Sawtooth Software

RESEARCH PAPER SERIES

Summary of Van Horn's Comparison of Utility Constraint Methods for CBC in HB-MNL Estimation

Bryan Orme
Sawtooth Software, Inc.

3210 N. Canyon Rd., Provo, Utah
+1 801 477 4700
www.sawtoothsoftware.com

Summary of Van Horn's Comparison of Utility Constraint Methods for CBC in HB-MNL Estimation

Bryan Orme, Sawtooth Software

November 2024

(Kevin van Horn has done work in Bayesian inference and Bayesian computation since the late 90's, with applications in speech recognition, market research, and time series forecasting. He spent six years at market research firm The Modellers designing, implementing, and improving a variety of Bayesian models and related simulations and experimental design software.)

Executive Summary

This short article is a summary of a more complete article by Kevin Van Horn, of Bayesium Analytics, entitled, "Sign/Order Constraints on HMNL Part Worths: A Comparison of Approaches". Sawtooth Software engaged Van Horn in this effort, to further explore and compare different approaches to enforcing utility constraints in HB-MNL estimation of utilities for conjoint analysis (HMNL and HB-MNL refer to the same model).

Van Horn finds:

- HB-MNL models for CBC (Choice-Based Conjoint) data estimated using Stan's Hamiltonian Monte Carlo (HMC, with no-u-turn sampling—NUTS) and Sawtooth Software's CBC/HB (using Metropolis Hastings) did not differ much at all in terms of predictive accuracy.
- Sawtooth Software's utility constraint approach, Simultaneous Tying (ST), for constraining ordered attributes like price and speed performed as well in terms of predictive validity (within 1.96 standard deviations) as six different approaches Van Horn implemented for utility constraints using Stan's HMC. Van Horn performed both ELPD (expected log predictive density, using posterior draws) as well as out-of-sample predictions of choice (using posterior point estimates). For out-of-sample predictive accuracy, Sawtooth's approach tied for best predictive accuracy.
- Convergence statistics that academics and advanced Bayesian analysts often rely upon (ESS and Rhat) were considerably worse for Sawtooth Software's Metropolis Hastings algorithm with ST than Stan's HMC using some of the approaches Van Horn implemented for utility constraints.

Introduction and Motivation

For a previous research effort (as reported by Orme 2024), Sawtooth Software asked Van Horn to investigate speed, convergence, and predictive validity of Stan HMC, Variational Bayes, and Sawtooth's CBC/HB algorithm (HB-MNL using Metropolis Hastings). For CBC datasets up to a maximum of 27 parameters to estimate, Van Horn found that Stan HMC and Sawtooth's CBC/HB led to essentially equally good predictions and equivalent mean parameters, given the same amount of run time. Variational Bayes was considerably faster, but could sometimes lead to suboptimal results. Stan HMC led to better formal convergence diagnostics than CBC/HB.

As a follow-up to that previous research effort, Sawtooth Software asked Van Horn to investigate different methods of constraining utility parameters for ordered attributes in Stan HMC, comparing them to the method Sawtooth uses for implementing constraints in CBC/HB.

In the HB-MNL models often used to analyze discrete choice surveys, utilities (part worths) for some attributes have an obvious a priori ordering: holding all else equal, a lower price is (usually) better than a higher price; more memory is better than less memory; a longer warranty is better than a shorter warranty. For certain modeling situations (e.g., small sample sizes, noisy data, WTP estimation, and product or portfolio optimization) some researchers prefer to constrain attributes with known a priori preference order. In Sawtooth Software's most recent customer feedback survey, we found that our customers used utility constraints for 30% of their CBC studies conducted over the previous 12 months (Sawtooth 2024).

This raises the question of how one should modify the standard HB-MNL to enforce such ordering constraints. In 2000, Sawtooth Software's Rich Johnson investigated various methods for imposing utility constraints in CBC (Choice-Based Conjoint) studies, finding that a method he devised of Simultaneous Tying (ST) worked quite well (Johnson 2000). Since then, Sawtooth Software has offered ST as its default method for imposing utility constraints within CBC, ACBC (Adaptive CBC), and MaxDiff utility estimation using HB-MNL.

