A probabilistic estimation framework for predictive modeling analytics

by C. V. Apte
R. Natarajan
E. P. D. Pednault
F. A. Tipu

IBM ProbE (for probabilistic estimation) is an extensible, embeddable, and scalable modeling engine, particularly well-suited for implementing segmentation-based modeling techniques, wherein data records are partitioned into segments and separate predictive models are developed for each segment. We describe the ProbE framework and discuss two key business solutions that have been built using ProbE: the IBM Underwriting Profitability Analysis for insurance risk management, and the IBM Advanced Targeted Marketing for Single Events for direct mail database marketing.

IBM ProbE (for probabilistic estimation) is a customizable data mining engine that is being developed to enhance the IBM predictive modeling products and services. Viewed from the broadest perspective, ProbE might best be described as an extensible, embeddable, and scalable segmentation-based modeling engine. Although virtually any predictive modeling technique can be implemented within ProbE's software environment, ProbE's application programming interfaces (APIs) are particularly well-suited for implementing segmentation-based modeling techniques, wherein sets of data records are partitioned into segments and separate predictive models are developed for each segment.

This style of modeling is popular among data analysts and applied statisticians, and it is usually approached as a sequential process in which data are first segmented (using, for example, unsupervised clustering algorithms), and predictive models are then developed for those segments. The drawback of this sequential approach is that it ignores the strong influence that segmentation exerts on the predictive accuracies of the models within each segment. Good segmentations tend to be obtained only through trial and error by varying the segmentation criteria.

ProbE is able to perform segmentation and predictive modeling within each segment simultaneously, thereby optimizing the segmentation so as to maximize overall predictive accuracy and thus to produce better models. Currently, ProbE includes a top-down tree-based algorithm for constructing segmentations, as well as a collection of other algorithms for constructing segment models. The latter includes stepwise linear regression and stepwise naive Bayes algorithms for general-purpose modeling, and a joint Poisson/log-normal algorithm for insurance risk modeling.

IBM ATM-SE (Advanced Targeted Marketing for Single Events) is an application built on top of ProbE for mining high-dimensional customer interaction and promotion history data in order to construct customer profitability and response likelihood models for the retail industry.[1] An evaluation of ATM-SE was recently conducted with Fingerhut Inc., a leading U.S. direct-mail retailer and a sophisticated user of predictive analytics in their targeted marketing efforts. The segmentation-based response models pro-

0018-8670/02/$5.00 © 2002 IBM

duced by ProbE either equaled or slightly outperformed Fingerhut's proprietary models, in a completely automated mode. The outcome of this evaluation is significant because numerous vendors and consultants have attempted to beat Fingerhut's in-house modeling capability in the past, but previously none had succeeded. Moreover, ProbE achieved this result in a fully automated mode of operation with no manual intervention. Although further development and testing is still needed, early indications are that ProbE will be able to consistently produce high-quality models for this application on a fully automated basis without requiring costly manual adjustments of the models or the mining parameters by data mining experts, a necessary step in making data mining attractive to medium-sized businesses.

A key feature of ProbE is that it can be readily extended so as to construct a wide range of predictive models within a segment. For example, in the IBM UPA (Underwriting Profitability Analysis) application,[2] a joint Poisson/log-normal statistical model is used to simultaneously model both the frequency with which insurance claims are filed, and the amounts (i.e., severities) of those claims for each segment. Using this class of segment models, the segments identified by ProbE would thus correspond to distinct risk groups whose loss characteristics (i.e., claim frequency and severity) are estimated in accordance with standard actuarial practices.

A second example is found in the ATM-SE application for predicting customer response to promotional mailings. To predict the expected revenues that would be generated by a customer targeted in such mailings, segment models were constructed using least-squares linear regression with forward stepwise feature selection to select the variables that appear in the regression equations. Using this class of segment models, ProbE would construct piecewise-linear models in which the segments correspond to regions of the response surface that are approximately linear and the boundaries between segments correspond to nonlinearities detected in that surface.

To predict the probability of a customer responding to a promotional mailing, segment models were constructed using naive Bayes methods with forward stepwise feature selection to select the variables that appear in the conditional probability equations. Using this class of segment models, ProbE would construct piecewise naive Bayes models in which the segments correspond to regions of the response surface in which the naive Bayes independence assumptions are locally valid and the boundaries between segments correspond to interactions among features detected in the response surface that violate the naive Bayes assumptions.

In addition to being extensible with respect to segment models, ProbE also permits extensions to be made to its segmentation algorithms. This degree of extensibility was achieved through careful design of ProbE's APIs. In particular, a single API is used to implement all predictive modeling algorithms, including segmentation algorithms. This model API is general enough to permit a very wide range of predictive modeling techniques to be implemented within ProbE. No matter what kind of predictive models are used within each segment, the same segmentation algorithms are used in ProbE to optimize the predictive accuracies of the resulting ensemble of models independent of their internal details.

