Sample Code for 3 Discrete Choice Models: Documentation

Multinomial Logit

Limited Consideration

Bayesian Learning

Introduction

I struggled to find sample code for the classic Bayesian learning model in Erdem and Keane 1996. I am sharing my own version of their model with the hope that it's helpful to others.

I am also sharing examples of other discrete choice models that I used in the same project.

This content is most useful for those already familiar with these models but seeking a coded example.

Documentation is provided here and within the script files.

All code is written in the Julia programming language.

Multinomial Logit

Generic multinomial logit model.

Documentation is contained in the script.

Estimation:

- Gibbs sampling for population parameters governing the distribution of individual utility parameters.
- Bayesian MCMC for individual utility parameters (α_{ij}, β_i).

Limited Consideration Model

Based loosely on Goeree 2008.

Setting:

- Individuals choose between 3 products (no outside option) based on choice sets that may not include all of the products.
- There are two exogenous product characteristics: product quality (an indicator/intercept for the product chosen) and a continuous choice set observable variable.
  - Product quality affects choices exclusively through the choice utility.
  - The choice set variable affects choices exclusively through the choice set formation process.
- Choice set technology: the probability of a choice being in a choice set is a logistic function and is independent of other products.
- Choice utility is based on product quality and typical logit choice model assumptions.

Estimation:

- Simulated maximum likelihood using Newton's method.
- Key likelihood feature: choice probabilities are integrated over all possible choice sets.
- The gradient and Hessian are calculated using automatic differentiation.

Limited Consideration Model

Choice Set Parameters

β

Utility parameters

$\alpha_1, \alpha_2, \alpha_3$

Choice set formation technology

$$\phi_{ij} = \frac{\exp(X_{ij}\beta)}{1 + \exp(X_{ij}\beta)}$$

Individual choice set

$$\Pr([2,3]) = \prod_{m \in [2,3]} \phi_{im} \prod_{n \notin [2,3]} (1 - \phi_{in})$$

[2, 3]

Choices dictated by basic logit preferences

$$R_{it} = \underset{j \in [2,3]}{\operatorname{argmax}} \{\alpha_j + \epsilon_{itj}\}$$

Model Parameters

Utility parameters: $\{\alpha_1, \alpha_2, \alpha_3\}$

Choice set parameters: $\{\beta\}$

Limited Consideration Literature

Agents are matched to <u>choice sets</u> which are subsets of all choices.

Identification challenges: choices may be rationalized by both choice sets or preferences.

Barseghyan 2021 describes 4 modeling/identification approaches:

1. **Default**: assume full choice set is available to everyone.
2. **Additional information**: use other observables to construct choice sets.
   - Honka 2017 (survey of brand awareness)
3. **Exclusion restriction**: assume certain variables impact only choice sets but not preferences.
   - Goree 2008, Seiler 2016, Hortacsu 2017.
4. **Restrictions on the choice set formation process:**
   - Abaluck 2021 (2 choice sets)
   - Crawford 2020 (choice sets over time with panel data)
   - Goeree 2008 (parametric assumption)
   - Less restricted approaches: Lu 2021, Barseghyan 2021, Cattaneo 2020

Crawford 2021 describes 2 estimation approaches:

1. **Integrating out**
   - Calculate likelihood of observed choices over all choice sets.
   - More restrictive but allows for counterfactuals.
2. **Differencing**
   - Treats choice sets as nuisance parameters.
   - Less restrictive but does not permit counterfactuals.

Bayesian Learning Model

Simple learning model based loosely on Erdem and Keane 1996.

Setting:

- Individuals choose between 3 products (no outside option) based on uncertain beliefs about product quality.
- Individuals begin with prior beliefs about product quality which are updated in a Bayesian fashion based on noisy signals of the true product quality.
- Individuals learn from experience (past choices) and from advertising. Both provide noisy signals of the true quality for each product. Physicians update priors for product quality based on these signals.
- Individuals are myopic and physicians make multiple choices in a given period.

