



Sawtooth Software

TECHNICAL PAPER SERIES

**HB-Reg: Hierarchical Bayes
Regression Analysis (v.3)
Technical Paper**

HB-Reg for Hierarchical Bayes Regression v3

Sawtooth Software, Inc.
December, 2003

Introduction

In the analysis of marketing research data, there are many occasions when the researcher has a sample of respondents, stores, or other experimental units, and wishes to estimate separate regression coefficients for each unit.

Consider three examples:

1. In full-profile conjoint analysis, survey respondents give preference ratings for hypothetical product concepts. Regression analysis is often used, where the independent variables are columns of a “design matrix” describing the concepts, and the dependent variable consists of preference ratings. The researcher often wants to estimate regression coefficients that will be interpreted as “part-worths.” Because respondents are expected to differ in their values, the researcher wants to produce estimates for each respondent individually.
2. Survey respondents in a customer satisfaction study provide ratings of several companies. Some ratings are on “explanatory” variables, such as customer service, product durability, convenience of use, etc. Other ratings are more general, such as overall satisfaction with the companies’ products. One goal of the study is to infer the relative importance of each explanatory factor in determining overall satisfaction. Because respondents may differ, the researcher wants to produce estimates for each respondent individually.
3. During a pricing experiment in grocery stores, the prices of several products are varied systematically in different time periods, and sales of each product are measured with scanner data. The independent variables are product prices and other factors such as the presence of displays, coupons, and newspaper features. The dependent variables are product sales. Because it is believed that customers of different stores may behave differently, the researcher wants to estimate price effects and cross-effects for each store individually.

In each situation, separate regression estimates are desired for each individual (respondent or store). However, in each case there is likely to be a degrees-of-freedom problem, with many parameters to be estimated for each individual, but relatively few observations per individual.

In the past, researchers have often tried to handle this problem by ignoring heterogeneity among individuals, pooling all the data, and estimating a single set of regression coefficients that describe the “average” individual. However, an alternative solution has recently become available to marketing researchers with the introduction of “hierarchical Bayes” (HB) methods. Several articles (see for example Lenk, *et al.* 1996 and Allenby, *et al.* 1998) have shown that hierarchical Bayes estimation can do a creditable job of estimating individual parameters even when there are *more* parameters than observations per individual. This is done by considering each individual to be a sample from a population of similar individuals, and “borrowing” information from other individuals in the estimation for each one.

HB-Reg is appropriate for the situations described above. It 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.

Although HB methods are often computationally arduous, our software has the advantage of being written in compiled code (C++ and C#). Compiled code programs are usually considerably faster than those written in higher-level languages such as Gauss.

This is one of four HB products that Sawtooth Software has provided. The other three, CBC/HB, ACA/HB, and CVA/HB are specialized applications for use with data generated by Sawtooth Software's CBC, ACA, and CVA conjoint analysis software. Because they are used in narrowly defined contexts, they require relatively little data processing on the part of the user. HB-Reg is more general, being applicable to data from a variety of sources. To permit this generality we must assume that the user is capable of arranging input data in a format acceptable to HB-Reg.

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. For many of the technical improvements in version 3, we have benefited from direction provided by Peter Lenk, of the University of Michigan.

Capacity Limitations and Hardware Recommendations

The "full-size" version of HB-Reg can handle a maximum of 1000 independent variables and up to 1000 observations per individual, with no limit to the number of individuals. We also offer a lower-priced "Limited" version so users can get acquainted with the software inexpensively, in which the maximum number of independent variables is 20, and the maximum number of observations per individual is 90.

HB-Reg benefits from a fast computer and a generous amount of storage space. HB-Reg estimates individual coefficients by doing many thousands of Monte Carlo simulation iterations. In a typical analysis, you might first do 10,000 iterations just to achieve convergence of the estimation process. Then you might save the results of 10,000 subsequent simulations to be used for estimating individual coefficients.

What's New in HB-Reg Version 3?

The most obvious improvement for version 3 is the new Windows interface. The parameters and settings that previously were defined using the text-only file `hbreg.par` are now specified through the Windows interface. With HB-Reg version 3, you can now use project files (long file names are acceptable) stored in any folder on your system. We also provide a graphical representation of the average estimates of the parameters during estimation, which aids in determining convergence. The speed of computation for most problems you may run is also increased slightly compared to version 2. In addition to the cosmetic changes, we've also included a number of new features.

1. We have added a new .CSV file format for storing the final data file containing point estimates of betas. This .CSV file can be directly opened with ExcelTM, and each respondent record is

contained on a single line (row) in the file, with data labels provided at the top of the file. Some users may find this file format more convenient than the .bet file.

2. Users may now specify the prior variance and degrees of freedom for the prior covariance matrix. We recently implemented similar features in our CBC/HB software, but this flexibility is probably even more valuable for HB-Reg. With our other conjoint-based HB systems, we could make reasonable assumptions about the relative scaling of the dependent variable and the conjoint part worth coefficients. In contrast, HB-Reg is a generalized system for analyzing many types of data, and we cannot always make general assumptions regarding the scaling of the variables and the related measures of variance. The previous version of HB-Reg assumed a prior variance of unity for betas. If there was relatively sparse information available at the respondent level (which is often the case) and if the data were scaled much differently from the prior variance assumption, the estimation could be sub-optimal or even incorrect. We have also come to recognize that for data sets that are extremely sparse within the unit of analysis, even if the prior default variance is reasonable, we still can face problems for proper estimation. In the previous HB-Reg software, the setting for the degrees of freedom for the prior covariance matrix sometimes resulted in overfitting in such extreme situations. In these cases, too much emphasis was placed on obtaining good individual-level fit to estimation observations, rather than relying on information provided by the population parameters.

Based on interactions with Peter Lenk of the University of Michigan, we have modified two aspects related to the prior covariance matrix. The user now has the option to tune the prior variance (its default in HB-Reg is still 1.0) and we have increased the degrees of freedom for the prior covariance matrix by the constant 5 (the user now may also modify this value).

Based on our testing with both real and synthetic data sets, these modifications result in equally good estimation for well-behaved HB-Reg data sets as with v2 of the software and, after appropriate tuning of the prior variance and degrees of freedom, better estimation for data sets with more extreme scaling or those which push the limits of individual-level HB estimation. When using very few observations relative to the number of parameters to be estimated, the tendency toward overfitting can be significantly reduced and perhaps eliminated by choosing appropriate priors (usually by reducing the variance and optionally increasing the degrees of freedom for the prior covariance matrix).

3. Advanced users may override the default prior covariance matrix with a user-supplied prior covariance matrix. To override the default prior covariance matrix in the software, the user creates a text-only file containing the prior covariance matrix named STUDYNAME.mtrx and saves it within the same folder as contains the other study-related files.

4. In previous versions of HB-Reg, a seed for the random number generator was drawn based on the computer's clock. That is still the default, but users can now specify a specific seed to use so that results are repeatable.

5. Users can now constrain parameters, either with respect to one another (e.g. variable 2 must be greater than or equal to variable 5) or with respect to zero (e.g. variable 3 must be greater than or equal to 0). The method of applying parameter constraints is called "simultaneous tying" and is the same technique as used in our CBC/HB software. Details are described in this documentation and also in a technical paper called "Monotonicity Constraints in Choice-Based Conjoint with Hierarchical Bayes" available at www.sawtoothsoftware.com.

6. The “draws” no longer need to be saved to a file. Most users do not make use of the hundreds or thousands of separate “draws” available for each individual (saved to a .dra file). Users can now specify that the draws should not be saved, and in that case point estimates are developed “on the fly.” This saves disk space and time for computation, since the many draws do not need to be processed in the final step to produce point estimates.

The Basic Idea behind HB-Reg

HB-Reg uses Bayes methods to estimate the parameters of a randomized coefficients regression model. In this chapter we provide a non-technical description of the underlying model and the algorithm used for estimation. Further details are provided in the Appendix. To make matters clearer, we focus on one of the earlier examples, which we repeat here:

Survey respondents in a customer satisfaction study provide ratings of several companies. Some ratings are on relatively specific “explanatory” variables, such as customer service, product durability, convenience of use, etc. Other ratings are more general, such as overall satisfaction with the companies’ products. One goal of the study is to infer the relative importance of each explanatory factor in determining overall satisfaction. Because respondents may differ in what is important, the researcher wants to produce unique estimates for each respondent.

Regression analysis is a statistical technique that seeks a set of “weights” which, when applied to explanatory variables, predict or explain another variable of interest. The explanatory variables are often called “independent variables” and the variable we want to explain or predict is often called the “dependent variable.”

The weights are generally called “regression coefficients,” though in this context we might call them “part-worths” or “importance weights.” The regression equation is usually of the form

$$y = x_1 b_1 + x_2 b_2 + \dots + x_n b_n + e$$

Here the variable y is a respondent’s rating of a company (or product or service) on “overall satisfaction.” The x variables are that same respondent’s ratings of the same company on other variables that we believe may be important in determining how satisfied that customer is with that company. (In many contexts, we include an “intercept,” which is an estimate of y if all the independent variables had values of zero. That can be accommodated in this formula by letting one of the x ’s have constant values of unity.) The symbol “ e ” on the right side of the equation stands for “error,” and represents our inability to predict y with complete accuracy by adding up weighted sums of the x ’s. If the respondent has rated several companies, we have several values of y , several corresponding sets of x ’s, and several corresponding values of e .