ProbE is also designed to be an embedded system that can be incorporated into industry-specific application environments. For example, ProbE does not have a graphical user interface (GUI) of its own; instead, one would have to be supplied by the host application if so desired, as is done in the UPA and ATM-SE solutions. The interface to ProbE has been kept as simple as possible. Host applications provide ProbE with specifications of data mining tasks to be performed, and ProbE returns the results of those tasks upon completion. At present, communication is conducted through specification and results files; however, future extensions to ProbE will permit full integration with relational database systems, with task specifications and mining results communicated through database tables.

Another consideration in the design of ProbE is scalability. ProbE is designed to work with very large, out-of-core data sets. Work is also underway to develop a data-partition parallelized version of ProbE that would allow large data sets to be partitioned across multiple processors, with each processor accessing data only in the partition assigned to it and with only statistical summary information being exchanged among processors. Because this approach would minimize the amount of communication among processors, it is anticipated that it will achieve near-linear improvements in execution speed (i.e., increasing the number of processors by a factor of n decreases the execution time by a factor of n).

In the next section, we describe the tree-based segmentation strategy utilized in ProbE. In the following two sections we elaborate in detail on two core predictive modeling algorithms in ProbE: linear regression trees and naive Bayes trees. This pair of algorithms has been used successfully in building solutions for targeted marketing and financial credit risk scoring. We then describe a third predictive modeling algorithm in ProbE; a joint Poisson log-normal model that was developed specifically for an insurance risk modeling application. The last section contains our concluding comments.

For scalability reasons, ProbE is designed to handle massive out-of-core training data sets.

Tree-based segmentation

The tree-based segmentation algorithm that is incorporated into ProbE can be used in conjunction with a wide range of multivariate statistical models as the leaf models. Beginning with a single root node, an overall tree-based model is generated by recursively applying a *model expansion* procedure to the leaf nodes of the current tree. This model expansion step involves two distinct and complementary mechanisms. The first mechanism is a *node split* that is comparable to those traditionally used to build trees, and involves a univariate binary split of an existing leaf node into two descendant leaves. The second mechanism is a *leaf-model extension* that involves adding a single new feature to a multivariate statistical model that appears in a leaf node of the current tree. Examples of such multivariate leaf models include linear regression models and naive Bayes models.

An important aspect of the above approach is that node splits and leaf-model extensions are placed on the same footing in the model expansion process. For each leaf node, ProbE explores a set of possible node splits on each input feature, as well as possible leaf-model extensions for each input feature. The node split or leaf-model extension that produces the greatest model improvement at a leaf is then selected and incorporated into the tree. The model expansion process is then recursively applied until terminated either by a user-specified "stopping condition" or by an internal cross-validation heuristic.

Another important aspect of the above approach is that feature selection is performed within each potential new leaf model as it is being constructed during both node splitting and leaf-model extension. The features that can be selected are restricted to those that appear in the leaf-model extensions performed along the path from the root node to the potential new leaf node being constructed. Thus, leaf-model extensions specify the subset of features that are allowed to appear in a leaf model, while leaf-model feature selection determines which of these features are actually used. A best-first wrapper-based approach[3] is used for leaf-model feature selection. Unlike ProbE, previous methods that use multivariate statistical models in the leaves of trees (e.g., References 4, 5) do not perform feature selection on leaf models, or they do not use multivariate leaf models to identify good splits, or both.

For scalability reasons, ProbE is designed to handle massive out-of-core training data sets. The I/O cost of each data scan is therefore an important performance consideration when implementing learning algorithms within ProbE.

To minimize I/O costs in the case of tree-based segmentation, the algorithm implemented in ProbE does not employ sorting when constructing binary splits on numerical features. Instead, binary node splits for all features are constructed by first generating multiway splits on each feature. In the case of categorical features, the individual categories define the multiway splits. In the case of numerical features, the values of the features are binned into subintervals and the resulting subintervals define the multiway splits. Once leaf models have been constructed for a multiway split, the best possible binary split on the corresponding feature is determined using a bottom-up merging procedure analogous to that employed in CHAID.[6] Specifically, two segments of a multiway split are merged so that the resulting segmentation produces the minimum increase in the resulting model evaluation score. This merging procedure is applied iteratively until only two segments remain. These two segments then represent the best binary split for the corresponding feature.

To further decrease I/O costs, the node-splitting procedure used in ProbE also imposes two important requirements on the implementations of the segment model objects constructed at the leaf nodes. First, as the training data are scanned, each relevant segment model object must be able to update its internal data structures to extract relevant "sufficient statistics" in a storage and computationally efficient manner.

Second, it should be possible to combine the sufficient statistics of two or more model objects that correspond to different data segments in such a way that the sufficient statistics of the resulting segment model object can be obtained without having to rescan the training data. That is, the result should be the same as if the resulting model object were trained using the union of the original training data for the data segments that are being merged. This requirement enables the entire binary bottom-up merging procedure described above to be performed without having to rescan the training data.

The ProbE framework is valuable because these two conditions can be satisfied within the framework for a large class of statistical models, although in some cases it raises interesting computational research issues.

