



Sawtooth Software

TECHNICAL PAPER SERIES

CVA/HB Technical Paper

The CVA/HB Technical Paper

Copyright 2002, Sawtooth Software

The CVA/HB Module is a component within SMRT (SMRT stands for Sawtooth Software Market Research Tools) that uses hierarchical Bayes (HB) to estimate part worths for ratings-based full-profile conjoint analysis (CVA) studies. HB considers each individual to be a sample from a population of similar individuals, and “borrows” information from the population in estimating part worths for each respondent. With HB, CVA users can often achieve equivalent results (relative to OLS) using fewer tasks per person and/or fewer total respondents. Precisely how much improvement HB estimation offers over the standard estimation techniques depends on the project.

HB estimation takes considerably longer than OLS or Monotone regression. Computation time will usually vary from about 5 minutes to an hour for most CVA data sets.

Although the CVA/HB module will produce results for very small sample sizes, we caution against using it in those instances. CVA/HB will produce results even with data from a single respondent, but with very small sample sizes it will have difficulty distinguishing between heterogeneity and error. How many respondents are required for robust HB estimation depends on the study design and the nature of the sample. We have seen HB perform well with samples as small as 80 respondents for ratings-based conjoint studies. HB may perform consistently well with even smaller sample sizes, though we know of no series of studies to substantiate that claim.

In any case, we suggest not using HB blindly. It is prudent to design holdout choices within your CVA questionnaires so that you can assess the performance of alternative part worth estimation methods.

Background

The first traditional conjoint analysis applications in the early- to mid-1970s used non-metric estimation or OLS to derive part worths. These techniques served the industry well over the first few decades of conjoint analysis practice. Even so, conjoint researchers have always faced a degrees of freedom problem. We usually find ourselves estimating many parameters (part worths) at the individual level from relatively few observations (conjoint questions). It is often challenging to get respondents to complete many conjoint tasks, so researchers may sacrifice precision in the part worth estimates by reducing the number of conjoint profiles. It is precisely in those cases that HB can be most useful.

HB became available in about the mid 1990s to marketing researchers. HB significantly improves part worth estimates and produces robust results when there are very few or

even no degrees of freedom. Several recent articles (see for example Lenk, *et al.* 1996 and Allenby, *et al.* 1998) have shown that hierarchical Bayes can do a creditable job of estimating individual parameters even when there are *more* parameters than observations per individual.

It is possible using CVA/HB to estimate useful part worths for an individual even though that respondent has answered fewer tasks than parameters to estimate. This can occur if respondents quit the survey early. However, the researcher may choose this approach by design. The researcher might (using CVA's paper-and-pencil mode) randomly assign respondents a subset of a larger CVA design, so that across all respondents each task has roughly equal representation.

The CVA/HB Module estimates a hierarchical random coefficients model using a Monte Carlo Markov Chain algorithm. In the material that follows we describe the hierarchical model and the Bayesian estimation process. It is not necessary to understand the statistics of HB estimation to use this module effectively. The defaults we have provided make it simple for researchers who may not understand the statistics behind HB to run the module with consistently good results.

We at Sawtooth Software are not experts in Bayesian data analysis. In producing this software we have been helped by several sources listed in the References. We have benefited particularly from the materials provided by Professor Greg Allenby in connection with his tutorials at the American Marketing Association's Advanced Research Techniques Forum.

The Basic Idea behind HB

CVA/HB uses Bayes methods to estimate the parameters of a randomized coefficients regression model. In this section we provide a non-technical description of the underlying model and the algorithm used for estimation.

The model underlying CVA/HB is called "hierarchical" because it has two levels. At the upper level, respondents are considered as members of a population of similar individuals. Their part worths are assumed to have a multivariate normal distribution described by a vector of means and a matrix of variances and covariances.

At the lower level, each individual's part worths are assumed to be related to his ratings of the overall product profiles within the conjoint survey by a linear regression model. That is to say, when deciding on his preference for a product profile, he is assumed to consider the various attribute levels that compose that product, and add the value of each level to come up with an overall rating for the product concept. Discrepancies between actual and predicted ratings are assumed to be distributed normally and independently of one another.

