Sawtooth Software

TECHNICAL PAPER SERIES

The ACA/Hierarchical Bayes v3.0 Technical Paper

530 W. Fir St.
Sequim, WA 98382
(360) 681-2300
www.sawtoothsoftware.com

The ACA/HB Module for Hierarchical Bayes Estimation v3
Sawtooth Software, Inc.
July, 2006

Background and Introduction

After its introduction in 1985, ACA went on to become the most frequently used method for commercial conjoint studies in both the United States and Europe (Wittink, McLauchlan & Seetharaman, 1997). Today, ACA is still one of the most popular conjoint methods. ACA's distinguishing characteristic is the way it combines self-explicated data with conjoint data, permitting part-worth estimates for potentially large numbers of attributes from a relatively short interview.

The ACA/HB module extends ACA's capabilities by using a hierarchical Bayes (HB) algorithm to estimate individual part-worths. This provides two benefits:

> First, the ACA/HB module improves the stability of each individual's estimates by "borrowing" information from other individuals. With ACA, as with most conjoint applications, the number of parameters to be estimated for each individual is often large with respect to the number of observations. HB methods have been shown to produce dramatic improvements under such circumstances, for both ratings-based and choice-based conjoint applications (Allenby, Arora & Ginter 1995, Lenk *et al*., Johnson). We provide results below showing that similar improvements are available for ACA results.
>
> Second, the ACA/HB module provides an improved way of combining data from the self-explicated and conjoint sections of the interview. ACA has been criticized because of potential scale incompatibilities between the self-explicated and conjoint sections of the interview (Green, Krieger & Agarwal 1991). Initial versions scaled the self-explicated and conjoint data in a way that was approximately optimal on average, but likely to be sub-optimal for any particular respondent. More recent versions of ACA have combined the two kinds of data using weights computed for each respondent to provide optimal fit to ratings of holdout "calibration concepts." However, one of the options available with the ACA/HB module combines the two kinds of data in a way that makes no scaling assumptions whatever.

The ACA/HB module is computationally intensive. The time required will range from a few minutes to many hours, depending on the size of the data set. The ACA/HB module is used after field work is completed and the data are all available. Its main output is a set of individual part-worths written to an Excel™ friendly .csv file and also written to a file (.hbu format) that may be imported within Sawtooth Software's SMRT market simulator program.

The basic idea behind the use of HB is to recognize that each individual is a member of a group of more or less similar individuals, and that knowledge of the entire distribution of individuals' part-worths can enhance estimation for each individual. Individuals are assumed to be distributed multi-normally, and HB estimates the mean vector and covariance matrix for that distribution. It uses that distribution as a "prior" in the Bayesian sense, to enhance the estimation of each individual's part-worths.

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.

Capacity Limitations

The ACA/HB module has capacity even greater than that of the ACA or ACA/Web programs. The ACA programs are able to handle a projects with up to 30 attributes, each having up to 15 levels, so the total number of part-worths estimated for each respondent could be as large as 450. The number of independent parameters to be estimated is less than that, being equal to the total number of levels minus the number of attributes.

Advanced users may generate their own .ACD files based on studies not fielded using our software, so the following limitations may be of interest:

> The maximum number of attributes is 1000.
> The number of levels per attribute is only limited by the total parameters restriction.
> The maximum number of conjoint pairs is 1000.
> The maximum number of attributes appearing in a pair is 1000.

Note: Academic Lab Systems are limited to 10 attributes and 250 respondents.

The ACA/HB module does not make use of the "Calibration Concepts" section of the ACA interview for estimating part-worths. However, if those data are available they can be used to calibrate estimated part-worths for use in purchase likelihood simulations. If you do not plan to use purchase likelihood simulations, you can delete that section of the interview.

However, we recommend strongly that you include choice tasks elsewhere in the interview to be used as holdout concepts for testing the accuracy of prediction, and for adjusting the "exponent" if using Share of Preference or RFC simulation methods.

Hardware Recommendations

The ACA/HB Module requires a fast computer and a generous amount of storage space.

The ACA/HB Module estimates individual part-worths by doing many thousands of Monte Carlo simulation iterations. In a typical analysis, you might first do 5,000 iterations just to achieve convergence of the estimation process. Then you might use the results of each tenth of the next 10,000 iterations for later analysis.

Suppose you were estimating 50 part-worths for each of 500 respondents, a “medium-sized” problem. If you elect to “save random draws” for each respondent, each iteration saved would require about 100,000 bytes of hard disk storage. Saving the results for 1,000 iterations would require about 100 megabytes. Approximately the same amount of additional storage would be required for interim results, so the entire storage requirement for even a medium-sized problem could be several hundred megabytes.

There is a great deal of activity writing to the hard disk and reading back from it, which is facilitated by Windows’ ability to use extra RAM as a disk cache when running DOS programs. Also, the availability of RAM may be almost as critical as sheer processor speed.

This can all be handled easily by a modern computer with fast processor, plenty of RAM, and many gigabytes of hard disk storage. But use of ACA/HB will not be feasible with a more limited computer such as those typically available only a few years ago.

Understanding the ACA/HB Module

This section attempts to provide an intuitive understanding of the Hierarchical Bayes method as applied to the estimation of conjoint part-worths. For those desiring a more rigorous treatment, we suggest "Bayesian Data Analysis" by Gelman, Carlin, Stern, and Rubin.

Bayesian Analysis

In statistical analysis we consider three kinds of concepts: data, models, and parameters.

> In our context, data are the answers that individuals give to ACA questionnaires.
>
> Models are assumptions that we make about data. For example, we may assume that a variable is normally distributed, or that variable y depends on variable x, but not on variable z.
>
> Parameters are numerical values that we use in models. For example, we might say that a variable is normally distributed with mean of 0 and standard deviation of 1. Those values are parameters.

Often in conventional (non-Bayesian) statistical analyses, we assume that our data are described by a particular model with specified parameters, and then we investigate whether the data are consistent with those assumptions. In doing this we usually investigate the *probability distribution of the data*, given the assumptions embodied in our model and its parameters.

In Bayesian statistical analyses, we turn this process around. We again assume that our data are described by a particular model and do a computation to see if the data are consistent with those assumptions. But in Bayesian analysis, we investigate the *probability distribution of the parameters*, given the data.

To illustrate this idea we review a few concepts from probability theory. We designate the probability of an event A by the notation **p(A)**, the probability of an event B by the notation **p(B)**, and the *joint probability* of both A and B by the notation **p(A,B)**.

Bayesian analysis makes much use of *conditional probability*. Feller (1957) illustrates conditional probability with an example of sex and colorblindness. Suppose we select an individual at random from a population. Let A indicate the event of that individual being colorblind, and let B indicate the event of that individual being female. If we were to do many such random draws, we could estimate the probability of a person being both female and colorblind by counting the proportion of individuals found to be both females and colorblind in those draws.

We could estimate the probability of a female's being colorblind by dividing the number of colorblind females obtained by the number of females obtained. We refer to such a probability as "conditional;" in this case the probability of a person being colorblind is conditioned by the person being female. We designate the probability of a female's being colorblind by the symbol $\mathbf{p(A|B)}$, which is defined by the formula:

$$\mathbf{p(A|B) = p(A,B) / p(B)}.$$

That is to say, the probability an individual's being colorblind, given that the individual is female, is equal to the probability of the individual being both female and colorblind, divided by the probability of being female.

Notice that we can multiply both sides of the above equation by the quantity $\mathbf{p(B)}$ to obtain an alternate form of the same relationship among the quantities:

$$\mathbf{p(A|B)\ p(B) = p(A,B)}.$$

We may write a similar equation in which the roles of A and B are reversed:

$$\mathbf{p(B|A)\ p(A) = p(B,A)}.$$

and, since the event $\mathbf{(B,A)}$ is the same as the event $\mathbf{(A,B)}$, we may also write:

$$\mathbf{p(B|A)\ p(A) = p(A,B)}.$$

The last equation will be used as the model for a similar one below.