Linear regression trees

The combination of the above tree-based segmentation algorithm with stepwise linear regression modeling at the leaves yields an overall algorithm that we refer to as linear regression trees (LRT). The segment models in this case employ a Gaussian probability model

$$\theta_y(\boldsymbol{X}, t) \sim \mathcal{N}\left(a_0 + \sum_{j}^{J} a_j(t) X_j,\ \sigma(t)^2\right) \tag{1}$$

in each segment t, with the regression parameters computed using the well-known normal equations method (Reference 7 page 224, Reference 8 page 49). This method satisfies the two requirements of the tree-based segmentation algorithm discussed above because the sufficient statistics (in this case, means and covariances) can be obtained from a single pass over the training data. Furthermore, the sufficient statistics from two or more segment model objects can be combined to compute the regression parameters for the model of the resulting combined segment.

Our implementation of the normal equations method incorporates feature selection in a way that regularizes the computations and provides stable estimates of the nonzero regression coefficients. The feature selection algorithm is based on the use of holdout data. Specifically, each relevant data record that is scanned is randomly designated to be either a training record (used to determine the order in which features are introduced in the regression model), or a holdout record (used to determine the optimum subset size in this ordered set of features). Each linear regression model object has separate data structures for storing and updating the means and covariances of the relevant training and holdout data records.

After the sufficient statistics are obtained from a training data scan, a forward stepwise feature selection procedure is used with the training covariance matrix. Subsequently, the optimal number of features is determined from this ordering using the holdout means and covariances. Finally, the means and covariances for the training and holdout data are combined in order to obtain final estimates for the regression parameters with the selected features.

During bottom-up merging, the means and covariances of the training and holdout data for pairs of linear regression model objects are separately merged, and feature selection and parameter estimation is performed in the resulting model object, as described above.

The details of each of the above steps are discussed in the following subsections.

On-line updating and merging of mean and covariance estimates. Let $\boldsymbol{\xi} = \{\boldsymbol{x}, y\}$ denote the training data record, where $\boldsymbol{x}$ denotes the J explanatory features and y is the response. Typically, only the continuous explanatory features are included for the segment models, although categorical features may be incorporated by explicitly encoding them in the input data using dummy indicator variables in the usual way.

For a given set of data (training or holdout), let $\{\boldsymbol{\xi}_i\}_{i=1}^{M}$ denote the previously scanned records. The mean vector $\boldsymbol{\mu}_M$ and covariance matrix $\boldsymbol{S}_M$ are then given by

$$\boldsymbol{\mu}_M = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\xi}_i, \qquad \boldsymbol{S}_M = \sum_{i=1}^{M} (\boldsymbol{\xi}_i - \boldsymbol{\mu}_M)(\boldsymbol{\xi}_i - \boldsymbol{\mu}_M)^T \tag{2}$$

For analytic convenience, the constant normalizing factor for the covariance matrix is omitted.

When a new record $\boldsymbol{\xi}_{M+1}$ is scanned, the following updates can be used to incrementally update the values of $\boldsymbol{\mu}_M$ and $\boldsymbol{S}_M$ as follows:

$$\boldsymbol{\mu}_{M+1} = \frac{M\boldsymbol{\mu}_M + \boldsymbol{\xi}_{M+1}}{M+1},$$

$$\boldsymbol{S}_{M+1} = \boldsymbol{S}_M + \frac{(M+1)}{M}(\boldsymbol{\mu}_{M+1} - \boldsymbol{\xi}_{M+1})(\boldsymbol{\mu}_{M+1} - \boldsymbol{\xi}_{M+1})^T \qquad (3)$$

As noted earlier, each new record is randomly assigned to update either the training data means and covariances $(\boldsymbol{\mu}, \boldsymbol{S})$, or the holdout data means and covariances $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{S}})$.

Now consider two models, with M and N records, respectively, in a given subset of data (training or holdout) at the end of a data scan. For the merged model with $M + N$ records, we have

$$\boldsymbol{\mu}_{M+N} = \frac{M\boldsymbol{\mu}_M + N\boldsymbol{\mu}_N}{M+N},$$

$$\mathbf{S}_{M+N} = \mathbf{S}_M + \mathbf{S}_N + \frac{N(M+N)}{M}(\boldsymbol{\mu}_{M+N} - \boldsymbol{\mu}_N)(\boldsymbol{\mu}_{M+N} - \boldsymbol{\mu}_N)^T \qquad (4)$$

The means and covariances of the training and holdout sets are separately merged to produce the corresponding statistics for the merged model. The symmetric covariance matrices are stored in a packed format to save storage, and the rank-one updates in Equations 3 and 4 are implemented using standard routines in the BLAS library.[7]

Feature selection and linear regression. At the end of a training scan, a sequence of regression models is constructed by introducing explanatory features one at a time in a forward stepwise fashion in order to create a sequence of regression equations. Each explanatory feature is selected so as to maximally reduce the variance of the resulting regression model as measured on the training set. Excluded from consideration are those explanatory features that are highly collinear with respect to the explanatory features that have already been introduced earlier in the sequence. Such explanatory features are excluded in order to avoid numerical instability in the calculation of regression coefficients. Collinearity is detected by examining the model variance obtained when a regression equation is constructed that uses the explanatory features already introduced to predict the next candidate explanatory feature to be introduced. The ratio of the residual variance of the resulting regression model, divided by the original variance of the candidate explanatory feature, is calculated and compared to a threshold. If the ratio falls below the threshold, then that candidate explanatory feature is declared to be collinear and is omitted from further consideration. A threshold of 0.001 has been found to work well in practice, which corresponds to a situation in which the explanatory features already selected account for at least 99.9 percent of the variance observed in a candidate explanatory feature.

