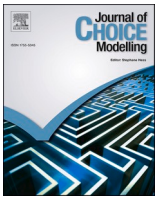
Journal of Choice Modelling 47 (2023) 100396

Contents lists available at ScienceDirect

Journal of Choice Modelling

journal homepage: www.elsevier.com/locate/jocm



Distribution-free estimation of individual parameter logit (IPL) models using combined evolutionary and optimization algorithms

Joffre Swait

Erasmus School of Health Policy & Management and Erasmus Choice Modelling Centre, Erasmus University Rotterdam, Campus Woudestein, Burgemeester Oudlaan 50, 3062, PA, Rotterdam, the Netherlands

ARTICLE INFO

Keywords:
Distribution-free estimation
Non-parametric estimation
Individual parameters logit
Mixed logit
Genetic algorithm
Evolutionary algorithm

ABSTRACT

When estimating random coefficients models from choice data, decisions relating to the multivariate density function assumed to describe preference heterogeneity across the population raise questions about stochastic (in)dependence between preference dimensions, uni- vs. multimodality, potential point masses, bounds and/or constraints on support regions, among other concerns. Parametric representations of population distributions have generally implied uncomfortable compromises to achieve estimation tractability. It would seem preferable to sidestep such issues by estimating individual preferences in a distribution-free manner, but this freedom of form implies a large number of parameters since we lose the parsimony enabled by parametric densities and must deal directly with estimation of individual decision maker preferences. I propose a hybrid distribution-free estimator for individual parameter logit models that uses a genetic algorithm as first stage, the solution from which becomes a starting point for a gradient-based search to obtain the final posterior maximum likelihood estimates of individual preferences. This estimator is described in detail, its parameter recovery capability is tested with Monte Carlo data generation simulations, and a case study is developed in some detail to illustrate its use in policy analysis. The estimator can be applied to both stated and revealed preference data, requiring only that sufficient choice replications be available for individual observation units consistent with extant estimation methods. Computational experience shows the estimator to require CPU times comparable to extant simulation-based estimation methods, meaning that its use is practical for the exploration of the parameter space through multiple trials.

1. Introduction

The Mixed Logit generalization of the Multinomial Logit (MNL) model is a key specification for capturing preference heterogeneity in a sample of individuals from choice data. This heterogeneity is generally described through a population-level multivariate density function h_β, which the analyst must provide prior to estimation of distribution parameters underlying the preference heterogeneity. The model was introduced to the literature through several empirical papers involving Kenneth Train (Revelt and Train 1998; Brownstone et al., 1999; Train 1998; Brownstone et al., 2000) just before the turn of the 21st century. One of the model's main attractions stems from Theorem 1 in McFadden and Train (2000), which established the specification's ability to approximate other choice model forms arbitrarily closely. Mixed Logit has been widely used in subareas of applied economics, among them transportation, marketing, environment, labor, and health, with its use spanning both revealed and stated preference data generating

E-mail addresses: Joffre.Swait.Jr@gmail.com, swaitjr@eshpm.eur.nl.

https://doi.org/10.1016/j.jocm.2022.100396
Received 11 January 2022; Received in revised form 14 November 2022; Accepted 15 November 2022
Available online 24 November 2022

mechanisms. The extensive recent review in Haghani, Bliemer and Hensher (2021), though not focused on the Mixed Logit *per se*, is nonetheless very informative with respect to its impact in the choice modeling discipline. Soekhai et al. (2019), in a systematic review focused on the application of choice experiments in the health care area, shows in their Table 2 that the use of the Mixed Logit model is increasing significantly over time in that domain.

For some applications, inferring the population preference distribution h_β is not *per se* the analyst's objective, rather it is obtaining reliable estimates of individual preferences to enable better predictions at the individual level, improving understanding about the joint distribution of marginal rates of substitution in support of product customization, improving estimates of welfare distribution implications driven by policy changes, eliminating potential confounds or bias while testing an (unrelated) substantive hypothesis, or even improving characterization of the existence and extent of non-compensatory evaluation rules through the parameters of the utility function (see Johnson and Meyer 1984, Swait and Adamowicz 2001a,b). Thus, for some modelers and policy analysts in certain applications, the interest in one or another parametric distributions to describe preferences in a population is just the means to the end goal of individual parameter estimates in a sample.

However, imposing parametric preference distributions on the data generation process nearly always raises questions about the desirability and tenability of the specific distributions adopted. To make concrete some of the difficulties surrounding this selection, consider these issues the analyst must think about to define a candidate h_β.

- *Dimensionality and Dependence Across Dimensions* – Is h_β to be defined as multivariate and expressing some form of dependence among specific preferences (e.g., Multivariate Normal joint distribution)? Or specified as $h_\beta = h_{\beta 1} \bullet h_{\beta 2} \ldots \bullet h_{\beta K}$, expressing that the preference dimensions are stochastically independent of one another? Or some version in between?
- *Uni- vs. Multi-Modal Distributions* – Certain quality attributes may evince strong positive or negative reactions in different people, suggesting bimodal population preferences. To accommodate such disparate preferences, mixtures of unimodal distributions can be considered, but this significantly adds to the complexity of the estimation problem since one would have to specify the number of modes for each attribute thus affected, and there might be multiple such attributes.
- *Discrete-Continuous Distributions* – This is related to multi-modality. Consider preference for a quality that is distributed in some continuous fashion in the population, but with a particular segment of consumers having a disproportionate representation of one or more discrete values (essentially, point masses at particular values, say, zero). This will lead to a preference distribution that is a mixture of a continuous distribution over the support region, plus point masses at specific values. Representation of this phenomenon is … complicated (see, e.g., Swait 2009).
- *Bounded or Unbounded Support Region* – In early empirical work with the Mixed Logit, it was common to specify the heterogeneity of all coefficients to be (independently) normally distributed; this included variables like price, for which the *a priori* expectation is that preferences should be strictly non-positive. Since the support region of the normal distribution is unbounded, there is *always* a non-zero probability of positive price coefficients. For such situations workarounds have been given in the literature, as discussed in Daly et al. (2012), among others. These workarounds carry their own issues and imply specific analyst burdens.

In response to such concerns, Train (2016) states that enabling greater flexibility in the specification of the parametric distribution of underlying preferences "… shifts the emphasis for future research from overcoming distributional constraints to developing richer datasets that can more clearly distinguish among the variety of shapes that the parameter distribution might take" (Train, 2016, p. 41). Seeking flexibility is certainly one direction that can be taken, but what if instead we were able to estimate individual preferences in a *distribution-free* manner, letting the data reveal these jointly distributed preferences as reflected therein, unencumbered by parametric distributional decisions often made simply for analyst convenience?

This question gives rise to the research objectives of this paper: (1) to conceptualize, implement and test a distribution-free estimator for individual-level preferences based on the logit model, and (2) to do so without imposing any *a priori* requirements on number of modes, peculiar discrete mass points, regions of support and independence across dimensions of preference, as would be required in estimation of population level multivariate density function h_β.

To meet these objectives I have elaborated a hybrid estimation procedure that uses an evolutionary algorithm in its first stage to directly explore the K-dimensional individual preference space for N decision makers, for each of which we have observed $R \geq 2$ choice replications. This initial broad search uses maximization of the *posterior log likelihood* as the figure-of-merit (FOM). In a second stage, a good (and hopefully, near to optimal) solution from the evolutionary algorithm can be submitted to a traditional optimization routine to finalize individual-level parameter estimates, based on the same FOM. The contribution I make, therefore, is to harness together the evolutionary algorithm and nonlinear optimization algorithm to maximize the posterior maximum likelihood, in the process creating a distribution-free estimator for the posterior individual parameters logit (IPL) model. This combination is able to handle the very high dimensionality (NxK parameters) imposed by the distribution-free requirement, and is able to do so mainly due to the evolutionary algorithm. Since a genetic algorithm (GA) is employed in the first stage, I'll term this the IPL-GA estimator. The proposed method is easily adapted to other kernels and model formulations, so it should be viewed as a more general tool than what is directly illustrated here.

The paper is structured as follows: 1) the implementation of IPL-GA is discussed in some detail in both its evolutionary and optimization phases; 2) a simulation study investigates the algorithm's ability to recover individual-level preference parameters under known circumstances; and 3) IPL-GA is applied to a fabric softener scanner panel data set to illustrate its use with revealed preference (RP) panel data having large choice sets at each purchase occasion (59 alternatives), many parameters ($K = 30$) and variable number of replications (R in [2,…,40]) per decision-maker. We close the paper with a discussion of selected topics, future directions, software availability and other considerations.