Under certain conditions, regression analysis can provide estimates of the b ’s for this respondent.

- Usually we assume the errors are random, have mean of zero, and are independent of the x ’s. If we are using “least squares” regression, we also assume the sum of squared errors is as small as possible.

- The respondent must have rated at least as many companies as the number of variables for which we seek importance weights. Another way of saying this is that the number of unknowns (the b 's) must be no larger than the number of data points (the y 's).
- The respondent's ratings on different variables (the x 's) must have some degree of independence from one another. For example, if there were two variables for which each company got equal ratings, we would have no way of deciding how importance should be allocated among those two variables.

Unfortunately, researchers doing customer satisfaction studies usually find that none of these conditions is satisfied.

The first condition is seldom satisfied because survey respondents tend to bunch their ratings at the top ends of the scale, so random variability tends to be smaller for highly rated products. Researchers have tried for many years to overcome this problem, one of the most recent attempts being Rossi *et al.* (1999). Although this is an important problem, its complete solution will probably require new methods of data collection that encourage respondents to discriminate more finely among companies they like.

The second condition is an even more serious impediment to the estimation of individual importance weights. The persons funding the research are often interested in a large number of possible explanatory variables, but it is usually not possible for each respondent to provide knowledgeable ratings of a large number of companies. Respondents get bored when asked to rate many companies on many attributes, and the quality of their output suffers. Also, many respondents are familiar with only a few companies, so their ratings of other companies contain little real information. Fortunately, HB methods can provide significant help in overcoming this problem. Unlike conventional regression analysis, HB-Reg can provide reasonable estimates for each respondent's importance weights, even when each respondent rates fewer companies than the number of variables for which weights are to be estimated. The ability of HB-Reg to provide reasonable individual-level estimates in this case may be enhanced by constraining the signs of the coefficients to be positive or negative.

Failure to satisfy the third condition is also a serious problem. Often respondents fail to distinguish among variables as precisely as researchers would like. The researcher may want to learn whether "reliability" is more or less important than "durability," but if those words mean nearly the same thing to a respondent, he or she is likely to produce identical ratings on each variable. This failure is known as "colinearity," and describes the condition in which ratings on one variable are predictable from ratings on others. For most efficient estimation, we would like a respondent's ratings on several variables to be completely independent of one another, but that is almost never true. Fortunately, HB methods can also provide significant help in overcoming colinearity.

The model underlying HB-Reg is called "hierarchical" because it has two levels. At the upper level, respondents are considered as members of a population of similar individuals. Their importance weights 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 importance weights are assumed to be related to his ratings by the simple equation above. That is to say, when deciding on his level of overall satisfaction with a company, he is assumed to consider several explanatory variables, multiplying his rating of that company on each variable by an importance weight and adding up those products.

Suppose there are N individuals, each of whom has rated products on n explanatory variables. If we were to do ordinary regression analysis separately for each respondent, we would be estimating $N \cdot n$ importance weights. With the hierarchical model we also estimate $N \cdot n$ importance weights, and we further estimate n mean importance weights for the population as well as an $n \times n$ matrix of variances and covariances for the *distribution* of individuals' importance weights. 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.

In particular, it becomes possible to estimate individual parameters even though each respondent has rated only a small number of products, and even though there may be considerable colinearity in a respondent's ratings on explanatory variables. For example, suppose an individual has given similar ratings to two variables, such as reliability and durability, so that an ordinary regression analysis might be unable to allocate importance between them. But since we assume this respondent is drawn from a distribution with known characteristics, we can use information about that distribution to resolve ambiguities for each individual.

The Hierarchical Model

To recapitulate, the model used by HB-Reg is called "hierarchical" because it has two levels.

At the higher level, we assume that individuals' regression weights 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 regression weights have the multivariate normal distribution,

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

where:

β_i = a vector of regression (or importance) weights for the i th individual,

α = a vector of means of the distribution of individuals' regression weights,

D = a matrix of variances and covariances of the distribution of regression weights across individuals.

At the lower level we assume that, given an individual's regression weights, values of the dependent variable 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 ,

\mathbf{x}_{ij}' = a row vector of values of independent variables for the j th observation for respondent i ,

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

Continuing the customer satisfaction example, this model says that individuals have vectors of importance weights β_i drawn from a multivariate normal distribution with mean vector α and covariance matrix \mathbf{D} . Individual i 's rating of overall satisfaction with the j th company y_{ij} is normally distributed, with mean equal to the sum of that respondent's ratings on the independent variables, each weighted by the corresponding importance coefficient (which is equal to the vector product $\mathbf{x}_{ij}' \beta_i$) and 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 regression weights, the matrix \mathbf{D} of the variances and covariances of that distribution, and the scalar σ^2 .

Iterative Estimation of the Parameters

The parameters are estimated by an iterative process which is quite robust, and for which final results do not depend on starting values. As initial estimates of each parameter we use values of zero or unity. We use zeros as initial estimates of the betas, 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 (further details are provided in the Appendix):

Using present estimates of the betas and \mathbf{D} , 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 and α , 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 use different methods for doing this, depending on the format of the input data. If every respondent has the same values for his explanatory variables (as is frequently the case in full-profile conjoint analysis) we use a "normal draw" procedure to get a new estimate of beta for each individual. That is to say, we draw a random vector from the distribution characterizing his regression weights. If every respondent can have unique values for his explanatory variables (as is usually the case in customer satisfaction research) 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 iterations, typically 10,000 or more. After we are confident of convergence, the process is continued for many further iterations, and the actual draws of beta for each individual as well as estimates of α , \mathbf{D} , and σ are saved to the hard disk. The final estimates of regression coefficients for each individual, and also of α , \mathbf{D} , and σ , are obtained by averaging those values that have been saved.

Parameter Constraints

There are modeling situations in which the researcher knows ahead of time that certain parameters must not be less in magnitude than others. As one example, 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 slopes are likely to yield nonsense results and can undermine users' confidence. As another example, a model may include company ratings on different aspects of service or product quality, where higher ratings imply greater satisfaction. If those variables are used to predict some overall outcome such as likelihood of purchase or overall rating for the company, one should expect all betas to be positive. Due to the sparse nature of the data and random noise, many of the individual-level betas may be negative, but the researcher may want to constrain them to be positive.

HB-Reg provides the capability of enforcing constraints between two parameters, or sign constraints for individual parameters. The same constraints are applied for all respondents, so constraints should only be used for variables that have unambiguous a-priori orders or signs.

Evidence to date suggests that constraints can be useful when the researcher is primarily interested in the accuracy of individual models (such as for classification or estimating "hit rates"). 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 choices within choice simulators. The use of constraints can also get in the way of hypothesis testing, where the researcher may require the unconstrained distribution of parameters.

In a paper available on the Sawtooth Software Web site (Johnson, 2000) we explored several ways of enforcing constraints with HB among part-worths in the conjoint analysis context. Realizing that most conjoint analysis users are probably interested in predicting individual choices as well as aggregate shares, we examined the success of each method with respect to both hit rates and share predictions. One of the methods seemed consistently successful was referred to in that paper as "Simultaneous Tying." We have implemented that method in HB-Reg. We call it "Simultaneous" because it applies constraints during estimation, so the presence of the constraints affects the estimated values.

Simultaneous Tying

This method 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) betas, with distribution:

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

where:

- β_i = unconstrained betas for the i th individual,
- α = means of the distribution of unconstrained betas,
- D = variances and covariances of the distribution of unconstrained betas.

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

We speak of “recursively tying” because, if there are several variables involved in constraints, 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 betas. However, since the constrained betas are of primary interest, we plot the constrained betas to the screen. Only the constrained betas are saved to the .bet and .csv files.

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 out-of-sample situations.

Using the Software

The first step in setting up an HB-Reg project is to create a text-only data file with the appropriate format (as described in this chapter). This file should have any Windows-acceptable name and extension .DAT.

HB-Reg distinguishes between two types of data files:

Type 1: data files for which all individuals have the same values for the independent variables. An example of a type 1 data file would be a paper-and-pencil conjoint analysis study in which every respondent is shown the same concepts. Since every respondent has the same independent variable values, that information can be provided once at the top of the data file and need not be repeated for each respondent. We also assume that with this type of data you will not want to estimate regressions for subsets of variables, but rather will want to include all of the independent variables.

Type 2: data files for which each individual can have different values for the independent variables. In fact, with such data, different individuals can even have different numbers of observations (or, in our example, rate different numbers of companies). An example of a type 2 data file would be a customer satisfaction study in which different individuals rate different companies, and since the independent variable values are ratings provided by the survey respondent, they will be different for each individual. With this type of data we presume you may want to experiment, trying out different subsets of independent variables, and we provide the capability of doing so without reformatting the input data.