Each relevant data record that is scanned is randomly designated to be either a training record, or a holdout record.

Once a sequence of explanatory features has been selected using the training set, a best subset of explanatory features is identified using the holdout set. Specifically, the best subset of explanatory features is the one whose corresponding regression model maximizes the likelihood of the response field as measured on the holdout set.

After selecting a best subset of explanatory features, the mean and covariance matrices of the training and holdout sets are merged using Equation 4 and the resulting merged matrices are used to re-estimate the coefficients and variances of the regression models that were constructed.

Many well-known methods can be used to implement the above calculations (e.g., Reference 8). In particular, a method based on Cholesky factorization can be used to simultaneously solve for the coefficients of the regression equations and to identify the best explanatory feature to be added next in the sequence of regression equations that are produced.[9]

Naive Bayes trees

The combination of the tree-based segmentation algorithm described above with stepwise logistic regression modeling at the leaves yields an overall algorithm that we refer to as naive Bayes trees (NBT). In contrast to LRT, the implementation of NBT with logistic regression segment models is more problematic because the usual logistic regression algorithms require several data scans to fit even a single model, and there is no efficient way to use holdout data for feature selection. Furthermore, there is no set of sufficient statistics that allows two or more individual logistic regression model objects to be combined in the bottom-up merging step in ProbE.

These difficulties are addressed by employing the naive Bayes assumption, which leads to a simplified logistic model in which interaction effects are omitted (see Reference 10, pages 92–93). Using this assumption, the parameters of the resulting logistic model can be estimated in a single pass over the training data. In addition, certain heuristics can be employed for feature selection purposes and for estimating degree-of-fit scores during bottom-up merging.

Naive Bayes model. In the case of naive Bayes models, the response variable y is assumed to be categorical. The conditional probability that the value of the response is category k given that explanatory features $\boldsymbol{x}$ fall into segment t can be expressed using Bayes' rule as

$$\theta_k(\boldsymbol{x}, t) = \frac{P(\boldsymbol{X} = \boldsymbol{x} | y = k, t)\pi_k(t)}{\sum_{k'=1}^{K} P(\boldsymbol{X} = \boldsymbol{x} | y = k', t)\pi_{k'}(t)} \tag{5}$$

where $\pi_k(t)$ is the prior probability that the value of the response is category k. Naive Bayes models assume that the covariates $\{X_j\}_{j=1}^{J}$ in $\boldsymbol{X}$ are conditionally independent given the response y:

$$P(\boldsymbol{X} = \boldsymbol{x} | y = k, t) = \Pi_{j=1}^{J} P(X_j = x_j | y = k, t) \tag{6}$$

This naive Bayes assumption greatly simplifies Equation 5 to the point that the relevant conditional probabilities can be estimated from a set of sufficient statistics obtained from a single training data scan.

Numerical covariates can be used in Equation 6 by employing parametric or nonparametric models for the relevant univariate conditional distributions and fitting them to the training data,[11] or as we have done, by discretizing and binning numerical features to obtain a corresponding derived categorical variable.[12] It should be noted that even simple uniform discretizations can be very effective for naive Bayes modeling[13] although we tend to prefer equiprobable (i.e., maximum entropy) binning.

If the covariate X_j takes on M_j values denoted 1, 2, . . . M_j, respectively, then the estimate $P_{jmk}(t)$ for $P(X_j = m | y = k, t)$ in Equation 6 is given by

$$P_{jmk}(t) = \frac{N_{jmk}(t) + \lambda_{jmk}}{N_k(t) + \sum_{m'}^{M_j} \lambda_{jm'k}} \tag{7}$$

where $N_{jmk}(t)$ is the number of data records in segment t in which the explanatory feature X_j has the value m when the response y has the value k, and where $N_k(t) = \sum_{m=1}^{M_j} N_{jmk}(t)$. The λ_{jmk} are smoothing parameters that are introduced to avoid probability estimates that are equal to zero. In particular, the parameter values $\lambda_{jmk} = 1$ correspond to Laplace smoothing. Note that the frequency counts $N_{jmk}(t)$ in Equation 7 are sufficient statistics for estimating $P_{jmk}(t)$, and can be accumulated in a single pass over the data.