Sawtooth Software recently asked Bayesian modeling expert Kevin Van Horn to investigate newer methods for imposing utility constraints that may be implemented within Stan, for both Hamiltonian Monte Carlo and Variational Bayes algorithms. Van Horn investigated the following possible methods, in the context of utility estimation for CBC studies:

- Sawtooth Software's ST Method as used in CBC/HB and ACBC/HB

- Stan HMNL[1]: Thermometer Coding with an Exponential Transformation to Ensure the Sign of the Utility Deltas between Adjacent Levels
- Stan HMNL: Thermometer Coding with a Hinge Function Transformation to Ensure the Sign of the Utility Deltas between Adjacent Levels
- Stan HMNL: Thermometer Coding with a Quasi-Hinge Function Transformation to Ensure the Sign of the Utility Deltas between Adjacent Levels (four different variants of this model)

A Quick Primer on Thermometer Coding:

Like dummy- and effects-coding, thermometer coding (unary coding) avoids linear dependency and achieves identification in the MNL solution by coding the K levels of an ordered attribute using K-1 columns in the design matrix.

But, thermometer coding is a bit different than dummy coding. For example, if there are four levels of an ordered attribute such as price, we code the four levels as three columns in the design matrix as follows:

Level 1 (ref):	0 0 0
Level 2:	1 0 0
Level 3:	1 1 0
Level 4:	1 1 1

With thermometer coding (as with dummy coding), we achieve identification by setting the utility of the reference level (level 1) to zero. However, each other K-1 utility parameter (associated with columns 1-3) represents the utility delta associated with moving from the previous level to this level.

By constraining the sign of the thermometer coding utility parameters, one can impose utility constraints. For example, constraining as negative the three parameters for the three columns above leads to monotonically decreasing part worth utility parameters for the four levels of price.

As Van Horn describes, utilizing thermometer coding and sign constraints makes it much easier than approaches similar to ST to formulate utility constraints in Stan for ordered part worth coding for CBC studies.

[1] Hierarchical Multinomial Logit

Positivity Transforms in Stan HB-MNL

Once we've coded an attribute like price for a CBC study using thermometer coding, we are only required to impose sign constraints on the utility parameters. Two positivity transforms that force a positive sign, and of course may force a negative sign (by negating the raw utility, applying a positivity transform, and then re-negating the result) are an Exponential (accomplished by exponentiating the drawn utility parameter) and a Hinge function (accomplished by mapping beta to max (0,beta), i.e., it replaces negative utilities with 0, so that only nonnegative utilities result). Another option is a quasi hinge function, which is any continuous-derivative approximation to the hinge function.

The Data Sets

Van Horn tested different methods of imposing utility constraints in Stan HML for the following four CBC datasets:

Name	Type	# Attributes	Total # levels	Sample Size
Cruise	Std. CBC	6	27	600 calibration, 600 holdout
HDTV	Std. CBC	7	22	2939
FK	Alt-Spec CBC	12	39	2940
SKU34	Alt-Spec CBC	28	80	753

Some datasets (Cruise, HDTV) were designed with out-of-sample validation in mind (either a cell of holdout respondents or a limited number of blocks of the experimental design allowed for k-fold out-of-sample validation). Other datasets (FK, SKU34) required that Van Horn construct calibration and holdout choice tasks for validation.

Because Van Horn's comparison to Sawtooth Software's CBC/HB with ST is the comparison of most interest to Sawtooth Software users, we report on those results below.

Algorithms and Settings

Van Horn reports that for Stan the parameters were estimated using Hamiltonian Monte Carlo (HMC) to sample from the posterior distribution, along with Variational Bayes (VB) to sample from an approximation to the posterior. He implemented the models in the Stan modeling language, compiling and running it via the CmdStanR interface.