Suppose there are N individuals, each of whom has rated conjoint profiles on n attribute levels. If we were to do ordinary regression analysis separately for each respondent, we would be estimating $N \cdot n$ part worths. With the hierarchical model we estimate those same $N \cdot n$ part worths, and we further estimate n mean part worths for the population as well as an $n \times n$ matrix of variances and covariances for the *distribution* of individuals' part worths. Because the hierarchical model requires that we estimate a larger number of parameters, one might expect it would work less well than ordinary regression analysis. However, because each individual is assumed to be drawn from a population of similar individuals, information can be “borrowed” from other individuals in estimating parameters for each one, with the result that estimation is usually enhanced.

The Hierarchical Model

To recapitulate, the HB model is called “hierarchical” because it has two levels.

At the higher level, we assume that individuals' part worths are described by a multivariate normal distribution. Such a distribution is characterized by a vector of means and a matrix of covariances. To make this explicit, we assume individual part worths have the multivariate normal distribution,

$$\beta_i \sim \text{Normal}(\alpha, D)$$

where:

β_i = a vector of part worths for the i th individual

α = a vector of means of the distribution of individuals' part worths

D = a matrix of variances and covariances of the distribution of part worths across individuals

At the lower level we assume that, given an individual's part worths, values of the dependent variable (responses to the conjoint questions) are described by the model:

$$y_{ij} = x_{ij}' \beta_i + e_{ij}$$

where:

y_{ij} = the dependent variable for observation j by respondent i

x_{ij}' = a row vector of values of dummy-coded independent variables for the j th observation for respondent i

e_{ij} = random error term, distributed normally with mean of zero and variance σ^2 .

This model says that individuals have vectors of part worths β_i drawn from a multivariate normal distribution with mean vector α and covariance matrix \mathbf{D} . Individual i 's rating of the conjoint profile for the j th task y_{ij} is normally distributed, with mean equal to the sum of that respondent's part worths characterizing that profile, which is equal to the vector product $x_{ij}' \beta_i$ with variance equal to some value σ^2 .

The parameters to be estimated are the vectors β_i of part worths for each individual, the vector α of means of the distribution of part worths, the matrix \mathbf{D} of the variances and covariances of that distribution, and the scalar σ^2 .

Iterative Estimation of the Parameters

The part worth parameters are estimated using an iterative process that is quite robust. Depending on the random seed you use, you will achieve slightly different part worths from subsequent runs. However, the differences should converge toward zero as the number of iterations increases.

As initial estimates of each parameter we use values of zero or unity. We use zeros as initial estimates of the betas (part worths), alpha, and the covariances, and we use unity as initial estimates of the variances and of sigma. Given those initial values, each iteration consists of these steps:

Using present estimates of the betas, \mathbf{D} , and sigma generate a new estimate of α . We assume α is distributed normally with mean equal to the average of the betas and covariance matrix equal to \mathbf{D} divided by the number of respondents. A new estimate of α is drawn randomly from that distribution.

Using present estimates of the betas, α , and sigma draw a new estimate of \mathbf{D} from the inverse Wishart distribution.

Using present estimates of α , \mathbf{D} , and σ , generate new estimates of the betas. We obtain a new estimate of beta for each individual using a Metropolis Hastings algorithm.

Using present estimates of α , \mathbf{D} , and the betas, generate a new estimate of σ . For this purpose we again use the inverse Wishart distribution.

In each of these steps we re-estimate one set of parameters conditionally, given current values for the other three. This technique is known as "Gibbs sampling," and eventually converges to the correct distributions for each set of parameters. Another name for this procedure is a "Monte Carlo Markov Chain," deriving from the fact that the estimates in each iteration are determined from those of the previous iteration by a constant set of probabilistic transition rules. This Markov property assures that the iterative process converges.