Although concrete concepts such as sex and colorblindness are useful for reviewing the concepts of probability, it is helpful to generalize our example a little further to illustrate what is known as "Bayes theorem." Suppose we have a set of data that we represent by the symbol $\mathbf{y}$, and we consider alternative hypotheses about parameters for a model describing those data, which we represent with the symbols $\mathbf{H_i}$, with i = 1, 2, ….

We assume that exactly one of those alternative hypotheses is true. The hypotheses could be any set of mutually exclusive conditions, such as the assumption that an individual is male or female, or that his/her age falls in any of a specific set of categories.

Rather than expressing the probability of the data given a hypothesis, Bayes' theorem expresses the probability of a particular hypothesis, $\mathbf{H_i}$, given the data. Using the above definition of conditional probability we can write

$$\mathbf{p(H_i \mid y) = p(H_i , y) / p(y)}.$$

But we have already seen (two equations earlier) that:

$$\mathbf{p(H_i , y) = p(y \mid H_i)\ p(H_i)}$$

Substituting this equation in the previous one, we get

$$\mathbf{p(H_i \mid y) = p(y \mid H_i)\, p(H_i) / p(y)}$$

Since we have specified that exactly one of the hypotheses is true, the sum of their probabilities is unity. The **p(y)** in the denominator, which does not depend on i, is a normalizing constant that makes the sum of the probabilities equal to unity. We could equally well write

$$\mathbf{p(H_i \mid y) \propto p(y \mid H_i)\, p(H_i)}$$

where the symbol $\propto$ means "is proportional to."

This expression for the conditional probability of a hypothesis, given the data, is an expression of "Bayes theorem," and illustrates the central principle of Bayesian analysis:

> The probability $\mathbf{p(H_i)}$ of the hypothesis is known as its "prior probability," which describes our belief about that hypothesis before we see the data.
>
> The conditional probability $\mathbf{p(y \mid H_i)}$ of the data, given the hypothesis, is known as the "likelihood" of the data, and is the probability of seeing that particular collection of data values, given that hypothesis about the data.
>
> The probability $\mathbf{p(H_i \mid y)}$ of the hypothesis, given the data, is known as its "posterior probability." This is the probability of the hypothesis, given not only the prior information about its truth, but also the information contained in the data.

The posterior probability of the hypothesis is proportional to the product of the likelihood of the data under that hypothesis, times the prior probability of that hypothesis. Bayesian analysis therefore provides a way to update estimates of probabilities. We can start-with an initial or prior estimate of the probability of a hypothesis, update it with information from the data, and obtain a posterior estimate that combines the prior information with information from the data.

In the next section we describe the hierarchical model used by the ACA/HB Module. Bayesian updating of probabilities is the conceptual apparatus that allows us to estimate the parameters of that model, which is why we have discussed the relationship between priors, likelihoods, and posterior probabilities.

In our application of Bayesian analysis, we will be dealing with continuous rather than discrete distributions. Although the underlying logic is identical, we would have to substitute integrals for summation signs if we were to write the equations. Fortunately, we shall not have to do so.

The Hierarchical Model

The Hierarchical Bayes model used by the ACA/HB Module is called "hierarchical" because it has two levels.

> At the higher level, we assume that individuals' part-worths are described by a multivariate normal distribution. Such a distribution is characterized by a vector of means and a matrix of covariances.
>
> At the lower level we assume that, given an individual's part-worths, his/her probabilities of responding in a particular way are governed by a multinormal distribution.

To make this model more explicit, we introduce some notation. We assume individual part-worths have the multivariate normal distribution,

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

where:

> $\boldsymbol{\beta_i}$ = a vector of part-worths for the ith individual
>
> α = a vector of means of the distribution of individuals' part-worths
>
> **D** = a matrix of variances and covariances of the distribution of part-worths across individuals

At the individual level, choices are described by the model:

$$\mathbf{y_{ih} = x_{ih}' \boldsymbol{\beta_i} + e_{ih}}$$

where:

> $\mathbf{y_{ih}}$ = the answer to question i by respondent h
>
> $\mathbf{x_{ih}'}$ = a row vector of values describing the ith question for respondent h (that is, a row of a design matrix.).
>
> $\mathbf{e_{ih}'}$ = an iid error term, distributed normally with mean of zero and variance σ^2 .

This model says that the answer of question i by respondent h is normally distributed with mean equal to a function of his part-worths (the product $\mathbf{x_{ih}'} \boldsymbol{\beta_i}$) and variance σ^2 .

The parameters to be estimated are the vectors $\boldsymbol{\beta_i}$ of part-worths for each individual, the vector α of means of the distribution of part-worths, the matrix **D** of the variances and covariances of that distribution, and the scalar σ.

Iterative Estimation of the Parameters

The parameters are estimated by an iterative process. That process is quite robust, and its results do not depend on starting values. However, to make the process converge as quickly as possible, we start-with estimates of the parameters that are reasonably close to final values.

Our initial estimates of the betas are "prior" part-worths obtained from the self-explicated section of the ACA interview.

Our initial estimate of alpha is the average of the initial betas.

Our initial estimate of **D** consists of variances and covariances of the initial betas.

Our initial estimate of sigma is 1.0.

Given those initial values, each iteration consists of these four steps:

Using present estimates of α**, D,** and σ**,** generate new estimates of the betas. This is the most interesting part of the iteration, and we describe it in detail below. A procedure known as a "Metropolis Hastings Algorithm" is used to draw the betas.

Using present estimates of the betas and **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 **D** divided by the number of respondents. A new estimate of α is drawn from that distribution (see the Appendix for details).

Using present estimates of the betas and α, draw a new estimate of **D** from the inverse Wishart distribution (see the Appendix for details).

Using present estimates of α**, D,** and the betas, generate a new estimate of σ. For this purpose we again use the inverse Wishart distribution (see the Appendix for details).

In each step we re-estimate one set of parameters conditionally, given current values for the other three. This technique is known as "Gibbs sampling," and converges to the correct distributions for each of the three sets of parameters. Since there is a lot of random variation, any particular iteration may provide either better or worse fit of the model to the data. However, the general trend is for successive iterations to provide better fit until improvement is no longer possible. When that occurs we consider the iterative process to have converged.

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 **D** are saved to the hard disk. The final values of the part-worths for each individual, and also of **D**, are obtained by averaging the values that have been saved.

The Metropolis Hastings Algorithm

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

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

We first calculate the relative probability of the data, or "likelihood," given each set of part-worths, $\boldsymbol{\beta_o}$ and $\boldsymbol{\beta_n}$. 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 each value of beta:

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

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 $\boldsymbol{\beta_o}$ and $\boldsymbol{\beta_n}$, given current estimates of parameters α and **D** (that serve as "priors" in the Bayesian updating). 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 value of beta:

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

Call the resulting values $\mathbf{d_o}$ and $\mathbf{d_n}$, respectively.

Finally we then calculate the ratio:

$$\mathbf{r\ = p_n\ d_n\ /\ p_o\ d_o}$$

Recall from the discussion of Bayesian updating that the posterior probabilities are proportional to the product of the likelihoods times the priors. The values $\mathbf{p_n}$ and $\mathbf{p_o}$ are proportional to the likelihoods of the data given parameter estimates respectively. The values $\mathbf{d_n}$ and $\mathbf{d_o}$ are proportional to the probabilities of drawing those values of $\boldsymbol{\beta}_\mathbf{n}$ and $\boldsymbol{\beta}_\mathbf{o}$, respectively, from the distribution of part-worths, and play the role of priors. Therefore, **r** is the ratio of posterior probabilities of $\boldsymbol{\beta}_\mathbf{n}$ and $\boldsymbol{\beta}_\mathbf{o}$, given current estimates of α , **D**, and σ, as well as information from the data.

