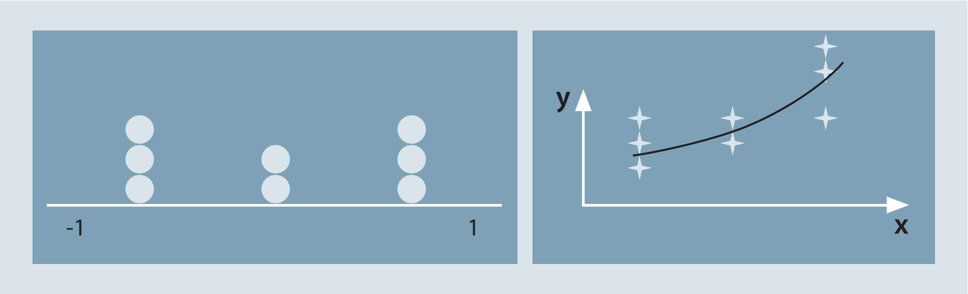
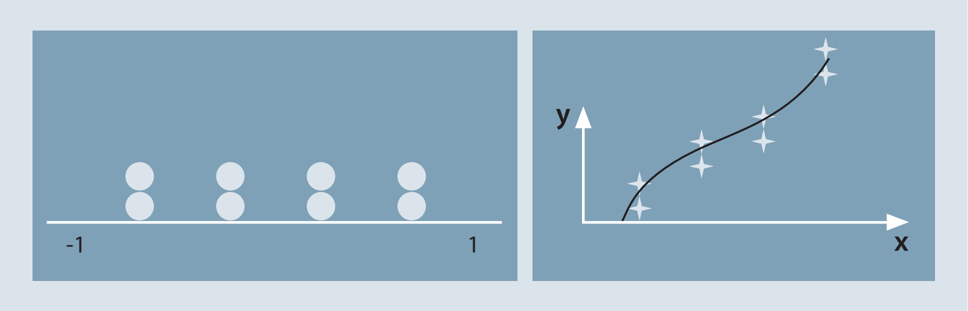


FIGURE 10: EXPERIMENTAL DATA FOR FITTING CUBIC RELATIONSHIP



Trying to fit a linear, quadratic or cubic fit to the unplanned data in Figure 7 will result in high variance of the regression parameters and potentially poor *extrapolation*, i.e., predictions with a very low degree of confidence in the region not covered by the observed data (e.g., at x = 1). The only way to

© 2012 Fair Isaac Corporation. All rights reserved.

boost confidence in a model and its predictions is to run experiments in the region of interest, while taking into account the envisioned model structure.

Even with large amounts of unplanned data, estimated effects can be *confounded*. Consider a simple authorization strategy assigning credit limits proportional to applicants' incomes. One wishes to model the profit of an account as a function of the control factor "credit limit" and income (which is not a control factor here), such that the optimal factor setting can be determined for a given applicant. However, without further information, regression analysis of the unplanned data generated under the business-as-usual strategy cannot attribute the variation of profit to the individual variations of income or credit limit, because the two are varied in unison. It is therefore also not possible to solve for the optimal authorization strategy. A simple and effective (but not necessarily optimal) tactic of experimental design, *randomization*, will solve this business problem, by assigning a randomly selected fraction of applicants to randomly distributed credit limits within a plausible neighborhood of the "business-as-usual" authorization limit.

Experimental design is the answer to these problems of *extrapolation* and *confounding* by allowing for efficient, systematic acquisition of data samples. The design of the experiments varies in complexity depending on the number of experimental factors, their suspected degree of non-linearity and order of interactions, and constraints on factor ranges and their permissible combinations. When designed appropriately, the number of experiments required to accurately estimate the desired effects will be minimized. The simplest (and most easy to interpret) approach is to run experiments where only one factor is varied at a time. However, this approach is inefficient if many factors need to be estimated. In these cases, fractional and full factorial designs are used (see references in this section). A large number of design approaches have been developed over time for different applications, including:

- Screening designs for main effects to identify factors with greatest impact on the outcome.
- Response surface designs for quadratic effects to optimize an outcome, often using a sequence of experiments.
- Optimal designs for nonstandard model structures and irregularly shaped operating regions.

A comprehensive description is outside the scope of this discussion.

Applications

Applications of experimental design range from testing of research hypotheses in the natural sciences to industrial process optimization for the agricultural, healthcare, chemical and electronics industries. Experimental design is widely used in research, usually to show the statistical significance of an effect that a particular factor exerts on the dependent variable of interest. In industrial settings, the primary goal is usually to extract the maximum amount of unbiased information regarding the factors affecting a production process from as few (costly) observations as possible. In marketing applications, the goal is usually to test as many marketing strategies as possible, within budgetary constraints on the number of experiments and constraints on the factor combinations that define a product.

References

C. F. Jeff Wu, Michael Hamada (2000), *Experiments: Planning Analysis, and Parameter Design Optimization*, John Wiley & Sons.

G. K. Robinson, (2000), *Practical Strategies for Experimenting*, John Wiley & Sons.

Chance (Nov. 3, 1997), Vol.10, *A Note on Multivariable Testing in Marketing Research.*

» Factor and Principal Component Analyses

Data with a large number of variables often exhibit a high degree of linear relationships (covariance or correlation) among the observed variables. It is often useful to try to use these relationships to help reduce the dimensionality (number of variables) of the data by squeezing out the redundant information (due to the related nature of the variables) represented by the many variables. The reduced dimensionality may facilitate expedient exploratory data investigation and modeling.

Principal component analysis and factor analysis are two related data analysis techniques that help reduce the dimensionality of the data by utilizing the linear relationship between variables. Factor analysis may provide further insight into potential grouping schemes for the observable variables. While the two techniques are not modeling techniques per se, the results of principal component analysis and factor analysis can be used as part of other modeling techniques. Both techniques, given their utilization of correlation or covariance between variables, are only applicable to continuous-valued variables[15]. Also, as the total amount of correlation between the variables in the data decreases, these techniques become less useful.

Principal Component Analysis

The goal of principal component analysis is to reduce the dimensionality of data by generating a sequence of linear combinations, called principal components, of the original observable variables. In other words, it tries to derive a smaller set of principal components that represent a larger set of observable variables in the data without loss of significant specificity. No explicit "meaning" need be associated with the principal components themselves. (See Factor Analysis below for further discussion on "meaning".)

The result of principal component analysis is a sequence of uncorrelated principal components that are ranked in terms of the amount of total variation (correlation/covariance) they explain. For highly correlated data, a few principal components may represent most of the variation. For data not highly correlated, many principal components may be needed to explain a majority of the variation. Depending on the ultimate goal of the dimension reduction and just how few of the principal components actually explain the majority of the variation in the data, the analyst must choose between simplicity (using fewer principal components) and comprehensiveness (explaining more of the variation).

Graphical examination of the observable variables against the dominant principal components often reveals the original correlation between the observable variables more clearly.

Factor Analysis

Factor analysis is related to principal component analysis in that its goal is also to search for a few representative variables to explain the observable variables in the data. However, the philosophical difference in factor analysis is that it assumes that the correlation exhibited among the observable variables is really the external reflection of the true correlation of the observable variables to a few underlying but not directly observable variables. These "latent" variables are called "factors" that drive the observable variables. When *conditioned* on the factors, there is no correlation between the observable variables.

[15] Two other techniques may help with dimension reduction for categorical data: correspondence analysis, a technique that tries to do dimension reduction of multivariate categorical data by working with the contribution to Chi-squared statistic from each cell of multivariate crosstabulations in a similar conceptual fashion as principal component analysis works with the correlation or covariance matrix; log-linear modeling (see section in this document), which is useful for detecting structure of relationship among categorical variables.

For example, the concepts of "ability to pay" and "willingness to pay," although difficult to observe directly, are two very general factors that may drive most of the credit risk variables we typically encounter. More specific and practical examples of factors in credit data are "revolving credit capacity," "revolving credit utilization," and "revolving credit experience."'

Factor analysis is the process by which various alternative choices are made towards generating the factors and selection of the factor scheme that most intuitively relates the original observable variables is made. In addition to choosing the trade-off between number of factors and amount of correlation/covariance to explain, there are additional choices of whether to allow the factors to be correlated (oblique) or uncorrelated (orthogonal).

Applications

As mentioned above, principal components and latent factors are often used as a dimensionality reduction technique to reduce the overall number of variables down to a fewer, manageable number of variables on which further modeling can be performed. Factor analysis is used in behavioral and social sciences as well as in the field of market research where it is appealing to collapse answers to many related survey questions into a few underlying factors.

Strengths

- Can summarize many dispersed continuous variables into a few summary variables.
- Pattern of correlation of the principal components and factors with the observable variables may reveal "structure" in the data and provide insight.
- Speculating about the nature of the factors may provide more insight into the original observable variables.

Weaknesses

- When a few principal components or factors are insufficient (as seems to be the case in credit data), the original difficulty of the high-dimensionality of the data returns.
- Inability to handle missing values or mixed variables (where some of the cases are special, non-interval scale values) reduces applicability on credit data.
- Do not address, or have to address via pre-processing, categorical or ordinal observable variables.
- Interpretation of factors and their relationship to the observable variables is rather subjective and arbitrary.
- Not applicable if the relationships between the observable variables are not linear (i.e., strong relationship but not identified as so by correlation or covariance measures).

References

Johnson, R. A. and Wichern, D. W. (1982), *Applied Multivariate Statistical Analysis*, Englewood Cliffs: Prentice-Hall, Inc.

Everitt, B. S. and Dunn, G. (1992), *Applied Multivariate Data Analysis*, New York: Oxford University Press.

» Genetic Algorithms

Genetic algorithms (GAs) are a class of *optimization algorithms* inspired by population genetics and the Darwinian principle of natural selection, commonly referred to as "the survival of the fittest." Given an *objective function*, the typical GA begins with a random population (generation) of solutions (chromosomes). Each solution is represented by a sequence of characters (genes) each having certain values (alleles). By mating and mutating the best solutions (as measured by some fitness value), the GA produces a new population of improved solutions (offspring). The average fitness of the population, as well as the fitness of the best solutions, improves at each generation. This process continues until the GA has determined an acceptable solution to the problem (as determined by the developer).

This process can best be illustrated with an example. Suppose we want to identify subsets of an applicant population that are most likely to be classified as good accounts in the future. We are given a data set that contains the historical performance of a number of applicants and the following predictive variables:

1. Applicant Age (1 = old and 0 = young)
2. Residential Status (1 = owns and 0 = does not own)
3. Checking Account Reference (1 = yes and 0 = no)
4. Credit Card Reference (1 = yes and 0 = no)
5. Derogatory Ratings on Credit Bureau Report (1 = yes and 0 = no)

Given this information, we can represent any subset of the population by a sequence of 0s and 1s. For example, the sequence 01110 represents individuals that are young, own their residence, have a checking account and a credit card, and do not have any derogatory ratings on their credit bureau report. For each subset of the population, we can define a fitness measure by counting the number of "good" and "bad" applicants in the subset and calculating the good:bad odds for the subset. The fittest subsets are those that have the highest good:bad odds.

Now that we have defined the form of the solutions (sequences of 0s and 1s) and a measure of how well each solution performs (the good:bad odds), we can generate a random set of solutions and let the GA run. Suppose we generate the following four random solutions (ranked by their corresponding odds):

01110	30:1	(young, owns, checking account, credit card, no derogs)
10101	2:1	(old, does not own, checking account, no credit card, derogs)
11011	2:1	(old, owns, no checking, credit card, derogs)
00010	1:1	(young, does not own, no checking, credit card, no derogs).

Using the principle of the survival of fittest, the GA selects the fittest solutions in order to produce a new generation of solutions. The GA uses two operations to generate new solutions. The first, crossover, is simply a swapping of values at certain positions of the sequences representing a pair of solutions. For example, the GA might generate two new solutions by swapping the first two positions of the fittest solutions above:

01|110 ➔ 10|110
10|101 ➔ 01|101.

The second operation is called mutation and is simply a random change in the value at some position in the sequence. For example, the GA might mutate the value at the third position of the fourth solution above by changing the 0 to 1:

00|0|10 ➔ 00|1|10.

© 2012 Fair Isaac Corporation. All rights reserved.

Through crossover and mutation, the GA generates a new set of solutions. Crossover allows the GA to preserve features of the best solutions of past generations while trying slightly different solutions. Mutation lets the GA produce novel solutions that it might not find using crossover alone. Through continuous use of crossover and mutation, the population of solutions will evolve, each generation more fit than the previous. For example, the initial population after several generations might evolve into the following:

11110	40:1
01110	30:1
10110	30:1
00110	20:1

At this particular generation, the GA has determined that based on historical data, applicants who are old, own their residence, have a checking account and a credit card, and do not have any derogatory ratings on their credit bureau report are most likely to perform well in the future.

Although the number of possible configurations of applicants in this example is finite and small (with only 32 possible combinations), like all optimization algorithms, GA is most useful in the search for an optimal configuration when the number of possible configurations is large and the cost of calculating the fitness of all possible solutions is prohibitive.

Applications

Flexible encoding allows genetic algorithms to be applied to a diverse set of problems in biology, computer science, engineering and operations research, image processing and pattern recognition, and the social sciences. Their highly parallel search mechanism makes them suitable for high-dimensional, highly non-linear, non-smooth objective functions that other optimization techniques find difficult to solve. In general, however, genetic algorithms will generally take longer to converge than other techniques, and as with other optimization techniques, are not guaranteed to find the globally optimum solution.

Strengths

- General-purpose technique that is applicable to a variety of problems.
- Generally finds a good solution.

Weaknesses

- Not guaranteed to find the best solution.
- Computationally intensive.

References

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading: Addison-Wesley.

Holland, J. H. (1975), *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.

Mitchell, Melanie (1999), *An Introduction to Genetic Algorithms*. Cambridge: The MIT Press.

Nygard, K. E., Ficek, R. E., and Sharda, R. (August 1992), *Genetic Algorithms: Biologically inspired search method borrows mechanisms of inheritance to find solutions*, OR/MS Today, pp. 28-34.

» Graphical Decision Models

Graphical paradigms play an important role in modeling and structuring decision problems.[16] The two most commonly used graphs to display decision models are influence diagrams and decision trees. The following types of nodes are used in both types of graphs:

- Decision nodes, drawn as rectangles, represent decisions.
- Chance nodes, drawn as ovals, represent uncertain events.
- Consequence or value nodes, drawn as rounded rectangles or diamonds[17], represent consequences.

Influence Diagrams

An influence diagram is a directed acyclic graph in which each node labels a single variable of the decision problem and the arcs represent two main types of relationships among the variables. Arcs into decision nodes signify that all variables labeled by the nodes from which the arcs emanate (called direct predecessor or parent nodes) are observed by the decision maker before the decision is made. These are sometimes called information arcs. Arcs into chance or consequence nodes represent possible probabilistic dependence on their direct predecessor and are usually referred to as dependence arcs.

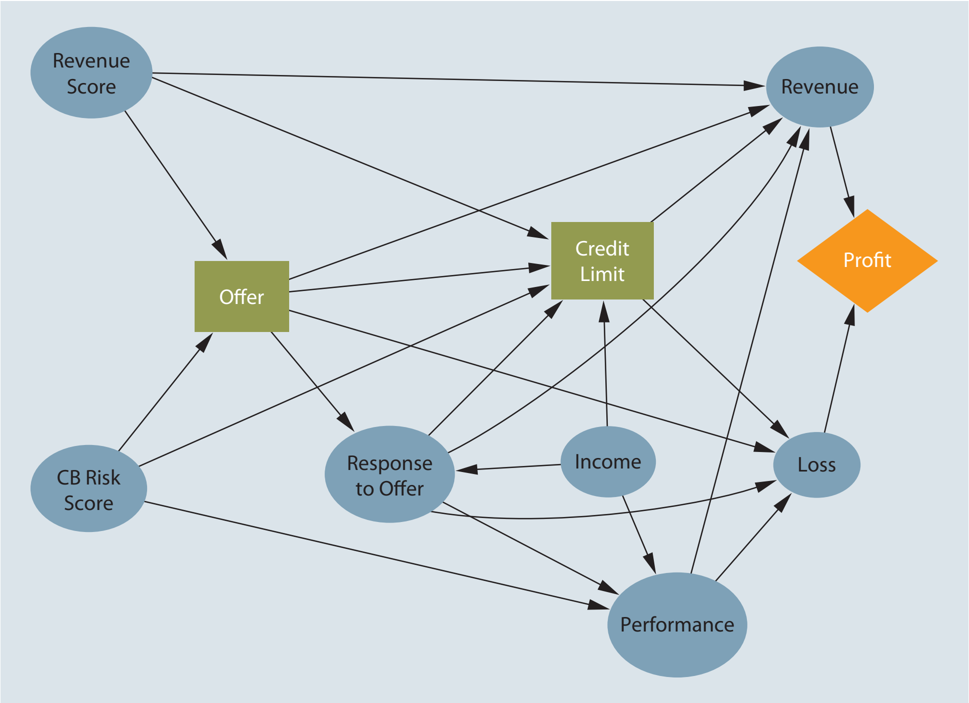
Figure 11 shows an influence diagram that illustrates these notions and the modeling power of influence diagrams. It is a simplified model for a two-stage credit card campaign decision problem. Before making the Offer decision, the only information available to the decision maker consists of the Credit Bureau Risk Score and the Revenue Score of the candidate, a fact captured by the two (information) arcs into the Offer node. Alternatives at the Offer node may be some combination of promotional rate and APR. At the Credit Limit decision, the decision maker has observed, in addition to the two scores, whether the customer responded to the offer, and, if so, the Income on his credit application. The arc from Offer to Credit Limit conveys that the decision maker remembers and is aware of the alternative chosen (which offer has been made) at the previous decision.[18]

[16] Decision problem formulation is discussed in the section titled, "Decision Analysis: Key Concepts and Tools."

[17] There is less convention for value node representation, and sometimes they are represented by triangles or octagons.

[18] Such an information arc, between two decision nodes, is called a *no-forgetting* arc.

FIGURE 11: AN INFLUENCE DIAGRAM FOR A TWO-STAGE CREDIT CARD CAMPAIGN DECISION PROBLEM



The rest of the arcs in the influence diagram are dependence arcs. A strong statement in an influence diagram is made by the lack of such arcs, which implies *conditional* independence. For example, two conditional independence statements made by the model in Figure 11 are:

Given the Revenue and Loss, the Profit is conditionally independent of everything else.

Given the Credit Bureau Risk Score, the (positive) Response, and the Income, the Performance ("Good" or "Bad") of the customer is conditionally independent of the Revenue Score, and both decisions.

Conditional independence statements drastically simplify model complexity, which in most cases is not only highly desirable, but essential to a tractable solution that also allows some insight.

Applications

Influence diagrams are a powerful tool in modeling decision problems, because they allow for the specification and visualization of the structure of fairly complex problems in a compact graph that conveys explicitly the assumed dependence, or independence, among variables, the sequence of decisions, and the flow of information to the decision maker. They are most effective in the early stages of modeling an unstructured problem, when data and other details are unavailable, as a communication tool between a decision analyst and a decision maker. In conjunction with *sensitivity analysis*, they allow the determination of what matters in a problem and what does not, and thus the construction of tractable models that allow insight into the problem and its solution.

Influence diagrams with only chance nodes, also called *Bayesian Networks*, allow, through formal mathematical interpretation of the structure of their graphs and transformation algorithms, powerful probabilistic inference in much larger models. Some algorithms are designed to discover the conditional independence and other structural details of an underlying model from large datasets of empirical data on the component variable. These algorithms—some of which are used at FICO—enable the simplification of models of large problems that would otherwise be intractable.

Strengths

- Allow for the visualization of complex problems in a compact way, particularly the dependence structure among variables.
- Effectively communicate the relationships between variables and the sequence of decisions.
- Serve as a formal framework for Bayesian inference and learning.

Weaknesses

- Detail behind each node in the graph is not readily apparent.
- Typically unable to capture the asymmetric structure of a decision problem[19].

Decision Trees

In contrast to influence diagrams, decision trees explicitly show any asymmetry in the structure of a decision problem. They also show the functional and numerical details for each node on the corresponding branches. Each branch emanating from a decision node corresponds to an alternative and each branch emanating from a chance node corresponds to a possible outcome.

Figure 12 shows a very small portion of a decision tree for the credit initiation problem described above. It explicitly shows the following asymmetries:

- When there is no Response, the immediate realization of Profit[20] is the end of this scenario; other events, like Income and Performance, are never realized, and the Credit Limit decision never gets to be made.
- When the customer applies but the decision maker decides to not grant credit, the immediate realization of Profit[21] is similarly the end of this scenario.

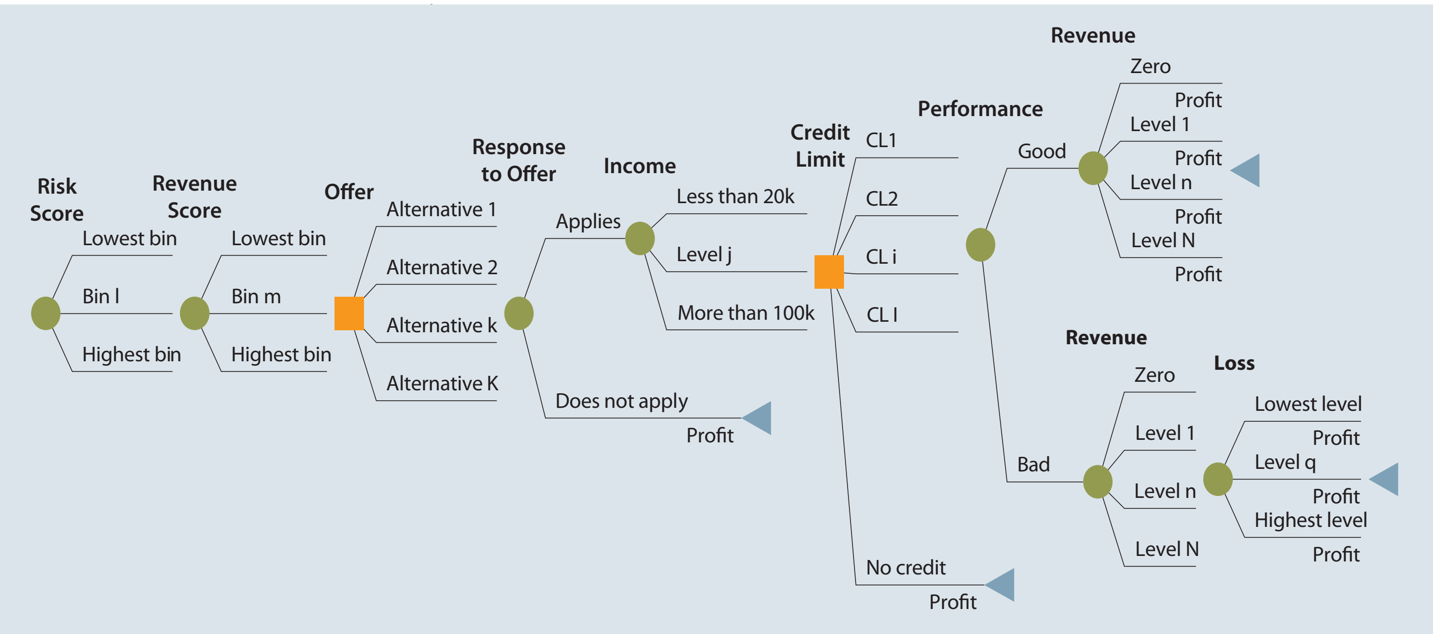
[19] Asymmetry in a decision problem refers to the very common situation where different scenarios do not have the same realization of variables or the same order of variables realized. In the example above, for instance, if the customer does not respond to the Offer, then the Credit Limit decision is never made and other events, like Income and Performance are never realized.