The observed negative log-likelihood of this model, which is used as the degree-of-fit score, is given by

$$\begin{aligned}
&\mathfrak{L}_{TR}(t) \\
&= \underbrace{-\sum_{k=1}^{K} \frac{N_k(t)}{N(t)} \left[\log \pi_k(t) + \sum_{j=1}^{J} \sum_{m=1}^{M_j} \frac{N_{jmk}(t)}{N_k(t)} \log P_{jmk}(t) \right]}_{\mathfrak{A}} \\
&\quad \underbrace{+ \frac{1}{N(t)} \sum_{i=1}^{N(t)} \log \left(\sum_{k'=1}^{K} \{\Pi_{j=1}^{J} P_{jx_{j,i}k'}(t)\} \pi_{k'}(t) \right)}_{\mathfrak{B}}
\end{aligned} \tag{8}$$

where $x_{j,i}$ denotes the value of X_j for the ith training data point. The exact evaluation of $\mathfrak{L}_{TR}(t)$ requires an additional training data scan because the sufficient statistics and the estimates of $\pi_k(t)$ and $P_{jmk}(t)$ allow only the term denoted by $\mathfrak{A}$ in Equation 8 to be evaluated exactly. In addition, when merging the sufficient statistics from two or more probability model objects, again only the $\mathfrak{A}$ term can be exactly evaluated for the resulting combined model object. In all cases, the evaluation of the term denoted by $\mathfrak{B}$ in Equation 8 requires a separate pass over the training data, with the contribution of each data point being evaluated and summed.

Natarajan and Pednault[14] describe a Monte Carlo heuristic to approximate the troublesome term $\mathcal{B}$ in Equation 8, which in conjunction with a BIC (Bayesian Information Criterion) penalized likelihood approach[15] can be used to obtain the best feature set for a given naive Bayes model using just two training data scans. Furthermore, along with some further heuristics, the binary merging step in ProbE can also be performed with just three training data scans, as discussed below.

Feature selection and computational heuristics. The forward-selection algorithm for introducing features in the naive Bayes model is similar to the one considered by Langley and Sage[16] but with a different feature selection criterion based on the maximum induced decrease in the observed negative log-likelihood of the training data. For each naive Bayes segment model, feature selection is implemented by performing a first data scan to collect the frequency counts in Equation 7. With these statistics and the Monte Carlo heuristic referred to earlier, estimates for $\mathcal{L}_{TR}$ can be obtained for ordering the covariate explanatory features via forward stepwise selection. Finally, a second data scan is used to evaluate $\mathcal{L}_{TR}$ exactly for all J feature subsets of size 1, 2, . . . , J according to the ordering of the features determined after the first data scan. The minimum value of the observed negative log-likelihood plus a BIC-penalty term[15] is then used to identify the best subset of features from this sequence. As shown in Reference 14, this approach leads to models that are comparable in predictive accuracy to those obtained using more computationally intensive algorithms, and it is quite practical in a ProbE implementation.

During bottom-up merging, the Monte Carlo heuristic is used to perform binary split construction using at most three training data scans. In the first phase, the frequency counts for each multiway split are collected in a single training data scan. The Monte Carlo estimate for $\mathcal{L}_{TR}$ is then obtained without feature selection. Next, the binary merging steps are carried out by merging the frequency counts from pairs of naive Bayes model objects, and then using the Monte Carlo heuristic at each step to estimate the negative log-likelihood degree-of-fit scores that are used to merge segment models until the best binary split has been identified for each feature. In the second phase, a training data scan is used to provide an exact evaluation of the degree-of-fit score (Equation 8) for each candidate binary split. From this exact evaluation, the best binary split among all features can be identified. Finally, in the third phase, this best binary split is introduced into the tree, and the naive Bayes models in the resulting segments are retrained using feature selection as described in the previous paragraph.

In contrast to the model expansion step in LRT where only one data scan may suffice, each model expansion step in NBT requires at least three data scans, despite the use of the Monte Carlo heuristic.

Predictive modeling for insurance risk management

A third class of models that has been implemented for use in the leaves of the tree are joint Poisson/lognormal models. This model class was developed for use in property and casualty (P&C) insurance risk modeling.

The P&C insurance business deals with the insuring of tangible assets, such as cars, boats, and homes. The insuring company evaluates the risk of the asset being insured, taking into account characteristics of the asset as well as the owner of the asset. Based on the level of risk, the company charges a certain fixed, regular premium to the insured. Actuarial analysis of policy and claims data plays a major role in the analysis, identification, and pricing of P&C risks.

Actuarial science is based on the construction and analysis of statistical models that describe the process by which claims are filed by policyholders (see, for example, Reference 17). Different types of insurance often require the use of different statistical models, the choice of statistical model being dictated by the fundamental nature of the claims process.

For property and casualty insurance, the claims process consists of claims being filed by policyholders at varying points in time and for varying amounts. In the normal course of events, wherein claims are not the result of natural disasters or other widespread catastrophes, loss events that result in claims (i.e., accidents, fire, theft, etc.) tend to be randomly distributed in time with no significant pattern to the occurrence of those events from the point of view of insurable risk (see Figure 1). Policyholders can also file multiple claims for the same type of loss over the life of a policy.

Claim filings such as these can be modeled as a Poisson random process,[17] which is the appropriate mathematical model for events that are randomly distributed over time with the ability for events to reoccur (i.e., renew).

In addition to modeling the distribution of claims over time, actuaries must also model the amounts of those claims. In actuarial science, claim amounts for property and casualty insurance are modeled as probability distributions. Two kinds of distributions are usually considered: those for the amounts of individual claims, and those for the aggregate amounts of groups of claims. In principle, aggregate loss distributions can be derived mathematically from the distributions of the individual losses that make up the sum. However, only in a few special cases can closed-form solutions be obtained for these mathematical equations. In most cases, approximations must be employed. Fortunately, actuaries typically consider large groups of claims when analyzing aggregate loss. The central limit theorem can therefore be invoked and aggregate losses can be reasonably approximated by normal (i.e., Gaussian) distributions.