If **r** is greater than or equal to unity, $\boldsymbol{\beta}_\mathbf{n}$ has posterior probability greater than or equal to that of $\boldsymbol{\beta}_\mathbf{o}$, and we accept $\boldsymbol{\beta}_\mathbf{n}$ as our next estimate of beta for that individual. If **r** is less than unity, then $\boldsymbol{\beta}_\mathbf{n}$ has posterior probability less than that of $\boldsymbol{\beta}_\mathbf{o}$. In that case we use a random process to decide whether to accept $\boldsymbol{\beta}_\mathbf{n}$ or retain $\boldsymbol{\beta}_\mathbf{o}$ for at least one more iteration. We accept $\boldsymbol{\beta}_\mathbf{n}$ 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 $\mathbf{p_n}$ will be larger than $\mathbf{p_o}$, which will tend 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 values that best fit 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 part-worths that best characterizes our uncertainty about 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 $\boldsymbol{\beta}_\mathbf{o}$ to $\boldsymbol{\beta}_\mathbf{n}$. 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' part-worths. 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.

Saving results for thousands of iterations can consume a large amount of storage space. If the only use for those estimates will be to average them to obtain a single best estimate for each respondent, then that average can be accumulated "on the fly" without actually storing the intermediate results.

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.

The Role of Constraints

We may know from other sources that every respondent should prefer one attribute level to another. For example, everyone should prefer higher quality to lower quality, and lower cost to higher cost. However, unless such order relationships are actually enforced, estimated part-worths are frequently found to violate them.

Many authors have pointed out that conjoint results can be improved by constraining part-worths (Srinivasan *et al*., VanderLans *et al*., Allenby *et al*.1995). ACA usually profits less than other conjoint methods from imposing constraints, because the self-explicated information from the initial part of the ACA interview already has a strong influence on the final part-worths. However, every study known to us that has examined the question has found constraints to have at least moderate benefit, even for ACA.

For this reason, we have built the capability of enforcing order constraints into ACA/HB. There is a simple and elegant way of doing this that we shall describe below. However, first we feel it necessary to point out that such constraints are incompatible with the probability model underlying HB estimation. Therefore, when constraints are used, this technique should probably not be referred to as "Hierarchial Bayes." ("Hierarachical Sawtooth" would be more correct!).

We use a method known as "rejection sampling" which considers only those beta candidates that conform to the desired constraints. It is our understanding that estimates of beta obtained in this way are proper, and in fact we find they are consistently superior to unconstrained estimates in predicting holdout choice concepts. However, we also understand that when betas are constrained in this way, the covariance estimates in the **D** matrix are not correct. We have urged our academic colleagues to come up with a practical way of enforcing constraints that is more appropriate for the HB model. Until that happens, we believe that the method we use will produce results that are almost always better than those of unconstrained solutions.

The self-explicated portion of the ACA interview asks the respondent about relative preferences for each level of each attribute, and also about attribute importance, measured in terms of the differences in part-worth between the best and worst level of each attribute. (SSI Web v6 permits dropping the self-explicated importance questions, in which case relatively large sample sizes are desirable and ACA/HB is required for data analysis.) ACA/HB has the capability of constraining estimated partworths so as to have corresponding orders within attributes as well as corresponding differences between attributes. (It should be noted that the relative preferences for each level of each attribute typically provide more useful information than the between-attribute importance ratings. In some situations, the between-attribute importance ratings can misinform part-worth estimation. For this reason, you can choose to constrain using only the within-attribute information, though that is not the default software setting.)

This is done in the selection of each new candidate $\boldsymbol{\beta}_\mathbf{n}$ in the Metropolis Hastings algorithm. If constraints are employed, then the covariance matrix for the jumping distribution is taken to be proportional to the identity matrix rather than to **D**, and the random draws of each element of $\boldsymbol{\beta}_\mathbf{n}$ are tested to see whether they conform to the desired constraints, considering the current values of all the other elements. Up to 10 trial draws are attempted for each element. If none is acceptable the old value for that element, which always satisfies the constraints, is retained until the next iteration.

Estimation Models

Early versions of ACA (Version 3 and before) used a regression scheme for estimating part-worths, which consisted of two matrix equations:

$$\mathbf{I} \quad \boldsymbol{\beta} \quad - \quad \mathbf{p} \quad = \quad \mathbf{e}_1$$

$$\mathbf{X} \quad \boldsymbol{\beta} \quad - \quad \mathbf{y} \quad = \quad \mathbf{e}_2$$

where **I** is the identity matrix,

X is a design matrix describing the pairs

p is a vector of “priors” from the self-explicated section

y is a vector of answers to the pairs section, after subtracting a constant corresponding to the value of the mid-scale position.

$\mathbf{e_1}$ and $\mathbf{e_2}$ are vectors of residuals.

The first equation expresses the difference between beta and the self-explicated prior part-worths as a vector of differences $\mathbf{e_1}$, while the second equation expresses the difference between the respondent's answers to paired comparison questions and the models predictions of them as a second vector of differences, $\mathbf{e_2}$. Minimization of the sum of squares of elements of $\mathbf{e_1}$ has the effect of making the estimated part-worths similar to the priors. Minimization of the sum of squares of $\mathbf{e_2}$ has the effect of making the predicted answers ($\mathbf{X}\ \beta$) similar to the actual answers. In the early versions of ACA, part-worths were estimated by minimizing the sum of those two sums of squares.

ACA/HB offers two options: one is whether to use the above regression tableau, which attempts to minimize the sum of squares of *both* $\mathbf{e_1}$ and $\mathbf{e_2}$, or to use just the bottom part of the tableau, minimizing only $\mathbf{e_2}$. The second option is whether or not to employ constraints. These two options result in four possible estimation models:

Fit Pairs Only with Constraints uses the self-explicated information only to provide initial approximations to the betas and information about how elements of beta should be constrained, both within and between attributes. However, no consideration is given to minimizing $\mathbf{e_1}$. In this approach, the self-explicated information affects the final solution only in an ordinal way, so no assumptions need be made about the variances of elements in $\mathbf{e_1}$. This approach seems to solve ACA's long-standing problem of the potential for scale incompatibilities between its two kinds of data. Fortunately, this method also appears to produce the best part-worths of any estimation method for ACA that we have examined. This is the default method for ACA/HB.

Fit Pairs & Priors with Constraints fits both the self-explicated and paired-comparison information, minimizing the sum of squares of elements of $\mathbf{e_1}$ plus those of $\mathbf{e_2}$. It also constrains the part-worths to have desired within-attribute orders and between-attribute importance differences. This method of estimation is similar in philosophy to earlier versions of ACA. It seems to work quite well, though usually not quite so well as the somewhat simpler method described immediately above.

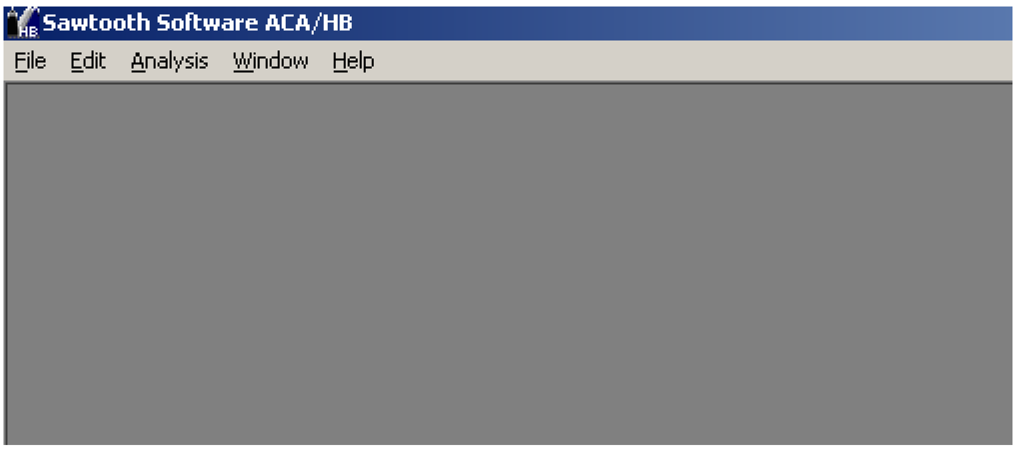
Fit Pairs Only without Constraints uses the self-explicated information only to provide initial approximations to the betas. Although this may speed convergence, the self-explicated information has no effect on the final solution. This method is the most "pure" of the four possibilities, since it avoids problems of scale incompatibility and does not use constraints that are incompatible with the underlying probability model. Unfortunately, however, it seems to be the worst of the four possibilities in predicting holdout concepts.