2. A hybrid evolutionary + gradient search method for estimating the individual parameters logit model

To briefly recap the Mixed Logit model before continuing to the proposed estimation method, the goal is to characterize the preference heterogeneity (1xK) row vector β_n for a person n, as jointly characterized over a population of decision makers by some parametric distribution h_β, often taken to be the Multivariate Normal (MVN) distribution:

$$\beta_n \sim MVN\left(\overline{\beta}, \Sigma_\beta\right) , \tag{1}$$

where $\overline{\beta}$ is the mean and Σ_β is the covariance matrix of the preference distribution. This description of the preferences is then applied to a "kernel" choice model P_{an}, a an alternative, here assumed to be the MNL model, thus:

$$P_{an|\beta} = \frac{exp(\beta X_{an})}{\sum_{j \in C_n} exp\left(\beta X_{jn}\right)}, a \in C_n, \tag{2}$$

X_{an} a Kx1 vector of explanatory variables, C_n the choice set of alternatives over which selection is exercised. This expression of the choice probability is conditional on a realization $\beta_n = \beta$ from density function h_β, so the unconditional probability must be obtained by integrating over the multivariate preference distribution and parameter space θ:

$$P_{an} = \int_\theta P_{an|\theta}\, h_\theta \, d\theta \tag{3}$$

Expressions (1)–(3) characterize the Mixed Logit model. Variants have been proposed (e.g., the Multinomial Probit model as kernel), but these have not been as widely used as the above specification. Different numerical and simulation methods exist for estimation of K preference parameters as well as parameters of density h_β, with choice of method being driven by considerations such as the dimensionality K of the preference vector, the structure of the covariance matrix (i.e., the correlation structure assumed among preference components) and the number of parameters implied by the representation, the number of measurement replications per respondent/decision-maker, and the number of decision makers. Simulation combined with nonlinear optimization methods has proven to be the most common method of estimation (simulated maximum likelihood, see Train 2009).

The proposed method shifts its focus from the multivariate prior population preference density function h_β, instead placing it on the direct maximization of the sample posterior log likelihood to obtain individual preference estimates. This is also the focus of Train (2009), Chapter 11, wherein he explores the implications of estimating the individual parameters logit model vis-à-vis the mixed logit model in expression (3). The reader is directed to Train for a clear exposition of the differences between methods. In a direct sense, the proposed approach frees us from the need to specify the details of the prior population distribution by directly estimating the posterior preference parameters. Frischknecht et al. (2014), who focus on the estimation of individual parameters logit using a weighted maximum likelihood method, is also a direct precursor work to this article.

Stage 1: Estimation of Individual Preferences Through Evolutionary Search Methods

Classical nonlinear unconstrained optimization search methods, such as the Newton-Raphson, BHHH, gradient/steepest descent or secant methods (see Berndt et al., 1974 for BHHH; Gill et al., 1981 on other methods), work on the principle of moving from a current to an improved point in the solution space, where improvement is measured through the figure-of-merit (FOM) or objective function. This is repeated until a (perhaps local) optimum is found. These methods all make use of first- and/or second-order information in the form of the gradient and Hessian of the FOM (or approximations of these in the secant methods), which gives them their search power but is also the source of their weakness: this information "biases" them to seek out locations where the gradient is zero, regardless of whether these be local optima, global optima, or even saddle points.

Evolutionary algorithms work on quite another principle: they seek to improve a set of candidate solutions (the population, or *candidate pool*) following genetic/evolutionary principles, motivated by biological or evolutionary metaphors to generate better solutions. The pool advances from one generation to another by building on the strong members (i.e., those that provide good FOMs) in the hopes of producing an even better population of solutions. This is done repeatedly until convergence and/or stopping criteria are satisfied.

Common operators used in the Genetic Algorithm (GA), arguably the most common evolutionary paradigm, to improve population strength are essentially different reproduction actions:

1) *Pairing* (splicing) of two members through some selection process, where mating produces offspring that inherit different components from each parent;
2) *Recombination* of two members, essentially a convex combination of the parental components;
3) *Cloning* of (strong) members, thus passing them through into the next generation; and
4) *Mutations* applied to any of the outcomes of actions 1–3, which introduces random exogenous perturbations to the pool to avoid stagnation of the population.

The key to evolutionary methods lies in the mechanics of selection and reproduction, which therefore vary (sometimes widely) from one method to another. Besides the GA, common implementations using different evolutionary metaphors are differential

evolution (DE) (e.g., Storn and Price 1997) and particle swarm (PS), among others. I refer the reader to the more general literature on the topics of evolutionary and genetic algorithms (e.g., Banzhaf et al., 1998; Fogel 2006; Goldberg 1989; Schmitt 2001, among many others), but recommend Dorsey and Mayer (1995) and Chaterjee et al. (1996) for their focus on statistical parameter estimation problems.

Evolutionary methods have a long history in the optimization literature, particularly for problem types that have difficult, discontinuous and/or non-differentiable objective functions (see Holland 1975; De Jong 1975). They were employed to solve optimization problems in biology, engineering and operations research (see, e.g., Goldberg 1989), where they are routinely employed to this day for what can be termed "global search" since they are viewed as methods that are more likely to help achieve global optimality, either by avoiding local optima, or being less likely to get "trapped" by them.

The application of evolutionary methods in statistics and econometrics has grown in the past 30 years,[1] but they still do not seem well diffused in econometric empirical work (though see Gilli and Winker 2008; Drake and Marks 2002, with a focus on finance and macroeconomic forecasting). One of the reasons for this may be the heavy reliance on maximum likelihood methods in econometrics, both as a methodology and as an orientation/perspective on the necessary requirements for estimators. In this view, anything short of the global optimum is unsatisfactory as the basis for hypothesis testing, inference and prediction. However, as emphasized by both Dorsey and Mayer (1995) and Chaterjee et al. (1996), one perspective on evolutionary algorithms is that due to the more global nature of their search behavior, they can provide "better" starting points for optimization algorithms to achieve global optimality. It is from this perspective that the algorithmic development herein is positioned.

I have not found papers in the evolutionary algorithm literature oriented towards the estimation of individual preferences using the type of disaggregate choice data commonly associated with choice modeling. The one near exception is Padmanabhan and Barfar (2021), who seek to infer preferences for groupings of television shows associated with political and social interests, using 1) disaggregate but anonymous TV panel viewership data, and 2) aggregate voting or political affiliation data with no specific relationship to the viewership data other than temporal and geographic commonality. Their goal is principally the identification of clusters of TV shows that can be used to target associated attitudes.

The focal estimation problem we address has *many* parameters, specifically *N*x*K* preferences, *N* the number of respondents and *K* the number of preference parameters. For a sample of 500 respondents used to estimate 10 marginal utilities per decision maker, that translates to 5 000 parameters … ! This size challenge is shared by other classes of problems, among them image processing, statistical engineering and hydrology (see Holloman et al., 2006). Evolutionary algorithms, such as the one examined here, have played a role in tackling such high-dimensional parameter estimation problems. In discussing this class of problems, Holloman et al. say that they are "… plagued by multimodality or other problematic structures in the likelihood or posterior" (Holloman et al., 2006, p. 878). The estimation problem we address here shares these difficulties, which require satisfactory addressing.

I will describe the implementation of the genetic algorithm used in IPL-GA in some detail, to make concepts as concrete as possible. It is also my hope that this specificity and subsequent algorithmic details will enable the diffusion of the method. However, it is first necessary to make a number of definitions. Let Π_g be the population/pool for generation g, defined as

$$\Pi_g = \{\pi_{1g}, \ldots, \pi_{Mg}\} \tag{4}$$

$$\pi_{mg} \text{ be a point in the } K-\text{dimensional preferences space } \Omega, \text{ the } m-\text{th member of the } g-\text{th generation pool;} \tag{5}$$

for $g = 0,1, \ldots,G$, $m = 1, \ldots,M$. For a given pool member m in the g-th generation pool, and assuming independence within subject across replications, the likelihood of decision-maker n's observed choices over R choice replications is given by

$$l_{n|mg} = \prod_{r=1}^{R} \prod_{a \in C_{nr}} \left[P_{an}\left(\pi_{mg} \middle| X_{anr}\right)\right]^{\delta_{anr}}, \tag{6}$$

where $P_{an}()$ is the choice probability model (e.g., expression 3) evaluated at pool member π_{mg}, with independent variables X_{anr} on choice set C_{nr}. The δ_{anr} are choice indicators, equal to 1 for the single chosen alternative and 0 otherwise. The posterior preferences $\widehat{\beta}_{ng}$ for decision-maker n are calculated by applying Bayes Theorem over the M pool members in Π_g, thus:

$$\widehat{\beta}_{ng} = \frac{\sum_m \pi_{mg} \cdot l_{n|mg}}{\sum_m l_{n|mg}}, n = 1, \ldots, N. \tag{7}$$

The posterior log likelihood for the sample is calculated from the following expression, using these posterior preferences:

$$L_{Ng} = \sum_n lnl_n\left(\widehat{\beta}_{ng}\right), \tag{8}$$

where

[1] A simple Web of Science search on the term "econometrics genetic algorithm" yields just 40 documents from 1996 to 2021, indicating that during that rather extended period little work in genetic algorithms has focused on econometric issues.

$$l_n(\widehat{\beta}_{ng}) = \prod_r \prod_a \left[P_{an}(\widehat{\beta}_{ng}|X_{anr})\right]^{\delta_{anr}} \tag{9}$$

Expression (8) defines the objective function, or FOM, being optimized, while expressions (6) and (7) show how the posterior preferences are related to candidate pool members at generation *g*. As the generations evolve, the impacts of changes therein filter through as changes (and hopefully improvements) to the posterior log likelihood.