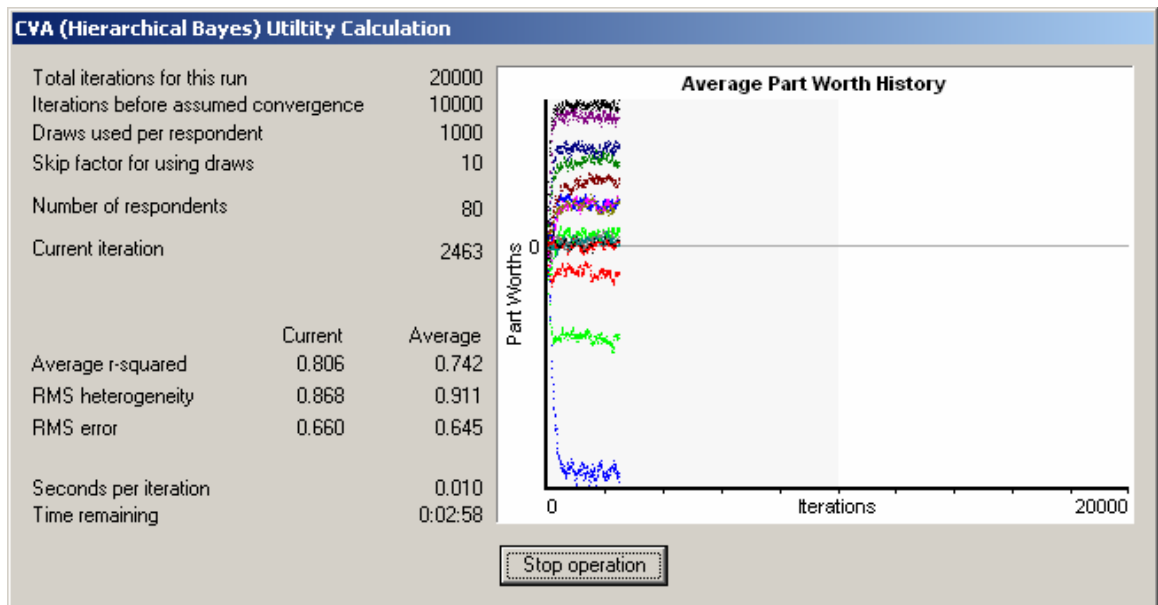
This process is continued for a large number of “burn-in” iterations, typically 2,000 or more. During the burn-in process, the values will vary and trend for a while until convergence is reached, after which the estimates oscillate randomly without demonstrating any remaining trend. During estimation, the software provides a graph of the history of the part worth estimates across iterations. This provides an intuitive visual indication to help the researcher assess convergence.

For many data sets, convergence is achieved within the first 2,000 burn-in iterations. Because some data sets may take longer to converge, we have provided a conservative default of 10,000 iterations prior to assuming convergence. However, there is no guarantee that convergence will occur within the first 10,000 iterations for every data set.

After we are confident of convergence, the process is continued for many further iterations, and numerous “draws” of beta for each individual are used (averaged) to provide point estimates of the part worths. The default is to average 1,000 separate draws of the part worths for each individual. By default, we use only every 10th draw, so 10,000 additional iterations are performed once convergence is assumed.

Monitoring the Computation

While the computation is in progress, information summarizing its current status and history is provided on the screen, like the example below:



This run uses 10,000 initial iterations (the grey background area in the chart), followed by 10,000 further iterations during which each tenth iteration is to be used.

Near the bottom of the screen is an estimate of the total time of 2 minutes and 58 seconds left to complete this computation. The total time for this run is just over 3 minutes.

Note that this is a small problem, with only 80 respondents and 22 conjoint tasks per respondent. Very large problems may require upwards of an hour or more.

The graph of part worths by iteration is the most colorful and prominent aspect of the reporting. This graphic provides insight into whether the part worth values have roughly stabilized and achieved convergence. The horizontal axis charts the iterations, and the vertical axis reflects the part worth estimates. The dummy-coded estimates of average betas for the respondent sample are plotted, with the first level of each attribute omitted. The gray area reflects the burn-in iterations. The final iterations are those after assumed convergence in which we use the draws to develop point estimates of respondents' part worths.

When the plot of part worths portrays a series of essentially horizontal lines with *no remaining trend*, this indicates convergence. With large sample sizes, the part worth history appears tighter with relatively little variation in the part worths from iteration to iteration. But with small sample sizes, the part worth history appears to have much more noise. With small sample sizes, convergence may have occurred even though there may appear to be quite a bit of noise between successive iterations. Again, the critical aspect is whether there is any remaining *trend* in the successive part worth estimates.

HB is probably slower than most other iterative processes with which you may be familiar, and when using different random seeds won't get *exactly* the same answer every time. But it would achieve the same result if a very large number of iterations were used, to as much precision as desired. However, in practice, HB users elect to use fewer iterations than would be required to get the same answer each time.