Fit Pairs & Priors without Constraints fits both the self-explicated and paired-comparison information, minimizing the sum of squares of elements of $\mathbf{e_1}$ plus those of $\mathbf{e_2}$. Since no constraints are used, this method of estimation is very similar to earlier versions of ACA. It works reasonably well, but not quite so well as either method employing constraints. This is the most effective approach for the user who wishes to forego constraints so as to remain completely true to the HB model.

Using the ACA/HB Module

Opening a Project or .ACD File

After the initial splash screen, you see the main menu with its **File**, **Edit**, **Analysis**, **Window** and **Help** options. You can open a recently opened ACA/HB study file (studyname.acahb) by selecting from the list of recently used projects. Or, you can click **File | Open...** to open another ACA/HB project or an existing studyname.ACD file (the data file produced by ACA or ACA/Web systems or that you create using your own resources).



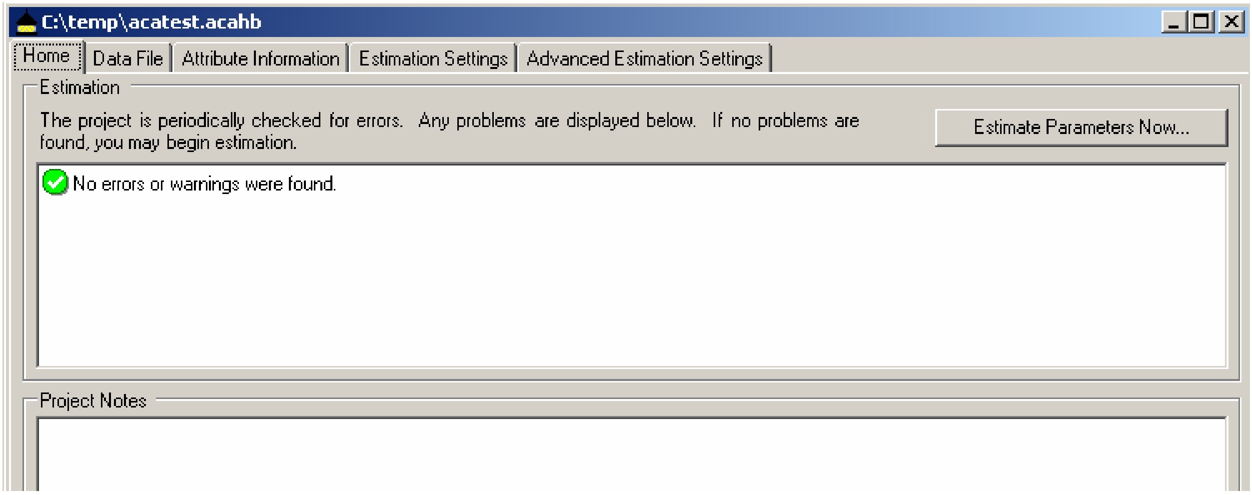
When you click **File | Open...**, the following options are available:

> **Open a data file in the .ACD Format**
> If you collected ACA data using Sawtooth Software's ACA or ACA/Web systems, you should use that system to export a studyname.ACD file. Click this option and browse to the studyname.ACD file.
>
> **Open an existing ACA/HB project**
> Click this option to open an existing ACA/HB v3 project with a .acahb extension.

Saving the Project

Once you have opened a project using either of the methods above and have configured your desired settings for the HB run, you can save the project by clicking **File | Save**. The settings for your ACA/HB run are saved under the name **studyname.acahb**. If you want to save a copy of the project under a new name (perhaps containing different settings), click **File | Save As** and supply a new project (study) name. A new project is stored as **newstudyname.acahb**.

After you open an existing project or a .ACD file, the main project window is displayed, with five main tabs: **Home**, **Data File**, **Attribute Information**, **Estimation Settings**, and **Advanced Estimation Settings**.



Two panels are shown on the *Home Tab*. The first reports any error messages for the study. The second is a workspace in which you can write notes, or cut-and-paste information from your HB runs for your documentation and review.

The *Home Tab* also includes the ***Estimate Parameters Now...*** button, which is the button you click when you are ready to perform HB estimation.

Performing HB Estimation

When you click ***Estimate Parameters Now...***, two things happen. First ACA/HB makes temporary binary data files that can be read much faster than the studyname.ACD file. Preparation of data files takes a moment, and then you see a screen like the following:

```
ACA/HB Build Process (6/21/2006 1:53:47 PM)
=====================================================
Data File: C:\temp\ACATEST.ACD

Attribute             Levels
-----------------------------------------------------
Brand                 3
Warranty              3
Microprocessor        3
Lab assistants        2
Ergonomic keyboard/m  2
Hard drive            3
RAM                   3
Modem/Internet acces  2
Price                 3

The number of parameters to be estimated is 15.

Constraints will be used on all attributes.

Build includes 80 respondents.
```

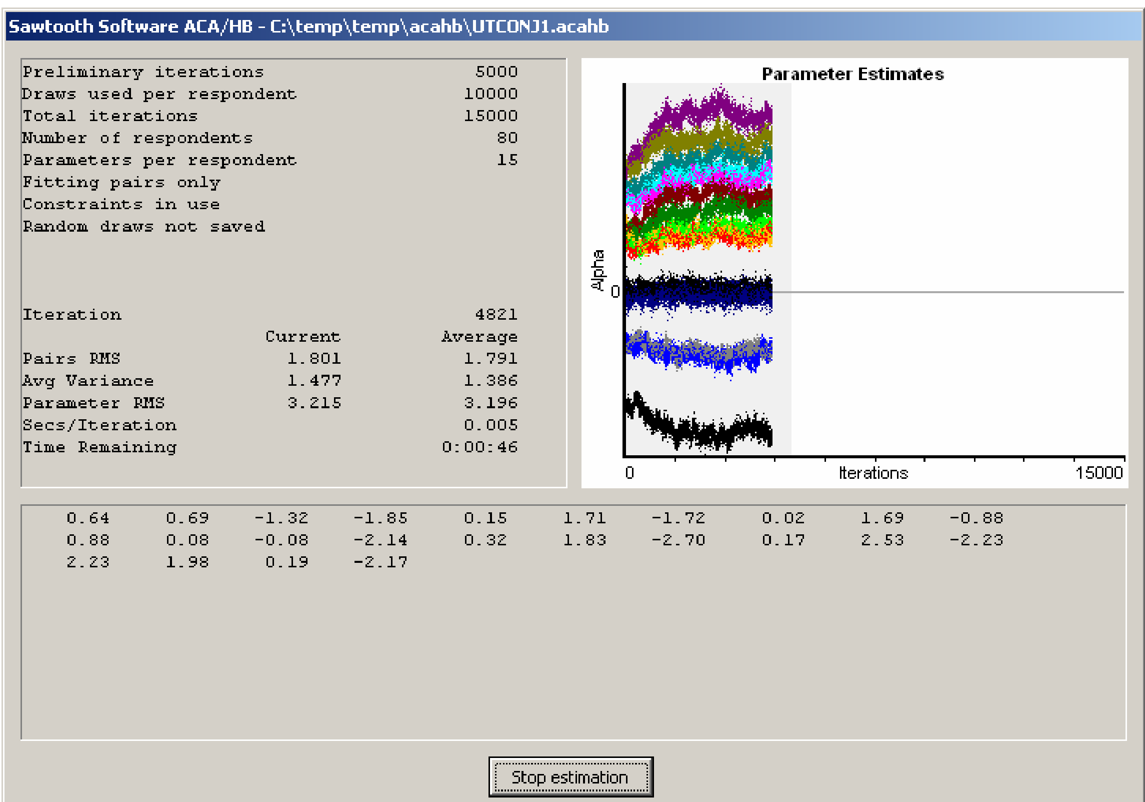
```
There are 1440 pairs in total, or an average of 18.0 pairs per respondent.

The build file 'C:\DOCUME~1\BB~1\LOCALS~1\Temp\tmpA52.tmp' was built successfully.
```