[20] The fixed cost of sending the offer, say, which is implicitly modeled.

[21] An implicitly modeled fixed cost that also would include in this case the evaluation of the application.

© 2012 Fair Isaac Corporation. All rights reserved.

FIGURE 12: A SECTION OF THE DECISION TREE DEPICTION OF THE CREDIT CAMPAIGN MANAGEMENT DECISION PROBLEM



Applications

Decision trees preceded influence diagrams by many years and are still indispensable when a highly asymmetric decision problem needs to be structured and modeled graphically. They are useful when used in conjunction with influence diagrams.

Strengths

- Details associated with each node are readily apparent in the graph.
- Asymmetric structure is readily displayed.

Weaknesses

- Decision trees become unwieldy for decision problems with even a moderate number of variables or a few stages.
- Conditional dependence and independence among variables are not readily apparent in the graph.

References

Clemen, R. T, (1996), *Making Hard Decisions: an Introduction to Decision Analysis*, Duxbury Press, 2nd ed.

Covaliu, Z. and Oliver, R. M. (1995), "*Representation and Solution of Decision Problems using Sequential Diagrams*," Management Science, 41.

Marshall, K. T. and Oliver, R. M. (1995), *Decision Making and Forecasting*, McGraw-Hill.

© 2012 Fair Isaac Corporation. All rights reserved.

Oliver, R. M. and Smith, J. Q. (1988), "*Influence Diagrams, Belief Nets, and Decision Analysis*," Proceedings of an International Conference, Wiley.

Raiffa, H. (1968), *Decision Analysis,* Addison-Wesley.

Shachter, R. (1988), "*Probabilistic Inference and Influence Diagrams*," Operations Research, 36.

© 2012 Fair Isaac Corporation. All rights reserved.

» Link Analysis

Computer-based link analysis is a set of techniques for exploring associations among large numbers of objects of different types. These methods have proven crucial in assisting human investigators in comprehending complex webs of evidence and drawing conclusions that are not apparent from any single piece of information. These methods are equally useful for creating variables that can be combined with structured data sources to improve automated decision-making processes. Typically, linkage data is modeled as a graph, with nodes representing entities of interest and links representing relationships or transactions. Links and nodes may have attributes specific to the domain. For example, link attributes might indicate the certainty or strength of a relationship, the dollar value of a transaction, or the probability of an infection.

Some linkage data, such as telephone call detail records, may be simple but voluminous, with uniform node and link types and a great deal of regularity. Other data, such as law enforcement data, may be extremely rich and varied, though sparse, with elements possessing many attributes and confidence values that may change over time.

Various techniques are appropriate for distinct problems. For example, heuristic, localized methods might be appropriate for matching known patterns to a network of financial transactions in a criminal investigation. Efficient global search strategies, on the other hand, might be best for finding centrality or severability in a telephone network.

Link analysis can be broken down into two components—link generation, and utilization of the resulting linkage graph.

Link Generation

Link generation is the process of computing the links, link attributes and node attributes. There are several different ways to define links. The different approaches yield very different linkage graphs. A key aspect in defining a link analysis is deciding which representation to use.

Explicit Links

A link may be created between the nodes corresponding to each pair of entities in a transaction. For example, with a call detail record, a link is created between the originating telephone number and the destination telephone number. This is referred to as an explicit link.

Aggregate Links

A single link may be created from multiple transactions. For example, a single link could represent all telephone calls between two parties, and a link attribute might be the number of calls represented. Thus, several explicit links may be collapsed into a single aggregate link.

Inferred Relationships

Links may also be created between pairs of nodes based on inferred strengths of relationships between them. These are sometimes referred to as soft links, association links, or co-occurrence links. Classes of algorithms for these computations include association rules, Bayesian belief networks and context vectors. For example, a link may be created between any pair of nodes whose context vectors lie within a certain radius of one another. Typically, one attribute of such a link is the strength of the relationship it represents.

Time is a key feature that offers an opportunity to uncover linkages that might be missed by more typical data analysis approaches. For example, suppose a temporal analysis of wire transfer records indicates that a transfer from account A to person X at one bank is temporally proximate to a transfer from account B to person Y at another bank. This yields an inferred link between accounts A and B. If other aspects of the accounts or transactions are also suspicious, they may be flagged for additional scrutiny for possible money laundering activity.

A specific instance of inferred relationships is identifying two nodes that may actually correspond to the same physical entity, such as a person or an account. Link analysis includes mechanisms for collapsing these to a single node. Typically, the analyst creates rules or selects parameters specifying in which instances to merge nodes in this fashion.

Utilization

Once a linkage graph, including the link and node attributes, has been defined, it can be browsed, searched or used to create variables as inputs to a decision system.

Visualization

In visualizing linking graphs, each node is represented as an icon, and each link is represented as a line or an arrow between two nodes. The node and link attributes may be displayed next to the items or accessed via mouse actions. Different icon types represent different entity types. Similarly, link attributes determine the link representation (line strength, line color, arrowhead, etc.).

Standard graphs include spoke and wheel, peacock, group, hierarchy and mesh. An analytic component of the visualization is the automatic positioning of the nodes on the screen, i.e., the projection of the graph onto a plane. Different algorithms position the nodes based on the strength of the links between nodes or to agglomerate the nodes into groups of the same kind. Once displayed, the user typically has the ability to move nodes, modify node and link attributes, zoom in, collapse, highlight, hide or delete portions of the graph.

Variable Creation

Link analysis can append new fields to existing records or create entirely new data sets for subsequent modeling stages in a decision system. For example, a new variable for a customer might be the total number of email addresses and credit card numbers linked to that customer.

Search

Link analysis query mechanisms include retrieving nodes and links matching specified criteria, such as node and link attributes, as well as search by example to find more nodes that are similar to the specified example node.

A more complex task is similarity search, also called clustering. Here, the objective is to find groups of similar nodes. These may actually be multiple instances of the same physical entity, such as a single individual using multiple accounts in a similar fashion.

Network Analysis

Network analysis is the search for parts of the linkage graph that play particular roles. It is used to build more robust communication networks and to combat organized crime. This exploration revolves around questions such as:

- Which nodes are key or central to the network?
- Which links can be severed or strengthened to most effectively impede or enhance the operation of the network?
- Can the existence of undetected links or nodes be inferred from the known data?
- Are there similarities in the structure of subparts of the network that can indicate an underlying relationship (e.g., modus operandi)?
- What are the relevant sub-networks within a much larger network?

- What data model and level of aggregation best reveal certain types of links and sub-networks?
- What types of structured groups of entities occur in the data set?

Applications

Link analysis is increasingly used in law enforcement investigations, detecting terrorist threats, fraud detection, detecting money laundering, telecommunications network analysis, classifying web pages, analyzing transportation routes, pharmaceuticals research, epidemiology, detecting nuclear proliferation and a host of other specialized applications. For example, in the case of money laundering, the entities might include people, bank accounts and businesses, and the transactions might include wire transfers, checks and cash deposits. Exploring relationships among these different objects helps expose networks of activity, both legal and illegal.

Strengths

Link analysis often makes information accessible that is not apparent from any single data record.

Weaknesses

Link analysis is as much an art as a science, and just configuring a link analysis can be a major endeavor.

References

Newman, M. E. J. (2003), *The structure and function of complex networks*, SIAM Review, Vol. 45, pp. 167-256.

Goldberg, H. G. and Wong, R. W. H. (1998), *Restructuring Transactional Data for Link Analysis in the FinCEN AI System*, Proceedings of the 1998 AAAI Fall Symposium on Artificial Intelligence and Link Analysis.

Wasserman, S., Faust, K., and Iacobucci, D. (1994), *Social Network Analysis: Methods and Applications*, Cambridge University Press.

» Log-linear Models

Log-linear models provide a systematic approach to the analysis and modeling of the observed cell frequency of occurrence in a cross-tabulation, developed purely for understanding the structure and modeling of *categorical* data.

Cross-tabulation Analyses

In their most general form, log-linear models are used to predict the cell frequency of occurrence in a cross-tabulation of *independent variables*. Hypotheses about the relationships between variables in the table can be tested by including parameters representing various levels of relationship complexity in the model. Such analyses are analogous to the analysis of the *correlation* structure of continuous variables.

One of the more useful areas of cross-tabulation analysis in business contexts involves models for measuring changes in behavior over time. Typical applications are analyzing the sequence of products purchased by customers from one purchase occasion to the next and the analysis of brand switching behavior over time.

The table below illustrates the type of data that are used in purchase sequence analyses. These data could be produced from a historical transaction file in a number of ways, such as recording the sequence of the next two products purchased subsequent to a fixed date by each customer in the file.

OBSERVED FREQUENCY COUNTS

		Second Product Purchased				
		A	B	C	D	TOTAL
First Product Purchased	A	1,701	6,472	6,921	3,190	18,284
	B	2,278	8,954	9,387	4,463	25,082
	C	1,632	5,799	6,879	3,021	17,331
	D	1,520	5,371	6,219	2,599	15,709
	TOTAL	7,131	26,596	29,406	13,273	76,406

The most basic question that arises from such a table is whether the product purchased first is independent of the product purchased next. The log-linear representation of this model of independence is:

$$\log m_{ij} = u + u_1(i) + u_2(j)$$

In this model, the logarithm of fitted cell counts m_{ij} can be decomposed into the sum of an overall mean *u*, a row effect $u_1(i)$ and a column effect $u_2(j)$ (hence the term "log-linear"). Independence of row and column effects is represented by the absence of an interaction term. The test of independence in this case is the familiar *chi square test* of independence. For these data, the chi square statistic is 51.7[22]. With 9 degrees of freedom, the p-value is much less than 0.0001.

[22] Pearson's chi square statistic X^2, calculated as follows: $X^2 = \sum \frac{(Observed - Expected)^2}{Expected}$. The expected value of each cell under the model of independence is simply the product of the row probability, the column probability and the sample size.

However, because of the large sample size, significance levels hold little meaning in this example. A more fruitful approach to understanding the data is to conduct an examination of the sources of the lack of fit with the model of independence and draw conclusions from the findings of that investigation. A simple but valid way to examine the sources of lack of fit is to produce a table of standardized residuals of the chi square statistic, as follows:

STANDARDIZED RESIDUALS

		Second Product Purchased			
		A	B	C	D
First Product Purchased	A	-0.13	1.35	-1.38	0.24
	B	-1.30	2.39	-2.71	1.61
	C	0.36	-3.01	2.56	0.19
	D	1.40	-1.32	2.23	-2.48

The calculated value in each cell is its signed contribution to the overall chi square statistic[23]. This table shows that the association between first and second purchase is primarily due to the loyalty of users of products B and C (and their antipathy to the other product in the pair) and movement from product D to product C.

When "structural zeros"[24] are present in the cross-tabulation, the calculation of the chi square statistic becomes more challenging, in that there is no closed form expression for estimated cell counts. Consequently, some form of iterative procedure for estimating cell counts is required. Typically, maximum *likelihood estimates* (MLEs) are produced using a *non-linear optimization algorithm*. Luckily, the likelihood function for log-linear models has a single maximum, so that global convergence is guaranteed. Another useful procedure for producing estimated cell counts—one that does not require the specification of the likelihood function—is Iterative Proportional Fitting (IPF).

[23] That is, $\frac{Observed - Expected}{\sqrt{Expected}}$. This value is analogous to a z-score for continuous data.

[24] In contrast to "sampling zeros," which are zero counts that occur due to insufficient sample sizes.

Logit Models

Log-linear models can also be applied to predict a categorical response (actually, the logit[25]) in a cross-tabulation of *predictor variables* and an *outcome variable*. This technique is analogous to regression modeling of continuous dependent variables. An example of logit modeling using two independent variables (age and residence) and a *binary outcome* variable (taking the values "good" or "bad") follows.

OBSERVED FREQUENCY COUNTS

		Outcome = Good **Residence**		
Age		Own	Rent	TOTAL
	< 30	65	480	545
	30 +	680	375	1,055
	TOTAL	745	855	1,600
		Outcome = Bad **Residence**		
Age		Own	Rent	TOTAL
	< 30	35	120	155
	30 +	120	125	245
	TOTAL	155	245	400

OBSERVED LOGIT

		Residence		
Age		Own	Rent	TOTAL
	< 30	0.62	1.39	1.26
	30 +	1.73	1.10	1.46
	TOTAL	1.57	1.25	1.39

We might hypothesize that the relationship between the "good" or "bad" outcome and the predictor variables can be represented by the overall mean effect (the overall mean of log-odds) and the main effects of each predictor (the deviations of row total or column total means of log-odds from the overall mean):

$$\textit{Expected Logit}_{age_i,\ resid_j} = u + u_{age_i} + u_{resid_j}$$

[25] A logit is the natural log of the odds of occurrence of a binary outcome. For a *binary outcome* Y with values 1 and 2 where $p = \Pr(Y = 1|\mathbf{x})$, logit $(p) = \log_e\left(\frac{p}{1-p}\right)$.

© 2012 Fair Isaac Corporation. All rights reserved.

Using Maximum Likelihood Estimation to estimate the parameters, the model obtained is:

$$\mathit{Expected\ Logit}_{ij} = 1.398 + \left[-.0345 \mid age < 30\right] + \left[.0345 \mid age \geq 30\right] + \left[.1450 \mid resid = own\right] + \left[-.1450 \mid resid = rent\right]$$

The estimate of the logit of the cell representing Age < 30, Residence = Own is therefore:

$$\mathit{Expected\ Logit}_{Age < 30,\ resid = own} = 1.398–.0345+.1450+1.5085$$

Estimated values of the logits for each combination of age and residence category, along with fitted cell counts based on those estimates, follows.

FITTED FREQUENCY COUNTS

Outcome = Good
Residence

		Own	Rent	TOTAL
Age	< 30	82	463	545
	30 +	663	392	1,055
	TOTAL	745	855	1,600

Outcome = Bad
Residence

		Own	Rent	TOTAL
Age	< 30	18	137	155
	30 +	137	108	245
	TOTAL	155	245	400

FITTED LOGIT

Residence

		Own	Rent	TOTAL
Age	< 30	1.51	1.22	1.26
	30 +	1.58	1.29	1.46
	TOTAL	1.57	1.25	1.39

The main effect parameters provide us with information about the direction and magnitudes of the main effects, e.g., those under 30 are expected to have lower logit values than those above 30, and owners are expected to have higher logit values than renters. The magnitude of the effects (vs.) indicates that the residence factor is a stronger main effect than age. The difference between observed value (0.62) and expected value (1.51) is substantially different for this cell as well as others. To test the significance of the difference between expected and observed frequencies of occurrence, a Likelihood Ratio chi square value[26] is computed. A significant chi square value leads us to reject the hypothesis that a main effects model provides a sufficient characterization of the structure of this data. In this case, the chi square value is 24.71, which corresponds to a p-value of 0.000001 with the 1-degree of freedom in the main effects model. Therefore, the main effects model is rejected, and we conclude that a significant interaction between age and residence is present.

A simple way of gaining an understanding of the source of the interaction effect is to calculate standardized residuals for the fitted main effects model relative to the observed counts.

STANDARDIZED RESIDUALS

Outcome = Good **Residence**			
Age		Own	Rent
	< 30	-1.87	0.79
	30 +	0.66	-0.85
Outcome = Bad **Residence**			
Age		Own	Rent
	< 30	3.79	-1.44
	30 +	-1.44	1.62

In this case, the major source of the interaction is clearly the count of young owners with bad outcomes, which has a much higher observed value than the main effects model would predict. One interpretation of this result is that the overall positive effect of ownership on outcome does not apply among the young.

One might have foreseen this result by comparing the observed logit values in the cells to their marginal row and column values. If the main effects model is appropriate, the cells should reflect the direction and magnitude differences of the marginal totals, when in fact the observed logits show reversals in patterns and differences in magnitudes.

Applications

The example above illustrates the value of testing the fit of log-linear models in various applications. Log-linear models are most frequently encountered in the social sciences, where the need to understand relationships between categorical data is often required. Marketers have used log-linear models for response modeling, with trees as a pre-processor to reduce the number of variables.

[26] The Likelihood Ratio chi square statistic or likelihood ratio criterion χ^2_L is generally used as the goodness of fit measure for logit models, where $\chi^2_L = 2.\sum\left((Observed).log_e\left(\frac{Observed}{Expected}\right)\right)$.

© 2012 Fair Isaac Corporation. All rights reserved.

Strengths

- Provides methods for analyzing categorical data that are analogous to correlation and regression analyses of continuous data.
- One of the more effective approaches for detecting low-dimensionality interactions between variables.
- Makes no assumptions about the distribution of the predictor data.
- Appealing as a segmentation tool, as it identifies unique segments of data.
- Provides an interpretation of the direction and magnitude of relationships in multi-dimensional tables.

Weaknesses

- Data get sparse quickly as dimensionality increases.
- Model is usually limited to low level of dimensionality, unless a very large sample of data is available. To be effective, this technique needs to be combined with a variable reduction pre-processor.
- Requires data to be categorical.

References

Bishop, Y. M., Fienberg, S. E., Holland, P. W. (1975), *Discrete Multivariate Analysis: Theory and Practice, Cambridge*, MA: The MIT Press.

Everitt, B. S. (1977), *The Analysis of Contingency Tables*, London: Chapman and Hall.

Feinberg, S. E. (1977), *The Analysis of Cross-Classified Categorical Data*, Cambridge, MA: The MIT Press.

Magidson, *J., CHAID, LOGIT, and Log-linear Modeling*, Marketing Research Systems, 11-130, Datapro Research Corporation, May 1988.

© 2012 Fair Isaac Corporation. All rights reserved.

» Mathematical Programming

An optimization problem consists in minimizing (or maximizing) an objective function subject to constraints. A large class of optimization problems can be represented and solved by Mathematical programming techniques. Mathematical programming originated in the 1940s, when the term "programming" was still synonymous with scheduling or planning. Mathematical programming solutions are utilized when there is no closed, algebraic solution for determining the optimum value of the objective function or when the derivation of an algebraic solution requires more time and effort than a mathematical programming technique.

In its most general form, the goal of mathematical programming is to

$$\begin{array}{ll} \text{Minimize:} & f(x) \\ \text{Subject to:} & g_i(x) = 0, \text{ for } i = 1, 2, \ldots, p \\ & h_j(x) \geq 0, \text{ for } j = p+1, p+2, \ldots, \end{array}$$

The parameters *x* are often called *decision variables*. Finding the minimum value of *f* is equivalent to finding the maximum value of –*f*, so that any maximization problem can be converted to a minimization problem.

Specializations of this general formulation include:

- Linear programming (LP): the special case when *f*(x), *g*(x), and *h*(x) are all linear functions.
- Quadratic programming: *f*(x) is at most a quadratic function, and *g*(x), and *h*(x) are linear functions.
- Integer programming: the special case when the x's are required to be integer values, or take values from some discrete (finite) set.
- Non-linear programming (NLP): the objective function *f*(x) and/or the constraint functions *g*(x) and *h*(x) are nonlinear functions.
- Unconstrained optimization: *g*(x) and *h*(x) are an empty set.

When the objective function being optimized is well behaved, the outcome of the optimization is the identification of the optimal feasible solution, and the combination of decision variables, x*, that define the optimal solution. In predictive modeling, LP and NLP problems estimate the values of parameter coefficients, while in decision modeling, they are used to identify the optimal decision.

Linear Programming

The most widely used algorithm to solve a linear programming problem is the simplex method. Discovered in the 1940s, this method derives its name from the geometry of the solution—the "simplex" is the feasible region described by the linear constraints. At least one solution lies at a vertex of the simplex.

Consider the following example: we wish to establish the most profitable production plan for two products, X and Y. One unit of X yields a profit of $4, and one unit of Y has a profit of $5. The resources available for production (raw material quantity and machine hours) are limited. One unit of product X requires one ton of raw material and 4 hours of machine time, product Y needs 3 tons of raw material and 3 hours of machine time.

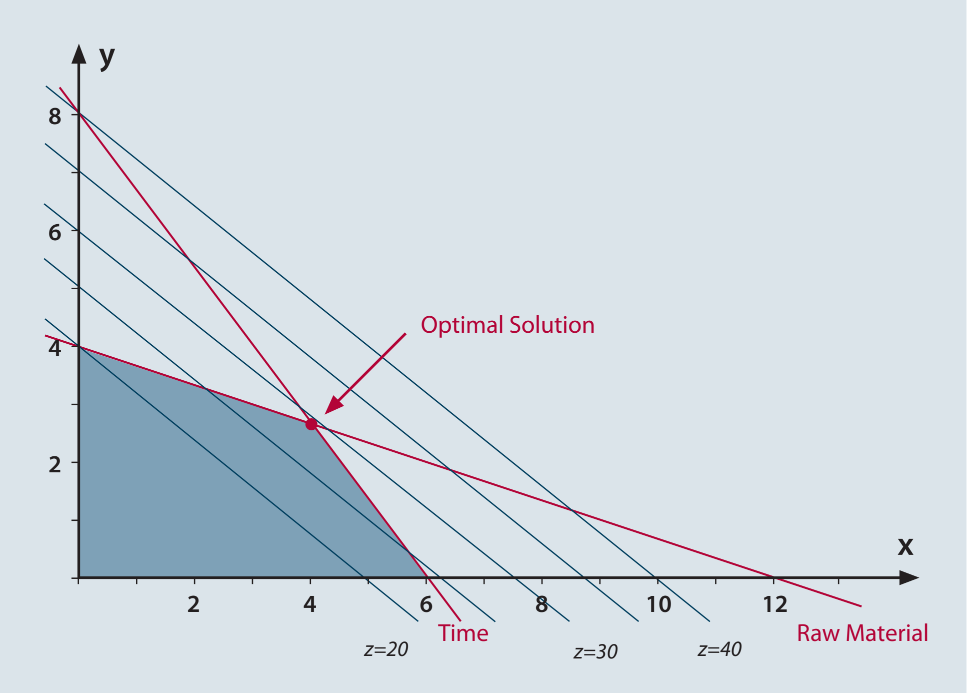
In mathematical terms, we can formulate this problem as an LP problem with two decision variables, *x* and *y*, representing the quantity of each product:

$$\begin{array}{llll} \text{Maximize:} & z = & 4x + 5y & \text{(total profit)} \\ \text{Subject to:} & & x + 3y \leq 12 & \text{(raw material limit)} \\ & & 4x + 3y \leq 24 & \text{(machine time limit)} \\ & & x \geq 0, y \geq 0 & \text{(cannot produce negative quantities)} \end{array}$$

© 2012 Fair Isaac Corporation. All rights reserved.

