Discrete Choice Models in R

Daniel Guhl & Sebastian Gabel

Institute of Marketing
School of Business and Economics
Humboldt University Berlin

and

So1 GmbH Berlin

Laundry Detergent Choice in Supermarket

Orange Juice Choice in Experiment

If these were your only options, which would you choose?

1 / 16

Brand	Hohes C	Albi	Valensina	NONE: I wouldn't choose any of these.
Price	1,69 €/L	1,09 €/L	1,99 €/L	
FairTrade label	No	Yes	Yes	
Package type	Plastic bottle (PET)	Carton (Tetra Pak)	Plastic bottle (PET)	
	○	○	○	○

Session Choice at eRum... you picked well ;)

10:45-12:15	Methodology 1 (McKinsey session)	Business 1	Packages 1
15:00-16:30	Methodology 2	Business 2	Packages 2
18:20-19:45	Methodology 3	Data workflow 1	BioR

Discrete Choice Models: Background

- **Discrete Choice Models: model and model assumptions (Train, 2009)**
  - Decision maker i obtains utility from alt. j at time t: $U_{ijt} = V_{ijt} + \epsilon_{ijt}$
  - Decision maker is utility maximizer: $U_{ijt} > U_{ikt}, \ \forall k \neq j$
  - Error term is distributed iid type I EV: $P_{ijt} = \frac{\exp\{V_{ijt}\}}{\sum_k \exp\{V_{ikt}\}}$
  - Mixed Logit (MXL) or random effects MNL: decision makers have heterogeneous preferences: $V_{ijt} = x'_{ijt} \cdot \beta_i$ with β_i $MVN(\mu, \Sigma)$
- **Discrete Choice Models are widely accepted**
  - Daniel McFadden: 2000 Nobel Memorial Prize in Economic Sciences (*"for his development of theory and methods for analyzing discrete choice"*)
  - Often used in academia (Chandukala *et al.*, 2007)
  - Often used in practice (e. g., Apple vs. Samsung: The $2 Billion Case, see Netzer/Sambandam, 2014)

Discrete Choice Models in `R`

- **Inference (and selected packages):**
  - Classical (Maximum Simulated Likelihood): `mlogit`, `gmnl`
  - Bayesian (Hierarchical Bayes): `bayesm`, `ChoiceModelR`
- **Stan (probabilistic programming language, see Carpenter et al., 2016):**
  - `Stan` uses Hamiltonian Monte Carlo (NUTS, see Hoffmann/Gelman, 2014); more efficient than Gibbs/Metropolis-Hastings
  - `Stan` is open source, written in C++, and provides interfaces to many other prog. languages including `Python`, `R`, `Stata`, `Matlab`, `Julia`, ...
  - Nice features, e. g. ShinyStan, loo, parallel chains, model checking, output summary, plots, ...
  - But: there is no implementation available for DCM, especially MXL
- **Why is `Stan` a good idea?**
  - MXL models can be difficult to estimate
  - `R` packages use optimized implementations of the "Allenby/Train-method" (see Train, 2009); but what about "non-standard models"?
  - Ben-Akiva/McFadden/Train (2016): *"Stan is the best general-purpose method, but ..."*-argument

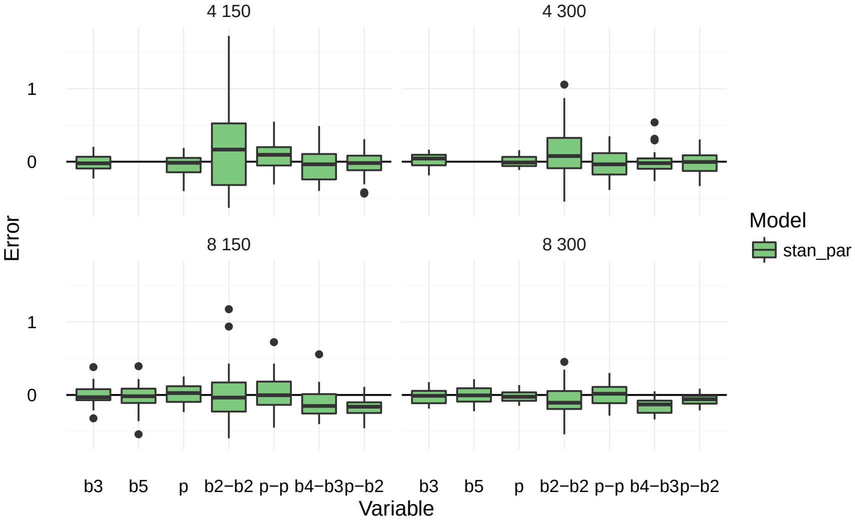
(Our First) MXL in Stan

```
data {
  int<lower = 1> N;                                          // number of obs.
  int<lower = 1> K;                                          // number of par.
  int<lower = 2> J;                                          // number of alts.
  int<lower = 1> I;                                          // number of resp.
  int<lower = 1, upper = J> y[N];                            // dep var
  int<lower = 1, upper = I> id[N];                           // resp. index
  matrix[J, K] x[N];                                         // x arrays
}

parameters {
  vector[K] mu;                                               // mean
  cov_matrix[K] Sigma;                                        // cov matrix
  vector[K] beta[I];                                          // reps. par
}

model {
  mu ~ normal(0, 100);                                        // priors
  Sigma ~ inv_wishart(K, diag_matrix(rep_vector(1.0, K)));

  for (i in 1:I) {
    beta[i] ~ multi_normal(mu, Sigma);                        // sample beta_i
  }
  for (n in 1:N) {
    y[n] ~ categorical_logit(x[n] * beta[id[n]]);             // logLik
  }
}
```