The first portion of the report identifies the source .ACD file. Next, the attribute list is shown, indicating the number of levels for each attribute. The number of parameters to be estimated and number of respondents included is displayed (the number of parameters is equal to the total number of levels minus the total number of attributes, due to dummy coding). The number of respondents and total pairs in the data set are also reported.

If you are satisfied with the way your data have been prepared, click ***Continue with estimation*** to begin the HB iterations. If not, click ***Do not estimate now*** to return to the main menu.

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:



These are results for an actual data set, obtained relatively early in the computation. The information at the top of the screen describes the settings that were chosen before the computation was begun. This run uses the default settings of 5,000 initial iterations, followed by 10,000 further iterations during which each iteration is used, but the random draws themselves are not saved to disk.

At the time this screen print was made, the 4,821st iteration had just been completed. A graphic shows a history of the estimates of respondent parameters (elements of alpha) to this point in the computation. This graphic is useful for assessing whether convergence has been reached. The graphic is divided into two regions, a gray region at the left, which represents the initial "burn in" iterations, prior to assuming convergence, and the white region at the right in which the subsequent draws are used to create point estimates of the parameters for each respondent.

The information in the two columns in the middle-left of the screen provides a detailed summary of the status of the computation, and we shall examine those values in a moment. Also, an estimate of the time remaining is shown. 46 seconds are required to complete this computation. This information is updated continuously.

At the bottom of the screen is the ***Stop estimation*** button. When this is pressed, the current iteration is finished and the current status of the computation is saved to disk for potential re-starting later. If the ***Stop estimation*** button is clicked during the second stage of estimation (the gray region of the graphic, after 5,000 iterations in this case) after we've assumed convergence and begun to use subsequent draws, the run will be halted and the current status saved, but the results from previous iterations will be deleted. When the computation is restarted all of the iterations during which results are to be used will be repeated.

We now describe the statistics displayed on the screen. There are two columns for most. 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:

new average = .01*(new value) + .99*(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 iterations have about 99% influence. Because the values in the first column tend to jump around quite a lot, the average values are more useful.

We should explain that during the iterations the part-worths are re-coded so the first level for each attribute has a value of zero (and thus can be ignored) and others are measured with respect to it.

When the part-worths are prepared for output, they are re-transformed so as to have an average of zero within each attribute. The statistics on this screen describe properties of part-worths for levels other than the first, expressed as differences from the first level, for each attribute.

If you are fitting priors (we are not in this example), the first statistic shown on the screen is **Prior RMS**. This is the root mean square error when using the current betas as estimates of the "prior" part-worths computed from each respondent's self-explicated preference and importance judgments. It's a measure of the extent to which the current estimates of part-worths are different from those estimated from the initial part of the ACA interview. This has a small value during the first few iterations, because those self-explicated estimates are taken as the initial values for estimating the betas. This statistic is only available if you choose to fit the priors data. If you choose the "pairs only" option, this line is not shown.

Pairs RMS is the root mean square error when using the current betas to estimate respondents' answers to the paired-comparison questions. Pairs RMS normally decreases during the first few iterations and then fluctuates around a constant value for the rest of the computation. This measure is the most useful indicator of how well the betas fit the data. When self-explicated data are also fitted, Pairs RMS is normally larger than Prior RMS. One reason for this is that the paired comparison answers are predicted by adding and subtracting several part-worths, so they have greater opportunity for error than the individual part-worth estimates themselves

Avg Variance is the average of the among-respondent variances for the estimated part-worths. It provides a generalized measure of the amount of heterogeneity among respondents.

Parameter RMS is the root mean square of all part-worth estimates, across all part-worths and over all respondents. It provides a generalized measure of the magnitudes of the part-worth estimates.

The studyname.log file contains a history of these measures, and may be inspected after the iterations have concluded, or at any time during a run by clicking ***Stop estimation*** to temporarily halt the iterative process. If values for the final few thousand iterations are larger than for the preceding few thousand, that should be considered as evidence that more iterations should be conducted before inferences are made about the parameters.

At the bottom of the screen are current estimates of average part-worths. The entire "expanded" vector of part-worths is displayed (up to the first 100 part-worths), including the final level of each attribute that is not counted among the parameters estimated directly.

Because ACA/HB uses a normal rather than a logit model it is more difficult to assess convergence than for CBC/HB, where the sizes of several statistics are more directly related to goodness of fit. For ACA/HB the primary indicators of goodness of fit are Pairs RMS and Avg Variance. We would, of course, like Pairs RMS to be small and Avg Variance to be large. Both statistics usually improve early in the iterative process: Pairs RMS usually decreases and Avg Variance usually increases. They soon stop improving, however, and begin to fluctuate randomly. Their failure to improve may be taken as evidence of convergence.

For users of CBC/HB it may be useful to remark that we have found ACA/HB to converge much more rapidly than CBC/HB. One reason may be that the "prior" estimates of the betas provided by the self-explicated part-worths are quite good, so the iterative process starts from an advantageous position. A second reason may be that the graded paired comparison scales in ACA provide more information in each answer than the choices made in CBC.

We have found 5,000 initial iterations usually to be more than sufficient for convergence with ACA/HB. 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 5,000, and then examining retrospectively whether these measures have been stable for the last several thousand iterations. If values for the final few thousand iterations display any trend whatsoever, that should be considered as evidence that more iterations should be conducted before inferences are made about the parameters.

Restarting

The computation may be thought of as having three stages:

> The initial iterations before convergence is assumed, and during which iterations are not saved for later analysis (the first 5,000 in the example above)
>
> The final iterations, during which results are saved for later analysis (the final 10,000 in the example above)
>
> If you have saved random draws, further time is required to sort them into order by respondent and compute average part-worth estimates for each respondent.

Your ability to restart the computation differs during these three stages. During the first two stages you may halt the computation at any time by clicking ***Stop estimation***. The current state of the computation will be saved to disk and you will later be able to restart the computation automatically from that point. If you click ***Stop estimation*** during the first stage, there will be no loss of information. If you do so during the second stage, you will be able to resume iteration at the same point, but any results already saved or accumulated will be lost, and you will have to do more iterations to replace those results.

Calibrating Part-Worths

The part-worths estimated by ACA/HB are appropriate for estimating shares of preference in scenarios involving competing products, but they are not appropriate for simulations of “purchase likelihood.”

If your ACA questionnaire included a “Calibration Concept” section, then those answers can be used to scale each respondent's part-worths so as to be useful with the “Purchase Likelihood” option of Sawtooth Software conjoint simulators.

The scaling is done with linear regression, predicting logits of each respondent's answers from his/her un-scaled utilities for those concepts. Part-worths are then scaled by the regression coefficient, and a fraction of the intercept is added to each part-worth, so that their sum contains the entire intercept term. The original file of part-worths is retained, and the rescaled values are saved in a new file.