And here is a graphical representation of this problem:

FIGURE 13: GRAPHICAL SOLUTION OF AN LP PROBLEM IN TWO VARIABLES



In Figure 13, the simplex is represented by the shaded region of the graph. The simplex method generates a sequence of feasible solutions by moving from one vertex of the simplex to the next. The algorithm terminates when no adjoining vertices with higher objective function values can be found. In our case, the optimal feasible solution is at the point (4, 8/3).

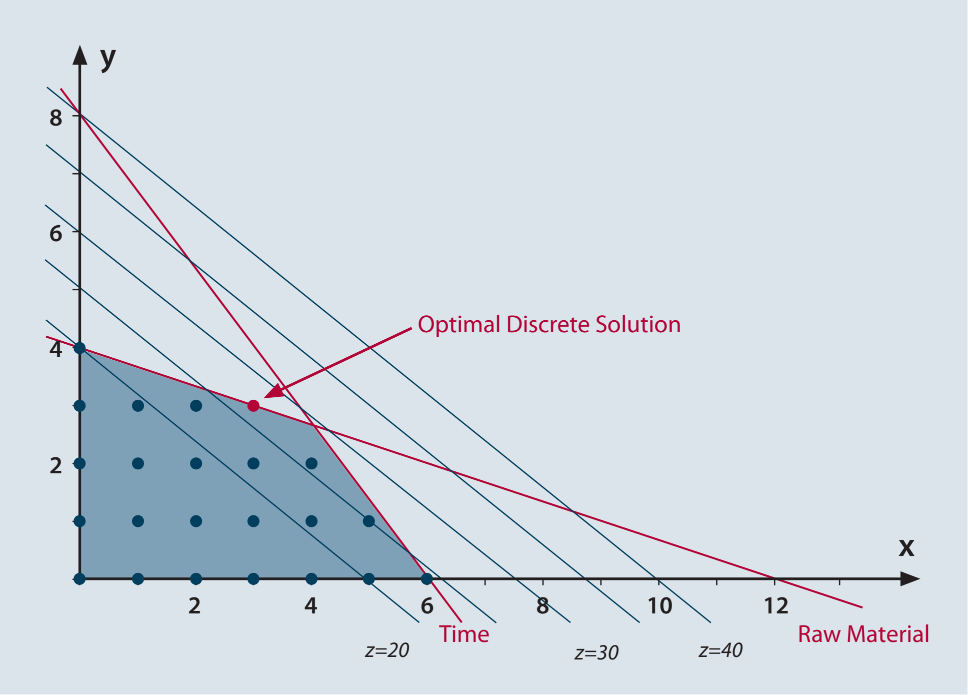
In the worst case, the simplex method requires a number of iterations exponential in the number of unknowns. The search for more efficient algorithms has led to the use of interior-point algorithms, which visit points within the interior of the feasible region, and are polynomially, rather than exponentially, complex.

Mixed-integer Programming

In many economic situations fractional solution values are not a desirable outcome. In our example we really are interested in optimizing the number of finished products (we cannot sell a unit of Y that has gone through 2/3 of the production process). In mathematical terms this corresponds to adding the condition that the decision variables *x* and *y* take only discrete (integer) values. Graphically this results in the following picture (Figure 14) where the integer feasible solutions are marked by dots:

© 2012 Fair Isaac Corporation. All rights reserved.

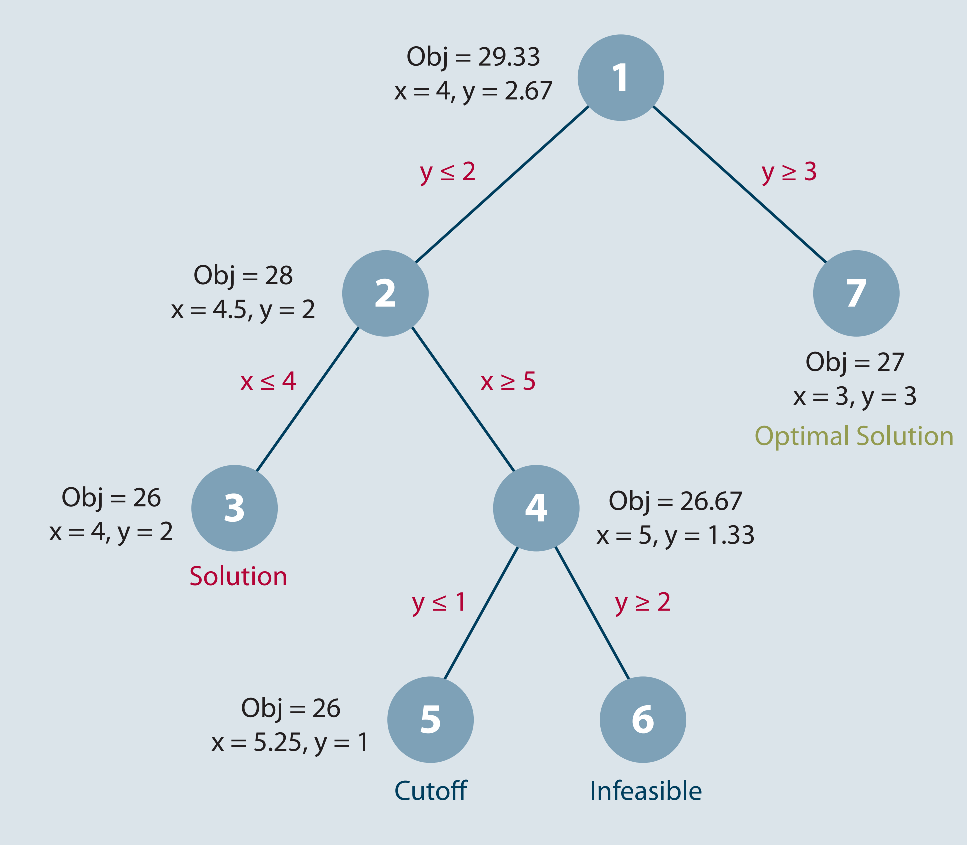
FIGURE 14: GRAPHICAL SOLUTION OF A MIP PROBLEM IN TWO VARIABLES



This type of problem belongs to the class of (Mixed-)Integer Programming. The standard technique for solving Mixed-Integer Programming problems is a tree-search algorithm known as Branch-and-bound where at each node an LP problem is solved. The following Figure 15 shows an example of what the search tree may look like for our small problem.

© 2012 Fair Isaac Corporation. All rights reserved.

FIGURE 15: MIP SEARCH TREE EXAMPLE



The number of nodes in the search tree grows exponentially with the number of decision variables. Solving problems of realistic size is possible only by combining the branch-and-bound search with other techniques, such as logic deductions simplifying the problem, analysis of the mathematical substructures, and reformulations improving its numerical properties.

Non-linear Programming

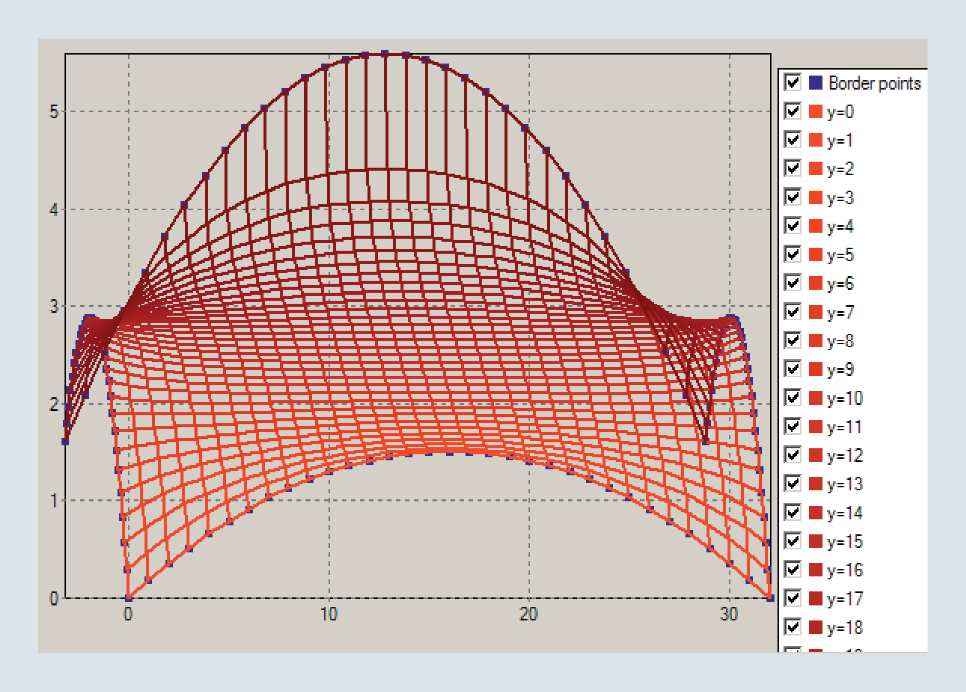
A variety of approaches exist to solving a non-linear programming problem. The degree of difficulty is driven by the complexity of the objective function, which may be difficult or computationally costly to compute, and by the inequality constraints, which impose discontinuities on the solution. One of the greatest challenges for NLP algorithms is to avoid getting trapped in "local optima", that is, values that are optimal within a confined neighborhood, but sub-optimal globally.

An NLP algorithm begins with an initial guess (or estimate) for the values of the decision variables. The algorithm proceeds by adjusting the estimate iteratively, using information about the objective function in the neighborhood of the current estimate. NLP algorithms differ in terms of the type of information they use for making the iterative adjustments. Some of the more widely used methods include:

- Gradient steepest descent: Iterations occur in the direction of the gradient. This approach converges slowly and unreliably.

- Conjugate gradient: Uses the difference between the gradient in the last iteration and the current gradient to infer the curvature of the objective function and derive a superior direction.
- Quasi-Newton: Computes the gradient directly and approximates the Hessian, or second derivative, to provide an improved search direction.
- Genetic algorithms (refer to the section on this topic).
- Simulated annealing.

FIGURE 16: SOLUTION GRAPHIC FOR AN NLP PRODUCED BY XPRESS-IVE



Applications

Mathematical programming is utilized widely to solve prediction and decision problems in the areas of finance, operations management, economics and the physical sciences. NLP techniques are often hidden within commonly used multivariate statistical software programs (e.g., maximum likelihood estimation for log-linear models) and in decision optimization software.

FICO makes wide use of mathematical programming. A linear program was at the heart of INFORM11, FICO's previous predictive modeling technology. The Scorecard Module in FICO® Model Builder, which incorporates the next generation of INFORM technology, uses a quadratic programming algorithm to optimize model weight. Linear programs are utilized in FICO's proprietary strategy optimization software, FICO® Decision Optimizer. Non-linear programs (in the form of genetic algorithms) are also utilized by FICO's Data Spiders™ module for Model Builder to identify optimal sets of predictive variables and by FICO's Adaptive Random Trees module to generate segmented model schemes.

State-of-the-art mathematical programming solvers are orders of magnitude better in terms of speed, size of model that they can solve, and numerical reliability than a method coded from its description in a textbook. For example, in the 1980s Intel decided to showcase their new

© 2012 Fair Isaac Corporation. All rights reserved.

286/287 processor/co-processor pair, which performed floating point operations in hardware, by programming a 'textbook' LP algorithm on a 16 processor pair hypercube. The Xpress LP solver was still faster running on a single 286 using software emulation for the floating point operations—a platform some 500 times slower than that used by Intel. Xpress is one of the two state-of-the-art solvers available today and is many thousands of times faster than such textbook methods, and around 10 times faster than the next tier of its competitors.

As well as the solver or 'Optimizer' it comprises a model builder, Mosel. Essentially this takes an algebraic formulation such as those given above, combines it with user data and poses a specific numerical model instance for solution, and analyses solutions found. Its ease-of-use and flexibility is of paramount importance to users for whom ensuring model correctness and maintenance is as important as speedy model solution.

The Xpress mathematical programming system was acquired by FICO in January 2008 from Dash Associates, who had originally developed it (Ashford, 2007).

Strengths

- Many techniques are available, so if one does not work for a particular problem, another might.
- LPs and NLPs handle a wide variety of objective functions and constraints.
- Mathematical programming is a well-researched area, so that guidance is available in the literature to help determine appropriate techniques for particular problems.

Weaknesses

- For NLPs, there is seldom a guarantee that a particular technique will converge to a solution for a particular problem or that the solution converged to will be a global minimum.
- Because many of these methods for solving NLPs are iterative, they can be computationally intense and require long execution times.

References

Ashford, R. (2007), Mixed integer programming: a historical perspective with Xpress-MP, *Ann. Oper. Res*, 149, pp5-17.

Burden, R. L., and Faires, J. D. (1993), *Numerical Analysis*, PWS Publishing.

Vasek Chvatal. (1980) *Linear Programming*. W. H. Freeman and Company.

Cuthbert, T. R. (1987) *Optimization Using Personal Computers*, John Wiley.

Dennis, J. E., and Schnabel, R. B. (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall.

Gallant, A. R. (1987), *Nonlinear Statistical Models*, John Wiley.

Hillier, F. S., and Lieberman, G. J. (1986), *Introduction to Operations Research*, McGraw-Hill.

Guéret, C., and Heipcke, S., and Prins, C., and Sevaux, M. (2002), *Applications of Optimization with Xpress-MP, Dash Optimization*. http://www.dashoptimization.com/applications_book.html

Pierre, D. A. (1986), *Optimization Theory with Applications, Dover*.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1986), *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press.

» Multiple-objective Decision Analysis

Rarely in realistic decision problems[27] is there a single objective. When there are two or more objectives, they typically conflict, in the sense that some strategy is optimal—performs best—with respect to one objective, while a different strategy is optimal with respect to another.

When multiple objectives are at stake, they must somehow be aggregated into a single measure of performance, to which a decision rule can be applied, unless a subjective decision rule is left to the decision maker's choice. One way to reconcile conflicting objectives is through tradeoffs. The decision maker needs to articulate his or her preferences in terms of tradeoffs among the objectives.

Explicit tradeoff factors allow decision makers to specify how much they are willing to give up in one objective to gain a unit in another. For example, in a credit card portfolio, where both loss and volume are important, a tradeoff could measure how much the portfolio manager is willing to risk in expected loss in order to increase expected volume of 1,000 accounts.

The limitation of tradeoff factors is that their value is implicitly constant throughout the applicable range of the objectives, which typically is not true. For example, in the portfolio management example, the manager may be willing to increase expected losses by $10,000 to increase volume from 15,000 to 16,000 accounts, but only by $5,000 to increase volume from 100,000 to 101,000 accounts. Another shortcoming of tradeoff factors is in their failure to capture interactions among objectives.

Efficient Frontiers

A simple and very effective way to graphically depict the tradeoffs among objectives is by using efficient frontiers. Given a decision model, a frontier represents, in the space of two or more objectives or attributes[28], the set of all achievable points by a specific strategy.

Figure 17 shows, for example, the expected-Volume-expected-Profit frontier associated with an accept-reject strategy in a credit portfolio origination decision, based on a single risk score. It illustrates that the lower the score cutoff, above which applicants are accepted, the higher the volume. In the high range of score values, where decreasing the cutoff mainly accepts "Good" applicants, the profit also increases. In the low range, however, continued decrease of the score cutoff results in reduced profit, because more and more "Bad" applicants are accepted. Only the thicker portion of the frontier is efficient in the sense that, for any given level of expected Profit, all decision makers would prefer a higher, rather than a lower, expected Volume.

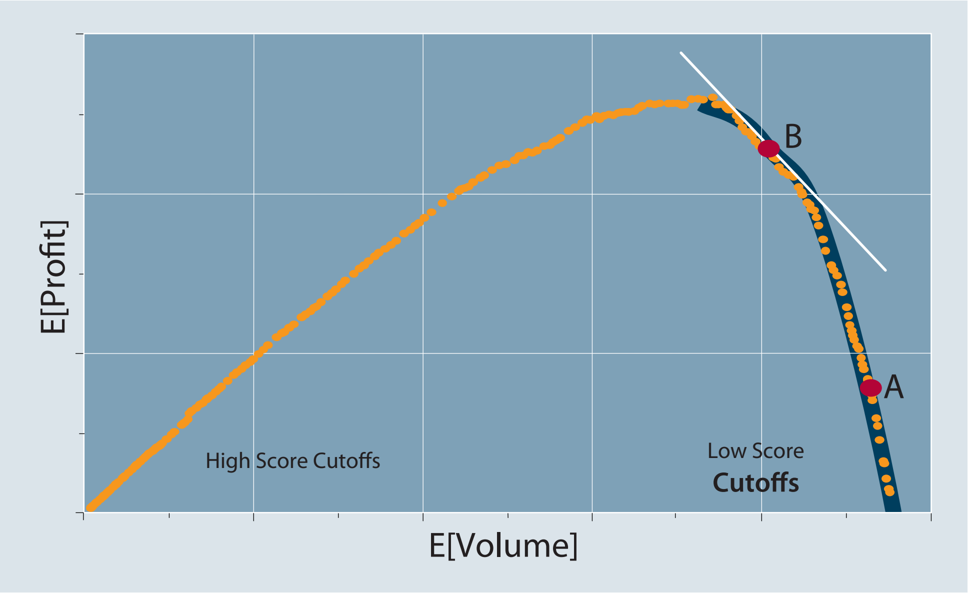
The choice of the operating point on the efficient frontier should be determined by the decision maker judgmentally, by considering the subjective tradeoff between profit and volume. A portfolio manager for whom volume is relatively more important, will choose a lower score cutoff, corresponding to point A, while another, for whom volume is relatively less important, will choose a higher score cutoff, corresponding to point B. In fact, if the portfolio manager can assess his or her tradeoff factor between Profit and Volume, the optimal point on the efficient frontier would be the one where the slope of the tangent line equals the tradeoff value.

[27] Decision problem formulation is discussed in the section "Decision Analysis: Key Concepts and Tools."

[28] An attribute in this context refers to the quantity and appropriate scale that measures the achievement of an objective. For example, cost in dollars is the objective when minimizing expected cost.

© 2012 Fair Isaac Corporation. All rights reserved.

FIGURE 17: EFFICIENT FRONTIER IN PROFIT-VOLUME SPACE



Tradeoffs can be regarded as a way to assign weights of importance to the various objectives. Multi-attribute utility theory provides a framework to assign these weights systematically, such that interactions among objectives, as well as risk attitude, are also taken into account. The decision rule is then to choose the strategy that maximizes the expected multi-attribute utility[29].

Applications

Multiple-objective decision analysis has been applied explicitly in many areas of government and medical decision making, and only in relatively few specific business areas, such as location analysis. In the credit and financial services industries, it has become more and more important in recent years to explicitly model objectives such as loss, market share and risk, in addition to net profits.

References

Bunn, D. (1984), *Applied Decision Analysis*, McGraw-Hill.

Clemen, R. T. (1996), *Making Hard Decisions: an Introduction to Decision Analysis*, Duxbury Press, 2nd ed.

Haimes, Y. Y., Li, D, and Tulsiani, V. (1990), "*Multiobjective Decision-Tree Analysis*" Risk Analysis, 10, 1.

Keeney, R. L., and Raiffa, H. (1976) *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, J. Wiley and Sons.

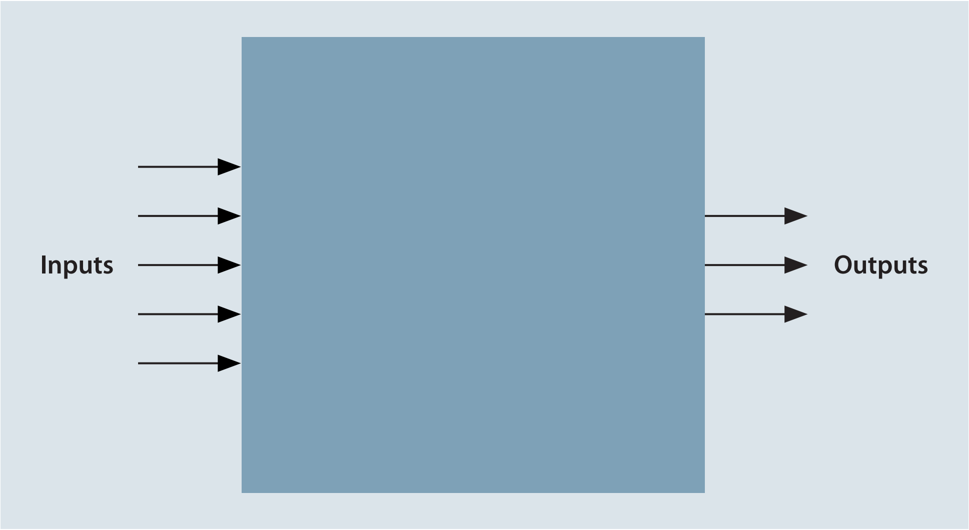
Marshall, K. T. and Oliver, R. M. (1995) *Decision Making and Forecasting*, McGraw-Hill.

[29] Refer to the section "Utility Theory" for further discussion.

» Neural Networks

A neural network[30] (NN) is an information processing structure that transforms a set of inputs into a set of outputs (see Figure 18). The manner in which a NN performs this transformation is inspired by researchers' understanding of how the human brain and nervous system process information. More specifically, a NN is a collection of simple processing units linked via directed, weighted interconnections (see Figure 19). Each processing unit receives a number of inputs from the outside world and/or other processing units, weights these inputs based on the weights of the corresponding interconnections, combines these weighted inputs, produces an output based on this combined input, and passes this output to other processing units via the appropriate weighted interconnections (see Figure 20). Mathematically, this process can be represented by a function that maps the set of inputs to a set of outputs. In general, this function is *non-additive* and *non-linear*.