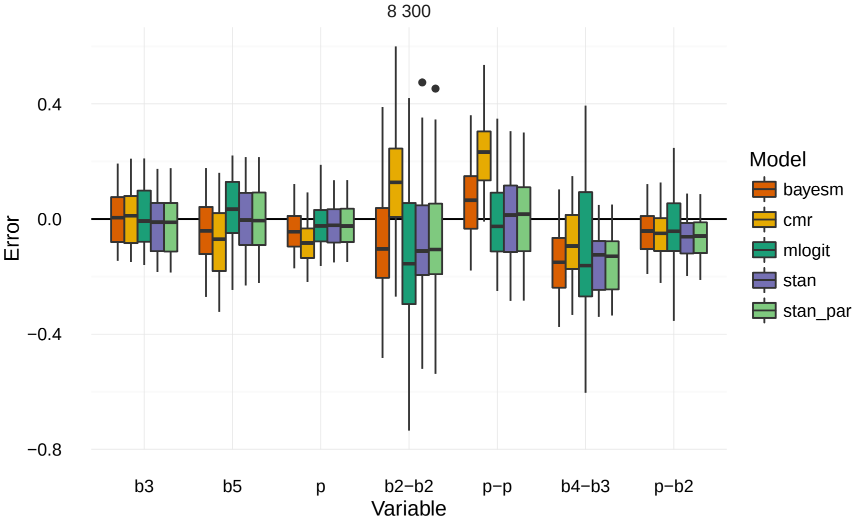
Simulation Study Setup

- **General setup:**
  - Number of alternatives $J \in \{4, 8\}$ (market shares: $ms_4 \in [14\%, 40\%]$ and $ms_8 \in [8\%, 24\%]$) and households $I \in \{150, 300\}$
  - Number of variables $K \in \{4, 8\}$ (price + $(J-1)$ alternative specific constants)
  - Number of estimated coefficients $C = K + K \cdot (K+1)/2$ (means + full covariance matrix)
  - Number of replications $R \in \{1, \ldots, 30\}$
- **Variables:**
  - $asc_{j_1, j_2} = \begin{cases} 1 & \text{if } j_1 = j_2 \\ 0 & \text{otherwise} \end{cases}$
  - $price \sim \mathrm{U}(3, 5)$
  - $\beta_i \sim \mathrm{MVN}(\mu_{nj}, \Sigma_{nj})$
- **Approaches:**
  - ours: `Stan` (single-core), `Stan` (multi-core, $n = 4$)
  - `mlogit`, `bayesm`, and `ChoiceModelR` (cmr)

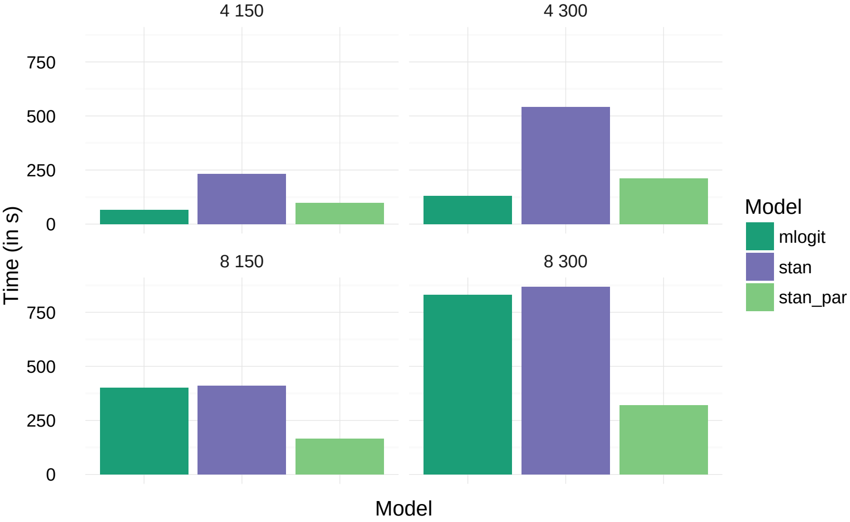
Simulation Study Results (1)