Using the Results

At the end of the computation, several files are available containing the results:

Files named **studyname.hbu** and **studyname.csv** contain final part-worth estimates for each respondent. These are the averages of hundreds or thousands of draws either saved during the final stage of the iterations or accumulated on the fly, if those were not saved. The **studyname.csv** file is a comma-separated text-only file that may be directly opened with Excel™. Either of these files, perhaps after minor modification, may be used in a conjoint simulator. If using Sawtooth Software's market simulator (within the SMRT platform), the **studyname.hbu** file may directly be imported into your SMRT study, under ***Analysis | Run Manager | Import*** within SMRT's menu system.

If you have not saved random draws to disk, a file named **studyname.hbv** contains within-respondent variances for each respondent's part-worths. Its format is similar to that of **studyname.hbu**.

A file named **studyname.cov** contains successive random draws of the variances and covariances of the population distribution. The final point estimate of the variances and covariances is obtained by averaging the draws during the final stage of the iterations. Only the elements on and above the diagonal of the covariance matrix are saved in this file.

Finally, if you have saved random draws, a **studyname.dra** file is available with estimates of each respondent's part-worths for all the iterations from which those values were saved. This may be a very large file, since it contains potentially thousands of estimates of part-worths for each respondent. The data are arranged in order by respondent. This file can provide the raw data for analyses you may undertake using statistical software packages.

How Good Are the Results?

We have tested the ACA/HB module on many ACA data sets, and we shall report some of those results here. We are mainly interested in three questions:

> How well do part-worths estimated by ACA/HB predict holdout choices, as compared to conventional estimation methods offered by ACA?
>
> How successful are the various combinations of analysis options offered by ACA/HB?
>
> Do ACA/HB part-worths differ systematically from those estimated by conventional methods? Are they more or less vulnerable to the “Number of Levels” effect, or, separately, do they show a general flattening or sharpening of attribute importances?

We shall describe analyses of several data sets contributed by ACA users, and we wish to thank our colleagues at KBA Consulting Group, Lodestone Research, and TRAC, Inc. for permitting our use of those data sets.

We begin by examining the first two of these questions, using data sets containing hold-out choice concepts. Then we consider the third question, using other data sets that feature differing numbers of attribute levels.

Predicting Holdout Choices

The data set that we have examined most closely is one reported by Orme, Alpert, and Christensen (1997) in which 80 MBA students considered personal computers, using 9 attributes, each with two or three levels. We shall refer to this as “MBA Data.” Several kinds of data were collected in that study, but we shall consider only ACA data plus first choices from five holdout tasks, each of which contained three concepts.

ACA Version 3 and earlier estimated part-worths by combining the self-explicated and paired-comparison data in the same regression tableau, as described in the “Estimation Models” above. ACA Version 4 estimates part-worths by combining data from the two sections of the questionnaire with weights chosen to best-predict answers in a separate section of the questionnaire.

When applied to the holdout concepts, the two kinds of ACA part-worths produced these hit rates:

	MBA Data Hit Rate %
ACA Version 3 (“equal weights”)	67.50
ACA Version 4 (“optimal weights”)	66.25

Perhaps surprisingly, Version 3 did a slightly better job than Version 4 in this comparison, although with 80 x 5 = 400 choices, the difference is very small. In comparison, here are hit rates for several combinations of ACA/HB analysis options:

	MBA Data Hit Rate %
Pairs Only With Constraints	71.50
Pairs + Self-Explicated With Constraints	70.75
Pairs + Self-Explicated, No Constraints	69.35
Pairs Only, No Constraints	69.00

It is noteworthy that all four of these combinations are more successful than the conventional methods of estimation offered by ACA, either in Version 3 or Version 4.

Among these combinations, the two involving constraints are both better than the corresponding option without constraints. Finally, the most successful, by a narrow margin, is the combination that fits only the paired-comparison data, but uses the self-explicated data only as a source of information for within-and-between-attribute constraints on the part-worths.

We also examined a fifth combination, which is *not* offered as an option in ACA/HB. This is "Self-Explicated Only", which ignores the paired-comparison data, using just the self-explicated data, but benefiting from the "borrowing of information" characteristic of all HB applications. This hit rate was 63.50, less successful than either of the conventional ACA methods, and far less successful than "Pairs Only, No Constraints." This finding contradicts the belief sometimes stated that the self-explicated section of ACA contains most of the information in the ACA interview, and that the paired comparison section contributes comparatively little. Just the opposite appears to be true in this case: although the self-explicated section is certainly valuable, it provides less information than the paired comparison responses. (However, it should be noted that this is an exceptionally "clean" data set with few attributes, in which respondents were highly motivated to provide thoughtful answers. The result might have been different with a larger and more confusing number of attributes, or if respondents had been less well motivated.)

We also examined an important methodological question. The lengthy computational time of ACA/HB could be reduced if covariances among part-worths were assumed equal to zero. A version of the algorithm was tested in which this simplifying assumption was made. The time per iteration was cut almost in half, but the decrement in performance was substantial, with hit rates for the two combinations employing constraints each cut by approximately 3 percentage points. Based on this result, and similar results for other data sets, we decided not to provide a zero-covariance option.

The second data set contains information for 136 respondents, each of whom considered seven ACA attributes with a total of 20 levels, and also answered 12 choice tasks, each with three concepts. We again compare hit rates for two conventional ACA methods and several ACA/HB combinations:

	Second Data Set Hit Rate %
ACA Version 3 ("equal weights")	68.60
ACA Version 4 ("optimal weights")	66.80
Pairs Only With Constraints	68.68
Pairs + Self-Explicated With Constraints	68.52
Pairs + Self-Explicated, No Constraints	68.54

With this data set ACA3 again beat ACA4, and three combinations of ACA/HB were all approximately equal in performance to ACA3. Pairs Only With Constraints is again the winner by a very slight margin, but the main conclusion to be drawn from these results is that all five estimation methods do quite well, and there is little difference among them.

(Incidentally, the fact that ACA Version 3 outperforms ACA Version 4 in these two comparisons should not be interpreted as strong evidence for the superiority of Version 3. We regard these two results as something of a curiosity. We know that Version 3 is more vulnerable to the "Number of Levels" effect than Version 4, and that this effect is even stronger in responses to choice tasks. That artifact may be responsible for the slightly better performance of Version 3 in these cases. We have seen many examples of the superiority of Version 4 in other data sets, and although the difference in performance is relatively slight, Version 4 is usually superior.)

A third data set consists of ACA data for 960 respondents on 10 attributes with a total of 47 levels, together with first choices on six holdout tasks, each with three alternatives. Every respondent got the same holdout tasks, so we can look not only at hit rates, but also the accuracy of share predictions.

Here are hit rates and Mean Absolute Errors for ACA Version 4, plus several combinations of ACA/HB options:

	Third Data Set	
	Hit Rate %	MAE
ACA Version 4 ("optimal weights")	45.12	7.92
Pairs Only With Constraints	50.50	6.55
Pairs + Self-Explicated With Constraints	49.76	6.76
Pairs + Self-Explicated, No Constraints	48.66	5.87

As with the MBA data, all of these options provided better hit rates than conventional ACA estimation. ACA Version 3 was not examined in this comparison, nor was the Pairs Only, No Constraints" combination for ACA/HB. "Pairs Only With Constraints" was again most successful, and for the combinations fitting self-explicated data, the one using constraints was again the more successful.

For measuring success in prediction of shares we use mean absolute error (MAE) between actual and predicted shares of choice. The share predictions were done using the Randomized First Choice method, adding Gumbel attribute error with standard deviation of unity (Huber, Orme, and Miller 1999). All ACA/HB combinations were again superior to conventional estimation by ACA Version 4. However, among those combinations there was an interesting reversal. The method with the least successful hit rate had the most successful (smallest) MAE. We have seen the same result with other data sets: using constraints usually helps hit rate, but sometimes degrades share predictions.

As a fourth example, we constructed an artificial data set for which the "true" values were known, and then tested the ability of each ACA/HB combination to recover those values.