Van Horn reports using all default settings for Stan HMC except the following:

- init: set to 1 (initialize all parameters randomly between -1 and 1 in the unconstrained parameter space) instead of the default of 2. The default setting often resulted in initialization so far out into the tails of the posterior that numerical difficulties arose.
- chains, pchains: set to 4 chains run in parallel.
- iter_warmup: 1000.
- iter_sampling: 2000.

For Stan's variational Bayes, Van Horn used all default settings except the following:

- init: set to 1.
- iter (max # of optimization iterations): 60,000.
- algorithm: "meanfield." This enforces a diagonal covariance matrix for the approximating multivariate normal, which allows for differing posterior variances, but no nontrivial correlations.

On failure (nonzero return code), Van Horn reran the VB algorithm, up to a maximum of 3 tries. Failure can occur because VB's gradient-ascent algorithm finds that it is not converging to a (local) maximum.

Results:

For the Cruise data set (the same robust CBC dataset used in the 2016 Sawtooth Prize modeling challenge), Bryan Orme of Sawtooth Software provided the results (draws and point estimates) of a ST run, with the price attribute constrained. This run was based on 50K burn-in iterations and 50K used iterations. To mimic the usual process for CBC practitioners, for predictive accuracy results, Van Horn utilized point estimates of the part-worth utilities rather than draws. To ensure an apples-to-apples comparison of predictive accuracy, point estimates of draws were also used for the Stan HMC results.

Van Horn estimated six different Stan HMC models, using various approaches to implementing sign constraints (as described above).

Task-Level Cross-Validation

Sawtooth Software's ST had an RLH (Root Likelihood) task-level cross-validation predictive accuracy of 0.4122 (note: higher RLH is better). Across the six Stan HMC approaches Van Horn implemented for modeling the parameters and constraining price, the worst holdout prediction RLH was 0.4110 and the best 0.4128. (The differences in prediction did not meet the $p<=0.05$ significance level.)

Out-of-Sample Prediction of Holdout Tasks

Sawtooth Software's ST had an RMSE (Root Mean Squared Error) of prediction to the shares of choice across the 21 holdout tasks for the 600 holdout respondents of 0.0353 (note: lower error is better). Across the six Stan HMC approaches Van Horn implemented for modeling the parameters and constraining price, the worst out-of-sample predictive validity (RMSE) was 0.0361 and the best was 0.0353.

Convergence Diagnostics

For practitioners who report summary part worth utilities (via point estimates) and make predictions using market simulators, predictive accuracy is usually the most critical test of the quality and usefulness of the model. That said, academics and experts in Bayesian modeling care a great deal about convergence and quality of the posterior lower-level draws, which directly represent the uncertainty around each respondent's point estimates.

> Higher ESS (effective sample size) is a measure of the quality and independence of the posterior draws (lower autocorrelation, better representation of the full posterior distribution, leading to better estimation of the mean).
>
> Rhat measures how similar within-chain and between-chain variation are.

Sawtooth Software's ST run had some bad ESS values. The average ESS values were low, and in some cases uncomfortably low. The average Rhat values were a bit higher than desirable, with a few specific Rhat values quite high.

Thus, if researchers care about these ESS and Rhat measures of convergence, Sawtooth Software's HB utilizing Metropolis Hastings sampling with ST for implementing utility constraints should probably not be used.

Notes on Variational Bayes

Variational Bayes (VB) was problematic, with many failures to converge. Van Horn reports that although VB shows some promise for fast approximate model estimation, its unreliability poses a significant obstacle to its production use.

Should Sawtooth's Simultaneous Tying Always Be Used?

For Sawtooth Software customers, Van Horn's findings offer a great deal of comfort—that the HB approach with Sawtooth's utility constraints algorithm developed nearly 25 years ago still provides just as good of predictive results as newer algorithms (Hamiltonian Monte Carlo and various methods of constraining utilities using thermometer coding). However, we would note that the findings van Horn reports were based only on an ordered attribute

with five levels. We should also note that ST leads to unreliable upper-level estimates, so if upper-level estimates are needed for statistical testing, ST should be avoided.