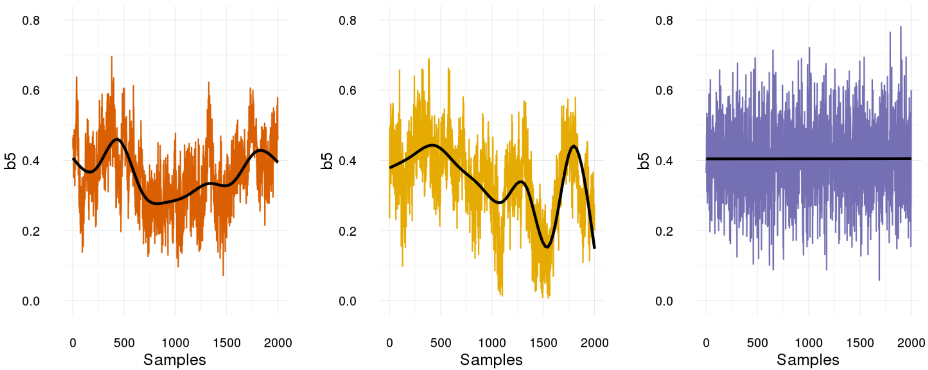
Simulation Study Results (2)



Runtime


4 150
4 300
8 150
8 300
750
500
250
0
750
500
250
0
Time (in s)
Model
Model
mlogit
stan
stan_par

Effective Samples

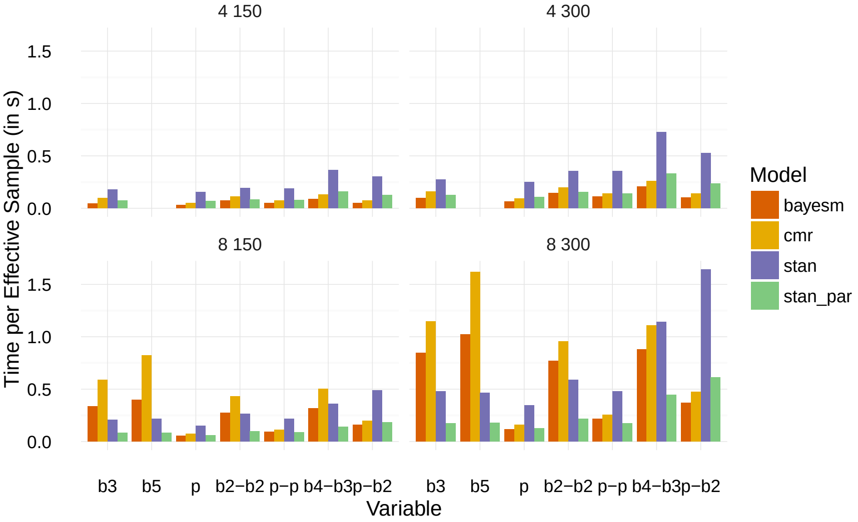


- 162 s for 40,000 samples
- 225 effective samples

- 223 s for 40,000 samples
- 115 effective samples

- 842 s for 4,000 samples
- 2,490 effective samples

Time per Effective Sample



Willingness-to-pay for a Fair Trade Label

- Willingness-to-pay (WTP) is an essential measure in QME (Netzer/Sambandam, 2014)
- Marginal rate of substitution between attribute and price: $WTP = \frac{\partial V / \partial x}{|\partial V / \partial price|}$
- Scholars advocate estimation of DCM in WTP-space (Sonnier *et al.*, 2007)
  - Preference-space: $V_{ijt} = x'_{ijt} \cdot \beta_i - \alpha_i \cdot price_{ijt}$
  - WTP-space: $V_{ijt} = \lambda_i \cdot (x'_{ijt} \cdot \omega_i - price_{ijt})$
- MVN-prior directly on WTP (ω) instead of preferences (β)
- No `R`-package available for estimating DCM in WTP-space "out-of-the-box" in a bayesian framework (but there is `gmnl` using MSL)
- However, product of parameters makes estimation more challenging
  - Even higher correlation in standard samplers
  - More iteration
  - Extreme thinning (*"...kept every 100th draw"*)