As described earlier, the transition of Π_g to Π_{g+1} occurs by a selection and reproduction process. Two key concepts underlie this updating: a) members of the pool have a "fitness" or "eligibility measure" for reproduction; and b) the selection process is elitist, in that high fitness members are more likely to reproduce, while low fitness members are less likely or even prohibited from reproducing. An obvious metric for fitness is the contribution that the pool member makes towards improving the FOM (expression 8). The GA implementation herein employs a closely related fitness measure f_{mg} that counts how often pool member *m* produces the highest individual likelihood across the *N* sample members:

$$f_{mg} = \sum_n \mathbf{1}\left(l_{n|mg} \geq l_{n|m'g}, all\ m' \neq m\right), m = 1, \ldots, M, \tag{10}$$

where $\mathbf{1}(A) = 1$ if event A is true, $= 0$ otherwise, and $l_{n|mg}$ is given by expression (6). The reason for using this indirect measure is to make the fitness measure easy/cheap to compute, as opposed to the direct use of the individual posterior log likelihood (expression 9).

With this background, the updating of one generation to another occurs according to the following algorithmic logic[2]

Algorithm. UCP (Update Candidate Pool)

Algorithm UCP (Update Candidate Pool)

1: Sort pool members of Π_g in decreasing fitness (*f*) order, based on measures (10). Denote the sorted pool as $\widetilde{\Pi}_g$.

2: Discard the lower *d*% of pool members from further consideration, so $\widetilde{\Pi}_g$ has $M(d) \leq M$ members.

3: For $s=1,\ldots,M$ *// create next pool* Π_{g+1}

3.1: To generate the *s*-th new pool member, draw at random with replacement a mating pair (L)eft and (R)ight from $\widetilde{\Pi}_g$, L from the first $[M(d)-1]$ elements and R from $\{\pi_{g,L+1},\ldots,\pi_{g,M(d)}\}$.

3.2: Draw at random a mating strategy for new member *s*, following the probability distribution $(\alpha_P,\alpha_R,\alpha_{CL},\alpha_{CR})$, respectively the probabilities for Pairing, Recombination, Clone Left, and Clone Right $(\alpha_P+\alpha_R+\alpha_{CL}+\alpha_{CR}=1)$. Implement the selected mating strategy as follows:

3.2.1: (*Pairing*) Create $\pi_{g+1,s}$ by splicing together π_{gL} and π_{gR} at a randomly drawn element *r* in $\{1,\ldots,K\text{-}1\}$. The progeny inherits certain properties uniquely from each parent.

$$\pi_{g+1,sk}=\pi_{gLk} \text{ for } k=1,\ldots,r\text{-}1 \text{ and } \pi_{g+1,sk}=\pi_{gRk} \text{ for } k=r,\ldots,K \tag{11a}$$

3.2.2: (*Recombination*) Create $\pi_{g+1,s}$ by combining all characteristics from the mating pair with a mixing weight *w* drawn from U[0,1).

$$\pi_{g+1,s}=w\pi_{gL} + (1\text{-}w)\pi_{gR} \tag{11b}$$

3.2.3: (*Clone Left*) Create $\pi_{g+1,s}$ by copying π_{gL} in its entirety.

$$\pi_{g+1,s}=\pi_{gL} \tag{11c}$$

3.2.4: (*Clone Right*) Create $\pi_{g+1,s}$ by copying π_{gR} in its entirety.

$$\pi_{g+1,s}=\pi_{gR} \tag{11d}$$

3.3: (*Mutation*) With randomly drawn probability γ_g drawn from U[0,1), replace $\theta_{g+1,s}$ either fully or partially with a randomly generated point in bounded parameter space Ξ, a subspace of the *K*-dimensional parameter space Ω limited by minimum and maximum box constraints.

4: End

Sorting the starting pool by fitness, discarding the lower *d* percentile of the pool before reproduction and choosing mating pairs as a random selection of a higher fitness member with a lower fitness member (what might be termed "uplift"), jointly make this an elitist genetic algorithm. Elitism accelerates the improvement of the candidate pool with respect to the fitness measure (10) and the objective function (8), and is at the heart of the GA's ability to find good solutions.

[2] Note that in the algorithm outline, indentation indicates nesting of instructions, and is thus substantive.

Allowing mutations is a means to preserve diversity in the candidate pool, which is critical to enabling the algorithm to be more global or expansive in its exploratory behavior. The likelihood of mutations is controlled through the probability threshold γ_g, which has been made to vary over generations in this implementation. This promotes wider search performance in earlier generations; in later generations it puts a damper on mutation likelihood, which can be useful to stabilize the pool to preserve good solutions without adding noise by needlessly replacing good pool members. In the implementation, threshold γ_g evolves according to this functional form, which is a direct modification of a proposal by Kim and Lee (2012):

$$\gamma_g = max\left(\mu_{min}\ , \mu_{max} \cdot \chi_g^q \cdot exp\left(\frac{c \cdot g}{K^2}\right)\right), g = 1, \ldots, G, \tag{12}$$

where.

$\mu_{\min}$, $\mu_{\max}$ are the analyst-specified minimum and maximum mutation rates;
χ_g is a measure of diversity of the pool at generation g, defined below;
q is a power exponent on diversity, which can be either negative or positive;
c is a rate parameter dictating the decrease (if $c < 0$) or increase (if $c > 0$) in mutation rate between subsequent generations.

Equation (12) builds in a floor for the mutation rate, which may be useful to preserve some degree of change in the pool membership even in later generations. Finally, note that a useful implementation should allow γ_g to be fixed across generations.

The diversity measure is defined so as to capture the multi-dimensional dispersion of the members of the pool, and is implemented thus:

$$\chi_g = \frac{D_g}{r_{max,g}}, g = 1, \ldots, G, \tag{13}$$

where.

D_g is the average Euclidean distance between points in the pool, which is scaled relative to $r_{\max,g}$; $r_{\max,g}$ is the radius of the circle enclosing the pool members in Ξ-space and centered at the origin of the Cartesian coordinate system. This is the largest distance from the origin to a pool member.

Thus, as D_g increases the diversity of the pool increases, for a fixed radius.

Algorithm UCP defines the transition from one generation/pool to the next. The GA algorithm progresses through generations until some pre-specified stopping criteria are met. One straightforward strategy (termed Strategy = Best) is to execute G updates of Algorithm UPC, recording the posterior parameters (7) corresponding to the best solution found for measure (8), and using that as the Stage 1 Solution (S1S):

$$\widehat{\beta}_{n,S1S} \overset{\text{def}}{=} \widehat{\beta}_{n,g_{best}} \tag{14}$$

where g_{best} is the generation at which the best FOM was achieved. This strategy uses the evolutionary algorithm as an optimization tool.

A somewhat more elaborate solution strategy (Strategy = AVERAGE) is to average posterior parameters (7) over a number of generations (say, T»1 of them) after the exploration process has stabilized in a region where FOMs are relatively similar (i.e., achieved steady state, in direct analogy to MCMC estimation methods). Define the following measure of stability of the FOM during a window of H (≥2) successive generations:

$$\widetilde{\Delta}_{Hg} = \frac{\widetilde{\sigma}_{HNg}}{\widetilde{L}_{HNg}}, \tag{15}$$

where

$$\widetilde{L}_{HNg} = \frac{1}{H}\sum_{h=0}^{H-1} L_{N,g-h} \tag{16}$$

$$\widetilde{\sigma}_{HNg} = \sqrt{\frac{1}{H-1}\sum_{h=0}^{H-1}\left(L_{N,g-h} - \widetilde{L}_{HNg}\right)^2} \tag{17}$$

Stability criterion $\widetilde{\Delta}_{Hg}$ is the coefficient of variation (standard deviation over mean) of the FOM over H generations from (g-H+1) to the current generation g. If this criterion becomes smaller than or equal to the threshold $\overline{\Delta}$ at generation $g_s(\overline{\Delta})$, we deem the search stabilized. To promote achievement of stability by reaching steady state, it is customary to impose a burn-in period of g_{burn} generations, so require that $g_s > g_{burn}$. S1S is then defined as the average posterior parameters over generations g_s, …,g_s + T-1, according to expression (18):

$$\widehat{\beta}_{n,S1S} = \frac{1}{T}\sum_{g=g_s}^{g_s+T-1}\widehat{\beta}_{ng}, n = 1, \ldots, N, \tag{18}$$

with corresponding FOM

$$L_{Ng}^{S1S} = \sum_n lnl_n\left(\widehat{\beta}_{n,S1S}\right) \tag{19}$$

This last expression is used for both estimators (14) and (18).

It will be noted that the two solution strategies outlined above produce point estimates of the *N*x*K* preferences of interest. During the course of obtaining the Strategy = AVERAGE solution, one will have observed *T* points in *K*-parameter space for each respondent. Therefore, it is possible to construct the posterior parameter distributions for each individual, in direct analogy to a Bayesian MCMC method. We leave this further development to future research, and focus on estimators BEST and AVERAGE, particularly the latter.

The complete expression of the Genetic Algorithm follows:

Algorithm. EvGA (Search Figure-of-Merit Space)

Algorithm EvGA (Search Figure-of-Merit Space)

1: Initialize the pool Π_0 by sampling *M* points at random from Ξ-space, which is a (reasonably) bounded subspace of the parameter space Ω.

2: Initialize solution conditions ...

2.1: Stable ← False; t ← 0

2.2: $L_{best} \leftarrow -\infty$

2.3: Iterate ← True; $g \leftarrow 0$

3: While Iterate ...

3.1: $g \leftarrow g+1$

3.2: Execute Algorithm UCP to update Π_{g-1} to Π_g.

3.3: Calculate posterior individual parameters $\hat{\beta}_{ng}$(expression 7) for pool Π_g and n=1,…,*N*.