Over the years, we've had numerous discussions with experts at SKIM (including Kevin Lattery, Kees van der Wagt, Michael Smith, and Jeroen Hardon) regarding utility constraints in CBC for ordered attributes. They've reported to us that Sawtooth's simultaneous tying approach can slow down[2] considerably when the number of levels of the ordered attribute exceeds about 10-12. For that reason, they often code ordered attributes using thermometer coding (imposing sign constraints on the utility parameters involved in thermometer coding) when using Sawtooth or other software such as R or Stan to conduct HB estimation for CBC datasets.

In the final section of this article, the author tests how long it takes to run HB-MNL estimation for a typical CBC dataset, when the ordered attribute has 10, 20, and 30 levels. We find for a typical CBC dataset with a 30-level ordered attribute, 60K iterations could be performed overnight on a modest performance laptop computer.

Speed Checks for Sawtooth's CBC/HB with ST Constraints

How much can the slowdown be for Sawtooth's ST constraints algorithm for an ordered attribute with many levels? The author tested Sawtooth's CBC/HB routine with its default ST utility constraints on a CBC data set with random respondents (a worst-case scenario for resolving utility constraints, since a lot of reversals are expected).

CBC Test Dataset:

n = 1000
12 tasks per respondent
4 concepts per task
6 attributes: 5,5,5,5,5,(10 or 20 or 30) levels + None

CBC/HB Settings:

HB-MNL estimation
40K burn-in iterations/20K used (double the default 20K/10K settings)

Hardware specifications (modest by 2024 standards for a laptop):

[2] This slowdown in speed for ST is expected, given Van Horn's observation that the ST algorithm requires in the worst case a number of passes (to resolve potential out-of-order part-worth estimates) that grows proportional to the square of the number of levels. For example, with 10 levels of an ordered attribute, there are 10(10-1)/2=45 potential pairs of utilities to check and remediate. With 20 levels there are 20(20-1)/2=190 pairs.

Lenovo, Thinkpad X1 laptop
1800 Mhz, 14-core processor
32-GB RAM

Speed of Computation Results (60K total iterations)

10-level ordered attribute (w/ 5 other attributes, 30 total parameters to estimate):

0.4 hours No constraints
0.9 hours ST utility constraints

20-level ordered attribute (w/ 5 other attributes, 40 total parameters to estimate):

1.0 hours No constraints
6.0 hours ST utility constraints

30-level ordered attribute (w/ 5 other attributes, 50 total parameters to estimate):

1.5 hours No constraints
15.4 hours ST utility constraints

The utility constrained model takes around twice as long to run as the unconstrained model for a 10-level ordered attribute; six times as long for a 20-level attribute, and ten times as long for a 30-level attribute. Even in the worst case of a 30-level ordered attribute, the constrained run could still be completed overnight (15.4 hours) on a modest performance laptop (the author has a colleague whose laptop is nearly twice as fast). Thermometer coding with sign constraints would be faster, but these runtimes for the ST constraints algorithm seem acceptable in practice. And, it avoids custom coding the thermometer parameters in the data file and the extra complexity of dealing with thermometer coded utility parameters for building the market simulator.

(Note: these run times are when using CBC/HB's batch mode, so that the software was not slowed down by painting the runtime results and history of parameters graph to the screen.)

References:

Johnson, R. M. (2000), "Monotonicity Constraints in Conjoint Analysis with Hierarchical Bayes," Technical Paper available at: https://sawtoothsoftware.com/resources/technical-papers/monotonicity-contraints-in-choice-based-conjoint-with-hierarchical-bayes

Orme, Bryan (2024), "A Comparison of HB-MNL Estimation via Metropolis Hastings (Sawtooth Software's CBC/HB Program), Hamiltonian Monte Carlo (via Stan), and Variational Bayes (via Stan)," Sawtooth Software Research Paper, available at: https://sawtoothsoftware.com/resources/technical-papers/a-comparison-of-hb-mnl-estimation

Sawtooth Software (2024), annual customer feedback survey.

Van Horn, Kevin (2024), "Sign/Order Constraints on HMNL Part Worths: A Comparison of Approaches," available by writing the author at: kevin@bayesium.com