The first decision you must make is which way to format your data. Even though you may have type 1 data, you may wish to format it like type 2 data so that you are free to include different subsets of independent variables in the model. If your data are truly of type 1, you should get equivalent results formatting it either way, although the computation will be somewhat slower if it is formatted as type 2 data, because the data file will be larger.

Format for Type 1 Data:

When every respondent has the same design matrix, that information only needs to be included once

at the top of the file. Below are the first few lines of the SAMPLE1.DAT file included with the software. (This file contains artificial data, created to have certain properties which will be described later when discussing HB-Reg's output.)

```
10 5
0 1 1 1 -1
-1 -1 1 1 -1
1 1 -1 0 -1
1 1 0 1 -1
1 -1 1 0 -1
-1 0 1 1 1
1 -1 1 1 0
-1 0 0 0 0
-1 -1 0 1 -1
0 1 -1 -1 0
1 -2.53 -6.56 1.68 0.39 2.67 -6.27 -1.17 -3.75 -7.12 3.02
2 1.28 -3.44 8.21 5.72 2.16 -9.40 -3.26 -1.27 -0.04 4.86
3 -1.05 -1.12 2.41 1.11 5.53 -6.18 -1.38 -1.85 -1.82 1.65
4 2.28 0.32 4.74 5.99 7.87 -6.19 3.70 -2.66 -2.38 -1.04
5 5.12 0.12 2.69 6.64 5.00 -1.23 3.75 -0.60 -0.49 -0.17
(Etc. for as many respondents as needed)
```

All values are numeric and separated by blank spaces or tabs. We have formatted this file with each section starting on a new line, but that is just for ease of reading. You could put all the values on a single line, or they could be distributed among several lines in whatever way is most convenient for you.

The file starts with two integers that are the number of observations for each individual (10 in this case) and the number of independent variables (5 in this case).

Next comes the design matrix, in order of "variable within observation." The first row of the example gives values of the 5 independent variables for the first observation. This was a paired-comparison conjoint study (two product concepts shown per task), so design variables have values of plus or minus one and zero. (See the section later in this chapter on coding of independent variables.)

A line follows for each respondent which contains an identification number, followed by data. In this case the data are numbers with two decimal digits. Although we have shown data for only 5 respondents, these lines are followed by similar lines for other respondents.

Format for Type 2 Data:

When each respondent can have different values for the independent variables, then they must be included for every respondent. Below are the first few lines of the SAMPLE2.DAT file included with the software.

```
1      10
4.00  9.00  6.00  3.00  9.00 -23.42
4.00  9.00  6.00  9.00  8.00 -28.80
7.00  8.00  4.00  6.00  2.00 -13.17
5.00  4.00  4.00  7.00  6.00 -17.36
4.00  1.00  2.00  4.00  6.00 -11.72
2.00  7.00  3.00  4.00  1.00 -11.57
4.00  3.00  9.00  4.00  9.00 -16.66
2.00  8.00  6.00  3.00  2.00 -12.83
4.00  9.00  1.00  5.00  5.00 -22.14
1.00  1.00  2.00  1.00  9.00 -18.58
2      10
4.00  9.00  3.00  7.00  1.00  5.50
(Etc. for nine more 'observations' for this case, and as many subsequent
cases as needed.)
```

All values are numeric and separated by blank spaces or tabs. The file starts with an identification number for the first respondent (1), followed by a count of the number of observations for that respondent (10). We do not specify how many variables are defined per observation here, but will specify that later within the HB-Reg interface.

Like the sample1.DAT file, this file was created artificially to demonstrate certain properties.

Following the data for the first respondent we begin the record for the second respondent, with an identification number and a count of the number of observations to follow. That count happens to be 10 again, but it is not necessary for every respondent to contribute the same number of observations.

User-Specified Prior Covariance Matrix

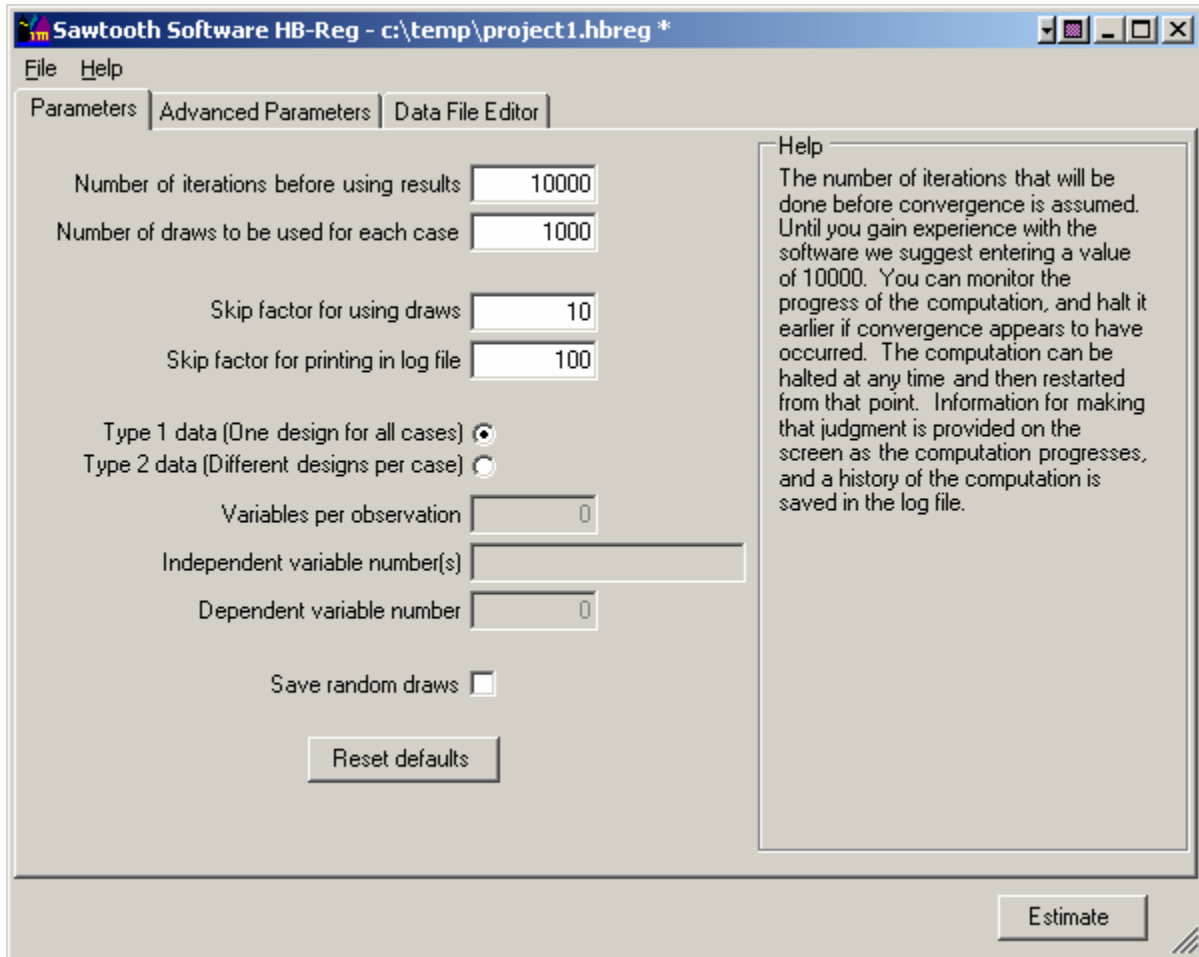
Advanced users can provide a user-specified prior covariance matrix, by creating a text-only, space-delimited file named STUDYNAME.mtrx containing an NxN matrix of values, where N is equal to the number of parameters to be estimated. The user-specified prior covariance matrix overrides the default covariance matrix as well as the prior variance setting provided under the *Advanced Parameters* tab.

By default, HB-Reg uses an identity matrix for the prior covariance matrix (1s across the diagonal, and 0s in all off-diagonal elements). This indicates a prior variance of 1 for all parameters.

There is no provision in HB-Reg for specifying prior means for parameters.

Setting Estimation Parameters

When you open or create a new data file, the following dialog is shown:



Number of iterations before using results is the number of iterations that will be done before convergence is assumed. Until you gain experience with the software we suggest entering a value of 10,000 (but without the comma). You can monitor the progress of the computation, and halt it earlier if convergence appears to have occurred. The computation can be halted at any time and then restarted from that point. Information for making that judgment is provided on the screen as the computation progresses, and a history of the computation is saved in a file named *studyname.log*.

Number of draws to be used for each case is the number of iterations for which results will be available for analysis. There is an option lower on the menu about whether you will save iterations (draws) or summarize them “on the fly.” Let’s assume for a moment that you decide to save them to disk. There is a trade-off between the benefit of statistical precision and the time required for estimation and potential difficulty of dealing with very large files. The default is to use results for 1,000 iterations. Slightly better results would be achieved by using information for more iterations. If you are not saving the draws to disk, the only cost for using more draws per

individual is time for computation. But, if you are saving the draws to disk, this can result in a very large file.