3.4: Calculate figure-of-merit L_{Ng} (expression 8) for pool Π_g.

3.5: If $L_{Ng} > L_{best}$, then … $L_{best} \leftarrow L_{Ng}$; $\beta_{best} = \hat{\beta}_g$.

3.6: Calculate stability criterion $\widetilde{\Delta}_{Hg}$(expression 15).

3.7: If not Stable and ($g > g_{burn}$ and $\widetilde{\Delta}_{Hg} \leq \widetilde{\Delta}_{stop}$), then Stable ← True.

3.8: If Stable and Strategy==AVERAGE, then
a) $t \leftarrow t+1$
b) Accumulate posterior individual parameters for this generation for subsequent evaluation of $\hat{\beta}_{n,S1S}$.

3.9: Iterate ← (Strategy==BEST and $g<G$) or (Strategy==AVERAGE and Stable and t<T) or (Strategy==AVERAGE and not Stable and $g<G$)

4: If (Strategy==BEST), then $\hat{\beta}_{n,S1S} = \hat{\beta}_{n,best}$, else calculate $\hat{\beta}_{n,S1S}$from expression (18).

5: Calculate FOM for $\hat{\beta}_{n,S1S}$from expression (19).

6: End

It is possible to halt the pursuit of optimal individual parameter estimates and perform prediction with the S1S. The classical statistician or econometrician might be dissatisfied with this intermediate solution, but I'd like to note that appropriate care in specifying the parameters of IPL-GA will lead to quite good solutions particularly for the averaging across stable generations (Strategy = AVERAGE). In my relatively extensive computational experience with this algorithm over a period of years, I found that the estimates provided by (18) have low gradient norms (though not necessarily zero) and are generally not "far away" from the optimal solution found in Stage 2 optimization searches. More on this topic subsequently.

Before turning our attention to the Stage 2 procedure to bring estimates to optimality, I first deal with the topic of imposing sign constraints on parameters in Stage 1. Stage 1 turns out to be a more convenient phase at which to impose constraints due to the simplicity of doing this, as we shall see below. The alternative is to use a constrained maximum likelihood procedure in the optimization phase, which is more difficult empirically and less tractable theoretically. With evolutionary algorithms it is far more common to handle constraints (e.g., sign restrictions) by "incentivizing" pool evolution to favor pool members with fewer and fewer constraint violations, eventually eliminating such violations entirely from the pool. This requires redefining fitness function (10) to reflect sign violations in pool members:

$$f_{mg}^{\rho} = \sum_n 1\left(l_{n|mg} \geq l_{n|m'g}\right) - \rho \sum_{k:\kappa_k \neq 0} 1\left(sgn\left(\pi_{mgk}\right) \neq \kappa_k\right), all\ m' \neq m, m = 1, \ldots, M, \tag{20}$$

where

$$\kappa_k = \left\{ \begin{array}{cc} -1 & \text{if } k \text{ must be} \leq 0 \\ 0 & \text{if no sign restriction} \\ +1 & \text{if } k \text{ must be} \geq 0 \end{array} \right\}, k = 1, \ldots, K, \tag{21}$$

expresses the sign requirements on dimension k, and

$$sgn(x) = \left\{ \begin{array}{cc} -1 & x < 0 \\ 0 & x = 0 \\ +1 & x > 0 \end{array} \right\} \tag{22}$$

Due to the second term in the penalized fitness measure f^{ρ}, pool members with sign violations become less likely to reproduce. Parameter $\rho \geq 0$ is a penalty weight that accelerates the purification of the pool. This value can be relatively large (10, 100, 1000, …, 10^6, …) depending upon the problem data. Setting $\rho = 0$ returns us to the original fitness function (10).

This scheme for handling simple sign constraints is very consistent with the evolutionary conceptualization of Stage 1. Experience has shown it works remarkably well with the type of choice data common to this problem setting. While mutations make it possible that sign violations are occasionally re-introduced, the penalized fitness measure usually soon rids the population of undesirable candidates. Calibrating/adjusting the penalty ρ to the particular data helps to (relatively) quickly produce pools with no sign violations, and the elitist reproduction rules help to keep them that way. The key to successful employment of this form of constraint imposition is to drive the sign violations to zero early in the evolutionary search, which can be accomplished through judicious choice of the penalty parameter ρ and by performing multiple trials to explore different regions of feasible space. In addition, it should be noted that using larger pool sizes to improve posterior preference estimates also implies that the absolute number of sign violations to be eliminated will likely increase, which may require increasing penalty ρ.

I have also implemented a bound-constrained nonlinear search in Stage 2, which supports this Stage 1 penalized approach. Together, these methods promote the elimination of sign violations if bounds are imposed. On the issue of imposing sign constraints in the case of coefficients such as price, it should be noted that this may generate the same issues of unstable WTP (see McFadden 2022) that can arise from the use of lognormal distributions in parametric heterogeneity distributions due to individual posterior estimates around the boundary value of zero.

Stage 2: Completing the Estimation of Individual Preference Parameters Through Gradient-Based Search Methods

As noted earlier, many algorithms exist for optimizing continuous, twice-differentiable figure-of-merit functions such as expression (8). I do not intend to discuss the technical details of these algorithms since they are extensively described in other sources (see, e.g., Gill et al., 1981, Dennis and Schnabel 1983, for gradient and gradient secant methods, Berndt et al., 1974 for the BHHH algorithm). What I wish to discuss is how these algorithms have been adapted to the problem of estimating individual preference parameters for a sample of N decision-makers, starting from the S1S.

In the maximization of a function $f(X|\theta)$, X being a data matrix and θ a parameter vector, the nonlinear search methods to which we refer make use of a gradient vector $\nabla_{\theta} f(X|\theta)$ to decide good directions of search. Thus, optimization proceeds by repeated movement over some set of θ-points that lead to the norm of $\nabla_{\theta} f(X|\theta)$ being small enough to stop the search at a stationary point. That final θ is taken as the (possibly local) optimal solution, θ*.

The issue to hand is that we effectively have a functional with N individual parameter vectors β_n to be estimated, rather than a parametric characterization of these via some distribution function h_{β}. In the search approaches I have explored to date, the search direction is determined at the sample level and the step size is assumed to be the same for all individuals. In detail, the updating of the posterior individual preference estimates proceeds by simply applying the same step size to the individual preferences of all $n = 1, \ldots, N$ decision-makers, thus (I assume no boundaries to the parameter space):

$$\beta_{n,j+1} = \beta_{nj} + \varphi \cdot \Delta_j, \tag{23}$$

where j is an iteration of the search, Δ_j is the multidimensional direction of movement for all n at iteration j, and φ is the scalar step-size to be taken. For the gradient/steepest ascent search method, the direction Δ_j at iteration j is

$$\Delta_j = \sum_n \nabla_{\beta} lnl_n(\beta_{nj}), \tag{24}$$

whereas for the BHHH method the dimensional changes are given by (Berndt et al., 1974)

$$\Delta_j = A_j^{-1} \left(\sum_n \nabla_{\beta} lnl_n(\beta_{nj}) \right), \tag{25}$$

where A_j is a KxK crossproduct matrix (or the Jacobian of the log likelihood function) defined as

$$A_j = \left(\frac{\partial ln\ l_n}{\partial \beta_{nj}}\right)^t \left(\frac{\partial ln\ l_n}{\partial \beta_{nj}}\right). \tag{26}$$

The partial derivatives in (24-26) are the gradient of the *individual* log likelihood functions $ln\ l_n$ (expression 9), evaluated at the posterior taste estimates for the individual, and contains one column per parameter, producing thus an *N*x*K* matrix. This makes the crossproduct matrix *K*x*K* in expression (26), which results in a dimensionally consistent matrix product in (25), and yields a uniform *K*x1 change vector for all individuals. One of the characteristics of A_j is that it is everywhere positive definite, guaranteeing that its inverse exists for obtaining direction Δ_j (Berndt et al., 1974).

Customarily, for the gradient and BHHH search methods, one first attempts $\varphi = 1$ to obtain a better FOM; if no improvement pertains, successively reduce φ, e.g., by halving the step size or performing an optimization assuming a quadratic function approximation, until a FOM improvement is obtained.[3] The crucial point about updating equation (23) is that the direction and step size are the same for all decision-makers. Future work should examine the possibility of making the updates individual-specific.

So, Stage 2 proceeds as described below:

Algorithm. GradSrch (Optimize Figure-of-Merit)

Algorithm GradSrch (Optimize Figure-of-Merit)

1: Initialize $\beta_1 = \hat{\beta}_{S1S}$.
2: Converged ← False; j ← 1
3: While not Converged
3.1: Calculate direction vector Δ_j according to (24) or (25), as appropriate.
3.2: Update individual preferences using (23) to obtain β_{j+1}.
3.3: Calculate the FOM L_N at β_{j+1}.
3.4: Test stopping on a) maximum iterations, b) change in FOM below a threshold, and c) Euclidean norm of the gradient below a threshold.
1) If (b) and/or (c) pertain, Convergence ← True.
2) Meeting any of conditions (a)-(c) leads to Exit loop.
3.5: $j \leftarrow j+1$
4: If converged, $\hat{\beta}_{S2S} = \beta_j$.
5: End

The global search by Algorithm EvGA plus the local search by GradSrch is a potent combination to help identify the global optimum. However, once we move beyond the MNL model and linear-in-parameters utility function, the likelihood for which is guaranteed to have a single (therefore, global) optimum, we cannot be sure that maximum likelihood (ML) solutions reported by econometric estimation software are, in fact, global optima. Unfortunately, few choice modelers understand just how little knowledge we have about the topography of the likelihood function based on the interplay of a highly nonlinear model form with low density (i.e., sparse) choice data. Practically speaking, current best practice is to start ML optimization from Q random points (generally, Q is relatively small, likely below 10, and perhaps most often, simply one), and accept as global optimum the best solution thus found. Another option is to employ an exploratory algorithm less prone to getting stuck at local optima, e.g., Stage 1 in the current algorithmic combination, to provide good starting points for the optimization. In a sense, these first-stage algorithms are being used to map out the topography of the parameter space, with some care to avoid getting trapped by local optima. Despite these "global search" capabilities, however, these methods still cannot eliminate convergence to local optima, so it is still advisable to try multiple random starts.