When the computation ends, a text report of the diagnostics after every 1,000th iteration is shown. These diagnostics are described in the next section.

CVA/HB Diagnostics

We now describe each of the statistics displayed at the left half of the screen. There are two columns for each. In the first column is the actual value for the previous iteration. The second column contains an exponential moving average for each statistic. At each iteration the moving average is updated with the formula:

$$\text{new average} = .01 * (\text{new value}) + .99 * (\text{old average})$$

The moving average is affected by all iterations of the current session, but the most recent iterations are weighted more heavily. The most recent 100 iterations have about 60% influence on the moving averages, and the most recent 500 iteration have about 99% influence. Because the values in the first column tend to jump around quite a lot, the average values are more useful.

The first statistic displayed on the screen during computation is **Average r-squared** which is short for the average squared correlation between each respondent's predicted

and actual data for (in this case) the 22 observations of the dependent variable. The average r-square will be zero initially and improve throughout the early part of the computation. Actually, the present value of 0.806 is about as high as it will get in 20,000 iterations, suggesting that this process may already have converged.

Note that the average r-square from CVA/HB will always be less than the average r-square from OLS estimation. The goal in CVA/HB is not to maximize the r-squared for each individual, but to strike an effective balance between fitting the individual's data and tempering those estimates by population parameters. To the degree that an individual's data are internally consistent with his answers, relatively *less* information is borrowed from the population parameters. For respondents whose data are internally inconsistent, relatively *more* information is borrowed from the population parameters. It usually turns out that the additional information provided from the population parameters actually improves the estimate of the individual's preferences, as evidenced by predictability of holdout observations.

The next statistic is **RMS heterogeneity** which is short for "root mean square heterogeneity." Recall that we estimate the variances and covariances for the part worths among respondents. RMS heterogeneity is just the square root of the average of those variances.

The next statistic is **RMS error** which is a measure of the average error in predicting each respondent's response to the conjoint question from his/her part worths. This is nearly the same thing as the sigma parameter that we estimate, except that this value is computed directly from the data, whereas sigma is estimated by making a normal draw from an estimated distribution.

As iterations progress, all of these statistics change systematically for a while: Average r-square and heterogeneity *increase* at first, while RMS error *decreases* at first. Eventually they level off, thereafter oscillating randomly around their final values. Lack of trend may be taken as evidence of convergence. However, studying the pattern of part worths that is graphed may provide the best visual clue as to convergence. The part worths in the chart presented earlier seem to be leveling off with little trend, suggesting that convergence may have occurred.

Using Constraints

Conjoint studies frequently include product attributes for which almost everyone would be expected to prefer one level to another. However, estimated part worths sometimes turn out not to have those expected orders. This can be a problem, since part worths with the wrong relationships (especially if observed at the summarized group level) are likely to yield nonsense results and can undermine users' confidence.

CVA/HB provides the capability of enforcing constraints on orders of part worths within attributes. The same constraints are applied for all respondents, so constraints should

only be used for attributes that have unambiguous *a priori* preference orders, such as quality, speed, price, etc.

Evidence to date suggests that constraints can be useful when the researcher is primarily interested in individual-level classification or the prediction of individual choices, as measured by hit rates for holdout choice tasks. **However, constraints appear to be less useful, and indeed can be harmful, if the researcher is primarily interested in making aggregate predictions, such as predictions of shares of preference.** Most research is principally concerned with the latter. Another concern is that constraints can bias the apparent importances of constrained attributes in market simulations, relative to unconstrained attributes.

CVA/HB employs a technique called *Simultaneous Tying*. In a paper available on the Sawtooth Software Web site (Johnson, 2000), the author explored different ways of enforcing constraints in the HB context. He found the method of simultaneous tying to perform best among the techniques investigated.

Simultaneous tying features a change of variables between the “upper” and “lower” parts of the HB model. For the upper model, we assume that each individual has a vector of (unconstrained) part worths, with distribution:

$$\beta_i \sim \text{Normal}(\alpha, D)$$

where:

β_i = unconstrained part worths for the *i*th individual

α = means of the distribution of unconstrained part worths

D = variances and covariances of the distribution of unconstrained part worths

For the lower model, we assume each individual has a set of constrained part worths, \mathbf{b}_i where \mathbf{b}_i is obtained by recursively tying each pair of elements of β_i that violate the specified order constraints.