First, to develop reasonable part-worths for each respondent, ACA/HB was run on the MBA data with 20,000 initial iterations, followed by 50,000 during which every tenth was averaged. The resulting part-worth estimates were taken as "true" values for a fictitious sample of 80 artificial respondents.

Next, a data simulator was used to construct a data file containing new answers to the paired comparison questions occurring in the MBA data set. This was done for each respondent by accumulating the true utility difference for products in paired comparison, adding a normal random perturbation with standard deviation of unity, adding a constant of 5, and then rounding the result to be an integer between 1 and 9.

That data file was analyzed by three different combinations of ACA/HB options. In each case convergence seemed to occur within the first 3,000 iterations. A total of 25,000 iterations were done by each combination of options, and after each 1,000 iterations the average part-worths were assessed by three measures, each computed over all respondents and all part-worths within respondent:

Correlation between "true" and estimated part-worths,
RMS error between "true" and estimated part-worths,
Scale Factor, the ratio of RMS element sizes of estimated/true values.

Here are results for this data set:

Recovery of Part-Worths in Artificial Data Set

	Correlation	RMS Error	Scale Factor
Pairs Only With Constraints	0.990	0.240	1.001
Pairs + Self-Explicated With Constraints	0.987	0.342	0.744
Pairs + Self-Explicated, No Constraints	0.978	0.426	0.693

The first option, “Pairs Only With Constraints,” is most successful with respect to all three measures. It produces the highest correlation between true and estimated part-worths, has the smallest RMS error between true and estimated values, and produces estimates that are scaled almost exactly correctly.

The second option, “Pairs + Self-Explicated With Constraints,” is second best. Its correlation is quite high, but the estimated part-worths are scaled about 25% too small. This is due to scale incompatibility between the self-explicated “priors” and the paired-comparison responses. In trying to fit both kinds of data, the algorithm has produced estimates that are scaled down to better fit the self-explicated data, and as a result the fit to the paired-comparison answers has been sacrificed.

The third option, “Pairs + Self-Explicated, No Constraints” has a somewhat lower correlation, and has also produced scaled-down estimates that fit the self-explicated information better at the expense of the paired-comparison answers.

The Shape of ACA Part-Worths

The first versions of ACA made assumptions about the shape of the part-worths within an attribute. In early versions of ACA, the only option in the priors section was to rank-order attribute levels. Prior part-worths therefore had equal spacing among levels within an attribute. Since the pairs information made no assumptions about the shape of the utility function within an attribute, the final part-worths (after combining priors and pairs) were not so regular, but the influence of the priors biased the results toward equal spacing among levels.

Starting with version 4 of ACA, two changes were made to help relax the assumption of equidistant part-worths within attributes. First, the “optimal weighting” option customized the weight applied to priors and pairs to best fit each respondent’s calibration concept evaluations. Therefore, if the equidistant prior part-worths did not fit the data well, the priors were given less influence in a person’s final part-worths. Second, an option was provided for rating levels within each attribute rather than ranking them. The metric information from ratings avoided the equal spacing assumption, but doesn’t seem to have found widespread use in practice. It can seem unnatural for some respondents to provide rating data for levels of attributes with known orders (such as price). Confused or inattentive respondents can provide incorrect information (reversals) that conflict with rational ordinal relationships. Moreover, requiring ratings leads to longer surveys, especially with ordered attributes, for which rankings could be assumed.

Using ACA/HB seems to offer a better way to combine the information from priors and pairs. The default method for ACA/HB (pairs only with constraints) uses only the rank-order information from the priors. Once the ordinal relationships are established, the shape of the utility function across the levels within each attribute is determined solely from the metric pairs information.

In large ACA designs, it is common for only a subset of the total attributes (the most important ones) to be taken forward to the pairs section. With traditional ACA estimation, if the only information about levels within an attribute comes from prior rankings, the final part-worths for attributes not included in the pairs section have equal spacing. For large designs, this systematically biases less important attributes toward equal spacing within the levels. With ACA/HB, this bias is less likely, since information about the relative values of levels within an attribute not included in a given respondent's pairs section is borrowed from respondents whose pairs section included that attribute.

How ACA/HB Handles Attributes Not Included in Pairs

One complaint regarding ACA (especially with particularly large designs) is that it can lead to too little discrimination between attributes in terms of attribute importance. Consider a design including 20 attributes, but for which only 12 attributes (the most important ones for each respondent) are taken forward to the pairs section. Let's further assume that a four-point importance scale was used, and that all eight attributes not taken into pairs for a given respondent were given a rating of one (least important). Under traditional ACA estimation, the final part-worths for this respondent will reflect equal importance across the least important eight attributes. In contrast, ACA/HB borrows information from other respondents regarding the relative importance among those eight attributes, and more discriminating, reasonable importances can result.

Comparing ACA Version 4 and ACA/HB Importances

In this section, we'll limit our analysis to four real ACA data sets, with the following numbers of attributes, respectively, 10, 9, 21, 18. The number of levels within attributes varied from 2 to 5. We computed part-worths using hierarchical Bayes estimation, in three different ways: a) fit priors and pairs, b) fit priors and pairs plus constraints, and c) fit pairs only plus constraints (default method). We zero-centered the average importances within each of the four studies, then computed correlations among the importances for these three methods and for ACA version 4.

Correlations among Attribute Importances

	ACA v4 (OLS)	ACA/HB fit priors and pairs	ACA/HB fit priors and pairs + constraints
ACA/HB fit priors and pairs	0.973		
ACA/HB fit priors and pairs + constraints	0.970	0.988	
ACA/HB fit pairs + constraints	0.958	0.984	0.998

It is clear from this table that there is a great deal of similarity between the patterns of importances derived from the different methods.

There were a total of 58 attributes across these four data sets. We can gain an overall feel for the degree of discrimination among the attributes by computing the standard deviation across the 58 attribute importances (zero-centered within each study):

Standard Deviation of Attribute Importances

ACA Priors	1.2
ACA v4 (OLS)	2.0
ACA/HB fit priors and pairs	2.5
ACA/HB fit priors and pairs + constraints	2.2
ACA/HB fit pairs + constraints	2.2

This analysis suggests that the "slope" of the attribute importances is similar, but a bit "steeper" when using HB estimation than ACA v4. A steeper slope leads to greater discrimination between more important and less important attributes. Overall, these two analyses suggest that the ACA/HB importances are quite similar to the ACA v4 OLS importances, for attributes with five or fewer levels.

ACA/HB and The Number of Levels Effect

The Number of Levels (NOL) effect is well documented in conjoint literature. The NOL effect occurs when some attributes are defined using more levels than others. If an attribute is defined using more levels (while holding the total range of variation constant) it tends to get more importance. The bias can be quite large for traditional full-profile conjoint methods (such as card-sort and choice-based conjoint).

One of ACA's strengths is that it is less susceptible than other conjoint methods to the NOL effect. Still, a NOL effect can be demonstrated for ACA. The NOL effect may be less severe in ACA because of its use of the priors information. The priors' stated importances are not affected by NOL in the case of ordered attributes, where respondents are only shown the best and worst levels of each attribute. It has recently been shown (Orme, 1998) that the optimal weighting procedure in ACA v4 helps reduce the NOL effect even more than the estimating method used in earlier ACA versions. The question we will examine in this section is to what degree the importances computed under ACA/HB are subject to the NOL effect. Again, we examine the four real ACA data sets (each with between 2 to 5 levels per attribute) used in the previous section.

Using the 58 rows of zero-centered (within study) average attribute importances as observations, we estimated a multiple regression model for each estimation method with the following form:

$$Y = b_0 + b_1X_1 + b_2X_2$$

Where:

Y = estimated importance for attribute *i* for method *j*
X_1 = number of levels in the attribute
X_2 = attribute's average prior stated importance

If $b_1 > 0$, then the method demonstrates a NOL effect. If $b_2 > 1$, then the slope of the estimated attribute importances for that method is greater than the slope from the stated importances in the priors section.