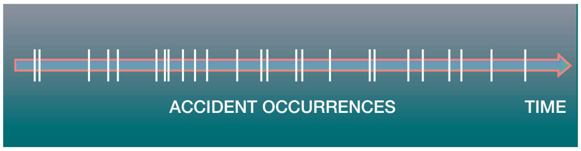
On empirical examination of large volumes of automobile claims data, claim amounts were found to have highly skewed distributions. Most claims were small in value relative to the maximum amounts covered by the policies, but a significant proportion of large claims were also present. When the claim amounts were logarithmically transformed, the skewness virtually disappeared and the resulting distributions were found to be highly Gaussian in shape. These properties are the defining characteristics of log-normal distributions, an example of which is illustrated in Figure 2.

For Poisson random processes, the time between claim events follows an exponential distribution. Moreover, no matter at what point one starts observing the process, the time to the next claim event has the same exponential distribution as the time between claim events. From these properties and the additivity properties of Poisson random processes, it can be shown that the probability density for the time T (i.e., the total earned exposure) until the $(k + l)$th claim filing (where k is the number of settled claims and l is the number of open claims) is given by

$$f(T|k + l) = \lambda^{k+l} e^{-\lambda T} \tag{9}$$

The maximum likelihood estimate used by ProbE for the frequency parameter λ is thus the same one that is typically used by actuaries for estimating frequency:

Figure 1 Accident occurrences over time



Reprinted with permission from C. Apte, E. Grossman, E. P. D. Pednault, B. Rosen, F. Tipu, and B. White, “Probabilistic Estimation Based Data Mining for Discovery Insurance Risks,” *IEEE Intelligent Systems* **14**, No. 6, 49–58 (© 1999 IEEE).

$$\hat{\lambda} = \frac{k + l}{T} = \frac{\text{Total Number of Claims}}{\text{Total Earned Exposure}} \tag{10}$$

In the case of claim amounts, the joint probability density function for the severities $s_1, \ldots, s_k$ of k settled claims is given by:

$$f(s_1, \ldots, s_k) = \frac{1}{\prod_{i=1}^{k} \sqrt{2\pi}\, \sigma_{\log}\, s_i} \cdot exp\left(-\frac{\sum_{i=1}^{k} (\log(s_i) - \mu_{\log})^2}{2\sigma_{\log}^2}\right) \tag{11}$$

where $exp(x)$ stands for e to the power x.

The estimates of the mean log severity $\mu_{\log}$ and the variance of the log severity $\sigma_{\log}^2$ are likewise the ones typically used for log-normal distributions:

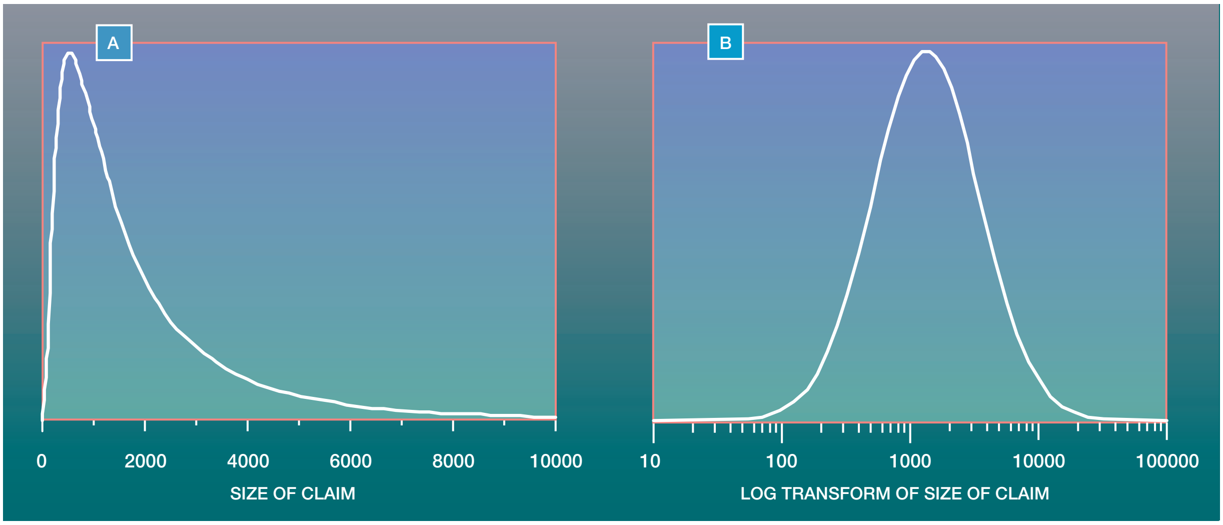
$$\hat{\mu}_{\log} = \frac{1}{k} \sum_{i=1}^{k} \log(s_i) \tag{12}$$

and

$$\hat{\sigma}_{\log}^2 = \frac{1}{k-1} \sum_{i=1}^{k} (\log(s_i) - \hat{\mu}_{\log})^2 \tag{13}$$