With this model, we consider two sets of part worths for each respondent: unconstrained and constrained. The unconstrained part worths are assumed to be distributed normally in the population, and are used in the upper model. However, the constrained part worths are used in the lower model to evaluate likelihoods.

We speak of “recursively tying” because, if there are several levels within an attribute, tying two values to satisfy one constraint may lead to the violation of another. The algorithm cycles through the constraints repeatedly until they are all satisfied.

When constraints are in force, the estimates of population means and covariances are based on the unconstrained part worths. However, since the constrained part worths are of primary interest, we plot the constrained part worths to the screen. Only the constrained part worths are saved to the utility run for use in the market simulator.

When constraints are in place, measures of fit (average r-squared) are *decreased*. Constraints always decrease the goodness-of-fit for the sample in which estimation is done. This is accepted in the hope that the constrained solution will work better for predictions in new choice situations. Measures of scale (Avg. Variance and Parameter RMS), which are based on unconstrained part worths, will be *increased*.

How Good Are the Results?

Before showing the results of a study demonstrating the benefit of HB, we should point out that CVA/HB is the fifth in a series of Sawtooth Software's products that use HB to estimate individual coefficients. Previous products are CBC/HB and HB-Sum for use in estimating individual part worths in choice studies using a multinomial logit formulation, ACA/HB for estimating individual part worths from ACA data using a linear regression formulation, and HB-Reg for generalized regression problems.

For CBC/HB, ACA/HB, and HB-Reg we have done a performance review, finding HB estimation to be as good as or better than the alternative in every case. That evidence is available in three technical papers that can be downloaded from the sawtoothsoftware.com Web site (Sawtooth Software 1998, 1999a, 1999b).

We now present an example from a real conjoint data set to demonstrate CVA/HB's usefulness. This example is but one of many data sets that have demonstrated HB's superiority relative to traditional methods such as OLS. The margin of superiority reported here is quite typical of other data sets we've seen and those reported in the literature.

This example is from a study reported by Orme *et al.* (1997). Respondents were 80 MBA students from three universities. The subject of the study was personal computers, and nine attributes were studied, each with two or three levels. Each respondent did a full-profile card-sort in which 22 hard-copy cards were sorted into four piles based on preference, and then rated using a 100 point scale. A logit recode method was used for the dependent variable, both for ordinary least squares regression and also by CVA/HB.

In addition, each respondent saw five full-profile holdout choice sets, each containing three product concepts. These choice sets were constructed randomly and uniquely for each respondent. Respondents rank-ordered the concepts in each set, but the results we report here were based only on first choices. (Hit rates in the original paper are based on implied paired comparisons, whereas those reported here are based on triples, and are therefore lower.) We have computed hit rates for predicting holdout choices:

Ordinary Least Squares	72.00%
CVA/HB	74.50%
CVA/HB with Constraints	75.75%

We see that using HB improves results relative to OLS. Since we have not imposed constraints on the first two sets of part worths, this is a fair comparison, and CVA/HB has a 2.5% margin of superiority*. Constraining utilities to conform to rational preference orders further improves the hit rate. The constrained CVA/HB reflects a 3.75% margin of superiority relative to the OLS solution.

(* Note: In the HB-Reg manual, we report that unconstrained HB-Reg yields a 73.50% hit rate for this data set. CVA/HB and HB-Reg use the same procedure, and the difference seen here is due to the random starting point and/or the number of iterations used.)

It is not a given that researchers should impose utility constraints. Whereas hit rates are usually improved by imposing utility constraints, share prediction accuracy is often not improved and sometimes even damaged by constraints. Please see an article (Johnson, 2000) listed in the References that follow for further evidence regarding constraints.

Details of Estimation

Above we attempted to provide an intuitive understanding of the HB estimation process, and to avoid complexity we omitted some details that are provided here.

Gibbs Sampling

The model we wish to estimate has many parameters: an alpha vector of population means, a beta vector for each individual, a **D** matrix of population variances and covariances, and a scalar sigma squared of error variances. Estimating a model with so many parameters is made possible by our ability to decompose the problem into a collection of simpler problems.