FIGURE 18: AN INFORMATION PROCESSING STRUCTURE



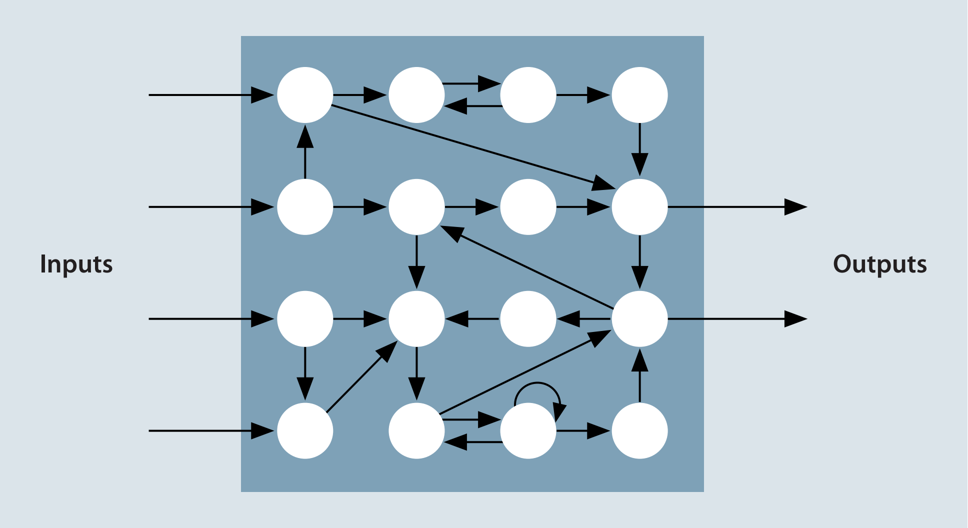
The development of a NN generally consists of the following three steps:

1. The first step is the definition of a network structure. The network structure is determined by the number of processing units and the manner in which these processing units are connected.
2. The second step is the definition of the computational aspects of the network or how the individual processing units combine and transform inputs. Once the developer has completed the first two steps, she has defined the form of the mathematical formula that relates the inputs to the outputs.
3. The final step is the determination of a set of weights such that the network performs a useful function. This process is generally iterative; data are presented to the network and the weights are updated after each presentation according to some mathematical rule. Hence, this iterative process is often referred to as training, adaptation, or learning.

[30] More accurately called artificial neural network.

© 2012 Fair Isaac Corporation. All rights reserved.

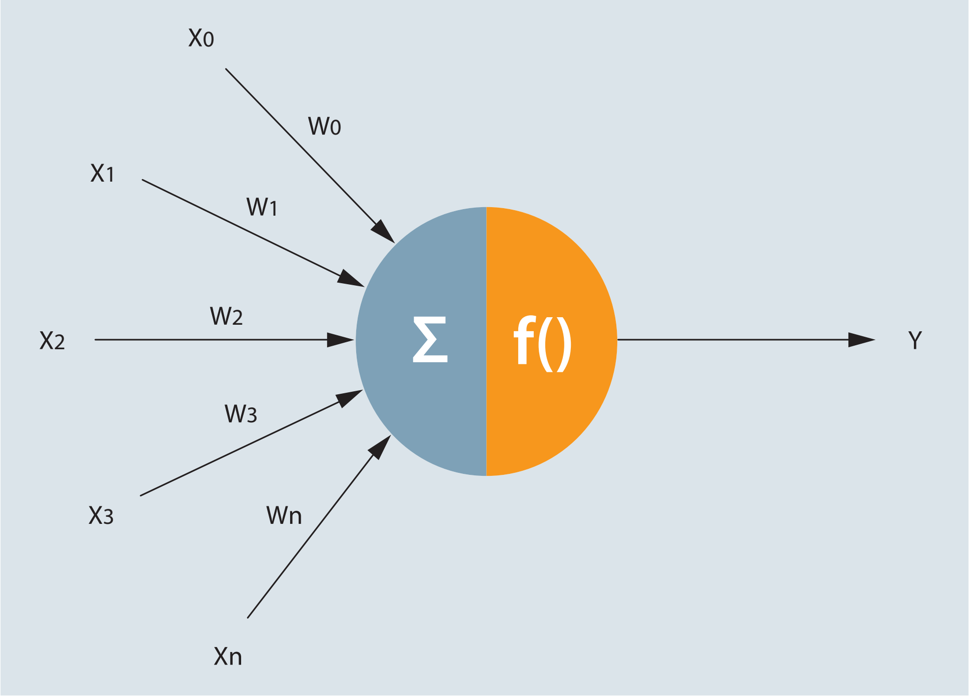
FIGURE 19: A COLLECTION OF SIMPLE PROCESSING UNITS LINKED VIA DIRECTED, WEIGHTED INTERCONNECTIONS



There are a variety of different NN paradigms, only a few of which are appropriate for statistical modeling applications. The most common neural network model used in commercial applications is the multilayer perceptron neural network. This neural network uses the back propagation mathematical algorithm to determine the optimal set of interconnection weights. The goal of the backpropagation algorithm is often the same as *least-squares regression*[31]. As a result, the backpropagation algorithm is often classified as an iterative, nonlinear least-squares regression technique.

[31] The objective of the backpropagation algorithm is to find the set of interconnection weights such that the mathematical formula represented by the backpropagation neural network minimizes the sum of the squared errors between the network outputs and the desired outputs over some data set.

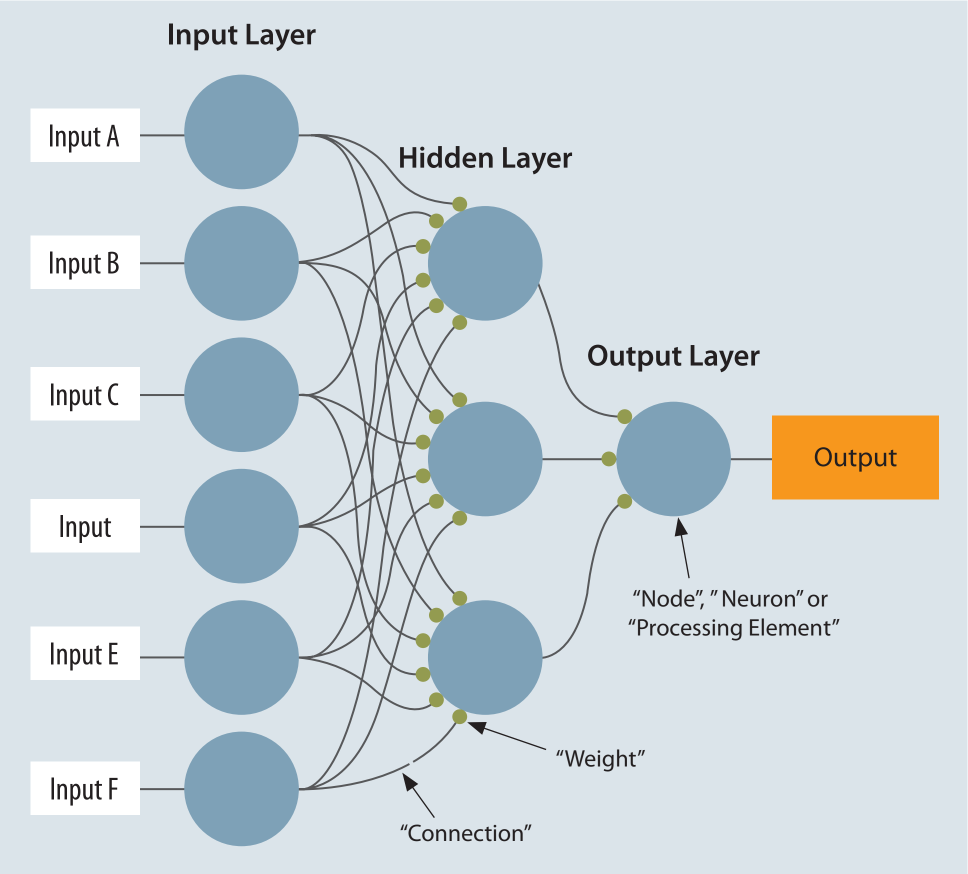
FIGURE 20: A SIMPLE PROCESSING UNIT



Let us look at an example of a typical NN developed for a credit scoring application. This network is represented graphically in Figure 21. The inputs may include information such as "age" and "owner." The value of the input "age" is simply the age of the applicant. The value of the input "owner" is 1 if the applicant owns their residence, otherwise its value is 0. The output of the network is a "score" that indicates the likelihood that an individual will be good in the future.

© 2012 Fair Isaac Corporation. All rights reserved.

FIGURE 21: FULLY CONNECTED FEEDFORWARD NN



Applications

Neural networks, and multilayer perceptron neural networks in particular, have been used to address a variety of problems, a few of which are listed below:

- Optical character recognition.
- Industrial adaptive control systems and robotics.
- Image compression.
- Medical diagnosis based on a set of symptoms.
- Statistical modeling.

Strengths

- Capture nonlinear interactions among input data and between input and output.
- Need fewer segments.
  - Mainly when there is lack of uniformly available data.
- Provide automatic feature extraction through hidden nodes.
- Have strong theoretic foundations and Bayesian connection.

© 2012 Fair Isaac Corporation. All rights reserved.

  - A single hidden layer neural network with sufficient hidden units can represent any continuous functional mapping to arbitrary accuracy.
  - For classification problem, neural networks can approximate any decision boundary to arbitrary accuracy.
  - Neural networks are universal non-linear discriminant functions.
  - Neural networks can appropriate posterior probabilities of class membership.
- Handle multi-outcome and continuous outcome models.
- Make reverse engineering of scores extremely difficult.

Weaknesses

- Provide little data insight and are difficult to interpret.
  - Automatic Reason Code Generator developed by FI helps reveal the insight.
- Can overfit the development data if used naively.
  - Deploying advanced statistical techniques such as penalized objective functions and *cross-validation* methods can alleviate this danger.
- Have few capabilities around constraining or otherwise engineering the weights.
- May be sensitive to the starting point due to the possibility of multiple locally optimal solutions.
  - Annealing techniques, such as training with different starting points and learning rates can help avoid local minimums or maximums.
- Are relatively computationally intensive.

References

Bishop, Christopher M. (1995), *Neural Networks for Pattern Recognition*, Oxford: Clarendon Press.

Haykin, Simon. (1994), *Neural Networks: A Comprehensive Foundation*, New York: Macmillan College Publishing Company.

Hecht-Nielsen, Robert. (1990), *Neurocomputing*, Reading: Addison-Wesley Publishing Company.

Hertz, John A., Krogh, Anders, S., and Palmer, Richard G. (1992), *Introduction to the theory of neural computation*, Redwood City: Addison-Wesley Publishing Company.

Masson, Egill and Wang, Yih-Jeou (1990), *Introduction to computation and learning in artificial neural networks*, European Journal of Operational Research 47, 1-28.

Masters, Timothy (1993), *Practical Neural Network Recipes in C++,* Boston: Academic Press, Inc.

Masters, Timothy, (1995), *Advanced Algorithms for Neural Networks*, New York: John Wiley & Sons, Inc.

Weiss, S. H. and Kulikowski, C. A. (1991), *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*, Morgan Kaufmann, San Mateo, California.

Zahedi, F. (1991), *An introduction to neural networks and a comparison with artificial intelligence and expert systems*, Interfaces, Vol. 21, No. 2 (March-April), pp. 25-38.

» Pattern Recognition

Pattern recognition can be defined as the categorization of input data into identifiable classes via the extraction of significant *features* or attributes of the data from a background of irrelevant detail. The historically most frequent areas of application are in spatial pattern recognition—3-D image processing, character and voice recognition, and in temporal pattern recognition—weather forecasting and financial time series forecasting.

The field of pattern recognition employs a large variety of technologies including regression, clustering, genetic algorithms, principal component analysis, trees and neural networks. Pattern recognition, like other fields in the data mining arena, is characterized by automated searches over a large number of observations and huge combinatorial spaces. While the problems traditionally addressed in pattern recognition—character recognition and image processing—are somewhat unique, the technologies employed are not.

Basic Components

The basic components of pattern recognition are feature extraction and *classification*. Feature extraction is the process of converting potentially huge amounts of raw data into a sufficient, manageable vector of features, via data compression and dimensionality reduction techniques. Features can be the original raw input but are more often transformations of that raw input. The transformations can be simple—the *main effects* as represented by the original inputs—or they can be quite complex—high order *interactions* across many inputs. (In FICO terms, a typical feature vector would be represented by the *attributes* of generated *predictor variables*—the predictors themselves could represent main or interaction effects.)

For example, the problem may be to identify authorized personnel in a secured building from a facial scan. Feature extraction from a facial snapshot entails segmenting the image into many small local regions, and extracting the most important features from within and across regions, e.g., the length of the ear lobe and the distance between ear lobe and corner of eye.

This process of feature extraction is geared towards improving the raw data for classification purposes. It might begin with a pre-processing step, which might standardize, scale, clamp or smooth the raw input data. The next step focuses on positing and assessing the value of a wide variety of potential transformations. The transformations most helpful in the classification process are retained. This wide set of potential transformations often are generated from one or more of:

- Domain expertise applied directly by the analyst to specify particular transformations.
- Automated creation of features through adaptive function estimation algorithms (neural networks, trees and Multivariate Adaptive Regression Splines (MARS) fall into this category).
- Automated creation of features through genetic algorithm searches over a family of possible transformations (FICO's Data Spiders™ technology falls into this category).
- An algorithmic approach to iteratively transform the raw inputs—that is to say a transformation with no closed, algebraic expression.
- The search across these potential transformations can also take a variety of forms from traditional optimization techniques to genetic algorithms to efficient combinatorial searches, which are often customized to the particular data source or transformation set.

Classification is the partitioning of the feature space into decision regions which correspond to classes, such that each instance of a feature vector can be classified as belonging to one of N classes. In the authorized personnel example, given a snapshot instance where the ear lobe is 3 mm and the distance between ear lobe and corner of eye is 12 mm, the image may be classified reliably as

© 2012 Fair Isaac Corporation. All rights reserved.

belonging to authorized personnel member X. A classification algorithm provides the parameter estimates for the features identified in the feature extraction step. The distinction between feature extraction and classification steps is blurring, as computing enhancements allow for greater automation of the entire process.

Pattern Recognition Technology

There are thousands of hybrid approaches, representing various combinations of technologies and practical guidelines, used in the field of pattern recognition. Most of the technologies are drawn from the fields of machine learning and statistics.

Feature extraction methods can draw on the experts' application domain expertise, but some rely on automated, exhaustive or filtered searches and evaluations across a very large combinatorial space. For example, one simplistic rule-based approach to a credit card fraud problem would be to find all 'rules' or subsets of data where the probability of fraud is "very high." For a data set with 100 predictors of 5 attributes each, that would mean searching across 5^{100} possible rule combinations. In order for searches of such magnitude to occur in real time, intelligent database management, in the form of query optimization and parallel data base servers, is generally required of the search software. Genetic algorithms have also successfully been used to make the search dilemma more tractable. On the statistics side, principal component analysis, clustering and *Bayesian networks* have been used to identify features and reduce the problem's dimensionality.

Classification technologies run the gamut of modeling technologies. A partial list includes discriminant analysis, linear and non-linear regression, tree methods and various types of neural networks. Trees, because of their history with rule-based systems in AI, seem to be a popular approach. Neural networks, which like tree methods make no assumptions about the data and capture interactions automatically, are also a commonly used approach for classification.

Applications

Pattern recognition techniques are often used for image processing, character and voice recognition, as well as weather forecasting and financial time series forecasting. Applications continue to expand with recent examples in the area of credit risk, marketing and fraud detection model development. Descriptive modeling of website behavior built by analyzing click-stream data is another area where pattern recognition has been successful.

Strengths

- Can increase the predictive power of classifiers substantially by finding valuable new patterns.
- Automated search capabilities inherent in most pattern recognition techniques can leverage analyst time and hasten the learning process for new data sources or classification problems.
- Wide field applicable to many problems across many different industries.

Weaknesses

- Patterns discovered might be spurious or not representative of future cases. Sample tuning can be an issue with some pattern recognition techniques.
- Definition of a "valuable" pattern might be unique to a particular problem. Borrowing pattern recognition techniques from a different problem without consideration can produce meaningless features and classifiers.

© 2012 Fair Isaac Corporation. All rights reserved.

References

Bezdak, James C. and Pal, Sankar K. (1992), *Fuzzy Models for Pattern Recognition*, IEEE Press, Piscataway, NJ.

Fayyad, Usama M., Piatetsky-Shapiro, Gregory, Smyth, Padraic, Uthurusamy, Ramasamy (1996), *Advances in Knowledge Discovery and Data Mining*.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, Great Britain.

Thearling, Kurt (October 1995), *From Data Mining to Database Marketing*, Data Intelligence Group (DIG) White Paper.

© 2012 Fair Isaac Corporation. All rights reserved.

» Recursive Scorecard Segmentation

Recursive Scorecard Segmentation is a FICO technology for developing an optimal segmentation to predict an outcome using a system of scorecards.

Why Segmentation?

Many powerful predictive modeling technologies, including FICO's Scorecard module, benefit from segmenting the population into more homogenous sub-populations prior to building models. This is done for a variety of reasons:

- **Interaction detection**: It can be difficult to capture interactions directly through easily interpretable variables; by segmenting the population wisely, it is possible to have the segmentation capture the interactions correctly.
- **Focus on key variables**: In some populations, certain variables become more relevant. For example, the segment of consumers with historical delinquency will benefit from having a large number of delinquency related variables come into the model, when these variables will be irrelevant for the population with no historical delinquency.
- **Palatability**: Related to the interactions, it is possible for variables to have complicated predictive patterns on an overall population, but straightforward patterns on subpopulations. This can lead to the ability to put variables with palatability issues into models where interpretability is required.

Qualities of a Good Segmentation

A good segmentation scheme needs to meet several criteria:

- The predictors or prediction pattern should be different on different sub-populations.
- The segmentation should produce an increase in the overall population predictive power.
- Each segment needs to be large enough to produce a statistically sound predictive model.
- Migration of accounts/cases between segments should not generate large or unexplainable jumps in the scores.

Challenges with Manual Segmentation

Traditionally, model segmentation has been a task performed by expert modelers testing alternative segmentation ideas. Examining any individual segmentation can be time consuming, since many candidate models need to be developed. Since each one is costly in time, this greatly limits the number of alternate segmentation schemes that can be tested. Typically modelers explore different segmentation until they run out of time, which leads at best to rough segmentation schemes that are unlikely to be optimal.

Qualities of a Good Automated Segmentation Algorithm

Determining a good segmentation system in an automated way can be challenging. The goals for a tool that automatically comes up with a segmentation scheme should include:

- Fast runtimes
- Accurate estimates of the ultimate value of the segmentation
- Ability to create a variety of alternate candidate segmentations, with significant differences between alternate candidates
- Ability to specify constraints in predictors for palatability
- Ability to impose initial splits or partial trees for business palatability
- Ability to produce starter models for each segment

© 2012 Fair Isaac Corporation. All rights reserved.

- Avoidance of overfitting during segmentation
- User-friendly reporting of segmentation schemes and models, helped by visualization

FICO has designed a recursive scorecard segmentation tool to meet all of these needs.

Algorithmic Approach

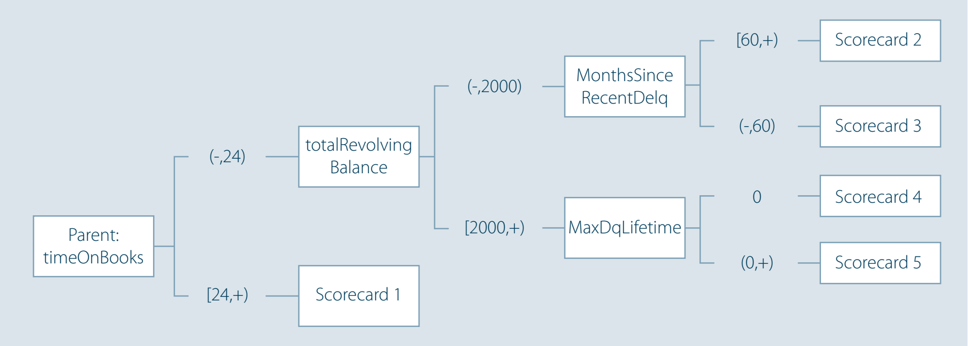
FICO has taken an approach similar to CART to build a segmentation tree by identifying the optimal split at each leaf, and then recursively splitting on each subleaf. In more detail:

The user (optionally) specifies an initial tree. The algorithm then:

- Recursively takes a leaf that has not yet been split
- Builds a model on the entire leaf, based on a list of candidate variables specified by the user
- Exhaustively examines each potential segmentation on that leaf from a user specified list of candidate segmentation variables and segmentation points:
  - Ensures there are robust enough counts on each side of the split to continue
  - Builds a model on each split population
  - Calculates the overall quality of the submodels at the parent leaf[32]
  - Identifies the best split (split with highest model quality)
  - If the split has significantly higher quality than the overall model, split on this variable
- Repeats on each leaf until
  - Each leaf is sufficiently small that it cannot be segmented any further and/or
  - No segmentation can be found on this leaf that increases model quality

This algorithm, similar to CART, is "greedy" in that it picks the best split at each leaf and moves on. It is possible that by using a sub-optimal split at a high level, a better overall tree could be found. However, academic research has shown that the greedy approach used in CART works well in practice, and FICO's research has also found that this approach produces very good trees in practice.

FIGURE 22: SEGMENTATION SCHEME EXAMPLE



[32] There are a variety of potential metrics that can be used to evaluate the quality of a model, such as K-S, GINI, ROC area, etc.

Note that this algorithm heavily relies upon the ability of FICO® Model Builder Scorecard module to automatically create strong models in each leaf. FICO has found that approaches which only approximate the value of each leaf frequently lead to poor segmentation trees.

FICO research has found that this algorithm produces superior segmentation to FICO's prior generation of automated segmentation search, and produces results in significantly less time, frequently forty to fifty times faster.

Applications

Segmentation has been successfully applied to broad-based bureau scores, master file-based scores, and application scores, among other applications. In all of these areas, we see:

- A large number of both good and bad exemplars, enabling robust models to be built on subpopulations
- Palatability constraints, where different characteristics may be more palatable in certain population segments
- Diverse populations, where different variables may be more predictive in different segments. For example, in broad-based bureau models, populations with small numbers of trade lines have very different predictive patterns than populations with many trade lines; populations with historical delinquencies have different key predictive variables than populations with no historical delinquency.