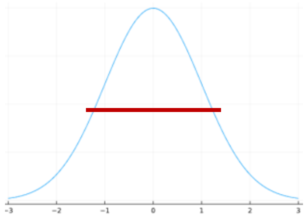
Estimation:

- Simulated maximum likelihood using Newton's method.
- The gradient and Hessian are calculated using automatic differentiation.
- The likelihood is integrated over fixed learning signals that are scaled according to the variance parameters.

Bayesian Learning Model

Previous Periods

Experience signals



$$Signal_{ijt} = Q_j + e_{ijt}$$

Advertising signals

Current Period

Updated Beliefs

Quality prior means: $\{Q_{0,1}, Q_{0,2}, Q_{0,3}\}$

Quality prior variance: $\{\sigma_{0,1}^2, \sigma_{0,2}^2, \sigma_{0,3}^2\}$

$$Q_{jt} = \frac{\sigma_{j1}^2}{N_j(t)\sigma_{j1}^2 + \sigma_\varepsilon^2} \sum_{s=1}^{t-1} Q_{js}^E d_{js} + \frac{\sigma_\varepsilon^2}{N_j(t)\sigma_{j1}^2 + \sigma_\varepsilon^2} Q_{j1},$$

$$\sigma_{jt}^2 = \frac{1}{\frac{1}{\sigma_{j1}^2} + \frac{N_j(t)}{\sigma_\varepsilon^2}},$$

Calculated using Bayes rule

Current period choices

$$\boxed{U(Q,p) = \theta_1 Q + \theta_2 Q^2}$$

Expected utility based on beliefs/ uncertainty of true quality, and payments

$$\boxed{\begin{aligned} &E[U|Q,p,\sigma_t^2,\sigma_e^2] \\ &= \theta_1 Q + \theta_2 Q^2 + \theta_2(\sigma_t^2 + \sigma_e^2) + \epsilon \end{aligned}}$$

Individual choice problem maximizes utility subject to quality and unobservables.

$$\boxed{R_{it} = \underset{j \in J}{\operatorname{argmax}} \{E[U|Q,\sigma_t^2,\sigma_e^2] + \epsilon_{itj}\}}$$

Additional experience signals based on choices

Model Parameters

True quality: $\{Q_1, Q_2, Q_3\}$

Quality prior means: $\{Q_{0,1}, Q_{0,2}, Q_{0,3}\}$

Quality prior variance: $\{\sigma_{0,1}^2, \sigma_{0,2}^2, \sigma_{0,3}^2\}$

Experience signal variance: $\{\sigma_e^2\}$

Advertising signal variance: $\{\sigma_a^2\}$

Utility parameters: $\{\theta_1, \theta_2\}$

Comments on the Julia Programming Language

Syntax is similar to Python and the pattern for optimized code is similar to c++.

Relevant features of the language; see Bayesian Learning Model.

- Use of structs and custom functions.
- Separated code into multiple files.
- A likelihood function that is compatible with automatic differentiation in the FiniteDiff package. That the gradient and Hessian can be computed quickly without providing the analytical expression.
- A Bayesian MCMC algorithm that loops over every field of a struct containing the model parameters.

Guidelines for optimized code in Julia:

- Use loops instead of vector operations.
- Avoid re-initializing frequently used arrays.
- Use functions when possible.
- Use structs to pass variables and avoid using global variables.

Cited Papers

Barseghyan L, Molinari F, Thirkettle M. Discrete Choice under Risk with Limited Consideration. American Economic Review. 2021 Jun;111(6):1972–2006.

Crawford GS, Griffith R, Iaria A. A survey of preference estimation with unobserved choice set heterogeneity. Journal of Econometrics. 2021;222(1):4–43.

Erdem T, Keane MP. Decision-Making under Uncertainty: Capturing Dynamic Brand Choice Processes in Turbulent Consumer Goods Markets. Marketing Science. 1996;15(1):1–20.

Goeree MS. Limited Information and Advertising in the U.S. Personal Computer Industry. Econometrica. 2008;76(5):1017–74.