Skip factor for using draws is a way of compensating for the fact that successive draws of the betas are not independent. A skip factor of k means that results will only be used for each k th iteration. Recall that only about 30% of the “new” candidates for beta are accepted in any iteration; for the other 70% of respondents, beta is the same for two successive iterations. This dependence among draws decreases the precision of inferences made from them, such as their averages. If you are saving draws to disk, because file size can become critical, it makes sense to increase the independence of the draws saved by conducting several iterations between each two for which results are saved. If 1,000 draws are to be saved for each respondent and the skip factor is 10, then 10,000 iterations will be required to save those 1,000 draws.

Skip factor for printing in log file controls the amount of detail that is saved in the *studyname.log* file to record the history of the iterations. Several descriptive statistics for each iteration are printed in the log file. But since there may be 20,000 or more iterations altogether, you will probably not want to record every one of them. We suggest recording every hundredth. In the case of a very large number of iterations, you might want to record only every thousandth.

Type 1 Data is checked when a single design matrix is used for all respondents (such as with a fixed, traditional conjoint design). When Type 1 Data are used, all independent variables in the design matrix are estimated. See the previous chapter for details of file layout.

Type 2 Data is checked when each respondent can receive unique independent variables. See the previous chapter for details of file layout.

Variables per observation. If using Type 2 data, you must specify how many total variables are available in the data file per “observation.” This includes all potential independent variables plus the dependent variable.

Independent variable number(s). If using Type 2 data, if all independent variables are to be used in estimation, you can leave this field blank. If you want to include a subset of the independent variables in your model, you can specify them in this field by listing the variable numbers, each separated by spaces. For example, to use independent variables 1, 2, 3, and 6 in the model, you can type **1 2 3 6**.

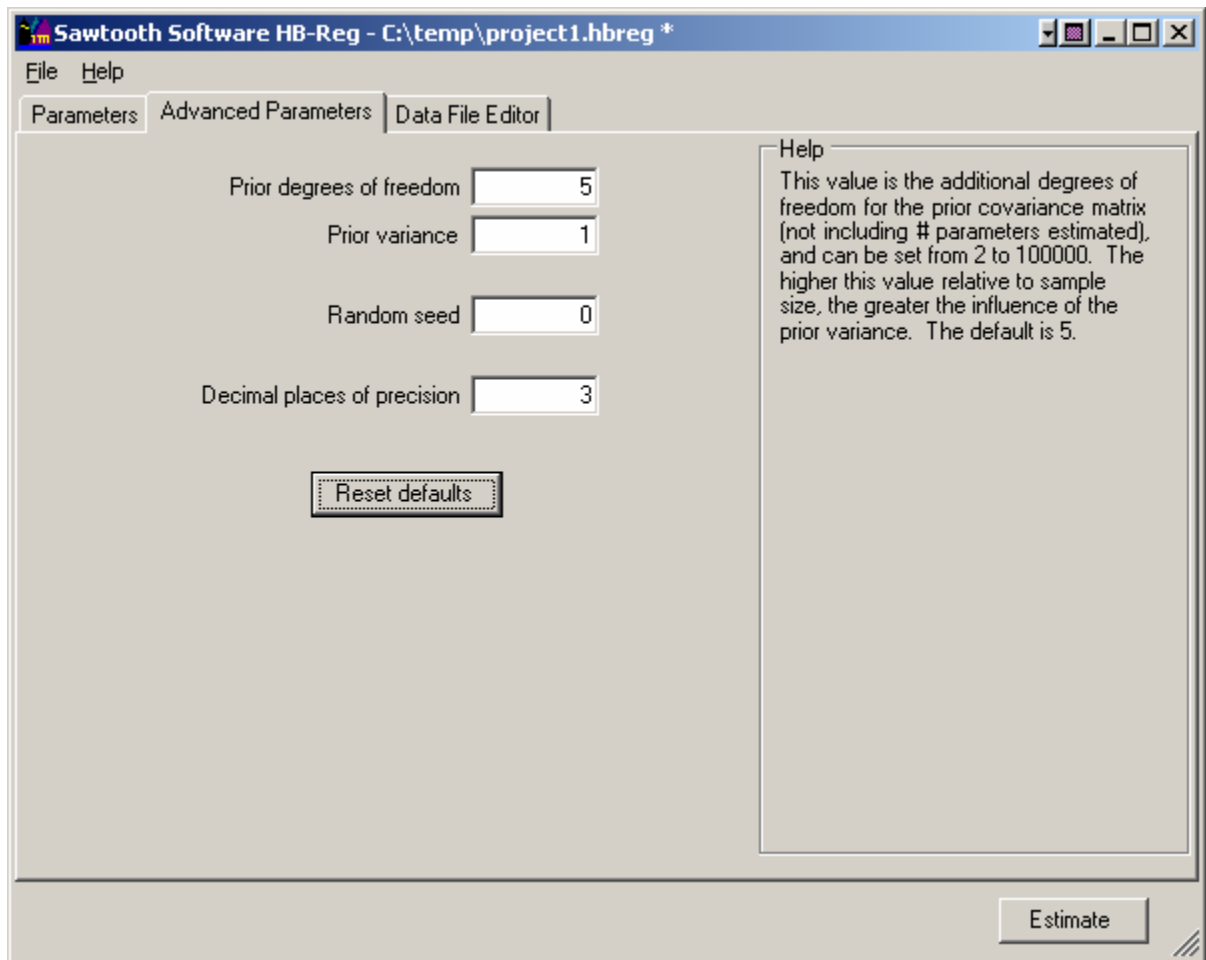
Dependent variable number. If using Type 2 data, you must specify which variable number is the dependent variable. For example, if the dependent variable is the 20th variable in the file, you should specify **20**.

Save random draws: Check this box to save random draws to disk, in which case point estimates of respondents’ betas are automatically computed by averaging each respondent’s draws after iterations have finished. The default is *not* to save random draws, but rather to have the means and standard deviations for each respondent’s draws accumulated as iteration progresses. In that case the means and standard deviations are available immediately following iterations, with no further processing. We believe that their means and standard deviations summarize almost everything about them that is likely to be important to you. However, some of the statistical measures reported in the .log file will not be available unless you save draws. If you elect not to save the draws the computation will be slightly faster because some reading and writing of files is avoided. Also, you will be spared computer storage problems that can occur with very large files.

Advanced Parameters

Most users will probably never change the defaults on the *Advanced Parameters* tab. However, we've provided additional settings to provide more flexibility to deal with extreme types of data sets and to give advanced users greater control over estimation.

When you click the *Advanced Parameters* tab, the following appears:



Prior degrees of freedom: This value is the additional degrees of freedom for the prior covariance matrix (not including the # parameters to be estimated), and can be set from 2 to 100000. The higher the value, the greater the influence of the prior variance and more data are needed to change that prior. The scaling for degrees of freedom is relative to the sample size. If you use 50 and you only have 100 subjects, then the prior will have a big impact on the results. If you have 1000 subjects, you will get the about the same result if you use a prior of 5 or 50. As an example of an extreme case, with 100 respondents and a prior variance of 0.1 with prior degrees of freedom set to the number of parameters estimated plus 50, each respondent's resulting part worths will vary relatively little from the population means (assuming there is relatively sparse information available within the unit of analysis). We urge users to be careful when setting the prior degrees of freedom, as large values (i.e. above 10) can make the prior exert considerable influence on the results.

Prior variance: The default is 1 for the prior variance for each parameter, but users can modify this value. You can specify any value from 0.001 to 1000. Increasing the prior variance tends to place more weight on fitting each individual's data, and places less emphasis on "borrowing" information from the population parameters. The resulting posterior estimates are relatively insensitive to the prior variance, except 1) when there is very little information available within the unit of analysis relative to the number of estimated parameters, and 2) the prior degrees of freedom for the covariance matrix (described above) is relatively large.

Random seed: In previous versions of HB, a seed for the random number generator was drawn based on the computer's clock. That is still the default behavior (a seed of "0" indicates to use a seed based on the computer clock), but users can now specify a specific seed to use (integers from 1 to 32000), so that results are repeatable. When using different random seeds, the posterior estimates will vary, but insignificantly, assuming convergence has been reached and many draws have been used.

Decimal places of precision: Determines how many decimal places of precision are used when writing the data to .dra and .bet files.

Coding Independent Variables

Although there is much that could be said about coding of independent variables, there are only a few points that we would like to emphasize.

One point concerns the scaling of your variables. All else equal, multiplying all values of a particular variable by any constant will result in the estimated regression coefficient being scaled by the reciprocal of that constant. For greatest numeric accuracy, it is useful for all variables to be scaled to be of comparable magnitude. For example, if you are including annual family income and number of dependents as two variables, it would be best to scale income in tens of thousands of dollars, rather than actual dollars, so the actual numeric values for those two variables are about the same size. If you do so, then you can specify shorter output fields without sacrificing precision.

(Note: advanced users can specify their own prior covariance matrix to deal with situations in which some betas have quite different expected posterior variances than others.)

Another point is that HB-Reg does not automatically compute an intercept. If you want an intercept term, you must include a variable that has the constant "1" for each observation. If you do that, the regression coefficient for that variable is the intercept.