Applications where consumers are more homogenous or palatability is not required may see less benefit from segmentation.

Strengths

- Segmented scorecards combine utmost predictive power with utmost palatability of the score formula
- Develops fully engineered scorecards at each node, speeding the path to finalized, implementation-ready models
- Good match between estimated value of segmentation and observed value after user-created scorecards created
- Outperforms prior generations of FICO segmentation tools
- Tests many more segmentations in much less time than would be possible in a manual approach

Weaknesses

- Optimal segmentation is not guaranteed

» Regression

Regression is a family of prediction modeling techniques. When "regression" is mentioned, care must be taken to understand which technique is being discussed to avoid misunderstanding. The goal of regression, as in many competing techniques, is to model the relationship between *predictor variables* and the desired *outcome variables* so that in the future, when the outcome variable is unknown, it can be estimated or predicted.

The method of arriving at the mathematical formulation depends on the structural assumptions of the relationship between predictors and expected outcome, as well as distributional assumptions regarding the outcome variable. Particular examples include the *least squares* method for *continuous outcomes* and logistic regression for *binary outcomes*. Many of these methods are particular instances of the family of Generalized Linear Models (GLMs), which are fitted via the *Maximum Likelihood* principle.

The simplest regression would, for instance, establish a straight-line relationship between applicants' age and their income. Regression can go far beyond this scenario by incorporating a greater number of predictor variables. There are more advanced regression techniques for modeling relationships between the predictors and the outcome that are curvilinear (i.e., quadratic or higher order polynomial) or *non-linear*.

It is worth noting that many vendors of solutions will compare their results to a "traditional statistical approach." In many cases this can be interpreted as one of the regression techniques.

Applications

Regression is probably the most widely used technique for building models involving continuous outcome variables.

Strengths

- Easy to interpret.
- Widely used, well documented.
- Can be a mixed model of continuous and categorical predictor variables.
- Allows for a wide range of statistical diagnostics and significance tests.

Weaknesses

- Regression cannot elegantly handle missing values on a variable-by-variable basis. Data must be lost, or some assumption made about the missing data to give it a value.
- Score weight patterns for categorical data cannot be made palatable.
- The model assumes fixed increments/decrements in the score values for variables on an interval scale.
- May not capture, or at least make readily apparent, interactions in data.
- Categorical variables may have to be represented by dummy variables, i.e., multiple variables which represent the absence or presence of each component attribute in the predictor variable.

Multiple Linear Regression

This approach is often used for predicting a continuous outcome (income, revenue, amount of purchases) from several predictor variables. It has been used on occasion for predicting categorical outcomes, but in general, this is a flawed approach and within this family, logistic regression is the preferred solution. In its simple form, the estimation of the coefficients to be applied to each predictor variable is computed simultaneously, i.e., the developer determines ahead of time what are the candidate predictor variables for the model.

© 2012 Fair Isaac Corporation. All rights reserved.

Strengths

- Multiple regression is a technique with which most people are familiar. Most statistical packages also include various significance tests (e.g., tests involving certain slopes) and diagnostic tests (e.g., residual test for normality).
- Irrespective of the distribution of the dependent variable, the least squares regression estimators are the best linear unbiased estimators (i.e., have minimum variance among all linear unbiased estimators).

Weaknesses

- Not robust when outliers are present in the data.
- The assumption that the outcome variable is normally distributed for each fixed combination of the predictor variables is not necessary for the least-squares fitting of the regression model, but is required, in general, for inference-making purposes. In this regard, the usual parametric tests of hypotheses and confidence intervals used in a regression analysis are "robust" in the sense that only serious departures of the distribution of the dependent variable from normality can yield spurious results. This is classified as a weakness only because for typical data encountered at FICO, this assumption is violated. For outcomes that are, in fact, normally distributed with respect to the predictor variables, this would not be a weakness.

Stepwise (Multiple Linear) Regression

As an extension to the standard case, *predictor variables* are sequentially added to and/or deleted from the solution until there is no improvement to the model. The *forward selection* method starts with an empty model and at each step adds the variable that would maximize the fit. The ba*ckward elimination* method starts with a model containing all potential predictors and at each step removes the variable that contributes least to the fit. The *stepwise* elimination method develops a sequence of regression models, at each step adding and/or deleting a variable until the "best" subset of variables is identified. Note that the term "stepwise" is sometimes used vaguely to encompass forward, backward, stepwise, as well as other variations of the search procedure.

Strengths

- Can be used to automatically select a reasonably good subset of possible scorecard characteristics.
- Allows user to demonstrably "exhaust" the data. By running the various Selections (Forward, Backward, Stepwise) separately, one can also gain helpful insight into the effect of the correlation of the predictive variables on parameter estimates.
- Fast and convenient method to screen a large number of variables and simultaneously fit a number of regression equations. Stepwise regression is probably the most widely used automatic search procedure.

Weaknesses

- Sometimes arrives at an unreasonable "best" set when the predictor variables are highly correlated or include several variables that represent transformations of the same source variables.
- Small changes in the data can result in very different models.
- As with all automatic methods, the resulting models will be tuned to the development sample, unless "engineered" for future use.

Logistic Regression

The *dependent variable* in this case is *categorical* while the predictor variables can be *continuous* or categorical or both. This method is statistically appropriate for modeling of *binary outcomes*. The usual objective is to estimate the likelihood that an individual with a given set of variables will respond in one way, or belong to one group, and not the other.

Methodology for performing stepwise logistic regression is available. It follows the same principle as stepwise linear regression.

Applications

Mostly used for modeling of binary outcomes (e.g., good/bad, high-revenue/low-revenue, response/no-response). Some packages support multinomial dependent variable prediction.

Strengths

- Can have mixed continuous and categorical predictor variables.
- Results are already in probability or in log odds scale.

Weaknesses

As is the case in most regression procedures, logistic regression is also sensitive to large correlations between the predictor variables in the model.

For categorical predictor variables that are converted to dummy variables, there is typically no mechanism for constraining the model coefficients to take on a directional pattern.

MARS (Multivariate Adaptive Regression Splines[33])

The 1988 brain child of Jerome Friedman, MARS combines properties of regression and tree techniques. Like regression, MARS attempts to optimize a fit of a dependent variable using the least squares method. Unlike regression, MARS allows for the specification of more complex terms than linear and additive ones in the model. Like trees, MARS partitions data, but unlike trees, MARS allows for the capture of linear and additive relationships and for the splitting over all nodes at each step, rather than just the currently terminal ones. Either *categorical* or *continuous* outcomes can be modeled using categorical or continuous predictors with this technique. A very simple example is a model predicting annual income (I) using the categorical variables Education Level (E) and Region (R) and the continuous predictor Age (A):

$$I = 10.0 + 0.5 \cdot A + [-5.0 | R=\text{rural}] + [5.0 | R=\text{urban and } E=\text{H.S.}] + [10.0 | R=\text{urban and } E>\text{H.S.}]$$

In this example, a 30 year old rural resident with a high school education would have a predicted annual income of $20,000 (10+.5*30-5) a year, while a 50-year-old urbanite with a college degree would have a predicted annual income of $45,000 (10+.5*50+10). Note that the first two terms are applied to the entire population, while the last three terms are applied only to specific regions of the data. The relationship between Region and Education Level, where the weights differ depending on the level of the two variables, is an example of an interaction.

[33] Splines are curves which are required to be continuous and smooth. Splines are generally n-degree piecewise polynomials whose function values and first n-1 derivatives agree at the points where they join (the abscissa values of the join points are called "knots"). MARS replaces the step function used in trees with a truncated power spline basis function in order to produce a continuous model.

© 2012 Fair Isaac Corporation. All rights reserved.

The MARS algorithm uses a forward stepwise approach to add terms that minimize variance, until a user-specified maximum number of terms is reached. At this point, a backward stepwise deletion strategy is employed to select the combination of terms that provides the best fit on a hold out or *cross-validated* sample.

Applications

Used for modeling of binary or continuous outcomes and the detection of interactions. Recognized in academia but not well known in industry. An approach for detecting interactions, recommended by Friedman, is to run a comparison of two MARS models, one with strictly additive terms, and the other including both additive and interactive terms. If the interactive terms improve the fit considerably, they can be considered significant.

Strengths

- Models both additive and non-additive relationships.
- Handles both continuous and categorical predictor variables and outcomes.

Weaknesses

- As is the case in most regression procedures, MARS is sensitive to large correlations between the predictor variables in the model.
- Provides limited capabilities to engineer solutions, control missing values and inject domain expertise.
- Depending on the model form, MARS is likely to be significantly more computationally intensive than additive modeling techniques.

References

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons.

Chambers, J. M., Hastie, T. J. (1992), *Statistical Models in* S, Pacific Grove, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

Collett, D. (1991), *Modelling Binary Data*, London: Chapman & Hall.

Friedman, J. H. (1991), *Multivariate Adaptive Regression Splines*, The Annals of Statistics, Vol. 190, No. 1, 1-14.

Mosteller, F., Tukey, J. W. (1977), *Data Analysis and Regression*, Addison-Wesley Publishing Co.

» RFM (Recency, Frequency, Monetary Value)

This approach used by direct marketers generally takes one of two forms:

- As a two- or three-dimensional matrix.
- As the basis of a "score."

In either case, RFM is a simple to create and simple to use construct designed to assist in targeting marketing efforts or in segmenting communications.

Three of the most important types of predictors of response among existing customers are:

- Recency, defined as the time since the customer last made a purchase.
- Frequency, defined as the number of purchases by a customer (often over a defined time period, such as the past 12 months).
- Monetary value, defined as the dollar amount purchased by the customer (often over a defined time period).

Response rates tend to be highest among groups of customers with low values of Recency (i.e., recent purchases) and high values of Frequency and Monetary value.

Perhaps the simplest use of RFM is to *intervalize* these measures and to then measure past response rates in each cross-classified cell of the resulting three-dimensional matrix. Response rates will usually vary substantially across cells; ranking the cells by past response helps the marketer determine groups more likely to respond in the future.

Simple ranking has some serious drawbacks. Depending on the total number of customers being examined and the strength of association of the R, F and M predictors, some of the cells may have very small counts. Response rates based on small counts will not be good estimates of the true response rates for those cells. Consequently, a simple ranking by response rates may not be the best ranking. One way to improve the ranking is to impose rules that enforce a logical order. A simple scheme is to override response ranks for any pair of cells where two of the RFM measures have the same value and the cell with the better value of the third measure has a lower response rate. Using such a scheme results in cell rankings that have higher intuitive appeal.

An RFM scheme can be developed even before response data are available. In such cases, initial decisions are made based on a hypothetical relation between RFM and response. Over time, response can be added to the matrix to give feedback for new strategies.

Literature in the direct marketing realm reveals attempts to parameterize the RFM approach into some functional form[34]. This way, it may be usable in more than just a cell-by-cell basis.

[34] If historical outcome data exist in addition to just the predictive data, one could attempt to fit the formula version of RFM as some form of a non-linear regression. The functional form would reflect that the outcome is proportional to the "frequency" and "monetary value" components and inversely proportional to the "recency" components. For example, one such proposed functional form seen in the literature is:

$$\textit{prediction} = \left(\frac{(1 + F).\,(1{+}M)^{P}}{R^{q}} \right)$$

with $0 \le p \le 1$ and $0 \le q \le 1$.

Applications

RFM is widely used in marketing applications. Even though it is a rough "model," some assert that it may find small pockets of responsiveness that generalized models may miss.

Strengths

- Good results without gathering performance (response) data.
- Easily understood. Appeals to the basic understanding of what predicts response.
- Easy to develop and easy to implement.
- Easy segmentation tool that results in groups distinguished by factors highly relevant to marketers (i.e., response and usage).

Weaknesses

- In its simplest form, makes no attempt to discover underlying relationships.
- Focuses on three variables only, which is near the maximum a cell-based scheme can support, unless sample sizes are extremely large. Additional factors (such as tenure or category purchase breadth) result in a multiplication of the number of cells, inevitably leading to unacceptably small cell sizes.
- No attempt is made to model the relationship between offer and response, which is critical in the effort to understand customer behavior and optimize offers in one-to-one marketing strategies.

» Rules-based Systems

Rules-based systems trigger actions or reach decisions based on combinations of conditions and/or events encountered during production operations. Rules may be defined by programmers or by business-level policymakers to reflect regulatory requirements, domain-specific best practices, enterprise business policies or process-control tasks. Rules can also be derived from data-driven analysis techniques and translated to a declarative format for ease of implementation and management. Rules-based systems differ from the other techniques discussed in this document in that they do not necessarily include statistical methodologies as a core component. While rules-based systems may utilize the results of historical data analysis, they often incorporate human judgment and arbitrary statements of policy in their design.

Declarative Rules

Declarative rules are written in a format stating specific actions that should occur whenever a triggering condition is recognized. The rule declaration format may vary between different systems or different types of rules within an application. Examples of declarative statements include the following general cases:

- IF (conditions) THEN (actions)
- WHENEVER (event) OCCURS DO (actions)
- PERFORM (actions) WHEN (conditions)
- CASE:

```
                    (conditions) : (actions)
                    (conditions) : (actions)
```

No matter the format, the goal is to set out explicit statements of application logic that can be easily reviewed, understood and quickly changed to meet changing business conditions.

Graphical Representations

Groups of related rules can sometimes be represented through alternative, graphically-oriented mechanisms such as decision tables and decision trees.

Decision Tables

A decision table lists condition values in rows and columns. The intersection of the conditions for a particular case leads to a grid cell that specifies what actions should be taken (most often setting a single value). An example of this format is the familiar postage rate chart. Knowing the type of service, weight, and delivery area of an item leads to the proper postage for that item. Each of the prices in the table is actually a rule established by the post office that assigns a value according to federally-mandated policies. Another example may be a simple matrix used to determine initial credit limits for a student credit card product. Figure 23 displays a decision table relating credit limit actions to income and credit card type conditions.

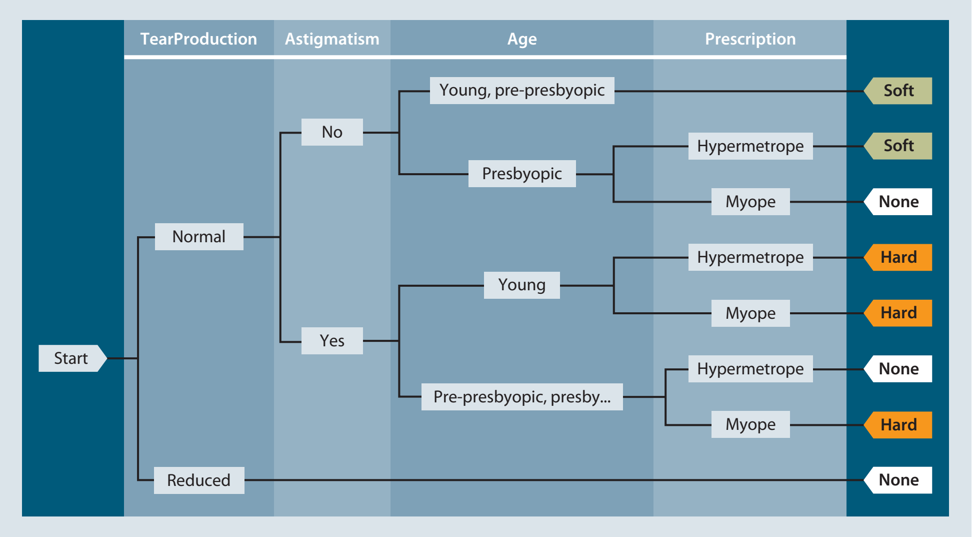
FIGURE 23. DECISION TABLES ALLOW DEFINITION, REVIEW AND UPDATE OF MANY RULES AT ONCE. EACH CELL IN THE TABLE SPECIFIES ACTIONS TO ASSIGN FOR A GIVEN COMBINATION OF CONDITIONS.

Card Type Condition	Student Bronze	Student Gold	Student Platinum
Income Condition	Credit Limit Action	Credit Limit Action	Credit Limit Action
7,500 - 9,999	1,000	1,500	2,000
10,000 - 19,999	1,100	1,600	2,100
20,000 - 29,999	1,200	1,700	2,200
30,000 - 39,999	1,500	2,200	2,700
40,000 - 49,999	2,000	2,500	3,000
50,000 - 59,999	2,500	2,800	3,300
60,000 - 69,999	3,500	3,800	4,000
70,000 - 79,999	4,000	4,500	4,800
80,000 - 89,999	4,500	4,700	5,200
90,000 - 99,999	5,000	5,200	5,700

Decision Trees

Decision trees allow applications to progress through a sequential series of tests, at each point taking one of a finite number of predefined branches leading either to another test condition or to a resultant action. Trees are useful for making sure that all possibilities are covered when testing against values that can have a limited number of potential values or ranges.

FIGURE 24: DECISION TREES ARE REPRESENTATIONS OF RULES THAT MUST BE CONSIDERED IN A SEQUENTIAL FASHION. EACH BRANCHING NODE INDICATES WHICH PATH SHOULD BE TAKEN FOR A CONDITION VALUE OR RANGE. THE END OF A BRANCHING PATH SPECIFIES THE APPROPRIATE ACTIONS.

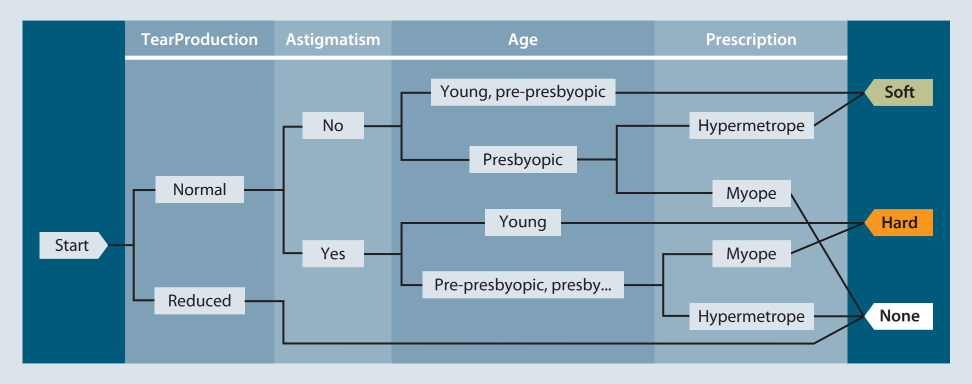


The 23 node decision tree shown below encodes the rules for prescribing contact lenses. The available actions are *soft*, *hard*, or *no* contact lenses. The sequential tests are made on *TearProduction*, *Astigmatism*, *Age*, and *Prescription*.

Directed Acyclic Graphs

Decision trees can be thought of as a kind of *Directed Acyclic Graph* constrained such that each node in the graph has one and only one predecessor. Unfortunately, this constraint prevents common decision logic from being reused or shared and, as such, very large trees can result. If we relax the constraint and allow nodes to have more than one parent, we can typically represent the same decision logic in a smaller structure.

FIGURE 25: BY ALLOWING NODES TO HAVE MORE THAN ONE PREDECESSOR, WE CAN REPRESENT THE DECISION LOGIC DESCRIBED BY THE TREE ABOVE 16 RATHER THAN 23 NODES.

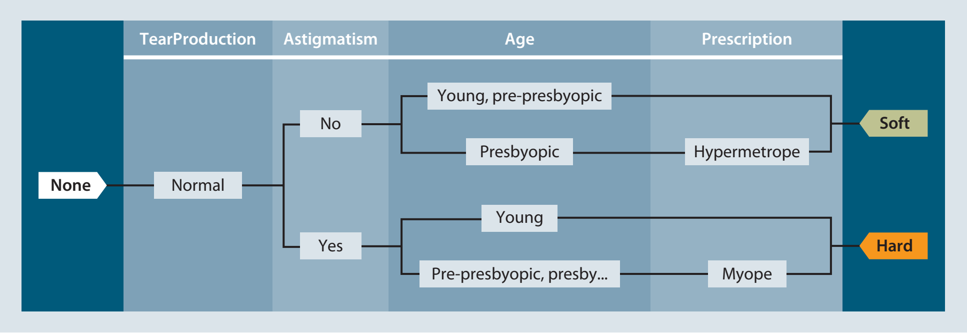


© 2012 Fair Isaac Corporation. All rights reserved.

Exception Graphs

Another kind of graph that can be used for representing decision logic is the *Exception Graph*. This is a DAG in which the node at the top (or left) of the structure represents an exception. In interpreting an Exception Graph, one would select the first (exception) action unless another path through the graph was possible, in which case the action at the end of the path would be selected.

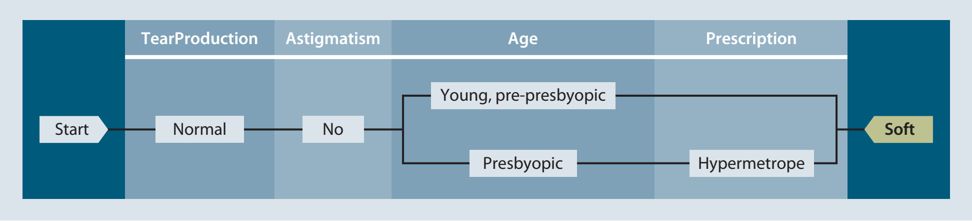
FIGURE 26: BY PROVIDING AN EXCEPTION, THE DECISION LOGIC CAN BE REPRESENTED BY 12 NODES.



Action Graphs

An *Action Graph* decomposes the overall decision logic into a collection of DAGs, each of which represents the decision logic for a single action. Often a very complex, difficult to understand decision tree can be decomposed into a number of small, easily interpretable action graphs. The figure below shows the decision logic lead to the "Prescribe Soft Contact Lenses" action.

FIGURE 27: THE ACTION GRAPH FOR "PRESCRIBE SOFT CONTACT LENSES" CONTAINS ONLY 7 NODES.



Development, Deployment and Maintenance

Modern rule systems are comprised of several components covering the stages of development, deployment and maintenance. The rule development environment contains all the facilities for authoring the rules; viewing them in relationship to each other; testing for functionality, performance and conflicts; and documenting the rule logic. Rule deployment typically involves the use of a rule engine that intelligently selects which rules need to be considered and executed based on transactional data and decisions that have been made by the engine to that point. The operations handled by the rule engine eliminate the difficult programming needed to ensure that system decisions can be reconsidered and executed at any time during processing. Ongoing maintenance is a key consideration in choosing to call on rules for application decision processing. Decision logic needs to be updated as corporate policies change, new business conditions are encountered, external regulations are introduced and so on. Because rules are maintained separately from the rest of the application code in their own repository, the rules that relate to a particular decision process can quickly be found, understood and altered without affecting the main program code. It is easy to think of this sort of specialized storage and maintenance in the same way as databases eliminate the problems of coding data values directly into program code.