Fig. 1 illustrates this abundantly clearly, by mapping out the posterior log likelihood values over about 300 generations for the 11 best S1S solutions from a 50-trial search for a particular DCE data set. Each trial corresponds to a different, randomly generated starting candidate pool. The prototypical evolution of each search is very similar, as the GA improves the candidate pool until it stabilizes after some 200 generations. These curves demonstrate the variety of solutions found, with significant variability in the posterior log likelihood, indicating the likely existence of many optima. And to remind the reader, these are only the 11 best S1S solutions, implying that the 39 searches not shown depict an even broader spectrum of search performance. In addition, the graph also shows the gap between the best S1S solution and the optimal S2S solution found by algorithm GradSrch starting from that best: the gap is on the order of about 200 log likelihood units, a non-negligible magnitude. Fig. 1 both illustrates the exploratory search behavior of the genetic algorithm during Stage 1 of the search, and the incremental but crucial benefit of the Stage 2 optimization search.

2.1. *Estimation of standard errors at the individual level*

The optimization of FOM L_N (expression 8) should be followed by calculation of standard errors for $\widehat{\beta}_{S2S}$ at the individual decision-maker level. Multiple methods are available: the straightforward method requires calculating the information matrix of the posterior log likelihood (expression 9) for each *individual* decision maker, over the *R* replications of the individual; another is to use the inverse of the *individual* cross-product matrix from the BHHH method (expression 26), based on the equality of this matrix and the information

[3] If φ becomes too small, usually the result of difficult topography of the likelihood function (e.g., saddle points, flatness around optima, collinearity between dimensions), it is necessary to take appropriate steps. These search methods are subject to scaling effects of the *X* matrix, so re-scaling of the independent variables so they are approximately of the same magnitude can be helpful in circumventing this issue.

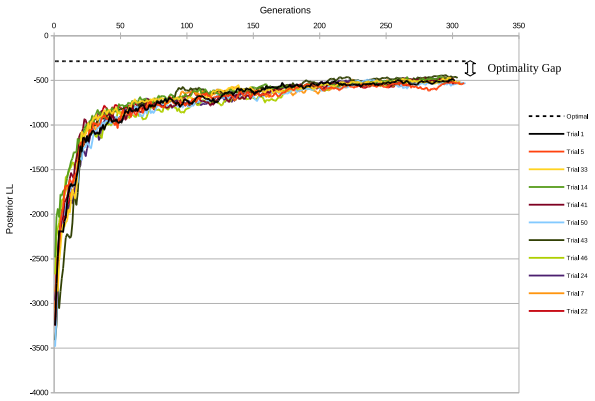


Fig. 1. Illustration of generational evolution of posterior log likelihood for best eleven feasible constrained solutions.

matrix for maximum likelihood estimators, under the assumption that the data generation process is correctly specified and that the global optimum has been found. These procedures can rely on numerical differentiation, or can use analytical expressions since the first- and second-order derivatives for the MNL model are readily derived or available from the literature (e.g., in Ben-Akiva and Lerman 1985, expressions 5.34–5.35 respectively specify its gradient and Hessian elements).

Computational experience with straightforward implementations of these two ideas has amply demonstrated to me the difficulties inherent to them in this estimation setting. Specifically, achieving optimality of the sample posterior log likelihood does not, of course, guarantee that each individual's posterior log likelihood is at an optimum. Empirically, this leads to individual-level information matrices often (even generally) not being positive definite, which means they're not proper estimators of the information matrix. This is a problem previously (and commonly) encountered in control systems theory, among others. It is addressed by Spall (2005), who proposes a combination Monte Carlo simulation and bootstrap approach to estimating both the matrices mentioned above. Initial testing with this method shows some improvement over the deterministic no-simulation method, but at least presently is still problematic from my perspective.

I have opted instead to directly apply the jackknife or bootstrap methods (Efron and Tibshirani 1993) to calculate individual decision-maker standard errors for parameter solutions. Bootstrapping and the jackknife have a long history in statistics and the sciences (e.g., Efron and Tibshirani 1993). Both methods involve 1) manipulations to the data set to configure alternative samples based on the original sample, 2) followed by application of Algorithm GradSrch to these samples to estimate the statistics of interest, namely the individual preferences, and 3) use of the variability in individual level preference parameters across samples as the basis for calculating standard errors.

The jackknife requires deleting one observation at a time from the original sample of N to create estimation sample $S_i = \{1,2, \ldots,i\text{-}1, i\text{+}1, \ldots,N\}$, $i = 1, \ldots,N$, each with (N-1) of the original observations, then executing Algorithm GradSrch starting at $\widehat{\beta}_{S2S}$ to obtain estimate $\widehat{\beta}_i$ of (N-1)xK individual preference parameters. This subsampling-estimation cycle is repeated N times. The standard error of a particular element k of person n's preferences is then given by

$$\sigma_{Jnk} = \sqrt{\frac{1}{(N-2)}\sum_{s=1,s\neq n}^{N}(\beta_{nks} - \overline{\beta_{nk}})^2} = \sqrt{\frac{(N-1)}{(N-2)}\left(E\left(\beta_{nks}^2\right) - E(\beta_{nk})^2\right)}, k = 1, \ldots, K, n = 1, \ldots, N, \tag{27}$$

where

$$E(\beta_{nk}) = \frac{1}{(N-1)}\sum_{s=1..N,s\neq n}\widehat{\beta}_{nks}, E\left(\beta_{nks}^2\right) = \frac{1}{(N-1)}\sum_{s=1..N,s\neq n}\widehat{\beta^2}_{nks}, k = 1, \ldots, K, n = 1, \ldots, N. \tag{28}$$

Expression (27) differs slightly from Efron and Tibshirani (1993, p. 141) by having a factor (N-1)/(N-2) rather than (N/(N-1)) inside the square root operator. This follows from the explicit consideration of the number of samples (N-1) in which a given sample member n is present, and defining the unbiased standard error estimator divisor to therefore be (N-2). For N > 10, the difference between these factors is minuscule in absolute terms.

The bootstrap uses a somewhat different scheme to produce samples: B independent bootstrap samples of size N are drawn by random sampling with replacement from among the original decision makers. Let these samples be indexed by $i = 1,\ldots,$B, and the resulting estimators termed $\ddot{\beta}_i$. Note that each individual n may be present in bootstrap samples a different number of times, which we will denote as $1 \leq B_n \leq B$. The bootstrap standard error calculation for person n and parameter k is (Efron and Tibshirani, 1993, p. 47)

$$\sigma_{Bnk} = \sqrt{\frac{B_n}{(B_n-1)}\left(E\left(\beta_{nks}^2\right) - E(\beta_{nk})^2\right)}, k = 1, \ldots, K, n = 1, \ldots, N, \tag{29}$$

where

$$E(\beta_{nk}) = \frac{1}{B_n}\sum_{s=1..B_n}\overset{\cdots}{\beta}_{nks}, E\left(\beta_{nks}^2\right) = \frac{1}{B_n}\sum_{s=1..B_n}\overset{\cdots}{\beta^2}_{nks}, k = 1, \ldots, K, n = 1, \ldots, N. \tag{30}$$

Despite the increased amount of computation time needed to calculate these standard error estimators, my experience has shown that the requirements are quite reasonable relative to the total estimation time for the sample sizes and replication numbers that are usual in Discrete Choice Experiments and panel RP data sets used by choice modelers. And an advantage not to be discounted is that the resampling methods make direct use of algorithms already developed and debugged to produce the point parameter estimates, and thus require little extra effort on the programmer's part to implement.

2.2. A note on forecasting

Working directly with individual posterior preferences requires that in-sample forecasting be done by sample enumeration. Out-of-sample forecasting with the same individuals is similarly handled since there is a one-to-one mapping between in- and out-of-sample decision-makers. To obtain confidence intervals for predictions and other quantities, individual preferences can be drawn from the individual-level distributions characterized through the standard errors calculated above. But the truth of the matter is that with individual posterior parameters for a sample of respondents, we are limited to 1) making statements only about the sample, and/or 2) weighting statistics of interest to the population level, though recognizing that these are necessarily conditioned on the observed sample choices used to calculate the individual posterior parameters. This places the burden of sampling strategy squarely on the shoulders of the analyst, where it should reside anyway.