The final point concerns the way categorical variables are coded. By "categorical variables" we mean variables like male/female or eye color, for which the various possible states or categories do not have obvious numeric values. HB-Reg has no built-in provision for automatically coding categorical data. We leave it to you to take care of that. There are two popular ways of doing so.

Often categorical variables are coded using "dummy variables." This means that a separate variable is devoted to each state, scored with a one if that state is present and a zero if that state is absent. It is desirable to delete one state of each variable, since otherwise there would be colinearity, with the sum of codes for each variable being exactly unity. Omitting one state for

each variable is equivalent to assuming that the regression coefficient for that state is equal to zero, and that the other coefficients for that variable measure effects with respect to the omitted state. This is the simplest and most straightforward procedure. Using dummy coding for categorical variables also makes it possible to constrain any of the “explicit” levels with respect to the “omitted” variable, as the omitted variable is zero.

“Effects coding” is a little more complicated, and consists of substituting a pattern of codes for each state. If a variable has k states, then effects coding, like dummy variable coding, devotes $k-1$ independent variables to that categorical variable. Examples of the patterns are as follows:

Two states (such as male/female)

If male, code	1
If female, code	-1

Three states (such as urban/suburban/rural)

If urban, code	(1 , 0)
If suburban, code	(0 , 1)
If rural code	(-1 , -1)

Four states (blue, green, brown, other eye color)

If blue, code	(1 , 0 , 0)
If green, code	(0 , 1 , 0)
If brown, code	(0 , 0 , 1)
If other, code	(-1 , -1 , -1)

As in dummy variable coding, the deleted state has an implied regression coefficient. However, rather than being equal to zero as in dummy variable coding, the deleted state has an implied coefficient equal to the negative of the sum of coefficients for the included states for that variable. Therefore, with effects coding the sum of coefficients for all (deleted and retained) states of a categorical variable is zero. This symmetry among the various levels is an attractive feature which has led many analysts to favor effects coding. *(Note: HB-Reg’s method of imposing constraints cannot be used to constrain an “explicit” level with respect to the “omitted” level for effects coding.)*

In ordinary least squares regression, results are independent of coding. That is to say, you could use dummy variable coding, insert a value of zero for the deleted state, and then zero-center all k of the (estimated and implied) coefficients, and you would get the same results as if you had effects-coded the variable. However, the same is not true for HB-Reg.

HB-Reg estimates the variance for each coefficient by computing the actual variance among hundreds or thousands of random draws for that coefficient.

With effects coding, the coefficient for the deleted state is taken as the negative of the sum of coefficients for the $k-1$ retained states. If those draws were to vary independently, then the variance of their sum would be the sum of their variances, and as a result the coefficient for the deleted state would have a larger variance than the coefficients estimated directly.

With dummy variable coding, the deleted state is given a value of zero. However, if after inserting zero as the value for the omitted state, the estimates are centered within each attribute, the coefficient for each deleted state becomes equal to the negative of the sum of the retained

states, divided by k . If estimates for the $k - 1$ retained states were to vary independently, the variance of their sum divided by k would be smaller than the average of their variances.

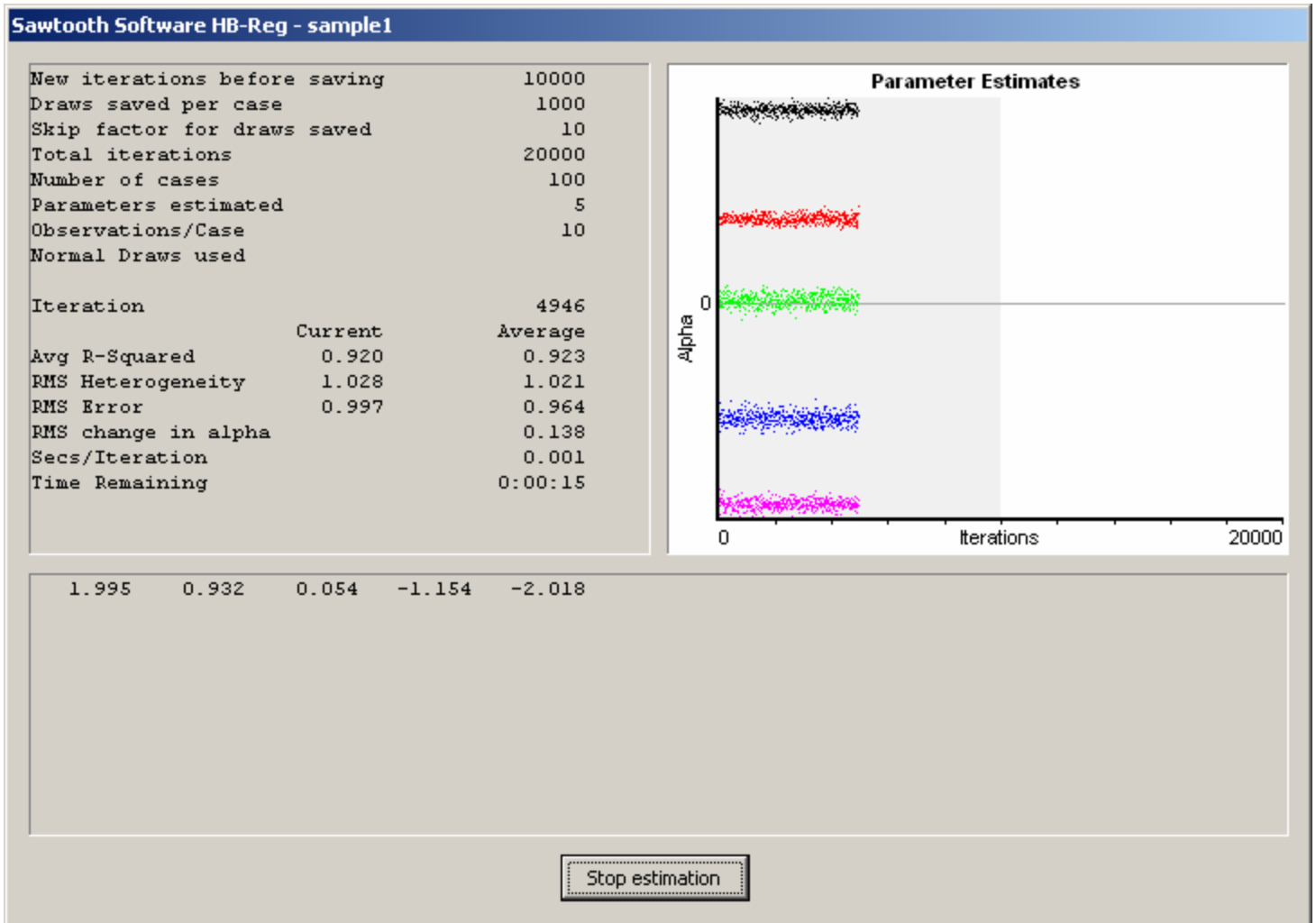
It turns out that the random draws are not completely independent, but do have some independent variation. Thus, if effects coding is used, the estimated standard errors for omitted levels will tend to be larger, and if dummy variable coding is used the estimated errors for omitted levels will tend to be smaller. Unlike ordinary least squares regression, with HB-Reg it makes a difference how the data are coded.

If there is a particular state of a categorical variable which it makes sense to regard as a baseline of some sort, then it would be appropriate to use dummy variable coding and delete that state, so coefficients for other levels express contrasts with that state. If you decide to use effects coding, the implied variances for the deleted levels will probably be slightly greater than for the included levels.

We have found with extremely sparse data that effects coding can sometimes lead to improper estimates of the omitted variable. The problems are enhanced as the number of mutually exclusive levels within a factor is increased. With choice-based data, the data can become especially sparse, so in our CBC/HB software we modify the prior covariance matrix to deal more effectively with effects coding (see the appendix of the CBC/HB software manual for details). With HB-Reg, the dependent variable is continuous rather than discrete, and therefore the data typically include more information within the unit of analysis. Even so, advanced users may implement similar procedures within HB-Reg as CBC/HB to deal with effects coding by overriding the defaults with a user-specified prior covariance matrix.

Monitoring the Computation

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



The information near the top of the upper-left panel verifies what you prescribed on the *Parameters* tab. This run uses 10,000 initial iterations, followed by 10,000 further iterations during which each tenth iteration is to be saved to disk. The data in the two columns further down in the upper-left panel provide a summary of the status of the computation, and we shall examine those values in a moment. At the bottom of this section is an estimate of the total time of 15 seconds required to complete this computation. Note that this is a tiny problem, with only 100 respondents, 10 observations per respondent, and 5 independent variables. More realistic problems will require from several minutes to an hour or more.

In the upper-right panel is a chart that plots the average estimates of alpha (the population means for the estimated parameters) on the y-axis, and time (by iteration) on the x-axis. (For a constrained solution, we plot the constrained estimates of beta.) The gray area represents the

burn-in iterations (prior to assuming convergence), and the white area represents the iterations used in developing the posterior estimates of the parameters (and optionally in saving draws to file). This graphic aids in determining whether convergence has occurred. Depending on how many respondents are included in the data set, the estimates of alpha may seem to vary quite a bit in subsequent iterations. The important point is that there be no noticeable trend by the time convergence is assumed and iterations begin to be used in developing posterior estimates (the final iterations represented by the white area on the graph). We can see with this small, hypothetical example that convergence occurs very early on in the process.