Equations 12 and 13 are used during training to estimate the parameters of the severity distribution for individual claims. These estimators presume that the individual severity distributions are log-normal. The usual unbiased estimators for the mean and variance of severity are used after data mining has been completed to estimate the parameters of the aggregate severity distribution:

Figure 2 The probability distribution for (A) size of claim, and (B) log transform of size of claim



Reprinted with permission from C. Apte, E. Grossman, E. P. D. Pednault, B. Rosen, F. Tipu, and B. White, "Probabilistic Estimation Based Data Mining for Discovery Insurance Risks," *IEEE Intelligent Systems* **14**, No. 6, 49–58 (© 1999 IEEE).

$$\hat{\mu} = \frac{1}{k} \sum_{i=1}^{k} s_i \tag{14}$$

$$\hat{\sigma}^2 = \frac{1}{k-1} \sum_{i=1}^{k} (s_i - \hat{\mu})^2 \tag{15}$$

Only fully settled claims are considered when applying Equations 12–15. The severity fields of unsettled claims are often used to record reserve amounts; that is, the money that insurers hold aside to cover pending claims. Reserve amounts are not actual losses and therefore are not used to develop models for predicting actual losses.

As mentioned earlier, negative log-likelihoods are calculated for each database record in a risk group based on Equations 9 and 11. The nonconstant terms in the negative log-likelihoods are then summed and used as the criterion for selecting splitting factors in the top-down identification of risk groups. The constant terms do not contribute to the selection of splitting factors and, hence, are omitted to reduce the amount of computation.

With constant terms removed, the negative log-likelihood score for the ith database record is:

$$\xi_i = \begin{cases} \lambda t_i & \text{for nonclaim records} \\ \lambda t_i + \log\left(\frac{\sigma_{\log}}{\lambda}\right) & \text{for open claim records} \\ \lambda t_i + \log\left(\frac{\sigma_{\log}}{\lambda}\right) + \frac{(\log(s_i) - \mu_{\log})^2}{2\sigma_{\log}^2} & \\ & \text{for settled claim records} \end{cases} \tag{16}$$

where t_i is the earned exposure for the ith record. Note that the Poisson portion of the model contributes an amount $\lambda t_i + \log(1/\lambda)$ to the score of each claim record and an amount λt_i to the score of each nonclaim record. The sum of these values equals the negative logarithm of Equation 9. The log-normal portion of the model contributes nothing to the scores of nonclaim records, and an amount $\log(\sigma_{\log}) + (\log(s_i) - \mu_{\log})^2/(2\sigma_{\log}^2)$ to the score of each settled claim record. The sum of these values equals the negative logarithm of Equation 11 with constant terms (i.e., $\Sigma_{i=1}^{k} \log(\sqrt{2\pi} s_i)$) removed. In the case of open claim records, an expected-value estimate of the log-normal score is constructed based on the scores of the settled claim records. After dropping constant terms from this expected value estimate, open claim records contribute an amount $\log(\sigma_{\log})$ to the log-normal portions of their scores.

If the database records for a risk group contain k settled claims and l open claims, then the sum of the above scores is given by:

$$\xi = \lambda \left(\sum_{i=1}^{N} t_i \right) + (k + l) \log \left(\frac{\sigma_{\log}}{\lambda} \right) + \left(\frac{1}{2\sigma_{\log}^2} \right) \sum_{i=1}^{k} (\log(s_i) - \mu_{\log})^2 \qquad (17)$$

In the above equation, N is the total number of database records for the risk group, the first k of which are assumed for convenience to be settled claim records. Equation 17 is then summed over all risk groups to yield the overall score of the risk model.

Note that the total numbers of open and settled claims, and the mean and variance of the log severity constitute the sufficient statistics of the above joint Poisson/log-normal model, and that negative log-likelihood degree-of-fit scores can be calculated from these sufficient statistics. Moreover, the sufficient statistics from two data segments can be easily merged, as required for ProbE's bottom-up merging process.

From the point of view of machine learning, the important thing to note about Equation 17 is that insurance-specific quantities such as earned exposure and claim status enter into both the equations for estimating model parameters and the equations for selecting splitting factors. Earned exposure effectively plays the role of a weighting factor, while claim status plays the role of a correction factor that adjusts for missing data in one of the two data fields to be predicted (i.e., the settled claim amount given that a claim was filed).

Equation 17 essentially replaces the entropy calculations used in many standard tree-based modeling algorithms. It should be noted that entropy is, in fact, a special case of negative log-likelihood. Its calculation need not be restricted to categorical or Gaussian (least-squares) distributions. The development of the joint Poisson/log-normal model presented above illustrates the general methodology one can employ to customize the splitting criteria of tree-based modeling algorithms to take into account data characteristics that are peculiar to specific applications.

The joint Poisson/log-normal probability model for modeling insurance risk provides the basis of the UPA solution.[2,18] This solution was benchmarked in partnership with Farmers Group, a large P&C insurer, and was found to perform very competitively. As with ATM-SE, the key advantage offered by UPA over traditional actuarial approaches is the high degree of automation in producing robust predictive models from large volumes of insurance data. UPA is typically run on millions of policyholder data records, each with several hundred attributes.