Out-of-sample predictions are a challenge based on the posterior taste distributions we have estimated through IPL-GA. The basic issue is that we have the multivariate distribution of posterior taste parameters for a sample of respondents rather than an estimate of the prior population distribution of tastes h_β. If we had h_β, we could integrate figures-of-merit like elasticities, marginal rates of substitution, consumer surplus, etc., over the multivariate density to obtain population estimates of statistics of interest. Since we don't have h_β, we are forced to consider methods such as synthetic samples and weighting to predict out-of-sample with the IPL-GA estimates. Again, either of these methods will produce statistics conditioned on the observed sample choices.

2.3. Backtracking to answer an important question: but does it work?

I undertook a Monte Carlo simulation exercise to investigate the capability of the hybrid evolutionary and gradient search algorithms to recover known individual-level parameters. Table 1 specifies the (many) details of the exercise, arrayed in the following order: 1) sample and choice process characterization; 2) parameters of Algorithm EvGA; 3) parameters of Algorithm GradSrch; and 4) replications of the core simulation. Two separate population preference structures were simulated: a) Unimodal preferences (UP), with all K preferences of N sample members drawn from independent uniform distributions U[-4,+4]; b) Bimodal preferences (BP), with sample members drawn with equal probability from either of two distributions, U[-5,-1] and U[1,5], for each of K preference dimensions. The BP experiment represents a two-segment population, one group having all negative preferences and the other all positive preferences, and is included to represent the possibility of multi-modal preferences. Below I present an overview of the most important details to give the reader a sense of the purpose and scope of the exercise, but will purposely leave other (lesser) details for perusal in the table.

The simulation has a sample size of $N = 500$ decision-makers, with individual preferences drawn at random from the distributions specified above. One implication of this selection of distribution is that all preference dimensions are independent (i.e., zero correlation). The number of preference dimensions is varied systematically over $K = 4$, 8, 16, and for the number of replications over $R = 4$, 8, 16, 32. The ranges for K and R are quite broad, once we consider that most models with parameter heterogeneity have 16 or so parameters including stochastic distribution parameters (such as variances), and present-day practice for DCEs is to implement around 16 replications at most. The choices for the individuals in each replication are simulated to obey the utility maximizing rule underlying the MNL model, using each person's true preference parameters to calculate systematic utility, which is then perturbed with a Gumbel (unit scale) stochastic error.

The combinations of the K and R factors define 12 individual conditions/cells of the (K,R) experiment, for each of UP and BP conditions. For each of the 12 cells, 25 data set replications are generated randomly according to the criteria given in Table 1. In the replications within a study cell, each individual maintains the same preferences throughout. The solution strategy adopted is to calculate average posterior parameters as the S1S, then optimize using the BHHH algorithm to obtain the S2S solution. Ten solution attempts using random starting candidate pools are made for each of the 25 data sets.

Stage 2 solutions $\widehat{\beta}_{S2S}$ were obtained for each data set using the full hybrid algorithm. These were compared to the true parameters in two ways: 1) the root mean square error normalized by true parameter range (referred to as RMSE_T), and 2) the average correlation between the estimated and true parameters at the individual level (RHO). These statistics are defined at the data set level, then averaged over replications within experimental cell.

Fig. 2 presents these summary statistics as a function of the number of replications R, for each of UP (Panel a1) and BP (Panel a2) experiments. In evaluating these statistics in the following paragraphs, I remind the reader that the number of parameters being recovered is very large ($=N\bullet K = 500$ K in this exercise) compared to usual simulation studies of this type, in which we would be considering the recovery of K parameters.

Table 1
Setup details for parameter recovery simulations.

Symbol	Description	Value, Range or Definition
Sample and Choice Process Characterization		
N	Sample size (number of decision-makers)	500
R	Number of choice replications per decision-maker	4,8,16,32
J	Number of alternatives a in a replication	3
K	Number of independent variables X_{nra}	4,8,16
X_{nra}	K predictor variables drawn from independent normals for each person, replication, and alternative	Attribute $X_{nrak} \sim$ Normal mean = 0, variance = 4, $k = 1, \ldots, K$; $\text{cov}(X_{nrak}, X_{nral}) = 0$, $l \neq k$
ε_{nra}	Gumbel (scale) error term to perturb utilities	~Gumbel(1)
V_{nra}^{true}	Systematic true utility of alternative a, replication r, person n	$= \beta_n^{true} X_{nra}$
U_{nra}^{true}	Full utility of alternative a, replication r, person n	$= V_{nra}^{true} + \varepsilon_{nra}$
δ_{nra}	Choice indicator for alternative a, replication r, person n	$= 1$ $if U_{nra}^{true} > U_{nrj}^{true}$ all $j \neq a$ $= 0$ otherwise
Population Taste Structure		
β_n^{true}	1xK vector of true individual preferences	Homogeneous: ~U[-4,+4] Heterogeneous: C1 (p = 0.5): ~U[-5,-1] C2 (p = 0.5): ~U[1,5]
Parameters for Algorithm EvGA		
Strategy	Estimator for individual preference parameters	AVERAGE
G	Number of generations simulated	Max(200,g_s + T)
M	Pool size	10,000
g_s	Burn-in generations	100
T	Number of required stable generations for implementing AVERAGE estimator	100
D	Lower fitness percentile of pool discarded before update	10%
$(\alpha_P, \alpha_R, \alpha_{CL}, \alpha_{CR})$	Probabilities of Pairing, Recombination, Clone Left and Clone Right	(0.20, 0.20, 0.30, 0.30)
$\mu_{\min}, \mu_{\max}$	Minimum and maximum mutation probability	0.05, 0.20
Q	Mutation adjustment rate based on diversity	0.5
C	Mutation time rate	−1.0
H	Width of window (in generations) for calculating stability criterion	10
$\overline{\Delta}$	Stability criterion threshold	Variable
Parameters for Algorithm GradSrch		
Method	Method for calculating directions during iterative updating	Gradient
	Maximum iterations	10000
	Convergence tolerance for norm of gradient	0.05
	Convergence tolerance for change in FOM	10^{-10}
Simulation Specification		
N_D	Number of datasets per (R–K) condition	25
N_{trials}	Number of solution trials per dataset	10

Taking these statistics in turn, Fig. 2a (Panel a1) demonstrates that with small K (=4) in the UP population, RMSE_T decreases sharply with replications, reflecting the benefit of more replications in improving the accuracy of recovering parameters. With $K = 8$, RMSE_T is much lower than at $R = 4$, then decreases somewhat with $R = 8$ replications, then increases marginally with $R > 8$. For $K = 16$, RMSE_T remains of comparable magnitude to the average RMSE_T of $K = 8$, so additional replications do not decrease the variability in parameter recovery. Both levels and patterns seem to hold generally for the Bimodal (BP) condition, suggesting that the estimation method may be robust to number of modes in preference distributions. The important takeaway may be that, whatever the underlying distribution, one should have sufficient replications at the individual level to support reliable estimation of individual preferences; how preferences are distributed at the sample level is of secondary importance, since they will be accurately recuperated if accurate estimates of individual preferences are obtained.

Fig. 2b presents graphs of RHO as a function of replications R, for different numbers of predictors K. We note to no surprise that for all K, the more data/replications R are available, the more accurate the estimator is at recovering the known individual parameters. For the UP condition, with $K = 4$ the proposed algorithm sequence recuperates true parameters over the 500 individual decision-makers with RHO ranging above 0.80 with as few as 4 replications per person. Increasing to $K = 8$ dimensions increases the error of recovery to an average RHO ranging from a minimum of 0.65 to a high of 0.91 for 32 replications. Finally, for $K = 16$ RHO varies from 0.48 to 0.81 as R varies from 4 to 32. For the BP condition, this same general pattern again holds, but interestingly does so with higher average RHO. This may be due to the construction of the two sub-populations in the BP condition, since one sub-population has all negative preferences and the other has all positive preferences. Nonetheless, these results suggest that the proposed estimation algorithm recuperates preferences relatively accurately, given sufficient choices at the individual level.

The combination of $8 \leq K \leq 16$ dimensions and $8 \leq R \leq 16$ replications is something of a "sweet spot" for DCE studies, and we see that the hybrid algorithm performs quite well in these conditions with UP: RHO$\geq$0.79 for $K = 8$ and RHO$\geq$0.60 for $K = 16$ (per Fig. 2b Panel b1); and BP: RHO$\geq$0.93 for $K = 8$, 16. It is the rare researcher who ventures into the region of $R \geq 16$, but this region is of course very helpful for success of the algorithm, since RHO is estimated to be around 0.8 for $K = 16$ parameters in the UP condition and 0.9 in the BP condition.