The results are as follows for various methods of computing ACA part-worths (standard errors in parentheses):

	b_1 (NOL)	b_2 (Slope)
ACA v3	0.71 (0.10)	1.52 (0.07)
ACA v4	0.22 (0.09)	1.54 (0.06)
ACA/HB fit priors and pairs	0.59 (0.12)	1.75 (0.09)
ACA/HB fit priors and pairs + constraints	0.48 (0.09)	1.60 (0.07)
ACA/HB fit pairs only + constraints	0.54 (0.10)	1.58 (0.08)

Examining b_1, we note that the ACA v4 part-worths are the least affected by the NOL effect, but the effect remains statistically significant ($t = 0.22/0.09 = 2.44$). All of the ACA/HB methods demonstrate a significant NOL effect, but in each case, the NOL effect is less than the effect for ACA v3 (though the differences are not statistically significant). It is worth noting that ACA v3's NOL effect has been shown to be significantly less than traditional conjoint methods.

How large is the NOL effect for ACA/HB methods relative to the priors? The average attribute importance across these four data sets was 6.9. In the case of the largest b_1 term for ACA/HB methods (0.59), we can infer that adding one level to an attribute increases its importance by $[(6.9+ 0.59)/6.9] – 1 = 9\%$. Doubling the number of levels for a 2-level attribute would result in an artificial increase in importance of 18%.

The b_2 term captures differences in the slope of the importances, after accounting for the NOL effect. All of the methods for computing ACA part-worths displayed steeper slopes ($b_2 > 1.0$) than the self-explicated importances. This analysis suggests that the ACA/HB importances are at least as steep as or a little bit steeper than the ACA OLS importances.

In summary, ACA/HB part-worths are subject to a slightly larger NOL effect than ACA v4 part-worths. Still, the NOL effect is probably less than when using ACA v3, which has been demonstrated to be far less vulnerable to the NOL effect than traditional full-profile conjoint methods. In our opinion, the benefits of using HB estimation, compared to ACA v4, outweigh the costs of a slightly larger NOL effect.

Conclusions

In all of the data sets we have examined, results from ACA/HB have equaled or surpassed results from conventional ACA methods. The second data set presented above is the worst case we have seen, and in that case all ACA/HB options essentially tied ACA Version 3 and exceeded ACA Version 4.

Our decision to make “Pairs Only + Constraints” the default option in ACA/HB was based on the results reviewed here as well as others that produced similar findings.

We regret that ACA/HB’s use of constraints appears to be at odds with the probability theory underlying the HB model. We hope our academic colleagues can provide a practical and theoretically correct way to achieve the same benefit. For the present, however, the practical evidence so strongly favors the use of constraints that we do not hesitate to recommend them. For users who wish to avoid constraints, “Pairs + Self-Explicated, No Constraints” works quite well, although it is not the best option we can offer.

Finally, for the purist who wishes to avoid constraints and not to use a regression model that combines observations with different variances, “Pairs Only, No Constraints” seems to work reasonably well, and appears to be completely unobjectionable statistically.

Appendix: Computational Procedures

In our attempt to provide an intuitive understanding of the HB estimation process, we omitted some details that we shall provide here.

With each iteration we re-estimate α, a vector of means of the distribution of individuals, the covariance matrix $\mathbf{D}$ of that distribution, a vector β_i of part-worths for each individual, and σ, the standard deviation of the errors of fit of the model to the data. We have already described the estimation of the betas in some detail. Here we provide details for the estimation of α, $\mathbf{D}$, and σ.

Random Draw from a Multivariate Normal Distribution

Often in the iterative computation we must draw random vectors from multivariate normal distributions with specified means and covariances. We now 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 a vector $\mathbf{u}$ and another vector $\mathbf{v} = \mathbf{T}\,\mathbf{u}$. Suppose the elements of $\mathbf{u}$ are normal and independently distributed with means of zero and variances of unity. Since for large $\mathbf{n}$, $1/\mathbf{n}\ \Sigma_n \mathbf{u}\,\mathbf{u}'$ approaches the identity, $1/\mathbf{n}\ \Sigma_n \mathbf{v}\mathbf{v}'$ approaches $\mathbf{D}$ as shown below:

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

where the symbol => 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 unit 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 $\mathbf{1/n}\ \ \mathbf{D}$. 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 $\mathbf{1/n}\ \ \mathbf{D}$.

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 $\mathbf{H}$ that combines the prior information with current estimates of α and β_i

$$\mathbf{H} = \mathbf{pI} + \Sigma_n (\alpha - \beta_i)(\alpha - \beta_i)'$$

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

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

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

$$\mathbf{S} = \Sigma_N (\mathbf{T}\,\mathbf{u_i})(\mathbf{T}\,\mathbf{u_i})'$$

Finally, our estimate of **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 **D**, except that σ^2 is a scalar instead of a matrix.

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

References

Allenby, G. M., Arora, N., and Ginter, J. L. (1995) "Incorporating Prior Knowledge into the Analysis of Conjoint Studies," *Journal of Marketing Research*, 32 (May) 152-62.

Allenby, G. M., Arora, N., and Ginter, J. L. (1998) "On the Heterogeneity of Demand," *Journal of Marketing Research*, 35, (August) 384-389.

Allenby, G. M. and Ginter, J. L. (1995) "Using Extremes to Design Products and Segment Markets," *Journal of Marketing Research*, 32, (November) 392-403.

Chib, S. and Greenberg, E. (1995) "Understanding the Metropolis-Hastings Algorithm," American Statistician, 49, (November) 327-335.

Gelman, A., Carlin, J. B., Stern H. S. and Rubin, D. B. (1995) "Bayesian Data Analysis," Chapman & Hall, Suffolk.

Green, P. E., Krieger, A. M., and Agarwal, M. K. (1991) "Adaptive Conjoint Analysis: Some Caveats and Suggestions," *Journal of Marketing Research*, 28 (May), 215-22.

Huber, J., Orme B. K., and Miller, R. (1999) "Dealing with Product Similarity in Conjoint Simulations," *Sawtooth Software Conference Proceedings*, Sawtooth Software, Sequim.

Johnson, R. M. (1998) "The CBC/HB Module for Hierarchical Bayes Estimation," Sawtooth Software Technical Paper.

Lenk, P. J., DeSarbo, W. S., Green P. E. and Young, M. R. (1996) "Hierarchical Bayes Conjoint Analysis: Recovery of Partworth Heterogeneity from Reduced Experimental Designs," *Marketing Science*, 15, 173-191.

Moore, W. L., Mehta, R. B. and Pavia, T. M. (1994) "A Simplified Method of Constrained Parameter Estimation in Conjoint Analysis," *Marketing Letters*, 5 (April), 173-81.

Orme, B. K. (1998) "Reducing the Number-of-Attribute-Levels Effect in ACA with Optimal Weighting," *Sawtooth Solutions*, Fall, 5-7.

Orme, B. K., Alpert, M. I., and Christensen, E. (1997) "Assessing the Validity of Conjoint Analysis – Continued," *Sawtooth Software Conference Proceedings*, 209-25.

Srinivasan, S., Jain, A. K., and Malhotra, M. (1983) "Improving Predictive Power of Conjoint Analysis by Constrained Parameter Estimation," *Journal of Marketing Research*, 20 (November) 433-38.

Van der Lans, I. A., Wittink, D. R., Huber, J, and Vriens, M. (1992) “Within-and Across-Attribute Constraints in ACA and Full Profile Conjoint Analysis,” *Sawtooth Software Conference Proceedings*, 365-80.

Wittink, D. R., McLauchlan, W. G., and Seetharaman, P. B. (1997) “Solving the Number-of-Attribute-Levels Problem in Conjoint Analysis,” *Sawtooth Software Conference Proceedings*, 227-40.