In the bottom panel are the current estimates of alpha, for each of the parameters in the model (up to the first 100 parameters are displayed).

We now describe each of the statistics displayed in upper-left panel. 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 is “Avg R-Squared” which is short for the average squared correlation between each respondent’s predicted and actual data for the 10 observations of the dependent variable. Because we start with estimated regression coefficients of zero, the average R-squared will be zero initially and improve throughout the early part of the computation.

The next statistic is “RMS Heterogeneity” which is short for “root mean square heterogeneity.” Recall that we estimate the variances and covariances for the betas (regression coefficients) among respondents. “RMS Heterogeneity” is just the square root of the average of those variances. The SAMPLE1 and SAMPLE2 data sets were both constructed to have heterogeneity of unity, so our estimation of this quantity is quite accurate.

The next statistic is “RMS Error” which is a measure of the average error in predicting each respondent’s dependent variable from his/her independent variables. 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. The SAMPLE1 and SAMPLE2 data sets were also constructed to have RMS error of unity, so this quantity is also estimated quite accurately.

The next statistic is “RMS Change in alpha.” Alpha is the estimate of average regression coefficients for the population from which the individuals are drawn. This estimate is updated on each iteration, and the RMS Change is the square root of the average squared change from one iteration to the next.

In the bottom panel are values identified as “Current estimate of alpha” (if constraints are in use, these values are mean constrained beta). This data set was constructed so that the expectations of these values are (2, 1, 0, -1, -2). As you can see, the current estimates are quite close. The estimates of alpha are not printed for each iteration, because they would change so rapidly that

they would be hard to read. These values are updated only as frequently as iterations are saved — in this case on every tenth iteration.

As iterations progress, all of these statistics change systematically for a while: Average R-squared and heterogeneity increase at first, while RMS Error and RMS Change in alpha decrease at first. Eventually they level off, thereafter oscillating randomly around their final values. Lack of trend may be taken as evidence of convergence. However, there is no good way to identify convergence until long after it has occurred. For this reason we suggest planning a large number of initial iterations, such as 10,000, and then examining retrospectively whether these four statistics have been stable for the last several thousand iterations. We also recommend you pay attention to the average estimates of alpha as plotted in the graphic, to see that they lack any trend over the last few thousand iterations.

The *studyname.log* file contains a history of these measures, and may be inspected after the iterations have concluded. If values for the final few thousand iterations are larger than for the preceding few thousand, that should be considered to be evidence that more iterations should be conducted before inferences are made about the parameters.

If you also run the SAMPLE2 data set, you will see a similar screen, with one difference. When respondents can have unique values for the independent variables, it is more efficient computationally to use a Metropolis Hastings algorithm than normal draws in updating estimates of the betas. With Metropolis Hastings algorithms, each successive candidate for a respondent's beta is obtained by making a random perturbation of his/her previous beta. The size of that perturbation has an important impact on speed of convergence, and is adjusted automatically during the computation. When a Metropolis Hastings algorithm is in use, the screen reports the current "jump size" as well as the proportion of respondents for whom the perturbed estimate was accepted as superior to its predecessor. That percentage is automatically maintained at about 30%.

Assessing Convergence

We have already noted that one of the easiest ways to try to determine convergence is to study the pattern of average estimates for the parameters as plotted in the visual graphic displayed during iterations. There is yet another way to assess convergence. After the computation has finished you can inspect its history by looking at a file named *studyname.log*. We specified that results of each 1,000th iteration were to be saved to the log file, and we reproduce that information here:

Iteration	AVG Rsq	Heter RMS	Error RMS	Change Alpha
1000	0.920	0.940	0.972	0.094
2000	0.926	1.015	0.966	0.164
3000	0.921	0.963	0.981	0.165
4000	0.930	1.233	0.918	0.117
5000	0.923	0.999	0.985	0.147
6000	0.926	1.026	0.955	0.133
7000	0.922	1.092	0.980	0.250
8000	0.922	0.991	0.963	0.137
9000	0.921	0.900	0.991	0.117
10000	0.920	1.053	0.994	0.124
11000	0.922	1.290	0.968	0.239
12000	0.922	1.039	0.961	0.088
13000	0.922	1.019	0.974	0.150
14000	0.922	0.977	0.965	0.076
15000	0.926	1.027	0.959	0.116
16000	0.931	1.019	0.915	0.141

17000	0.924	1.192	0.952	0.155
18000	0.926	1.108	0.949	0.150
19000	0.920	1.045	0.984	0.216
20000	0.922	1.101	0.968	0.081

As you can see there is no apparent trend in any of these four columns. Apparently this process converged long before the 10,000 iterations which we had allowed for that to occur.

You can also inspect the files named *studyname.alp* which contains each saved estimate of alpha (the vector of estimated means of the distribution of the betas) and *studyname.sig* which contains corresponding estimates of sigma. There is no apparent trend in either of these files, so it seems reasonable to conclude that the process has converged.

The *studyname.pnt* file contains a summary of point estimates of the parameters common to all respondents, which we reproduce here:

Estimated Means and Covariances for Distribution of Betas:

Point Estimate for Alpha (Mean of Distribution of Betas):
 1.995 0.879 0.049 -1.184 -2.061

Point Estimate for D (Covariances of Distribution of Betas):
 0.949 -0.015 -0.096 0.010 0.082
 -0.015 0.978 0.073 0.066 -0.066
 -0.096 0.073 1.098 0.067 0.027
 0.010 0.066 0.067 1.191 0.187
 0.082 -0.066 0.027 0.187 1.051

Point Estimate for Sigma: 0.965

This data file was constructed in such a way that the expectation for values of alpha is (2, 1, 0, -1, -2). Given that we are working with a sample of only 100 respondents, our estimates seem reasonably close.

Our expectations for the variances and covariances for this data set are that variances will be unity and covariances will be zero. Again, considering this small sample, it seems we have come reasonably close.

Finally, we expected Sigma to be unity, and the estimate of .965 also seems reasonable.

Another file is optionally produced which we have not yet described. It is named *studyname.dra*, and contains estimates of the betas for each respondent in each saved iteration. This file can consume a lot of space on your hard disk, and therefore, you may choose to not save draws. Even though the SAMPLE1 problem is tiny, with only 100 respondents and five variables, the SAMPLE1.DRA file occupies nearly seven megabytes of storage. If you are dealing with very large data sets, your analysis is more likely to be hindered by a scarcity of unfragmented disk space than by the time required for computation itself.

The *studyname.dra* file can be of use if you want to study the distributions of the betas, among or within respondents. We have summarized much of that information for you, as described in the next section. When you have finished using the *studyname.dra* file, you should delete or “zip” it so as to free up disk space for your next project.

Deciding Which Variables to Include

At the bottom of the *studyname.log* file is a summary which provides information about the relative influence of each independent variable. Here is the result for the SAMPLE1 run (assuming you choose to save draws):

Summary of Analysis

Variable	Mean of Beta Draws	Mean of Variance Draws	t Ratio for Mean of Betas	Mean Sq Between Cases	Mean Sq Within Cases	F Ratio Hetero- geneity
1	1.995	0.949	20.479	767.349	0.123	6237.25
2	0.879	0.978	8.888	773.739	0.143	5404.96
3	0.049	1.098	0.468	794.375	0.254	3123.81
4	-1.184	1.191	-10.849	871.575	0.253	3450.51
5	-2.061	1.051	-20.104	828.806	0.162	5109.51

The first column gives point estimates for the means of the distribution of respondents' regression coefficients. This is obtained by averaging the 1000 saved values of alpha which are available in the *studyname.alp* file. As can be seen, the values are reasonably close to the expected values of 2, 1, 0, -1, and -2, particularly considering that we are dealing with a small sample of only 100 respondents.

The second column gives the estimate of the variance of the distribution of each regression coefficient across respondents. This is obtained by averaging the 1,000 saved estimates for the variances.

The third column gives a t ratio that expresses the difference of each mean from zero. The standard error of the mean, estimated from the variance, furnishes the denominator for the t ratio. The t values give an indication of the *average* importance of each attribute in the regression equation, but reveal nothing about the value of that variable in differentiating among respondents. The importance of each variable in accounting for heterogeneity is given by the last three columns. The Mean Square Between Cases is a measure of the difference among respondents for that regression coefficient. The Mean Square Within Cases is a measure of the amount of random variation within respondents, obtained in different random draws for the same respondents. The ratio of those two variances is an F ratio, which measures the extent to which differences among respondents exceed differences within respondents. As you can see, these F ratios are all very large.

For Type 1 data (where everyone has the same design matrix) it is unlikely that you will want to do analyses using only subsets of the independent variables. However, for Type 2 data (where different respondents have different values for the independent variables) you may want to do so. In those cases, the t and F ratios from this table may be helpful in deciding on the relative importance of each variable. If a variable has a larger-than-average t ratio, or F ratio, it is probably more powerful than average in terms of accounting for variation in the dependent variable. A high t ratio indicates a variable on which individuals tend to agree, and a high F ratio indicates a variable on which they disagree, but which is important to them.