As a simple illustration, suppose we have two random variables, x and y for which we want to simulate the joint distribution. We can do so as long as we are able to simulate the distribution of either variable conditionally, given knowledge of the other. The procedure is as follows:

- (1) Draw a random value of x
- (2) Draw a random value of y , given that value of x
- (3) Draw a random value of x , given that value of y
- (4) Repeat steps 2 and 3 many times

The paired values of x and y provide a simulation of the joint distribution of x and y . This approximation of the joint distribution by a series of simpler conditional simulations is known as Gibbs Sampling.

With our model we are interested in the joint distribution of alpha, the betas, **D**, and sigma, so our task is more complicated, but in principle it is like the two-variable

example. We start with arbitrary estimates for each parameter. Then we estimate each of the four types of parameters in turn, conditional on the others.

We do this for a very large number of iterations. Eventually the observed distribution of each parameter converges to its true distribution (assuming the model is stated correctly). Then by continuing the process and saving subsequent draws we can capture the distribution of each parameter. Since our model involves normal distributions, the point estimate for each parameter is simply the mean of those random draws.

It remains to specify how the conditional draws are made in each iteration. For α , \mathbf{D} , and sigma, conventional techniques involving normal distributions are used. For the betas, we use a Metropolis Hastings algorithm.

Random Draw from a Multivariate Normal Distribution

Many times in the iterative process we must draw random vectors from multivariate normal distributions with specified means and covariances. We first describe a procedure for doing this.

Let α be a vector of means of the distribution and \mathbf{D} be its covariance matrix. \mathbf{D} can always be expressed as the product $\mathbf{T}\mathbf{T}'$ where \mathbf{T} is a square, lower-triangular matrix. This is frequently referred to as the Cholesky decomposition of \mathbf{D} .

Consider two column vectors, \mathbf{u} and $\mathbf{v} = \mathbf{T}\mathbf{u}$. Suppose the elements of \mathbf{u} are normal and independently distributed with means of zero and variances of unity. Since for large \mathbf{n} , $1/\mathbf{n} \sum_n \mathbf{u}\mathbf{u}'$ approaches the identity, $1/\mathbf{n} \sum_n \mathbf{v}\mathbf{v}'$ approaches \mathbf{D} as shown below:

$$1/\mathbf{n} \sum_n \mathbf{v}\mathbf{v}' = 1/\mathbf{n} \sum_n \mathbf{T}\mathbf{u}\mathbf{u}'\mathbf{T}' = \mathbf{T} (1/\mathbf{n} \sum_n \mathbf{u}\mathbf{u}')\mathbf{T}' \Rightarrow \mathbf{T}\mathbf{T}' = \mathbf{D}$$

where the symbol \Rightarrow means “approaches.”

Thus, to draw a vector from a multivariate distribution with mean α and covariance matrix \mathbf{D} , we perform a Cholesky decomposition of \mathbf{D} to get \mathbf{T} , and then multiply \mathbf{T} by a vector of \mathbf{u} of independent normal deviates. The vector $\alpha + \mathbf{T}\mathbf{u}$ is normally distributed with mean α and covariance matrix \mathbf{D} .

Estimation of Alpha

If there are \mathbf{n} individuals who are distributed with covariance matrix \mathbf{D} , then their mean, α , is distributed with covariance matrix $1/\mathbf{n} \mathbf{D}$. Using the above procedure, we draw a random vector from the distribution with mean equal to the mean of the current betas, and with covariance matrix $1/\mathbf{n} \mathbf{D}$.

Estimation of \mathbf{D}

Let \mathbf{p} be the number of parameters estimated for each of \mathbf{n} individuals, and let $\mathbf{N} = \mathbf{n} + \mathbf{p}$. Our prior estimate of \mathbf{D} is the identity matrix \mathbf{I} of order \mathbf{p} . We compute a matrix \mathbf{H} which combines the prior information with current estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_i$

$$\mathbf{H} = \mathbf{pI} + \sum_n (\boldsymbol{\alpha} - \boldsymbol{\beta}_i) (\boldsymbol{\alpha} - \boldsymbol{\beta}_i)'$$

We next compute \mathbf{H}^{-1} and the Cholesky decomposition

$$\mathbf{H}^{-1} = \mathbf{T T}'$$

Next we generate \mathbf{N} vectors of independent random values with mean of zero and unit variance, \mathbf{u}_i , multiply each by \mathbf{T} , and accumulate the products:

$$\mathbf{S} = \sum_N (\mathbf{T u}_i) (\mathbf{T u}_i)'$$

Finally, our estimate of \mathbf{D} is equal to \mathbf{S}^{-1} .