Incorporating Analytics

Business rule systems are adept at handling the functions of authoring, executing and updating application logic. However, they are typically not designed to assist in determining *what* rules should be implemented by a company. Data-driven analytics and strategy optimization are often employed for this purpose. Analytic techniques can select data items to use as decision making criteria and

© 2012 Fair Isaac Corporation. All rights reserved.

recommend thresholds and values that should be used in the rule tests. Predictive models can create mathematically derived functions that are called upon by the rule logic to implement different actions based on expected performance. As different strategies are considered, simulation and optimization software can suggest effective tests and actions that the company can choose to implement in its rule system.

Rules-based systems are effective at carrying out explicit decision processes that look at transactional data and take one or more actions based upon established guidelines. They make it easy to formalize expert judgments and apply them consistently to different situations. Because rules software is efficient at storing, executing and managing large numbers of decision points it is often used as the execution and implementation vehicle for data-derived strategies and models based on analytic methods.

Case-based Reasoning

Case-based reasoning is an alternative, yet related, approach to rules-based systems. In case-based reasoning systems, the knowledge base is not stored as a set of IF-THEN rules but as a set of historical cases. Case-based reasoning software allows the user to efficiently search this database for those cases most like the one being processed. For example, if the decision is whether to accept or reject a loan application, the case-based reasoning system would contain a database of past loan applications. After a new application is entered, the database would be searched for the closest matches. The matches would be tagged with the past accept/reject decision, the past actual outcome on booked accounts or both. Based on the outcome and management parameters set for approval, a decision would be made or the application would be referred to a human reviewer.

Applications

Successful applications of rules-based systems, such as the FICO® Blaze Advisor® business rules management system, can be found in the areas of insurance underwriting, claims management, personal and commercial lending, regulatory compliance, marketing, product promotion and selection, customer account management, benefits eligibility, healthcare diagnosis and patient administration, network management and manufacturing configuration processes.

Strengths

- Rules can encapsulate core pieces of personal or corporate knowledge and expertise so that it is not lost when an individual leaves the business.
- Decision strategies can be defined in the absence of historical data to analyze.
- Centralized rules can enforce management control and a way of exerting consistency in the decision making process.
- Declarative rules are often easier for business policy managers to understand and manage than statistical models.

Weaknesses

- The knowledge extraction process can be difficult and time consuming.
- Inconsistent or conflicting rules may be defined, leading to unpredictable or undesirable behavior of the decision model.
- Human judgment lends itself to non-formal decision processes that are unsubstantiated by experiential data. Rules allow policies to be established that conflict with statistical results.

References

Date, C. J. (2000), What Not How: *The Business Rules Approach to Application Development*, Addison-Wesley.

Ross, Ronald G. (1998), *Business Rule Concepts: The New Mechanics of Business Information Systems*, Business Rule Solutions, Inc.

Ross, Ronald G. (2003), *Principles of the Business Rule Approach*, Addison-Wesley/Pearson Education.

Von Halle, Barbara (2002), *Business Rules Applied: Building Better Systems Using the Business Rules Approach*, John Wiley & Sons.

Zahedi, F. (1991), *An introduction to neural networks and a comparison with artificial intelligence and expert systems*, Interfaces, Vol. 21, No. 2 (March-April), pp. 25-38.

© 2012 Fair Isaac Corporation. All rights reserved.

» Scorecard

Scorecards are FICO's proprietary predictive modeling technique. Scorecards are unique in their ability to account for business rules, legal and operational constraints, and biased or *missing data*, thereby generating scores that are not only highly predictive, but are also very interpretable and palatable to the business user.

Scorecards achieve these benefits through their flexible score formula that is based on Generalized Additive Models and augmentable with crosses between variables, and by employing non-linear constrained programming techniques to optimize score weights. The technology handles multiple objectives and constraints. These features have not been (and in some cases cannot be) implemented in other modeling techniques.

Applications

Scorecards have been successfully applied to all fields of predictive modeling served by FICO. Examples of applications include web log transaction, fraud detection, credit customer retention, insurance dollar loss reduction, direct mailing response maximization and mortgage risk assessment. This modeling technology is commercially available in the Scorecard Module of FICO® Model Builder.

Strengths

- Contains training algorithms that make no data structural assumptions other than that certain conditional score distributions are approximately normal.
- Contains training algorithm for Maximum Likelihood Estimation (for producing Engineered Logistic Regression scorecards) and for boosting classification performance of a score.
- Contains training algorithm that allows for two dependent variables, which enables, for example, developing a risk-adjusted response score.
- Supports score development for binary as well as ordinal/continuous dependent variables.
- Allows for constraints to be applied to score coefficient relationships (also called score weights restrictions or "score engineering") which provide for model stability and palatability.
- Provides properties of ridge regression via a weights penalty parameter, which can improve the performance on new data, especially for small samples.
- Offers Bootstrap unbiased estimation of validation performance, which is useful if development sample is too small to set aside a representative test sample.
- Allows for "Bagging" (Bootstrap Aggregation) of scorecards, a process in which many bootstrap replicas of a scorecard are fitted and then averaged.
- Allow problem variables to be de-emphasized by range restriction for a given set of variables while trading off overall model strength.
- Handles mixed variables (i.e., which take on continuous and categorical values).
- Handles missing values (i.e., no information) without having to drop the observation. Missing values are forced to take on the neutral predictive score weight.
- Provides automated and/or interactive fine and coarse binning of predictors, as required for developing Generalized Additive Models.
- Supports specification, engineering, and fitting of crossed binned predictors, which allows for interpretable interaction capture.
- Allows for automated step-wise variable selection based on predictive power and business rules.
- Allows for calculation of marginal weights for candidate variables not in the scorecard, providing insight into sources of potential model weakness.
- Accommodates alternative performance inference capabilities.

© 2012 Fair Isaac Corporation. All rights reserved.

- Provides log odds to score fitting and score scaling options.
- Score being approximately linear in log(odds) allows for multiple models to be aligned across segments of the population such that a score from any one of several models has the same interpretation.

Weaknesses

To capture interactions between raw input variables, features capturing these interactions must be constructed, and then included in the model as additive terms. Scorecard segmentation is an alternative. FICO provides other technologies (e.g., Data Spiders and Adaptive Random Trees) to automate the discovery of such features and segmentations.

» Sequential Decisions and Bayesian Learning

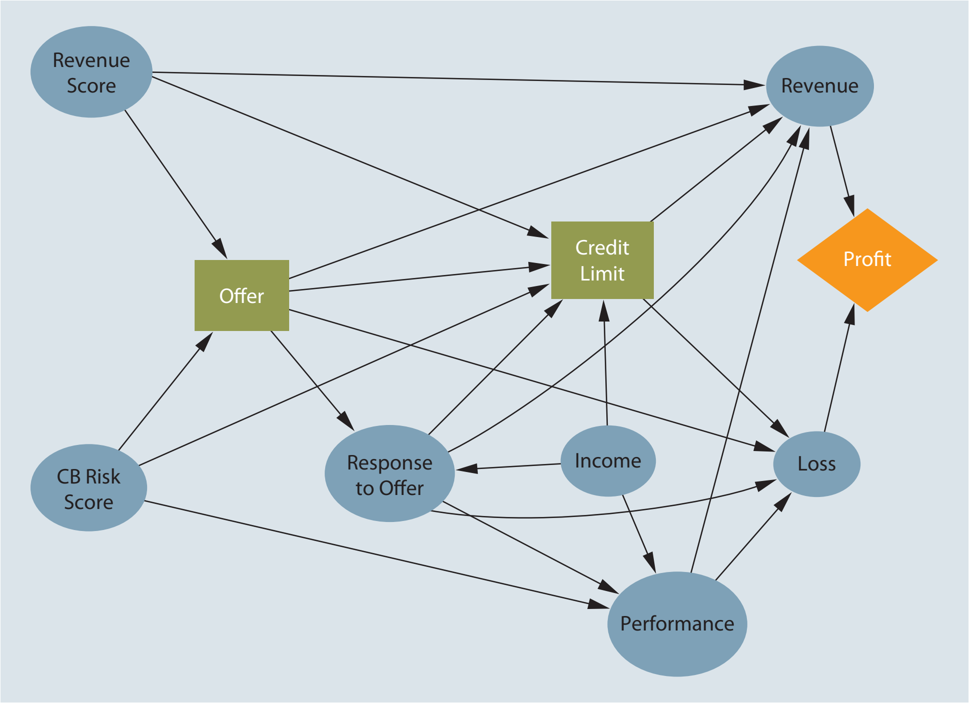
Most decision situations involve two or more decisions that need to be made at different points in time, which are closely related in affecting uncertain events, each other, consequences, and the decision maker's objectives. Sequential decision models are important because when choosing an alternative today, the decision maker has to know what additional alternatives he or she will have available to choose from in the future, as well as what type of information will be observable then that is still uncertain today.

In a customer-level relationship management environment, for example, the financial institution periodically makes proactive offers of new or improved products to customers. It also responds to customer requests for new products, such as a consumer loan, or an improvement to a current one, such as an increase in credit limit for an existing credit card. If the account manager only considers the present decision, using as information only the history and current status of the account, the choice may be far from optimal. This is because future possible decisions, such as subsequent credit limit increases or decreases, other loans, changes in pricing, etc., are not part of the model: the model is oblivious to anything beyond the time horizon of the present decision. The optimal resulting choice for the current decision may be quite different, and the long-run objectives significantly improved, if the manager considers today all, or at least most, of the future opportunities for offers or responses to requests. Such a consideration can only be done explicitly and quantitatively using a sequential decision model.

Sequential models are best depicted graphically using influence diagrams and decision trees. Refer also to the section "Graphical Decision Models" for a discussion of these paradigms. Figure 28 illustrates an influence diagram of a two-stage credit card campaign decision problem. Before making the Offer decision, the only information available to the decision maker consists of the Risk Score and the Revenue Score of the candidate[35]. At the Credit Limit decision, the decision maker has observed, in addition to the two scores, whether the customer responded to the offer, and, if so, the Income on his or her credit application. This model will provide much better strategies than two single-stage models considering the two decisions separately.

[35] Alternatives at the Offer node may be some combination of promotional rate and APR.

FIGURE 28: AN INFLUENCE DIAGRAM FOR A TWO-STAGE CREDIT CARD CAMPAIGN DECISION PROBLEM



Active Data Collection

Good sequential decision making is closely related to active data collection[36]. Some decisions in a sequential problem, particularly in the earlier stages, will typically be about, or related to, acquiring additional information (in the form of a survey, test, sampling, or other experiment) about uncertainties pertaining to the problem. In the customer-level relationship management example, the first offer (credit limit increase, for instance) should be optimized to address not only the decision maker's tradeoffs between the risk of default and the prospects of revenue, but also as an active collection of data designed to test the customer's use of the new limits.

The results of these sequential experiments should be used for continued optimal learning about the customer's behavior and for making future decisions to optimize long term objectives. The learning can be done by Bayesian updating of probability distributions.

Bayesian Learning

Bayesian inference is the fundamental learning mechanism in decision making, the only coherent way to update a decision maker's beliefs about uncertain events, based on newly available data or observations.

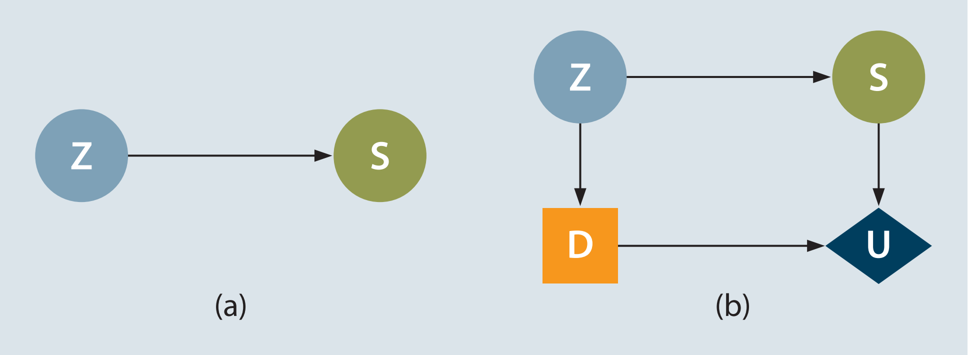
[36] Traditionally known, in a somewhat more restricted sense, as experimental design. Refer to the section on "Experimental Design" for further discussion.

© 2012 Fair Isaac Corporation. All rights reserved.

Even in simple, single stage decision models, Bayes rule is often useful. For example, when a risk scorecard is developed, we typically have available some prior ("population") probability of the risk performance, *p(Z)*, and the conditional score distributions, *f(s | Z)*, given "Good" or "Bad." This available information and the relationship between the performance and the score are shown in the influence diagram in Figure 29(a) by the arc from Z to s.

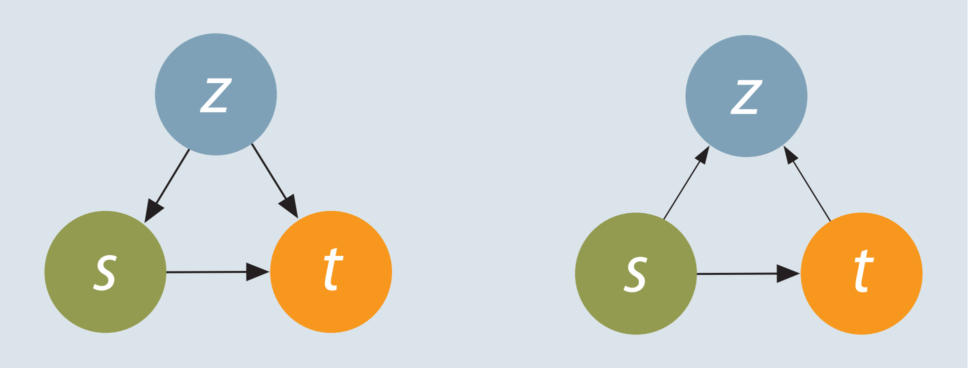
In the basic accept-reject decision, depicted in the influence diagram of Figure 29(b), however, we need to assess the posterior probability of Z given the score of an applicant, s. To go from (a) to (b), we need to use Bayes rule, $p(Z \mid s) \propto p(Z)\, f(s \mid Z)$, which is graphically represented by the reversal of the arc between the Z and the s nodes in the influence diagram.

FIGURE 29: BAYES RULE APPLIED TO A CREDIT GRANTING DECISION



If we had a second score available for the same individual, say t, we could use Bayesian revision to "combine" the information embedded in both scores so that we can improve the prediction of Z—and thus the quality of the decision. The process in this case is illustrated by the transformation of the influence diagram in Figure 30(a), showing the development of the scores, to the one in Figure 30(b), showing how they are combined to predict Z.

FIGURE 30: BAYESIAN REVISION USED TO IMPROVE THE PREDICTION OF Z



© 2012 Fair Isaac Corporation. All rights reserved.

Influence diagrams with only chance nodes, also called Bayesian networks, allow, through formal mathematical interpretation of the structure of their graphs and transformation algorithms, powerful probabilistic inference in much larger models then illustrated above. Some algorithms are designed to discover the conditional independence and other structural details of an underlying model from large sets of empirical data on the component variables. These algorithms—some of which are used at FICO—enable the simplification of models of large problems that would otherwise be intractable.

References

Chang, K. C., Fung, R., Lucas, A., Oliver, R. M., Shikaloff, N. (2000), "*Bayesian Networks Applied to Credit Scoring*," IMA Journal of Mathematics in Business and Industry 11.

Clemen, R. T. (1996), *Making Hard Decisions: An Introduction to Decision Analysis*, Duxbury Press, 2nd ed.

DeGroot, M. H. (1970), *Optimal Statistical Decisions*, McGraw-Hill.

Lindley, D. V. (1985), *Making Decisions*, Wiley.

Smith, J. Q. (1988), *Decision Analysis: a Bayesian Approach*, Chapman & Hall.

Winkler, R. L. (1972), *Introduction to Bayesian Inference and Decision*, Holt.

© 2012 Fair Isaac Corporation. All rights reserved.

» Support Vector Machines

Support vector machine (SVM) is a new technology for solving the *classification* problem. The technology has also been adapted to solve the non-parametric *regression* problem, but that will not be discussed here.

The term "machine" is used because SVM technology arose in the machine learning community, but the term machine is synonymous with score. The term "support vector" comes from the version of the problem where perfect discrimination between *binary outcomes* is possible. In that version of the problem, you find the high dimensional road of maximum width that separates the *binary outcomes* in input space. The input vectors that lie on the edges of the road are called the support vectors. They clearly play an important role in the theory.

A modeling technology can be defined by three elements. The first element is the *score formula*. The second element is the *objective function*, which needs to take into account both *training samples* and test samples to optimize the classification of new data. The third element is the optimization algorithm for finding the "fitted" parameters, which optimize the training sample objective function. SVM technology is described below via these three elements.

SVM Score Formula

The key construct in the SVM score formula is the *feature*. The SVM score formula is any linear combination of the features selected for the classification problem. In the standard theory, the features are linearly independent. However, *multicollinearity* can be handled with simple remedies.

There are two approaches to generating features. The first is based on domain knowledge, where a library of useful features develops over time. This approach is used for many domains of application from scorecards to speech recognition.

A second approach is more esoteric and implicit. Consider the case where the features are the same as the predictor variables. Then the score is a linear combination of the predictors—just like classical regression.

In this case, you can show that the optimal score coefficient vector is a linear combination of the input vectors in the sample. The coefficients of this linear combination form a dual parameter vector, whose dimension is the sample size. There is a dual formulation of the score development problem, which is an optimization conducted in this dual parameter space. The solution to this dual problem involves the inner products between all pairs of sample input vectors and the inner products between the sample input vectors and the vector associated with the account to be scored.

This unusual way of looking at the classical problem motivates the implicit approach to features. The concept of an inner product between two input vectors can be generalized using non-linear kernel functions. An example of a non-linear kernel function is the square of the quantity: usual inner product plus a constant. These non-linear inner products are easy to compute. If you just replace the inner products in the classical solution with the non-linear inner products, you get the optimal linear combination of certain non-linear features, which are uniquely associated with the non-linear kernel. These non-linear features never have to be computed. Only the inner products have to be computed, which involve only the kernel. The features exist implicitly. For the example kernel mentioned above, the implicit features are a large set of quadratic polynomials in the original input variables.

SVM Objective Functions

The objective function for the test sample is the *misclassification* cost. This can be generalized to the case of different costs for the two types of misclassifications, but most of the SVM literature is based on equal costs.

It is a very hard problem to find a linear combination of the selected features, which minimizes misclassification cost on the test sample. The SVM approach to this problem is to derive a very sophisticated training sample objective function, so that the resulting score validates very well with respect to misclassification cost on the test sample.

The SVM development objective function, which is to be minimized, has two components. The first component is the average of the squares of the score coefficients. This component is motivated by the case where the training sample of *binary outcomes* can be perfectly separated by a linear combination of features. Among all linear combinations, which accomplish the perfect separation, the best one is the one with minimum average of the squares of the score coefficients. Best is defined in terms of minimum test sample misclassification cost.

The second component is quite esoteric. It is based on the concept of a margin slack variable (MSV). MSV is a random variable defined on the training sample. It is defined in terms of the score, S, the score cutoff, c, and a cushion, u. Consider a "Good" observation. If the score exceeds the score cutoff by more than the cushion, then $MSV = 0$. Otherwise, $MSV = (c + u) - S$. Consider a "Bad" observation. If the score is less than the score cutoff by more than the cushion, then $MSV = 0$. Otherwise, $MSV = S - (c - u)$. Clearly, $MSV \geq 0$ and it is a measure of separation between the "Goods" and the "Bads." The smaller MSV is the better the separation. The second component of the SVM objective function is the expected value of MSV.

This second component of the objective function has to be expressed in terms of the score coefficients. There are two ways to do this—parametric and *non-parametric*. In the parametric approach one assumes that the score distribution can be expressed in terms of a few score moments. If the score distributions of the *binary outcomes* are both normally distributed, then the expected value of MSV can be expressed in terms of the means and variances of these distributions. These means and variances can be expressed in terms of the score coefficients. If the score distributions are Gamma distributed, then three moments are needed.

The non-parametric approach is more intricate. A very large quadratic program is formulated. The decision variables are the score coefficients, the score cutoff, and the margin slack variables for each observation. So there are more decision variables than there are observations. Let S_k and MSV_k denote the score and the margin slack variable for observation k. The second component of the objective function now becomes the arithmetic average of the margin slack variables over the observations.

The very large quadratic program also has many inequality constraints. The first set of constraints is that the margin slack variables are all non-negative. The second set of constraints involves the relationship between the score and the margin slack variables. For "Goods," the constraints are of the form $S_k \geq c + u - MSV_k$. For "Bads," the constraints are of the form $S_k \leq c + u - MSV_k$. These constraints are tight only when $MSV_k > 0$. The input vectors for the observations with $MSV_k > 0$ can be thought of loosely as support vectors, because these observations could not be classified correctly with a nice cushion. Of course, the score for observation k can be written as a linear combination of the score coefficients, so all the constraints remain linear in the decision variables.

© 2012 Fair Isaac Corporation. All rights reserved.

The two components of the objective function are combined by multiplying the first component by a tuning parameter, λ, and then adding the two components together. The tuning parameter, λ, determines the relative importance of the two components. In other technologies the first component has been called the penalty or regularization term.

For any particular score development, the choice of λ and u can be made to minimize the test sample misclassification cost. Of course, this will slightly bias the validation. An additional test sample is needed to get a purely unbiased estimate of misclassification cost.

SVM Optimization Algorithms

Once the SVM problem has been set up as a mathematical programming problem, standard mathematical programming algorithms can be used to solve it. In the non-parametric case, the problem is a very large quadratic program. Standard quadratic programming algorithms can be slow, so researchers have developed clever decompositions of the problem to speed things up.

Applications

Applications of SVM have begun to crop up in a variety of fields. The literature has mentioned applications to speech recognition, chemical classification and protein classification. In general, classification is a widely applied field, so the opportunities for this new technology are plentiful.

Strengths

- Score formula can be either simple or complex.
- Capture non-linear, non-additive relationships in data.
- No data structure assumptions in the non-parametric case.
- Can have both continuous and categorical predictors.
- Competitive with the state of the art for misclassification cost.
- Quadratic programming formulation allows the inclusion of score engineering.