Conclusion

Machine learning and predictive modeling-based solutions have been shown to be highly effective in solving many important business and industrial problems. However, until these techniques are made highly automated, scalable, and reliable, their use will remain limited, gated by the requirement to have trained analysts available to develop predictive models using the techniques.

The ProbE project represents a long-term effort in IBM Research to create highly automated, scalable, and reliable predictive modeling technology. With sufficient automation and reliability built into the overall framework, business solutions that utilize ProbE can be made available to a wider community of business application and solution developers.

With this goal in mind, a central theme of our ongoing research is to integrate the ProbE methodology into database middleware. Through integration, many of the strengths of database technology can be exploited, including the important feature of scalability via data-partitioned parallelism. We anticipate that to make data analytics ultimately widespread in their use, it is first necessary to make them widely available to the vast community of database application developers and users.

Cited references

1. C. Apte, E. Bibelnieks, R. Natarajan, E. P. D. Pednault, F. Tipu, D. Campbell, and B. Nelson, "Segmentation-Based Modeling for Advanced Targeted Marketing," *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, August 2001, ACM, New York (2001), pp. 408–413.
2. C. Apte, E. Grossman, E. P. D. Pednault, B. Rosen, F. Tipu, and B. White, "Probabilistic Estimation-Based Data Mining for Discovering Insurance Risks," *IEEE Intelligent Systems* **14**, No. 6 (November/December 1999), pp. 49–58.
3. R. Kohavi and G. H. John, "The Wrapper Approach," *Fea-*

ture Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, New York (1998), pp. 33–50.

4. J. R. Quinlan, "Learning with Continuous Classes," *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, World Scientific Press, Singapore (1992), pp. 343–348.
5. R. Kohavi, "Scaling Up the Accuracy of Naive Bayes Classifiers: A Decision-Tree Hybrid," *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA (1996), pp. 202–207.
6. G. V. Kass, "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Applied Statistics* **29**, No. 2, 119–127 (1980).
7. G. H. Golub and C. F. Van Loan, *Matrix Computations, Second Edition*, Johns Hopkins University Press, Baltimore, MD (1989).
8. A. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA (1996).
9. R. Natarajan and E. P. D. Pednault, "Segmented Regression Estimators for Massive Data Sets," *Proceedings of the 2nd SIAM International Conference on Data Mining*, SIAM, Philadelphia, PA (2002).
10. A. Agresti, *Categorical Data Analysis*, John Wiley & Sons, Inc., New York (1990).
11. G. John and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA (1995), pp. 338–345.
12. R. Kohavi and M. Sahimi, "Error-Based and Entropy-Based Discretization of Continuous Features," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA (1996), pp. 114–119.
13. C. N. Hsu, J. J. Kuang, and T. T. Wong, "Why Discretization Works for Naive Bayesian Classifiers," *Proceedings of the Seventeenth International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, CA (2000), pp. 399–406.
14. R. Natarajan and E. P. D. Pednault, "Using Simulated Pseudo Data to Speed Up Statistical Predictive Modeling," *Proceedings of the First SIAM International Conference on Data Mining*, SIAM, Philadelphia, PA (2001).
15. G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics* **6**, 461–464 (1978).
16. P. Langley and S. Sage, "Induction of Selective Bayesian Classifiers," *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA (1994), pp. 399–406.
17. S. A. Klugman, H. H. Panjer, and G. E. Wilmot, *Loss Models: From Data to Decisions*, John Wiley & Sons, Inc., New York (1998).
18. E. P. D. Pednault and C. Apte, "Probabilistic Estimation in Data Mining," *Data Mining for Scientific and Engineering Applications*, Kluwer Academic Publishing, New York (2001).

Accepted for publication March 24, 2002.

Chidanand V. Apte *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: apte@us.ibm.com).* Dr. Apte manages the Data Abstraction Research group at the IBM T. J. Watson Research Center. He received his Ph.D. degree in computer science from Rutgers University in 1984. His research interests include knowledge discovery and data mining, applied machine learning and statistical modeling, and business intelligence automation.

Ramesh Natarajan *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: nramesh@us.ibm.com).* Dr. Natarajan is a research staff member in the Data Abstraction Research group at the IBM T. J. Watson Research Center. He received his Ph.D. degree in chemical engineering from the University of Texas at Austin in 1984. He is currently working on statistical data mining applications, modeling algorithms for massive data sets, and database analytic extenders.

Edwin P. D. Pednault *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: pednault@us.ibm.com).* Dr. Pednault is a research staff member in the Data Abstraction Research group at the IBM T. J. Watson Research Center. He received his Ph.D. degree in 1987 at Stanford University, working with Robert C. Moore at SRI International on the mathematical foundations of automatic planning. His current research interests include data mining, statistical learning theory, and reinforcement learning.

Fateh Tipu *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: fateh@watson.ibm.com).* Mr. Tipu is an advisory software engineer in the Data Abstraction Research group at the IBM T. J. Watson Research Center. He received his M.S. degree in electrical engineering from the University of Wisconsin-Madison in 1991. His technical interests include software development, data mining, and computer-aided design (CAD) tools for logic design.