Estimation of Sigma

We draw a value of σ^2 from the inverse Wishart distribution in a way similar to the way we draw \mathbf{D} , except that σ^2 is a scalar instead of a matrix.

Let \mathbf{M} be the total number of observations fitted by the model, aggregating over individuals and questions within individual. Let \mathbf{Q} be the total sum of squared differences between actual and predicted answers for all respondents. Let the scalar $\mathbf{c} = \mathbf{p} + \mathbf{Q}$, analogous to \mathbf{H} above. We draw $\mathbf{M} + \mathbf{p}$ random normal values, each with mean of zero and standard deviation of unity, multiply each by $1/\sqrt{\mathbf{c}}$, and accumulate their sum of squares, analogous to \mathbf{S} above. Our estimate of σ^2 is the reciprocal of that sum of squares.

Estimation of Betas Using a Metropolis Hastings Algorithm

We now describe the procedure used to draw each new set of betas, done for each respondent in turn. We use the symbol $\boldsymbol{\beta}_o$ (for “beta old”) to indicate the previous iteration’s estimation of an individual’s part worths. We generate a trial value for the new estimate, which we shall indicate as $\boldsymbol{\beta}_n$ (for “beta new”), and then test whether it represents an improvement. If so, we accept it as our next estimate. If not, we accept or reject it with probability depending on how much worse it is than the previous estimate.

To get $\boldsymbol{\beta}_n$ we draw a random vector \mathbf{d} of “differences” from a distribution with mean of zero and covariance matrix proportional to \mathbf{D} , and let $\boldsymbol{\beta}_n = \boldsymbol{\beta}_o + \mathbf{d}$. We regard $\boldsymbol{\beta}_n$ as a candidate to replace $\boldsymbol{\beta}_o$ if it has sufficiently high posterior probability. We evaluate each posterior probability as the product of its density (the prior) and its likelihood.

We first calculate the relative probability of the data, or “likelihood,” given each candidate, β_o and β_n . We do not calculate the actual probabilities, but rather simpler values that are proportional to those probabilities. We first compute the sum of squared differences between the actual answers and our predictions of them, given each set of betas. The two likelihoods are proportional to the respective quantities for β_o and β_n :

$$\exp[-1/2 (\text{sum of squared differences})/ \sigma^2].$$

Call the resulting values p_o and p_n , respectively.

We also calculate the relative density of the distribution of the betas corresponding to β_o and β_n , given current estimates of parameters α , D , and σ . Again, we do not compute actual probabilities, but rather simpler values that are proportional to the desired probabilities. This is done by evaluating the following expression for each candidate:

$$\exp[-1/2*(\beta - \alpha)' D^{-1} (\beta - \alpha)]$$

Call the resulting values d_o and d_n , respectively. Finally we then calculate the ratio:

$$r = p_n d_n / p_o d_o$$

From Bayes’ theorem, the posterior probabilities are proportional to the product of the likelihoods times the priors. The values p_n and p_o are proportional to the likelihoods of the data given parameter estimates respectively. The values d_n and d_o are proportional to the probabilities of drawing those values of β_n and β_o , respectively, from the distribution of betas, and play the role of priors. Therefore, r is the ratio of posterior probabilities of β_n and β_o , given current estimates of α , D , and σ , as well as information from the data.

If r is greater than or equal to unity, β_n has posterior probability greater than or equal to that of β_o , and we accept β_n as our next estimate of beta for that individual. If r is less than unity, then β_n has posterior probability less than that of β_o . In that case we use a random process to decide whether to accept β_n or retain β_o for at least one more iteration. We accept β_n with probability equal to r .

As can be seen, two influences are at work in deciding whether to accept the new estimate of beta. If it fits the data better than the old estimate, then p_n will be larger than p_o , which will tend to produce a larger ratio. However, the relative densities of the two candidates also enter into the computation, and if one of them has a higher density with respect to the current estimates of α and D , and σ , then that candidate has an advantage.