Weaknesses

- Difficult to interpret unless the features are interpretable.
- Computationally intensive under the non-parametric case.
- Standard formulations do not include specification of business constraints.

References

Vapnik, Vladimir (December 1, 1997), *Support Vector Learning Machines*, NIPS tutorial notes.

Vapnik, Vladimir (1995), *The Nature of Statistical Learning Theory*, Springer.

Christianini, Nello and Shawe-Taylor, John (2000), *An Introduction to Support Vector Learning Machines and other kernel-based learning methods*, Cambridge University Press.

» Survival Analysis

Survival analysis is a collection of statistical methods where the outcome variable is the time until an event occurs. The event of interest may be death or relapse from remission in epidemiology, structural failure of a component in a manufacturing reliability test, attrition of a bankcard holder or mortgage prepayment. Time can be days, months or years from the start of the observation period until the event of interest occurs.

Basic Parameters of Interest in Survival Analysis

Four basic parameters of interest in survival analysis include:

1. The hazard function, *h*(t), specifies the probability that the event of interest occurs at time *t*, given that it has not occurred earlier. The failure event of interest can be death, structural failure of a component, customer attrition, mortgage prepayment, etc.
2. Related to the hazard function is the survival function, *S*(t), which gives the probability that the event of interest does *not* occur at time *t* or earlier, i.e., that the event has survived longer than time *t*.
3. The probability density/probability mass function, *f*(t), provides the unconditional probability of the event occurring at time *t*.
4. The mean residual life, *mrl*(t), measures an observation's expected remaining lifetime at time *t*. That is, given that the event has yet to occur at time *t*, the mean residual life provides an estimate of an observation's remaining lifetime.

Censored and Truncated Data

Most survival analysis applications involve censored data where the observation period of interest is incomplete for some records. Records can be right-censored or left-censored. A record is said to be right-censored when the observation period ends before the event of interest is observed, for example, subjects remain alive at the end of the study, or consumers continue to pay as agreed throughout the observation period. Left censoring arises when the event of interest has already occurred at the start of the observation period, for example, subjects have already relapsed at the beginning of a medical study but their exact time of relapse is unknown.

Truncated data occurs when the development data is sampled such that only records where the event of interest occurs either (i) after the start of the performance period (left truncation) or (ii) before the end of the performance period (right truncation) are included in the model development.

Unlike traditional *regression* methodologies, survival analysis statistical techniques feature mechanisms to handle censored and truncated data by incorporating the censoring and truncation information in the construction of the likelihood function. An observation with an exact event time provides information on the probability that the event occurs at time *t*, which is the probability density function *f*(t) for continuous timelines. For a right-censored observation, the event of interest has not occurred and thus the probability of it occurring is $P(X > C_r) = S(C_r)$, where *X* is the event time and C_r is the right censoring time. For a left-censored observation, the exact event time is unknown and all we know is that the event of interest has already occurred at the beginning of the observation period. The probability of a left-censored observation occurring is then $P(X < C_l) = 1 - S\ (C_l)$ for continuous timelines, where *X* is again the event time and C_l is the start of the observation period.

The likelihood function that accounts for censored data, assuming independent censoring times, is just the product of the exact timelines, right-censored, and left-censored components. The likelihood function for truncated data is similarly constructed, with an adjustment made to account for the fact that only records with event times (i) after a particular time point (left truncation) or (ii) before a particular time point (right truncation) are sampled. For censored non-truncated data, the likelihood function is defined as:

$$L \propto \prod_{i \in E} f(x_i) \prod_{i \in R} S(C_r) \prod_{i \in L} \left[1 - S(C_l) \right]$$

and for left-truncated data:

$$L \propto \prod_{i \in E} \frac{f(x_i)}{S(T_l)} \prod_{i \in R} \frac{S(C_r)}{S(T_l)}$$

and for right-truncated data:

$$L \propto \prod_{i \in E} \frac{f(x_i)}{1 - S(T_r)} \prod_{i \in L} \frac{S(C_l)}{1 - S(T_r)}$$

where E is the set of exact timeline observations, L is the set of left-censored observations, R is the set of right-censored observations, C_l and C_r are the left and right censoring times, respectively, and T_l and T_r are the left and right truncation times, respectively.

Semiparametric Proportional Hazards Regression Model

A popular semiparametric regression survival analysis technique for modeling time-until-event outcome variable is the proportional hazards model that was first proposed by D.R. Cox. The proportional hazards model can estimate time-until-event based on an individual's profile. For instance, a proportional hazards model may be built to estimate a consumer's time until mortgage prepayment, based on the individual's credit and demographic profile. This might include his/her risk score, loan-to-value (LTV) and income.

For an observation with profile $\vec{x}$, the hazard function for the proportional hazards model is defined as:

$$h(t, \vec{x}, \vec{\beta}) = h_o(t) \exp(\vec{x}^{\mathsf{T}} \vec{\beta})$$

The hazard function $h(t, \vec{x}, \vec{\beta})$ above expresses the hazard rate as a multiplicative relationship between the arbitrary baseline hazard function $h_o(t)$ and the exponential function of the covariates $\vec{x}$ multiplied by a set of regression coefficients $\vec{\beta}$).

The relative risk, also known as the hazard ratio, of an individual with a risk profile $\vec{x}$ versus one with a risk profile $\vec{x}^*$ is defined as $h(t, \vec{x}, \vec{\beta}) \,/\, h(t, \vec{x}^*, \vec{\beta}) = \exp(\vec{x}' \vec{\beta}) \,/\, \exp(\vec{x}^{*\prime} \vec{\beta})$. The proportional hazards model is so called for its hazard ratio being constant and thus proportional over time. The proportional hazards model is classified as a semiparametric model because the covariate effect is parameterized while the baseline hazard rate is left non-parameterized. Parameter estimates for the Cox proportional hazards model are obtained by maximizing its partial likelihood function—a function that does not require explicit specification of the baseline hazard function.

The main advantage with the proportional hazards model is that the analyst does not need to specify the distributional form of the survival function in fitting the model. In general, the proportional hazards model can provide reasonable estimates of time until the event of interest occurring when the basic shape of the underlying survival curve is not well understood.

© 2012 Fair Isaac Corporation. All rights reserved.

Discrete-Time Hazard Model and Time-to-Event Scorecard

For certain applications, it is not necessary to model time as a continuum, but time can be discretized into finite intervals, such as days, weeks, or months, without loosing significant information. For example, to calculate the net present value of a journal subscription, it may be reasonable to estimate monthly or quarterly attrition hazards. Or to predict the purchase of a slow-moving retail product, a weekly discretization may be adequate. In these cases, a discrete-time hazard model can be used.

A particular powerful and interpretable discrete-time hazard model can be developed using FICO's Time-to-Event (TTE) scorecard technology, which can generate very flexible while at the same time palatable survival models.

The TTE Scorecard is based on the method of maximum likelihood estimation. It seeks score weight estimates that maximize the likelihood of observing a given longitudinal data sample. The predictions are concerned with binary target events happening at discrete time intervals. The fundamental quantity modeled is the risk of event occurrence in each time period. The discrete-time hazard:

$$h(t_{ik}) = \Pr\{T_i = k \mid T_i \geq k\} \tag{1}$$

is the conditional probability that individual *i* will experience the event in period *k*, given that it did not experience the event earlier. To model a heterogeneous and time-dependent population where individuals can be distinguished from each other on the basis of predictors, we write:

$$h(t_{ik}) = \Pr\{T_i = k \mid T_i > k, X_{ik}\} \tag{2}$$

where *X* is a vector of possibly time-variant predictors, which can include both customer-level variables, such as time since subscription, or gender, as well as global variables, such as season of the year or advertisement spend. In retail marketing applications, where a task is to predict future purchase hazard for a target product, some of the *X* may be recencies and frequencies of related product purchases.

To specify the model, we express the hazard as a flexible regression function, which is a scorecard. The score is related to the hazard by the logit link function:

$$\text{logit}\, h(t_{ik}) = \log\left(\frac{h(t_{ik})}{1 - h(t_{ik})}\right) = S_{ik} = \sum\nolimits_j ß_j x_j \tag{3}$$

The coefficient $ß_j$ is an intercept term and $x_1 \equiv 1$. The $ß_j$; $j \geq 2$ are score weights. The score is an additive (in general, non-linear) function in the numeric parts of *X*. This is achieved by binning the raw numeric features and transforming them to indicator variables for the bins $x_j \in \{0,1\}$; $j \geq 2$. Score formula (3) achieves a fairly high level of flexibility for modeling hazard. Applying the inverse logit transformation, we obtain:

$$h(t_{ik}) = \frac{1}{1 + \exp(-S_{ik})} \tag{4}$$

Note that the predicted hazard can be time-variant, by virtue of including time-variant predictors into a scorecard.

To fit the model to data, we seek estimates of the ß's that maximize the likelihood of observing the sample data. It can be shown that with a suitable transformation of the data into a so-called "person-period format", logistic regression estimation (or equivalent, the Bernoulli Likelihood function in Scorecard) can be used to solve for the maximum likelihood estimates. This makes survival modeling accessible to users of logistic regression or scorecard.

© 2012 Fair Isaac Corporation. All rights reserved.

Parametric Regression Models

Parametric regression models for modeling time-until-event outcome variable include survival analysis methods where both the survival function and covariate effect are parameterized. When the survival function is correctly chosen, parametric regression survival analysis techniques may provide a better estimate of time until the event of interest occurs. The semiparametric proportional hazards model, however, can lead to better estimates than with an incorrectly chosen parametric regression model. In applications where the basic shape of the survival curve is understood, parametric regression models are preferred, as the survival distributional form is explicitly incorporated into the model.

Commonly selected distributions to model the parametric survival function include the exponential, Weibull, log logistic and log normal distributions. The exponential regression model assumes constant hazard rate over time for a given individual with profile $\vec{X}^*$. The *Weibull distribution* has a hazard rate that can be monotonic increasing ($\alpha > 1$), monotonic decreasing ($\alpha < 1$), or constant ($\alpha = 1$). Both the log logistic and log normal distributions feature a hazard rate that increases initially and then decreases.

Applications

Survival analysis statistical procedures are widely used in analyzing data from medical studies and manufacturing quality control tests. In the credit industry, survival analysis can be used to predict time until a bankcard holder attrites or time until a mortgage loan is prepaid. Instead of modeling whether the bankcard holder will attrite or whether the borrower will prepay his/her mortgage in a predefined observation window, survival analysis attempts to predict *when* the attrition or prepayment may occur.

Strengths

- Predicts the likelihood of an event occurring over time.
- Can be used to estimate *when* the event of interest occurs.
- Features mechanisms to handle censored data.

Weaknesses

- Might be less predictive compared to binary outcome models in predicting the likelihood of an event occurring over a predefined period.
- Can be cumbersome to calculate the hazard rate over time for every individual.

References

Klein, John P. and Moeschberger, Melvin L. (1997), *Survival Analysis: Techniques for Censored and Truncated Data*, New York: Springer-Verlag, Inc.

Hosmer, Davis W. Jr. and Lemeshow, Stanley (1999), *Applied Survival Analysis: Regression Modeling of Time to Event*, New York: John Wiley & Sons, Inc.

Kleinbaum, David G. (1996), *Survival Analysis: a Self-Learning Text*, New York: Springer-Verlag, Inc.

Singer, J. D., Willett, J. B. (1993). *It's about Time: Using Discrete-Time Survival Analysis to Study Duration and the Timing of Events.*

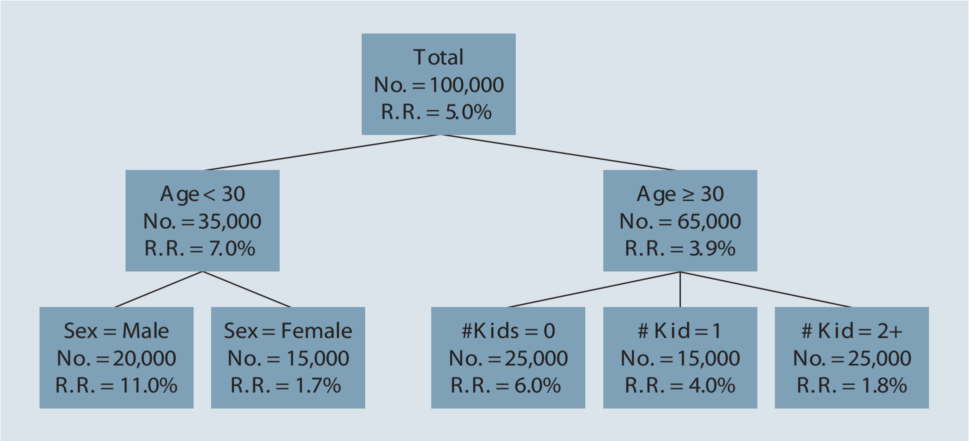
© 2012 Fair Isaac Corporation. All rights reserved.

» Tree Modeling Methods

Tree modeling technology includes *classification* and *regression* techniques, which rely on successive partitions of the data to predict the desired *outcome*. The resulting tree contains nodes or subsets in which observations within a node are similar based on the specified measure, while sample points in different nodes are dissimilar with respect to the same measure. These techniques are not to be confused with decision trees[37] or the implementation of simple if-then-else rules. However, the latter may well be the result of these modeling approaches.

An example of an objective might be to identify subsets most dissimilar in some *outcome variable* such as response. An illustration of results from a tree modeling approach, using a *dichotomous* response rate (R.R.) as the objective, is shown in Figure 31. Note that some techniques are limited to binary splits at each node; others can create multilevel splits. In this example, the total sample of 100,000 has an overall response rate of 5%. By testing a whole series of alternative splits, the technique determines that the variable which gives the "best" split is age, and partitions the sample into two groups having response rates of 7% and 3.9%. This process continues down through the structure until some stopping criterion is satisfied.

FIGURE 31. A TREE PARTITIONED BASED ON RESPONSE RATE



Applications

Some flavor of tree modeling technology is found in almost all general purpose data mining applications, and is used to solve a variety of prediction and classification problems. Typically, algorithms such as classification and regression trees technology, CHAID, and C4.5 are implemented in these applications. FICO has developed its own tree modeling application, and has integrated it into FICO® Model Builder. Model Builder implements a version of the classification and regression trees technology algorithm, modified as follows:

- Both binary and multilevel splits can be represented.
- Extremely large datasets can be processed.
- Trees can be constructed that make use of multiple outcome variables.
- The trees can be represented as XML and exported to FICO® TRIAD® Customer Manager, FICO® Blaze Advisor® business rules management system, or Blaze Decision system applications.

[37] Decision Trees are described in the section titled "Graphical Decision Models."

Tree modeling approaches continue to be an active area of investigation in the statistics and machine learning communities. Active areas of interest include the use of aggregate tree approaches such as bagging and boosting. Also, tree-based variants of Multi-Relational Data Mining algorithms appear to be gaining acceptance.

Strengths

- Simple trees are easy to interpret and are visually appealing.
- Capture interactions between variables.
- Do not require data distribution assumptions.
- Are relatively fast to compute.
- Handle missing values intuitively.
- Are robust and very resistant to outliers.
- Make use of locally predictive features.
- Provide information on the predictive power of the variables in the dataset.
- Guard against overfitting via tree pruning and error estimation techniques.
- Allow for fast application of the model to future data ("on-the-fly" variable computation).

Weaknesses

- Large trees can be difficult to interpret.
- In the absence of a validation mechanism, there is a tendency toward overfitting.
- Provide poor node-based estimates.[38]
- Contain a limited number of subsets (and therefore strategic options) for a reasonable sample size.
- Do not capture simple linear relationships efficiently.
- Are "unforgiving" in split selection. A single bad split can have significant negative impact on subsequent splits.
- Are "data consuming" and require large amounts of data.
- Produce sharp discontinuities in fitted surface.

References

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1984), *Classification and Regression Trees, Wadsworth*.

Breiman, L. (1996), *Bagging Predictors*, Machine Learning 26, pp. 123-140.

Crawford, Stuart L. (1989), *Extensions to the CART Algorithm*, International Journal of Non-Machine Studies, 31, pp. 197-217.

Friedman, J. H. (2001), *Greedy Function Approximation: A Gradiant Boosting Machine*, Annals of Statistics 29, pp. 1189-1232.

Kramer, S., Widmer, G. (2001), *Inducing Classification and Regression Trees in First Order Logic*. In Relational Data Mining, S. Dzeroski and N. Lavrac, editors. Springer, Berlin.

[38] To address this problem, the concept of bagging has been introduced by Leo Brieman. In bagging, many trees are built via bootstrap samples, and the new predictor becomes the average of the predictors from each bootstrapped model. This provides more accurate node-based estimates but considerably reduces the interpretability of the tree.

© 2012 Fair Isaac Corporation. All rights reserved.

Magidson, J. (1992), The CHAID *Approach to Segmentation Modeling: CHI-squared, Automatic Interaction Detection*.

Quinlan, J. R. (1993), C4.5: *Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA.

© 2012 Fair Isaac Corporation. All rights reserved.

» Utility Theory

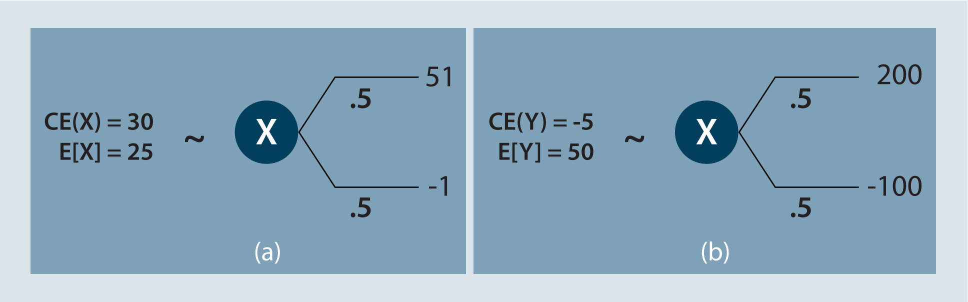
The most coherent way to incorporate decision makers' attitudes towards risk in making a decision under uncertainty, is to assess their utility function for the relevant consequence, e.g., profit, and then choose the strategy that maximizes the expected utility. Utility theory provides the underlying foundations and procedures for constructing a decision maker's subjective utility function.

Risk Attitude[39]

Most commonly, the decision maker chooses his or her certainty equivalent values for a number of specific simple lotteries, which have outcomes in the range relevant to the decision problem in question. A couple of simple lotteries are illustrated in Figure 32. The *certainty equivalent* for a simple lottery is the consequence (for example a dollar amount) that the decision maker is willing to accept in lieu of (be indifferent between it and) playing the lottery. For the lottery in Figure 32(a), the certainty equivalent (30) exceeds the expected value (25). For the lottery in Figure 32(b), the negative certainty equivalent is far less than the expected value. In the first case, the decision maker is said to exhibit risk proneness. In the second case, the decision maker is said to be risk averse. If given a choice, the risk averse decision maker will prefer:

- Any sure amount higher than 30 to the lottery in (a).
- The lottery in (b) to sure losses not higher than 5.
- Lottery (a) to lottery (b).

FIGURE 32: ILLUSTRATION OF THE SIMPLE LOTTERY



Single-objective Utility Function

The result of such an assessment procedure is a utility function of the type depicted in Figure 33. Concavity of the utility function represents a region of the domain of the decision maker's assets where the decision maker is risk averse, while convexity represents a region where the decision maker is risk prone. The shape and locus of the utility function depend, to a large extent, on the current assets of the decision maker. An individual that plays the state-lottery, for example, exhibits risk-prone behavior, because the expected value of the lottery is lower than the ticket cost. On the other hand, individuals pay insurance premiums, referred to as risk premium in the utility theory jargon, because they are typically risk averse in the range of values associated with houses, cars, etc.

[39] Also referred to as risk sensitivity.

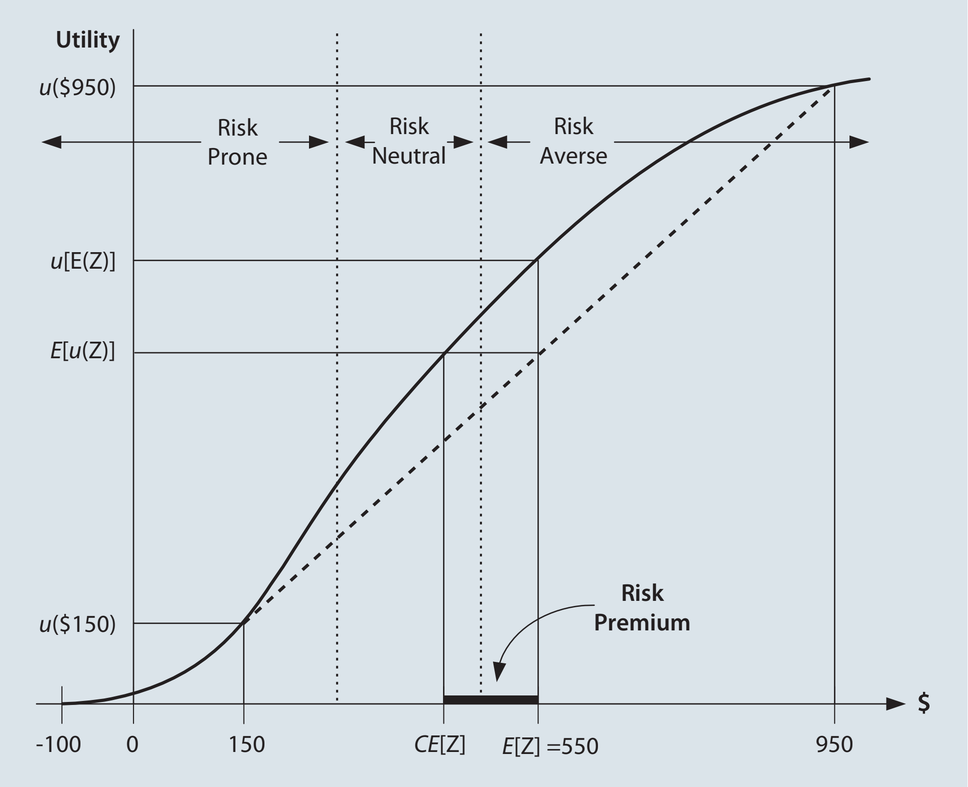
© 2012 Fair Isaac Corporation. All rights reserved.

By definition:

Risk premium = Expected value of lottery – Certainty equivalent of lottery.

Figure 33 graphically illustrates these notions for a 50-50 lottery, *Z*, in which the decision maker can win either $150 or $950. Clearly, *E*[*Z*] = $550. The certainty equivalent of the lottery, *CE*[*Z*], is the (certain) dollar value that has the same utility as the expected utility of the lottery, *E*[*u*(*Z*)] = .5*u*($150) + .5*u*($950). The risk premium is then *E*[*Z*] – *CE*[*Z*], the amount the decision maker is willing to "give up" to avoid the risk.