Setup

- **Setup:**
  - Data from choice experiment of Paetz/Guhl (2016)
  - 4 attributes: brand (Albi, Granini, HohesC, Valensina), packaging (carton, PET), FT label (yes, no), price (1.09, 1.39, 1.69, 1.99 Euro)
  - 200 respondents
  - 16 choices each
  - 4 alternatives (3 brands + no-buy)
- **Stan:**
  - We run `Stan` in parallel (4 chains)
  - 3,500 iterations incl. 1,000 for "warmup" = 10,000 draws
  - Successful convergence: all $\widehat{R}$-values are < 1.005
  - Runtime: 38 minutes
  - Efficiency: between 0.75 and 1.5 s/n_{eff}
- **gmnl**:
  - 4000 Halton draws
  - BHHH for optimization
  - Runtime: 43 minutes

WTP estimates ("standard errors" in parentheses)

	Stan		gmnl	
Attribute	**Mean**	**Std dev**	**Mean**	**Std dev**
No-buy option	-1.425 (0.059)	0.655 (0.053)	-1.434 (0.018)	0.633 (0.025)
Albi	-0.088 (0.036)	0.392 (0.035)	-0.066 (0.019)	0.460 (0.021)
Granini	0.024 (0.032)	0.352 (0.033)	0.055 (0.016)	0.319 (0.018)
Hohes C	0.240 (0.042)	0.512 (0.040)	0.234 (0.016)	0.541 (0.021)
FT label (yes)	0.258 (0.028)	0.345 (0.026)	0.260 (0.012)	0.352 (0.014)
Packaging (carton)	0.144 (0.040)	0.506 (0.035)	0.166 (0.013)	0.567 (0.019)

Status

- **Main idea similar to `rstanarm`:**
  - Pre-compiled `Stan` models for "standard" DCMs
  - Employs `rstan` as `R` interface to call `Stan`
- **Data handling:**
  - Provides functions to create data for `dcm`
  - Reshape `bayesm`, `ChoiceModelR`, and `mlogit` data
- **Supported models:**
  - "plain vanilla" MNL and MXL
  - MXL in WTP-space
  - MXL with log-norm price parameter
  - Models with μ as function of observables ("observed heterogeneity")
- **Outlook:**
  - About to release first version of `dcm` on github
  - More models (additional help + feedback is welcome!)
  - More convenience functions (e. g., analyzing output from `bayesm` or `ChoiceModelR` using `rstan`)

Conclusion

- `Stan` seems work well for estimating Discrete Choice Models
- Our implementation outperforms specialized and optimized implementations of standard models in `R`
- Speed comparisons are worthless w/o taking the effective sample size into account
- Additional flexibility is great for research ("non-standard-models")
- Coding models in `Stan` is easy + great resources online (e. g., manuals, tutorials, case studies, ...)
- However, high performance code needs (some) optimization + "tricks"

Thank you very much for your attention!

Questions?

References

Ben-Akiva, M., D. McFadden, & K. Train (2016): Foundations of Stated Preferences Elicitation Consumer Behavior and Choice-Based Conjoint Analysis. *Working Paper*.

Carpenter, B., A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, & A. Riddell (2016): Stan: A probabilistic programming language. *Journal of Statistical Software* (in press).

Chandukala, S. R., J. Kim, T. Otter, P. E. Rossi, & G. M. Allenby (2007): Choice Models in Marketing: Economic Assumptions, Challenges and Trends. *Foundations and Trends in Marketing* 2(2), 97–184.

Hoffmann, M. & A. Gelman (2014): The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* 15(1), 1593–1623.

Netzer, O. & R. Sambandam (2014): Apple vs. Samsung: The $2 Billion Case. *Columbia Case Works*, Columbia Business School.

Paetz, F. & D. Guhl (2016): Socially Conscious Consumption: Consumers' Willingness-To-Pay for Fair Trade Labels. *Proceedings of the DA2PL'2016*.

Sonnier, G., A. Ainslie, & T. Otter (2007): Heterogeneity distributions of willingness- to-pay in choice models. Quantitative Marketing and Economics 5(3), 313–331.

Train, K. (2009): *Discrete Choice Methods with Simulation*. 2nd edition, Cambridge University Press.

Server spec

- **CPU:**
  - E5-2690 v2
  - Intel Xeon (Ivy Bridge EP)
  - 3,0GHz
  - 25MB L3 Cache
  - DDR3 1866
- **RAM:**
  - 256GB DDR3 1866 DIMM
  - REG
  - ECC