Although these t and F ratios provided indications of the relative effects of variables in accounting for overall preference, we do not advocate using them for tests of statistical significance, since they would be biased upwards. The situation is similar to that of cluster analysis, where it is improper to do statistical tests on t or F statistics computed using the same data as was used to find maximally differing groups.

Following the table described above is a second table that reports the proportion of respondents for which a “pseudo-t” ratio is greater than an absolute value of 1.96. Each t value is computed by taking the point estimate of beta for each individual and dividing it by the standard deviation of the draws for that beta for that individual. Although this measure of t is somewhat *ad hoc* (not based on Bayesian theory), it has intuitive appeal.

Variable	% of respondents with t-ratio for beta absolute value > 1.96
1	93.0
2	56.0
3	26.0
4	54.0
5	94.0

A variable that has a significant t-ratio (critical value of 1.96 corresponds to 95% confidence) for very few respondents is of lesser value in the model. Most analysts would agree that variables one and five add substantial value to the model. Variable three is “significant” for 26% of the subjects. In this case, the user must make a judgment call regarding whether to include variable three in further analysis. It may be that these respondents comprise an important segment of the market for which variable three has a significant effect relative to the dependent variable.

How Good Are the Results?

In this chapter we present some findings from the analysis of both synthetic and real data sets. For synthetic data sets we already know the “right answers,” and we assess HB-Reg’s performance by measuring how well it recovers those known parameters. For real data sets we assess performance by the ability to predict holdout choices.

Before reviewing this evidence, we should point out that HB-Reg is one of four Sawtooth Software products that use HB to estimate individual coefficients. Other products are CBC/HB 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 CVA/HB for estimating individual part-worths for traditional conjoint analysis also using a linear regression formulation. For each of those products we have done a similar performance review, finding HB estimation to be as good 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 2002, 2003a, 2003b).

Synthetic Data Sets

Note: all the results presented in this section are based on the default settings for the prior covariance matrix as were used in HB-Reg version 2 and earlier.

We first review results from three kinds of synthetic data sets. Each contained 300 respondents, with five independent variables. Although there are some differences in details, each data set was constructed with similar steps:

Average “true” betas of (2, 1, 0, -1, -2) were chosen.

Idiosyncratic betas were chosen for each respondent by perturbing each element of the average betas by random normal heterogeneity with mean of zero and standard deviation of unity.

The independent variables were rectangularly distributed random integers in the range of 1 - 9, and were unique for each individual.

The dependent variable for each observation was constructed by summing the products of each respondent’s independent variables for that observation and his/her corresponding beta, and then to that sum adding normal error with mean of zero and standard deviation of unity.

Thus, we created data representing samples of respondents from a population normally distributed with alpha (the mean of the betas) = (2,1,0,-1,-2), variances of unity and covariances of zero, and with sigma of unity.

The number of observations per respondent varied from 1 to 10, and three different data sets were made for each number of observations per respondent. Each data set was used in a separate estimation run, and the three results for that number of observations were averaged for reporting.

In the second and third kinds of data sets we introduce colinearity and multiple populations. However, Table 1 provides results for the first analysis, in which the independent variables are independent of one another, and with a single population of respondents.

Table 1: Accuracy of Estimation with
No Colinearity, Single Population of Respondents

Obs/Ind	— RMS RECOVERY —			Ind Corr	- EST SDs FOR -	
	OLS AGG	HB AGG	HB IND		Heter	Error
1	0.313	0.263	0.992	0.826	1.383	5.213
2	0.177	0.087	0.856	0.870	1.062	0.895
3	0.193	0.049	0.717	0.910	1.020	1.093
4	0.187	0.046	0.564	0.947	1.043	1.018
5	0.144	0.019	0.418	0.971	1.033	1.044
6	0.125	0.025	0.319	0.983	1.033	0.976
7	0.122	0.013	0.249	0.989	1.023	0.999
8	0.117	0.009	0.207	0.993	1.049	1.011
9	0.095	0.009	0.175	0.995	0.979	1.009
10	0.078	0.014	0.160	0.996	1.060	0.989

The first three columns of the table measure the success at recovering respondents' betas, individually and in aggregate. Each statistic is a "root mean square" error of estimation, so smaller values are better.

The first column provides an evaluation for ordinary least squares regression, done by pooling all respondents' observations to do aggregate regressions. The parameters that we seek to recover in that regression are the average betas.

The second column provides an evaluation for HB-Reg's ability to recover the same average values. Note that every number in the second column is smaller (more favorable) than the corresponding number in the first column. We may conclude that HB-Reg has done a better job than ordinary regression at estimating the mean of the population, probably because it has been able to separate true heterogeneity from random error. Since ordinary regression is unbiased, we would expect this superiority to decrease as the sample size increases; but it should be noted that 300 is not an unusually small number of respondents for practical research projects.

The third column provides an evaluation for HB-Reg's ability to estimate individual betas. These numbers are quite a lot larger (worse) than the corresponding aggregate numbers. However, they decrease regularly as the amount of information per respondent increases. To assess the magnitude of these errors, consider the fourth column, which gives the average correlation between actual and estimated individual betas. With as few as 3 observations per individual, the average correlation is in the .90's. Of course, success will depend on the amount of error in the data. We have arbitrarily used error of unity. If the data had a different amount of response error, the results could be either better or worse.

The fifth and sixth columns give estimates of heterogeneity and random error, for which the correct values are both unity. The precision of these estimates is already reasonably good with as few as two observations per respondent, and continues to improve with more observations.

In this comparison, HB-Reg does a better job than ordinary regression at recovering the average parameters. It is able to make creditable estimates of individual parameters even with as few as three observations per respondent, an accomplishment denied ordinary regression, which requires that there be at least as many observations as parameters to be estimated per individual. However, this data set may be more favorable for HB-Reg than most real data sets would be, because it conforms to the underlying assumptions about heterogeneity, and presents no problems due to colinearity.

For the second group of data sets we introduce colinearity. Another 30 data sets were produced in exactly the same way, except that a common random value with large variance was added to the independent variable values for each observation, making the average correlation among independent variables approximately .85. This is a large amount of colinearity, perhaps approximating what might occur in a typical customer satisfaction study. Table 2 provides results for this analysis.

Table 2: Accuracy of Estimation with Colinearity, Single Population of Respondents

Obs/Ind	— RMS RECOVERY—			Ind Correl	-EST SD FOR-	
	OLS AGG	HB AGG	HB IND		Heter	Error
1	1.521	1.296	1.668	0.564	6.955	17.766
2	0.833	0.172	0.915	0.846	0.944	2.064
3	0.666	0.067	0.791	0.890	0.941	1.354
4	0.690	0.068	0.669	0.922	0.954	1.426
5	0.487	0.038	0.507	0.957	1.036	1.077
6	0.518	0.028	0.380	0.975	1.007	1.082
7	0.682	0.016	0.312	0.984	1.064	1.004
8	0.455	0.011	0.245	0.990	1.066	1.016
9	0.602	0.014	0.208	0.993	0.968	1.003
10	0.390	0.007	0.177	0.995	1.009	1.001

This table has many similarities to the previous table, but the presence of colinearity has impeded the recovery of the true parameters.

The first column again measures the ability of ordinary regression to recover average betas for the population. The errors are about five times as large as without colinearity, and again decrease uniformly as the number of observations increases.

The second column again measures the ability of HB-Reg to recover the same average betas. For a single observation per respondent its error is also about five times as large as without colinearity. But this improves dramatically as the number of observations increases. With only two observations per respondent the error is only about twice as great as without colinearity, and with three or more observations per respondent its errors are less than any tabled case for ordinary regression with no colinearity.

The third column, measuring HB-Reg's recovery of individual betas, again shows much larger errors for the case of a single observation per respondent. But these also improve rapidly with increases in the number of observations per respondent. The fourth column shows that with as few as two observations per respondent, average correlations between true and estimated individual betas are in the .80s, and with as few as four they are in the .90s. (Again, results would be different with different error levels in the data.)

Finally, the fifth and sixth columns show that estimates of heterogeneity and sigma are much worse than the case without colinearity, but the heterogeneity estimate is quite good with as few as two observations per respondent, and the estimate of sigma becomes reasonable when there are as many observations per individual as parameters estimated.

To summarize we see that colinearity is damaging to both ordinary regression and HB-Reg, although the impact on ordinary regression is much more severe. With as few as two observations per respondent, HB-Reg is able to produce good estimates of the population mean

and of the amount of heterogeneity. With three observations per respondent, HB-Reg is able to produce reasonable estimates of individual betas.

For the final set of artificial data sets we explore HB-Reg's ability to deal with data representing a mix of two populations, rather than the single population its model assumes. A group of 30 more synthetic data sets were created which did not have colinearity, but for which half of the respondents had average betas of (2,1,0,-1,-2) and the other half had betas of opposite sign. Thus, average betas for the population were zero, although those would not describe either sub-population accurately.