These simulations help establish that the proposed algorithm is capable of reliable preference parameter recovery at the individual

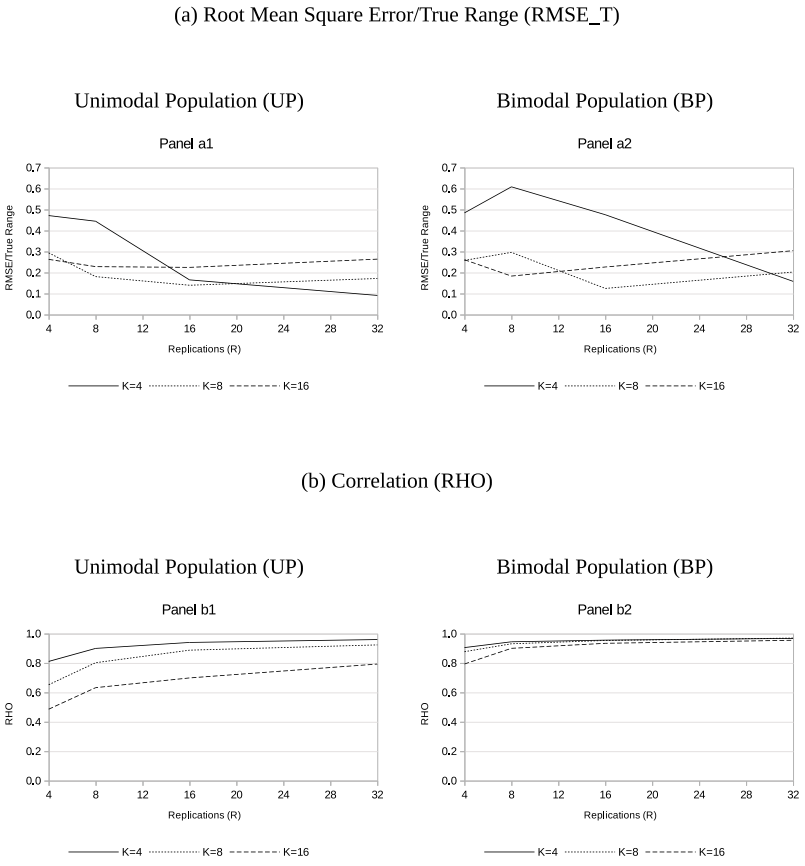


Fig. 2. Monte Carlo simulation study: Parameter recovery for unimodal and bimodal taste distributions.

level, particularly within bounds of 8–16 variables and 8–16 replications. More predictor variables than the range tested naturally impose a higher burden on the estimator for it to perform acceptably, and so require more replications than are likely to be available in stated preference studies. However, in some studies it may be possible to extend into these realms, as we will demonstrate in the following section of the paper.

3. An application of IPL-GA using revealed preference household-level scanner panel data

I now illustrate that the proposed estimator is practical for larger scale applications with revealed preference (RP) data. Scanner panel data from marketing is one of the better known panel RP data sources: participating households in focal markets have all their supermarket (and certain other retail channels) purchases recorded in proprietary databases. Historically, the marketing discipline has used these data to develop choice models to characterize basket purchasing patterns, preferences for product attributes, price elasticities and market structure analysis, impact of brand positioning, consumer segmentation, and so forth.

Swait and Erdem (2002) examined how temporal consistency in the marketing mix (product, price, channel and advertising) plus consistency in availability on store shelves impact buyer preferences. They employed fabric softener scanner panel data first used by Fader and Hardie (1996) to understand how product attributes influence choice. I also use these data, which pertain to a single supermarket for the period January 1990 to June 1992, and covers 59 different products or SKUs (stock keeping units). The attributes for each SKU are brand (# levels = 10), form factor (4), size (4), and formula (4). The marketing mix variables accounted for in the data, and collected independently of the purchases, are in-stock availability, regular price, promotion savings, display status and feature insert status. The data set has 594 households, which Swait and Erdem (2002) split randomly into two equal-sized data sets, one for estimation and another for out-of-sample prediction testing (thus, each with 297 observation units). I make use here only of their estimation data set. The interested reader is directed to the aforementioned papers for further information on the data.

The utility function has 30 predictor variables: nine brand constants, three form factor dummies, three size dummies, three formulation dummies, a regular price effect, a promotional discount (price cut) effect, display status, feature status, four measures pertaining to marketing mix consistency (see Swait and Erdem 2002), and brand, form, size and formulation "loyalty" measures (prior choice effects).

As a test bed for the IPL-GA algorithm, this is at the larger end of the problem size scale: $K = 30$, variable $2 \leq R \leq 40$ (mean = 6.2

purchases per household), and up to 59 alternatives in each choice set (product availability information was independently collected on a daily basis, which is an unusual feature of this data set compared to other scanner panel data). The N = 297 households made a total of 1 970 purchases, and there are 101,581 cases in the data. I estimate two models for this data set: 1) a "standard" Mixed Logit model with the common assumption of multivariate normal preferences with heteroscedastic diagonal covariance matrix (assuming, thus, independence between preference dimensions), estimated by simulated maximum likelihood with 2 000 Halton draws; and 2) a constrained IPL-GA model, for which no *a priori* distributional assumptions about preferences need be made, with 10,000 members in the candidate pool. In the latter model, the coefficients for Regular Price and Price Cut are, respectively, constrained to be non-positive and non-negative. Ten trials are made for the IPL-GA model to determine the reported solution.[4]

Table 2 presents the estimation results for these models. During estimation it was necessary to restrict certain variance parameters to zero in Model 1 (Mixed Logit), as noted in the table through the fixed -∞ values (note that the simulation estimator implementation estimates the natural logarithm of each variance). Model 1 has a posterior log likelihood of −2396.3, and Model 2 (IPL-GA Constrained) presents that measure at −2308.8. To put this result into context, the expectation is that these estimators should achieve comparable solutions, with the advantage generally being in favor of IPL-GA since it does not restrict heterogeneity distributions to any specific parametric family. An eyeball comparison of the means of the 30 parameters shows that only two (the Brand6 constant, and the Refill (base = Sheets) dummy) are markedly different across the models. Omitting these two parameters, the correlation between the estimated mean preference vectors from the two models is approximately 0.90.

Differences between the two models occur also at the parameter level. I highlight the parameter *Concentrated form factor*, which captures its impact relative to a base form (Sheets). In Model 1, the prior distribution of preference heterogeneity in this utility source is assumed to be an independent normal distribution with estimated parameters $(\mu \approx 0.02, \sigma^2 \approx 9.1)$, hence it necessarily has a single peak at μ. In contrast, the posterior preference estimates by IPL-GA feature a two-mode distribution, one with a negative mean and the other positive. A second example concerns the Regular Price parameter. First note that the simulation estimates of the prior for Regular Price can always produce positive prior and posterior coefficients values because the support region for the normal distribution is unbounded, but IPL-GA does not in this case because we specified a sign constraint. Fortunately, Model 1 shows that only one posterior price parameter is positive in this sample, so empirical use of the model can proceed. Several approaches, described earlier, could be used to deal with this condition, but we shall put this issue aside for purposes of this paper. By construction, the IPL-GA solution does not have any undesirable sign violations. More importantly, there is a notable difference in shape across the model inferences about price sensitivities: simulation estimates impose the symmetry implied by the normal prior density, whereas IPL-GA captures a distinct negative skew, allowing for a depiction of more negative price sensitivities than in Model 1. It is no doubt these kinds of differences between these heterogeneity estimators that are behind the somewhat better goodness-of-fit of the IPL-GA.

Model 3, an IPL-GA estimator using the AVERAGE operator but no optimization phase, is also presented in Table 2. To remind, the AVERAGE operator implements (18) to obtain the mean posterior parameters over T (=100 here) stable generations. The parameter values for Model 3 correspond to $\widehat{\beta}_{S1S}$, the Stage 1 Solution used as starting point for the optimization that leads to $\widehat{\beta}_{S2S}$, reported as Model 2. The posterior log likelihood of $\widehat{\beta}_{S1S}$ is −2553.36, compared to −2308.78 for $\widehat{\beta}_{S2S}$. The purpose of showing this last model is two-fold: first, to illustrate the benefit of the optimization performed in Stage 2, which was significant in this data set; but secondarily, to show that the AVERAGE operator yields a rather good solution which may have value on its own merits. A comparison of the parameters between Models 2 and 3 shows that they do not differ much in magnitude, though the former are optimal for the posterior log likelihood function. Future work should examine the relative benefit of employing only the first stage of the solution procedure, despite its lack of optimality.

We finish this example with Fig. 3, which shows the demand impact on Brand 4's portfolio (represented in the data through 15 SKUs, indexed 13–27 in the data) that arise from a pricing repositioning strategy: a) prices of all SKUs in the portfolio are jointly varied over the range [−20%,+20%] change, b) price cuts are eliminated/discontinued and c) pricing consistency over time is enhanced. Essentially, Brand 4's pricing strategy is changed from periodic discounting to always being the same undiscounted price, either below, at or above the current situation. The resulting changes in demand for Brand 4 are shown in the graph, as percent differences compared to the 0% condition (i.e., the status quo described in the data set, including price cuts and pricing consistency). Each of the three models in Table 2 was used to make predictions, using individual-level posterior parameter estimates. To remind, the Model 1 has no heterogeneity associated with a mean positive price consistency effect, but both the other models do allow for both positive and negative price consistency preferences. This latter result is to be expected, as argued in Swait and Erdem (2002): the positive effect for individuals who value price predictability, but a negative effect for those who value promotional savings. The IPL-GA (IPL-GA AVERAGE) model depicts the sample of decision makers as being 33% (43%) with negative price consistency preferences, hence 67% (57%) with positive preferences.