FIGURE 33: UTILITY FUNCTION ILLUSTRATION



Multi-attribute Utility Theory

When multiple objectives are at stake, they must be aggregated into a single measure of performance, to which a decision rule can be applied. One way to reconcile conflicting objectives is through explicit tradeoffs. The section on "Multiple-objective Decision Analysis" further discusses how tradeoffs can be articulated and represented and what their limitations are.

© 2012 Fair Isaac Corporation. All rights reserved.

Multi-attribute[40] utility theory (MAUT) provides a systematic framework to build a multi-attribute utility function which captures the relative weights of the attributes (objectives), their interactions, as well as the decision maker's risk attitude towards uncertainty in each of the individual attributes. The goal of MAUT is to construct a utility function of the form $u(v_1, v_2,..., v_k) = f(v_1, v_2,..., v_k)$, where v_k is the k-th attribute of concern.

The process of evaluating even a two-attribute general utility function (K = 2) becomes quickly intractable, and in practice analysts have used much simpler forms of $u(v_1, v_2,..., v_k)$. The most commonly used form is the additive utility function,

$$u(v_1,v_2,...,v_k) = \sum_k w_k u_k(v_k)$$

where $u_k(v_k)$ is the single-attribute utility function for attribute k, and w_k-s are the corresponding weights, which are all positive and sum to 1. This form requires strong assumptions of independence among attributes, which do not always hold.

Applications

Utility theory is used to capture the trade-offs and preferences of individuals or institutions in a coherent manner, particularly to make decisions under risk. The entire insurance industry is founded on the fact that most individuals are risk averse. Applications range from managing stock portfolios to determining insurance premiums.

Strengths

- Utility functions systematically capture risk attitude of the decision maker in choices under uncertainty.
- Utility functions allow quantitative modeling of qualitative objectives.
- Multi-attribute utility functions can capture the relative importance of several objectives as well as their interactions.

Weaknesses

- Constructing multi-attribute utility functions, except in the simplest form, is a very lengthy process to which few decision makers will submit.

References

Edwards, W., editor (1992), *Utility Theories: Measurement and Applications,* Kluwer.

Keeney, R. L., and Raiffa, H. (1976), *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, J. Wiley and Sons.

Kirkwood, C. W. (1977), *Strategic Decision Making*, ITP.

von Neumann, D., and Morgenstern, O. (1947), *Theory of Games and Economic Behavior*, Princeton University Press.

von Winterfeld, D., and Edwards, W. (1986), *Decision Analysis and Behavioral Research*, Cambridge University Press.

[40] An attribute in this context refers to the quantity and appropriate scale that measure the achievement of an objective. For example cost in dollars is a valid attribute if the objective is to minimize expected cost.

» Glossary

Some of the following glossary entries were written to clarify the terms as they are used in this paper and ignore their broader interpretation.

Additive

Generally referring to relationships that exhibit no high order *interaction* or *association*. Additive models are of the form:

$$y = f_1(\mathrm{x}_1)+f_2(\boldsymbol{x}_2)+ \ldots + f_n(\boldsymbol{x}_n)$$

where each of the functions $f_i(.)$ depends only on variable x_i.

Analysis of Variance

A technique for partitioning the variation in a continuous dependent variable(s) into variation due to the categorical or classification variables and variation due to random error. Analysis of variance may be written as a linear model to predict the dependent variable; model parameters are fit using a least squares method. Tests can be constructed to determine the significance of the classification variables.

Artificial Intelligence

A general term referring to those scientific fields concerned with the development of computer systems that exhibit "intelligent" behavior. Historically, artificial intelligence has included the fields of expert systems and knowledge-based systems. More recently, artificial intelligence has been broadened to include the fields of neural networks, genetic algorithms, fuzzy systems, case-based reasoning, artificial life, object-oriented programming, virtual reality and myriad other computer-based technologies.

Association

The relationship between two or more categorical variables in a cross tabulation.

The chi-squared test may be applied to measure the strength of evidence that an association exists. Log-linear models (refer to section on Log-linear Models) allow one to test hypotheses about the level of association between variables. Other tests exist to measure the strength of an association.

Association Rule

A data mining term generally used in the context of a database of transactions, where the rules represent associations between data items. The presence of one set of items in a transaction implies the presence of other items with some specified degree of confidence.

Attribute

A specific value that a variable can take, e.g., in home ownership, "renter" is an attribute; "under 25" is an attribute of age.

Bayesian Network

A purely probabilistic graphical model which represents the joint probability distributions of variables as nodes, and variable dependencies as arcs between nodes. Also referred to as a *probabilistic graphical model*, a belief network and a Bayesian Belief Network (BBN).

Binary Outcome

The modeling situation where the dependent variable has only two values, e.g., response/no response, good/bad.

Bivariate

Relating to two variables.

© 2012 Fair Isaac Corporation. All rights reserved.

Bootstrap Sample

A bootstrap sample is a sample of size *n* drawn with replacement from some source sample of size *n*. Some of the observations in the sample will be in a given bootstrap sample and some will not. The probability that a particular observation will appear in a given bootstrap sample is approximately 63.2%. Bootstrap samples are used in many modeling technologies to obtain reliable prediction estimates, usually by averaging the estimates across the samples, especially in cases where only small samples are available.

Categorical

A variable is said to be categorical when its values are categories which are not necessarily ordered. For example, occupation is a categorical variable. Continuous variables such as applicant age may be made categorical or intervalized by converting the values to such as "under 20," "20—30," etc. Categorical variables may alternately be referred to as discrete, classification, qualitative or nominal variables.

Characteristic

A variable that may be considered for inclusion in a *scorecard*. A characteristic can be a relatively simple value, for example total amount of monthly mortgage payments. Or it can be a much more complicated value, for example, debt ratio (the ratio of the total monthly payment obligations for mortgages, installment and revolving credit as a percentage of total monthly income from employer, interest and dividends).

Chi-square Test

A statistical test that attempts to assess the significance of differences in the actual cell frequencies and the expected cell frequencies in a cross-tabulation. Often called chi-square Goodness of Fit Test.

Classification

Referring to a family of techniques whose main objective is to generate functions to classify an outcome into one of two or more *categorical* outcomes as a function of a set of *predictor variables*.

Conditional

Referring to the observation or measurement of a phenomenon for a subgroup of cases with the value of one (or more) particular variable(s) is held constant.

Confounding

Referring to the condition where two ore more variables vary together in such a way that it is impossible to determine which variable is responsible for an observed effect.

Conjoint Analysis

Also referred to as feature trade-off analysis. A method for establishing respondents' utilities based on the preferences they express for combinations of product attributes and features. Price is typically one of attributes included.

Connection Weights

Referring to the weight associated with a connection between two nodes in a neural network. The target node contains a summation function of some form to add up the connection weights of the arcs arriving at the target node from the source nodes.

Continuous

A variable is said to be continuous or quantitative if its values are real numbers such as age, income, amount purchased.

© 2012 Fair Isaac Corporation. All rights reserved.

Continuous Outcome

The modeling case where the dependent variable has a continuum of values, e.g., revenue.

Constrained Optimization

A term referring to a branch of Operations Research where some objective has to be optimized (maximized or minimized) subject to some set of inviolable constraints.

Correlation

The extent to which there is a straight line relationship between two variables (i.e., to the extent to which income rises proportionately with age. The measure of correlation (the correlation coefficient) lies between -1.0 and +1.0, with +1.0 indicating perfect positive correlation, -1.0 indicating perfect negative correlation and 0 indicating the absence of a linear relationship. Specifically, correlation is covariance normalized by the product of the standard deviation of the two variables:

$$\rho_{ij} = \mathrm{Corr}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{Cov(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{Var\,(\boldsymbol{x}_i)\,.\,\mathrm{Var}\,(\boldsymbol{x}_j)}} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\,.\,\sigma_{jj}}}$$

A correlation matrix is a matrix of all correlation coefficients for a set of variables taken two at a time.

Covariance

A measure of the extent to which two numeric variables are linearly related. More specifically, it indicates the difference between the mean of the product of two variables and the product of the means:

$$\sigma_{ij} = Cov(x_i, x_j) = E\,(x_i\,.\,x_j) - E(x_i)\,.\,E(x_j) = E(x_i\,.\,x_j) - \mu_i\,.\,\mu_j$$

So if the values of x_i and x_j do not deviate too far from their respective means, the covariance of the two variables will also be small. If they deviate substantially, the covariance will be larger. Unlike correlation, magnitude of covariance is a function of the magnitudes of x_i and x_j. Covariance cannot indicate the degree to which the variables are related non-linearly.

Cross-validation

A method for estimating the reliability of a statistic generated from a small sample. In N-fold cross-validation, with n=1,...,N sample points, the statistic is computed N times from N-1 sample points; a different sample point is held out for each new computation. Similarly, in v-fold cross-validation, the sample is divided into v subsets, and the statistic is computed N/v times; each time a different subset is held out from computation. If the sampling distribution of the statistic being estimated is approximately normally distributed, confidence intervals for the estimate of the statistic can be approximated using the n-fold or v-fold statistic computations.

Data Mining

The class of methods used to extract patterns from data. Data mining has evolved with the phenomenal growth in the size of databases and the need to extract or "mine" information from them. This area represents a convergence of the fields of machine learning and statistics. Primary tasks of data mining include *classification, regression*, clustering, dependency modeling and pattern recognition. All the data analysis and modeling techniques discussed in this paper fall under the umbrella of data mining.

Data Structure Assumptions

Many statistical modeling techniques have specific data requirements. Some may involve the form of the data, e.g., must be categorical, or some have distributional assumptions. For instance, Discriminant Function Analysis is based on the assumption of normality for all the predictor variables.

© 2012 Fair Isaac Corporation. All rights reserved.

(Violation of distributional assumptions does not immediately invalidate the use of a technique but requires extreme care on the part of the analyst.)

Decision Variable
A controllable variable whose value is determined by the application and whose value forms part of the solution to the problem being solved.

Dependent Variable
See *outcome variable*.

Development Sample
A part of a population used to estimate or train a model. See also, *training sample*.

Dichotomous
Having only two values. See also, *binary*.

Dimension Reduction
Attempt at reducing the number of variables in data analysis by eliminating those that have no bearing on the analysis or by creating combination variables from correlated variables where fewer combined variables represent most of the information of the large number of original observed variables.

Directed Acyclic Graph
A directed graph containing no cycles, commonly referred to as DAG.

Euclidean Distance
Measure of geometric distance between two observations as measured by some function of the numerical values of the variables in the two observations. Specifically, the distance between an observation *P* with coordinates $(p_1, p_2,...,p_n)$ and an observation *Q* with coordinates $(q_1, q_2,...,q_n)$ is:

$$d(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2}$$

Extrapolation
The prediction of the value of an outcome variable outside the measured range of values of the predictor variables.

Feature
A feature is any single valued transformation of an input vector of *predictor variables*. Suppose the input vector is composed of time series of bills and payments. Examples of features would include: the balance last month, the ratio of last month's payment to last month's balance, the indicator variable of the event, {max delinquency = 1}, and any polynomial or spline transformation of the input vector. A shallow regression tree developed from the inputs and outputs could also be a feature.

Generalized Additive Model (GAM)
A modeling technology in which the score formula is represented by a sum of terms, where each term is a non-linear function of a *single predictor variable*. A fitting algorithm often associated with GAMs is the backfitting algorithm, which sequentially fits the non-linear functions to the data in a way that interactively decreases the model's misfit.

Goal Programming
An optimization technique used in operations research which aims to optimize several objectives (goals) simultaneously.

© 2012 Fair Isaac Corporation. All rights reserved.

Gradient
A vector of first partial derivatives of a function that is assumed to be differentiable at least once.

Heteroscedasticity
Refers to situations in which the variability of the model residuals is not constant. Most modeling procedures assume that the variability of the residuals is constant everywhere. If heteroscedasticity is observed, it may often be removed by transforming the *outcome variable* using a square root or a logarithm.

Hidden Markov Model (HMM)
Used for the stochastic modeling of non-stationary time-series data. An HMM can be regarded as a random generator of feature vectors. It consists of a set of states connected by probabilistic transitions. The model dynamically switches to a new state each time a new feature vector is observed. Every HMM consists of two key components: state transition probabilities that model the temporal correlation variability between the features vectors and output probabilities that model the characteristic variability of individual feature vectors. The power of HMMs results from the ability to combine the modeling of stationary stochastic processes producing observable features and the temporal relationships between these processes.

Hold-out Sample
A part of the population from which the *development sample* is drawn which is "held out" from the development and used as an independent check on results. Note that this is not independent in time from the development. Also referred to as a test sample or validation sample.

Independent Variable
See *predictor variable*.

Interaction
An interaction between variables is said to occur when the effect of one predictor variable (categorical or continuous) on the dependent variable depends on the observed level of a second predictor variable. For instance, the relationship between age and log (odds) may be increasing for Owners, but decreasing for Renters. Interactions are generally difficult to capture in a strictly additive model without some generation of additional variables to represent the interaction information.

Intervalized
See *categorical*.

Latent Semantic Indexing (LSI)
A set of techniques for text processing. The distinguishing feature of this approach is that principal component analysis (PCA) is used to map a high-dimensional space (where each distinct word defines its own axis) to a lower-dimensional space. The vectors for related words are close to one another, and the similarity between words or documents can be computed via the vector dot product operation. Also known as Latent Semantic Analysis (LSA).

Learning Vector Quantization (LVQ)
A supervised segmentation and classification technique developed by Teuvo Kohonen at the Helsinki University of Technology in the 1980s. Given a set of labeled training data, it learns a model for assigning class labels to data records. In brief, every data vector defines a point in a d-dimensional data space. One or more prototype vectors per class are placed into the same space. To classify a feature vector, the Euclidean distance to all prototype vectors is measured, and the exemplar is assigned to the class of the nearest prototype vector. The prototype vectors are iteratively moved in response to training exemplars for which the classifier's assignment disagrees with the training

© 2012 Fair Isaac Corporation. All rights reserved.

label. Specifically, the nearest prototype vector of the class that should have captured the exemplar is moved toward the training point. In some versions of the algorithm, the prototype vector that incorrectly captured the exemplar is also moved away from the training point. In practice, the learning process normally converges on a good solution rapidly

Least Squares
An estimation process which minimizes the squared difference between the predicted results and the actual results. Least squares estimation results are strongly influenced by *outliers*.

Line Optimization
Optimizing the *objective function* along a line.

Linear
Often referring to a straight line relationship between two variables. The relationship is generally characterized by a slope term and, in most instances, an intercept term as well.

Linear Combination
A method of generating a new variable from other variables by taking the sum of input variables each multiplied by some constant factor. For example, the variable *z* is a linear combination of *w*, *x* and *y*:

$$z = 1.20 \cdot w + 0.90 \cdot x + 2.40 \cdot y$$

Main Effect
The effect of one predictor variable on a dependent variable independent of any other predictor variable effects

Maximum Likelihood Estimation (MLE)
Method of parameter estimation used in logistic regression and log-linear modeling. MLE is used to estimate the parameters in a model such that the likelihood of the sample, or its probability of occurrence, is maximized. Since there is generally no closed form solution for the maximum likelihood estimate of the parameters, an iterative estimation process using a stop criterion is applied (several different computational methods exist). In instances of *nominal predictor variables*, this is equivalent to minimizing the Log Likelihood Ratio statistic across cells:

$$G^2 = -2.\sum \left((Observed).log_e \left(\frac{Expected}{Observed} \right) \right)$$

Misclassification Cost
The cost of misclassifying a class "j" object as a class "i" object. The costs may vary across object classes, e.g., the cost of misclassifying a "Bad" as a "Good" may be higher than the reverse.

Missing Values
Rarely does every field in every sample point have a valid value. In some cases, where the original data is captured from hard copy records, data is illegible, lost, never recorded and so for any particular sample point, variables may have missing values. Some techniques handle this very poorly (to the extent that data might have to be discarded). Others can finesse around this by substituting values (such as minimum or average). Others (INFORM is probably unique with the No Inform concept) handle it appropriately.

Multicollinearity
The situation where two or more of the independent variables are very highly correlated.

© 2012 Fair Isaac Corporation. All rights reserved.

Multidimensional

Observations having multiple variables whose values help distinguish one observation from another. Often referred to as *multivariate*.

Multinomial

Having more than two possible values.

Multivariate

See *multidimensional*.

Nominal Variable

Variable having values which are categorical and in no implied order. For example, the variable "residence" with values "own" and "rent" is nominal because its values cannot be ordered without further information on what the ordering should be. Compare against *ordinal* or *continuous*.

Non-additive

See *interaction*.

Non-linear

Usually referring to a relationship between two variables that varies more than just *linearly* over different regions of data. Characterization of this type of relationship takes more parameters than just slope. Usually, fairly complex functions are used to represent the non-linear shape of the relationship. For example:

$$f(\mathbf{x}) = \frac{1}{(1+e^{-x})}$$

Non-parametric Method

A method which makes no assumptions about the type of distribution from which the data came.

Objective Function

The formal statement of goal to be optimized by a mathematical programming optimization algorithm.

Odds

The ratio of frequency of occurrence two possible outcomes:

$$odds = outcome_1 : outcome_2 = \frac{outcome_1}{outcome_2}$$

Optimization Algorithm

A set of rules applied in a finite number of steps for identifying an (approximate) optimal solution to a stated *objective function* in terms of one or more *decision variables*. The problem may or may not be subject to a set of constraints on the decision variables.

Ordinal Variable

Variable which is categorical and whose values are ordered but without any implied distance between the values. Many survey responses are ordinal (answer between '1' for worst and '5' for best but not necessarily meaning '5' is five times better than '1').

Outcome Variable

The variable of interest to be modeled or predicted. Synonymous with performance, dependent or criterion variable.

© 2012 Fair Isaac Corporation. All rights reserved.

Outlier
Referring to an observation or case that lies so far outside of an expected pattern (e.g., very far from the fitted regression line) as to prompt further investigation into the possible problems with the observation itself or even with the "expected pattern."

Overfitting
See *sample tuning*.

Performance Inference
An attempt at guessing the probability distribution of an unknown dichotomous outcome variable for a group of accounts where such outcome has no chance of being measured. Good examples are good/bad outcome on declined applicants and response/no-response outcome on non-mailed names in a list.

Predictor Variable
The known items of information which are used to predict or estimate the values of the unknown or independent variables. See also, *independent variable*.

Probabilistic Graphical Model
See *Bayesian Network*.

Regression
Referring to a family of techniques whose main objective is to generate functions to predict, in most cases, some continuous outcome as a function of a set of predictor variables.

Ridge Regression
A regression technique in which the regression coefficients are biased toward zero in order to reduce *sample tuning*.

Sample Noise
Random variation exhibited in data that is unlikely to be repeated in other samples drawn from the same source and is not representative of any actual underlying population phenomena. See related item, *sample tuning*.

Sample Tuning
Sample tuning or overfitting refers to a pitfall of most modeling techniques which occurs when spurious relationships are identified which are not part of the underlying structure. A simple example might be where because of a small sample size, a relationship is identified in the data which isn't representative of the full population to which the solution will be applied. Note that taking large samples or census data do not obviate the need for engineering the solution to fit the future application. Judicious use of a holdout sample for cross validation and an alert analyst can usually avoid this danger.

Scorecard
A table of *characteristics*, each divided into exclusive *attributes*. Each *attribute* has a numerical value and sums up to a score

Score Formula
The mathematical formulation of the scoring function. The score formula has parameters (regression coefficients, score weights, connection weights, etc.), which define the score and are determined by some fitting algorithm applied to a *development* sample.

© 2012 Fair Isaac Corporation. All rights reserved.

Sensitivity Analysis

A tool of quantitative risk analysis for testing the effect of varying parameters in a model either singly or in combination to assess their effect on outcome. Analysis tools for apportioning the variation in the model to its sources include *correlation* coefficients, rank correlation and regression analysis.

Simulated Annealing

An *optimization algorithm* that attempts to find a good solution by random variations of the current solution. The search tries to avoid local minima by jumping out of them in early iterations. When the probability of accepting a worse solution nears zero, the algorithm seeks the bottom of its local minimum. This technique stems from thermal annealing, which attempts to obtain perfect crystallizations by a slow enough temperature reduction to give atoms the time to attain the lowest energy state.

Test Sample

See *hold-out sample*.

Training Sample

See *development sample*.

Uncertainty Analysis

A tool of quantitative risk analysis used to generate empirical histogram distributions for the outcome of a model, by means of sampling from the set of model input variables. Can be computationally intensive.

Weibull Distribution

The Weibull distribution is one of the most commonly used lifetime distributions in reliability engineering. It can take on the shape of other distributions, depending on the value of its shape or slope parameter.

Weight-of-Evidence

Quality index for a piece of information that indicates the strength of the information towards predicting the level of some binary outcome variable. The specific definition is the natural log of the ratio of the conditional probability of having an attribute given one outcome level over the conditional probability of having the attribute given the other outcome level:

$$WoE_i = \log_e \left(\frac{P\,(attribute_i | outcome_1)}{P\,(attribute_i | outcome_2)} \right)$$

For example:

$$WoE_{owner} = \log_e \left(\frac{P\,(owner | good)}{P\,(owner | bad)} \right)$$

© 2012 Fair Isaac Corporation. All rights reserved.

about FICO

FICO (NYSE:FICO) delivers superior predictive analytics solutions that drive smarter decisions. The company's groundbreaking use of mathematics to predict consumer behavior has transformed entire industries and revolutionized the way risk is managed and products are marketed. FICO's innovative solutions include the FICO® Score—the standard measure of consumer credit risk in the United States—along with industry-leading solutions for managing credit accounts, identifying and minimizing the impact of fraud, and customizing consumer offers with pinpoint accuracy. Most of the world's top banks, as well as leading insurers, retailers, pharmaceutical companies and government agencies, rely on FICO solutions to accelerate growth, control risk, boost profits and meet regulatory and competitive demands. FICO also helps millions of individuals manage their personal credit health through ***www.myFICO.com***. Learn more at ***www.fico.com***.

For more information	North America toll-free	International	email	web
	+1 888 342 6336	+44 (0) 207 940 8718	info@fico.com	www.fico.com

FICO, Blaze Advisor, TRIAD and "Make every decision count" are trademarks or registered trademarks of Fair Isaac Corporation in the United States and in other countries. Other product and company names herein may be trademarks of their respective owners. © 2006-2012 Fair Isaac Corporation. All rights reserved.