If the densities were *not* considered, then betas would be chosen solely to maximize likelihoods. This would be similar to estimating for each individual separately, and eventually the betas for each individual would converge to a distribution that fits his/her data, without respect to any higher-level distribution. However, since densities are considered, and estimates of the higher-level distribution change with each iteration, there is considerable variation from iteration to iteration. Even after the process has

converged, successive estimations of the betas are still quite different from one another. Those differences contain information about the amount of random variation in each individual's betas that best characterizes them.

We mentioned that the vector \mathbf{d} of differences is drawn from a distribution with mean of zero and covariance matrix proportional to \mathbf{D} , but we did not specify the proportionality factor. In the literature the distribution from which \mathbf{d} is chosen is called the “jumping distribution,” because it determines the size of the random jump from β_0 to β_n . This scale factor must be chosen well because the speed of convergence depends on it. Jumps that are too large are unlikely to be accepted, and those that are too small will cause slow convergence.

Gelman, Carlin, Stern, and Rubin (p 335) state: “A Metropolis algorithm can also be characterized by the proportion of jumps that are accepted. For the multivariate normal distribution, the optimal jumping rule has acceptance rate around 0.44 in one dimension, declining to about 0.23 in high dimensions ... This result suggests an *adaptive* simulation algorithm.”

We employ an adaptive algorithm to adjust the average jump size, attempting to keep the acceptance rate near 0.30. The proportionality factor is arbitrarily set at 0.1 initially. For each iteration we count the proportion of respondents for whom β_n is accepted. If that proportion is less than 0.3, we reduce the average jump size by a tenth of one percent. If that proportion is greater than 0.3, we increase the average jump size by a tenth of one percent. As a result, the average acceptance rate is kept close to the target of 0.30.

Readers with solid statistical background who are interested in further information about the Metropolis Hastings Algorithm may find the article by Chib and Greenberg (1995) useful.

References

- Allenby, G. M., Arora, N., and Ginter, J. L. (1998) "On the Heterogeneity of Demand," *Journal of Marketing Research*, 35, (August) 384-389.
- Allenby, G. M. and Ginter, J. L. (1995) "Using Extremes to Design Products and Segment Markets," *Journal of Marketing Research*, 32, (November) 392-403.
- Chib, S. and Greenberg, E. (1995) "Understanding the Metropolis-Hastings Algorithm," *American Statistician*, 49, (November) 327-335.
- Gelman, A., Carlin, J. B., Stern H. S. and Rubin, D. B. (1995) "Bayesian Data Analysis," Chapman & Hall, Suffolk.
- Green, P. E., Krieger, A. M., and Agarwal, M. K. (1991) "Adaptive Conjoint Analysis: Some Caveats and Suggestions," *Journal of Marketing Research*, 28 (May), 215-22.
- Huber, J., Orme B. K., and Miller, R. (1999) "Dealing with Product Similarity in Conjoint Simulations," *Sawtooth Software Conference Proceedings*, Sawtooth Software, Sequim.
- Johnson, R. M. (2000), "Monotonicity Constraints in Conjoint Analysis with Hierarchical Bayes," Technical Paper available at www.sawtoothsoftware.com.
- Lenk, P. J., DeSarbo, W. S., Green P. E. and Young, M. R. (1996) "Hierarchical Bayes Conjoint Analysis: Recovery of Partworth Heterogeneity from Reduced Experimental Designs," *Marketing Science*, 15, 173-191.
- Orme, B. K., Alpert, M. I. & Christensen, E. (1997) "Assessing the Validity of Conjoint Analysis – Continued," *Sawtooth Software Conference Proceedings*, Sawtooth Software, Sequim.
- Rossi, P.E., Zvi, G, and Allenby, G.M. (1999) "Overcoming Scale Usage Heterogeneity: A Bayesian Hierarchical Approach." Paper read at ART Forum, American Marketing Association.
- Sawtooth Software (1998) "The CBC/HB Module for Hierarchical Bayes Estimation," Technical Paper available at sawtoothsoftware.com.
- Sawtooth Software (1999a) "The ACA/HB Module for Hierarchical Bayes Estimation," Technical Paper available at sawtoothsoftware.com.
- Sawtooth Software (1999b) "HB-Reg: Hierarchical Bayes Regression Analysis Technical Paper," Technical Paper available at sawtoothsoftware.com.