Table 3: Accuracy of Estimation with No Colinearity, but Two Populations of Respondents

Obs/Ind	— RMS RECOVERY—			Ind Correl	-EST SD FOR- Heter	Sigma
	OLS	AGG	HB-IND			
1	0.307	0.255	1.578	0.412	3.840	2.890
2	0.200	0.164	1.300	0.680	3.058	1.078
3	0.162	0.074	0.981	0.826	3.148	0.708
4	0.155	0.043	0.701	0.913	3.195	0.965
5	0.131	0.034	0.502	0.957	2.900	0.919
6	0.153	0.019	0.352	0.980	3.094	0.987
7	0.138	0.017	0.271	0.988	2.929	1.030
8	0.089	0.014	0.215	0.992	2.989	0.980
9	0.126	0.008	0.181	0.995	3.103	1.000
10	0.097	0.011	0.161	0.996	2.871	0.988

Comparison of the first and second columns of Table 3 shows that HB-Reg is again better able to recover average betas for the population, with a margin of superiority that increases with the number of observations per respondent.

Individual betas are again recovered poorly with few observations per respondent, but with as many as three observations per respondent the average correlation between true and estimated individual betas is in the .80s, and with four the average correlation is in the .90s, a performance similar to its performance with colinearity (and which, again, depends on the arbitrary assumption of unit error). With the two sub-populations having betas with opposite signs, the true heterogeneity should be 3, and it is estimated quite accurately with two or more observations, as is the true value of 1 for sigma.

Summarizing these three kinds of data sets, we find that HB-Reg is consistently superior to ordinary least squares regression in its ability to recover the true population mean, and its margin of superiority increases with the number of observations per respondent. We find that even in the presence of colinearity or a mixture of normal populations, HB-Reg is able to produce reasonable estimates of individual betas with as few as three observations per respondent. HB-Reg is also able to produce quite reasonable estimates of the true amount of heterogeneity among respondents with as few as two observations per respondent.

HB-Reg is able to produce reasonable estimates of individual betas even when the number of observations per individual is less than the number of parameters to be estimated. Although this is an impressive accomplishment, we believe it is equally impressive that HB-Reg is able to estimate group parameters so accurately. With ordinary regression the analyst is required to pool observations from different respondents, and in doing so must confound heterogeneity and error. Recognizing the difference between these sources of variance seems to permit greater accuracy in the estimation of average betas for the population.

Real Data Sets

Note: the results presented in this section are based on the default settings for the prior covariance matrix as were used in HB-Reg version 2 and earlier.

We turn now to two small examples using real data sets, both examples of conjoint analysis.

The first is from a study reported by Orme *et al.* (1997). MBA students from three universities were respondents. The subject of the study was personal computers, and nine attributes were studied, each with two or three levels. There were a total of 80 respondents. 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. Those ratings were converted to logits, which were used as the dependent variable, both for ordinary least squares regression and also by HB-Reg. In these regressions each respondent contributed 22 observations and a total of 16 parameters were estimated for each, including an intercept.

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%
HB-Reg	73.50%

Neither of these sets of part-worths was constrained so that “obviously better” levels are at least as high as “obviously worse” levels. This can be done easily simply by tying offending pairs of values. Constraining part-worths in that way usually improves their performance in predicting choice. However, since we have not imposed constraints on either set of part-worths, this is a fair comparison, and HB-Reg has a 1.5% margin of superiority.

The second data set is from a study reported by Orme and King (1998) in which 280 individuals responded to an Internet conjoint study of credit cards with four attributes, each with three levels. There were a total of 9 parameters to be estimated for each respondent (including an intercept), and each respondent saw 9 concept cards, each of which was rated for likelihood of signing up on a 5 point scale. This design provided no extra degrees of freedom for error.

Each respondent also answered 9 paired-comparison questions dealing with the same attributes, to test the relative effectiveness of single-concept full-profile conjoint analysis to paired-comparison full-profile conjoint analysis. The authors concluded that the two data collection formats had equivalent performance. We consider only the single-concept data in our comparison.

Each respondent also saw three holdout tasks at the beginning of the survey, in which the preferred concept was selected for each set. The same questions were repeated at the end of the questionnaire, with rotation of concept position. The test-retest reliability was 83%.

We again compare hit rates for predicting holdout choices:

Ordinary Least Squares	78.50%
HB-Reg	79.83%

Again, neither set of part-worths was constrained, although either set could have been. HB-Reg again has a slight margin of superiority.

References

- Allenby, G. M., Arora, N., and Ginter, J. L. (1998) "On the Heterogeneity of Demand," *Journal of Marketing Research*, 35, (August) 384-89.
- 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) 327-35.
- 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, 253-66.
- 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, 173-91.
- Orme, B. K., Alpert, M. I. & Christensen, E. (1997) "Assessing the Validity of Conjoint Analysis – Continued," *Sawtooth Software Conference Proceedings*, Sawtooth Software, Sequim, 209-26.
- Orme, B. K. and King, W. C. (1998) "Conducting Full-Profile Conjoint Analysis over the Internet," Sawtooth Software Technical Paper, accessible from sawtoothsoftware.com web site.
- 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 (2002) "The CVA/HB Technical Paper," Technical Paper accessible from sawtoothsoftware.com web site.
- Sawtooth Software (2003a) "The CBC/HB Module for Hierarchical Bayes Estimation," Technical Paper accessible from sawtoothsoftware.com web site.
- Sawtooth Software (2003b) "The ACA/HB Module for Hierarchical Bayes Estimation," Technical Paper accessible from sawtoothsoftware.com web site.

Appendix:

Details of Estimation

Chapter 2 attempts to provide an intuitive understanding of the HB estimation process, and to avoid complexity it omits some details that we shall provide 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 \mathbf{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, \mathbf{D} , and sigma, so our task is a little 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. We differentiate between two types of data: (1) A fixed design matrix for all individuals, and (2) independent variables that can take different values for each individual. The draws of alpha, \mathbf{D} , and sigma are done the same way, regardless of type of data, but draws of the betas differ. With the first type of data we use “normal draws” for the betas. For the second type of data we use a Metropolis Hastings algorithm, which is more efficient in that case.

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} , $1/\mathbf{n} \sum_{\mathbf{n}} \mathbf{u} \mathbf{u}'$ approaches the identity, $1/\mathbf{n} \sum_{\mathbf{n}} \mathbf{v} \mathbf{v}'$ approaches \mathbf{D} as shown below:

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

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 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 α and β_i

$$\mathbf{H} = \mathbf{p} \mathbf{I} + \sum_{\mathbf{n}} (\alpha - \beta_i) (\alpha - \beta_i)'$$

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

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

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

$$\mathbf{S} = \sum_{\mathbf{N}} (\mathbf{T} \mathbf{u}_j) (\mathbf{T} \mathbf{u}_j)'$$

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 Normal Draws

Assuming that every individual has the same design matrix \mathbf{X} and individual \mathbf{i} has vector of data \mathbf{y}_i , then each β_i of regression weights for individual \mathbf{i} is distributed normally with mean μ_i and covariance matrix \mathbf{C} , where

$$\mathbf{C} = (\mathbf{D}^{-1} + \sigma^{-2} \mathbf{X}'\mathbf{X})^{-1}$$

$$\mu_i = \mathbf{C} (\mathbf{D}^{-1} \alpha + \sigma^{-2} \mathbf{X}' \mathbf{y}_i)$$

Each β_i is drawn using the random draw procedure described above, using mean vector μ_i and covariance matrix \mathbf{C} .

Estimation of Betas Using a Metropolis Hastings Algorithm

We now describe the alternative procedure used to draw each new set of betas, done for each respondent in turn. We use the symbol β_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 β_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 β_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 $\beta_n = \beta_o + \mathbf{d}$. We regard β_n as a candidate to replace β_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 \mathbf{p}_o and \mathbf{p}_n , respectively.

We also calculate the relative density of the distribution of the betas corresponding to β_0 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_0 and d_n , respectively.

Finally we then calculate the ratio:

$$r = p_n d_n / p_0 d_0$$

From Bayes' theorem, the posterior probabilities are proportional to the product of the likelihoods times the priors. The values p_n and p_0 are proportional to the likelihoods of the data given parameter estimates respectively. The values d_n and d_0 are proportional to the probabilities of drawing those values of β_n and β_0 , 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_0 , 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_0 , 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_0 . In that case we use a random process to decide whether to accept β_n or retain β_0 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_0 , 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 d of differences is drawn from a distribution with mean of zero and covariance matrix proportional to D , but we did not specify the proportionality factor. In the literature the distribution from which 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.

The iterative process has two stages. During the first stage, while the process is moving toward convergence, no attempt is made to save any of the results. During the second stage we assume the process has converged, and results for hundreds or thousands of iterations are saved to the hard disk. For each iteration there is a separate estimate of each of the parameters. We are particularly interested in the betas, which are estimates of individuals' betas. We produce point estimates for each individual by averaging the results from many iterations. We can also estimate the variances and covariances of the distribution of respondents by averaging results from the same iterations.

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.