Despite model differences, the results of these policy simulations are largely consistent across models: departure from the status quo will result in demand losses for Brand 4, according to Model 1 and IPL-GA, and largely losses for IPL-GA AVERAGE. For this latter model, portfolio-wide price decreases of greater than 15% are predicted to increase demand relative to the status quo. Price increases are quite impactful: a 20% consistent price increase in the regular prices of Brand 4 SKU's and elimination of discounts will result in a 41% decrease in demand according to Model 1, 32% for the IPL-GA, and 45% for the IPL-GA AVERAGE, *ceteris paribus*. Thus, the best

[4] Just over 24 h of elapsed time were needed to estimate the 10 trials, for an average of 145 elapsed minutes per trial. The computational platform was a dedicated 64-bit Linux OS machine with 32 MB RAM memory, 8 cores running at 3.4 GHz, using specialty Fortran95 code with parallel processing, and compiled with gfortran v9.3.

Table 2
Estimation results for fabric softener data set.

Estimator	Mixed Logit Simulation Estimator		IPL-GA Estimator	
Solution Type	Optimal		Optimal	AVERAGE
Constraint Status	Unconstrained		Constrained	Constrained
# Sign Violations in Posterior Parameters	1		0	0
	[95% Confidence Interval]		[0.025, 0.975] percentiles	
	Prior Mean	$ln\sigma^2$ [a]	Posterior Mean	Posterior Mean
Brand1	−0.7976 *[-1.6901,0.0949]*	0.7347 *[-0.2640,1.7334]*	−1.2891 *[-3.0441,2.1853]*	−1.4286 *[-3.1837,2.0458]*
Brand2	0.4264 *[-0.149,1.0018]*	−0.6988 *[-2.1299,0.7323]*	−0.5666 *[-3.0282,3.9875]*	−0.2580 *[-2.7197,4.2961]*
Brand3	−0.8618 *[-1.7305,0.0069]*	1.2052 *[0.4525,1.9579]*	−0.8375 *[-2.5987,3.3341]*	−0.9370 *[-2.6982,3.2347]*
Brand4	0.7746 *[0.3197,1.2294*	−0.4109 *[-1.1235,0.3017]*	0.4348 *[-2.3838,4.609]*	1.3049 *[-1.5136,5.4792]*
Brand5	−0.0235 *[-0.7325,0.6856]*	1.6693 *[1.3569,1.9817]*	0.0455 *[-2.5582,4.8209]*	0.1384 *[-2.4653,4.9138]*
Brand6	−7.7983 *[-10.8011,-4.7955]*	3.5076 *[2.9014,4.1138]*	−1.4542 *[-3.5413,3.6687]*	−1.3627 *[-3.4498,3.7601]*
Brand7	−1.6801 *[-2.2842,-1.076]*	1.4095 *[0.9635,1.8555]*	−1.0130 *[-3.7479,4.0865]*	−1.5811 *[-4.316,3.5185]*
Brand8	0.2217 *[-0.2603,0.7036]*	0.4141 *[0.0069,0.8214]*	−0.0252 *[-3.4187,4.556]*	0.3890 *[-3.0045,4.9702]*
Brand9	0.5017 *[-0.0906,1.0939]*	−0.0122 *[-1.2662,1.2419]*	0.3454 *[-2.3145,3.733]*	−0.0616 *[-2.7215,3.3259]*
Concentrated (base = Sheets)	0.0211 *[-0.3231,0.3654]*	2.2055 *[1.8165,2.5945]*	−0.3779 *[-3.9796,4.0768]*	−0.3097 *[-3.9114,4.145]*
Refill (base = Sheets)	1.2405 *[0.617,1.864]*	2.3367 *[1.9409,2.7326]*	0.9529 *[-1.726,5.338]*	1.4228 *[-1.2561,5.808]*
Liquid (base = Sheets)	−14.6333 *[-23.2576,-6.009]*	4.377 *[3.4504,5.3036]*	−2.5668 *[-4.7546,0.479]*	−1.7941 *[-3.9818,1.2518]*
Small (base = X-Large)	−1.7376 *[-2.3812,-1.094]*	−0.0502 *[-0.6348,0.5344]*	−1.1882 *[-3.8476,2.1397]*	−1.2292 *[-3.8885,2.0987]*
Medium (base = X-Large)	−0.1879 *[-0.6248,0.249]*	-∞	0.5989 [-3.7712,3.2605]	1.315 [-3.0551,3.9767]
Large (base = X-Large)	0.0923 [-0.2785,0.4631]	-∞	−0.2422 *[-2.6885,3.3755]*	1.0174 *[-1.4289,4.6351]*
Regular Scent (base = Unscen)	−0.0515 *[-0.3761,0.273]*	-∞	−0.5326 *[-3.0402,2.5168]*	−0.7935 *[-3.3011,2.2559]*
Light Scent (base = Unscen)	0.6695 *[0.3102,1.0289]*	0.675 *[0.2315,1.1185]*	0.9082 *[-3.4716,3.8746]*	0.8613 *[-3.5186,3.8277]*
Stainguard (base = Unscen)	−0.2220 *[-0.7875,0.3434]*	−0.0212 *[-1.5649,1.5225]*	−0.3425 *[-2.3859,2.5597]*	−0.3371 *[-2.3805,2.5652]*
Regular Price of SKU	−1.1377 *[-1.3479,-0.9275]*	−1.1810 *[-1.581,-0.781]*	−1.8543 *[-4.7623,-0.1589]*	−2.9618 *[-5.8742,-1.2708]*
Price Cut of SKU	1.8134 *[1.5996,2.0272]*	−0.9018 *[-1.497,-0.3066]*	2.9369 *[1.4391,4.7639]*	4.5703 *[3.0724,6.3972]*
Display	1.1801 *[0.8939,1.4663]*	-∞	1.2902 *[-1.1731,4.7912]*	1.0662 *[-1.397,4.5673]*
Feature	0.5915 *[0.2223,0.9608]*	-∞	0.4723 *[-1.9184,3.6426]*	0.3563 *[-2.0344,3.5266]*
Price Consistency	0.7308 *[0.4457,1.016]*	-∞	0.4154 *[-2.5626,2.9329]*	0.0607 *[-2.9174,2.5781]*
Display Consistency	0.5186 *[-0.0637,1.1008]*	-∞	0.7179 *[-1.5157,2.8254]*	1.1196 *[-1.114,3.2271]*
Feature Consistency	−0.4338 *[-0.9734,0.1058]*	-∞	−0.4992 *[-2.7134,2.1381]*	−0.3807 *[-2.595,2.2566]*
Shelf Availability Consistency	−4.6569 *[-5.8558,-3.4581]*	-∞	−3.4125 *[-4.9392,-1.6125]*	−1.096 *[-2.6227,0.704]*
Brand Loyalty	0.3113 *[0.121,0.5015]*	-∞	−0.2862 *[-3.2764,2.9256]*	0.1267 *[-2.8634,3.3385]*
Form Loyalty	0.2684 *[-0.0366,0.5735]*	−0.5719 *[-1.5482,0.4045]*	0.7552 *[-2.3037,3.378]*	1.1980 *[-1.8608,3.8209]*
Size Loyalty	0.1206 *[-0.0347,0.2758]*	-∞	−0.0890 *[-2.7896,2.9183]*	−0.1330 *[-2.8337,2.8743]*
Formula Loyalty	0.1752 *[-0.0363,0.3868]*	-∞	−0.0140 *[-3.9472,2.1158]*	0.0182 *[-3.915,2.148]*
Log Likelihood	−4963.87			
Posterior LL	−2396.31		−2308.78	−2553.36
Halton Draws or Pool Members	2 000		10,000	10,000
N	297			
# choice sets	1970			
# cases	101,581			

Notes.
[a] Heterogeneity given by independent normal distributions $N_k(\beta_k, \sigma_k^2)$, k a preference dimension. Mean and $ln\sigma^2$estimates shown in table.

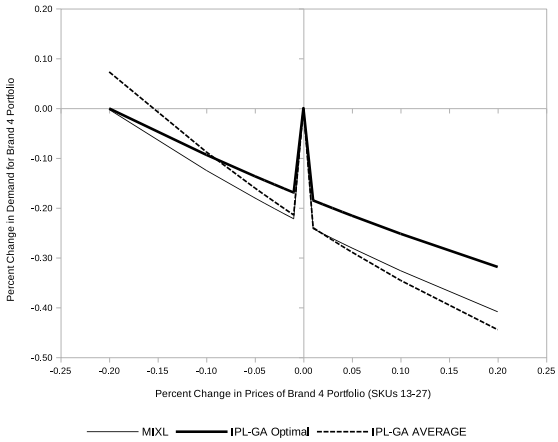


Fig. 3. Price repositioning analysis for brand 4 portfolio.

fitting model (IPL-GA) depicts the demand impact of the price repositioning to be less significant than predicted by the other models, but clearly indicates that the status quo pricing policy remains the best course to follow for this sample.

4. Summary and conclusion

This paper has examined the potential for employing evolutionary algorithms, specifically a genetic algorithm, as the front end of a two-stage estimation process to obtain posterior individual preference parameters for the logit model. The main distinguishing feature (and practical advantage) of this algorithm is that it is distribution-free: the analyst does not have to specify an arbitrary set of parametric densities to which the preference dimensions must adhere. The choice data, through replications at the individual decision maker level, will dictate the forms of posterior preference distributions supported by the kernel choice probability formulation. These distributions may turn out to be symmetric or skewed, single- or multi-modal, have point masses in addition to continuous representation, and are bounded by construction. In addition, correlations between some or all preference dimensions will arise as part of the solution process, not through the imaginations of analysts. The cost of adopting a distribution-free algorithm does create a large estimation problem, since NxK parameters are in play.

I have laid out the details of this two-stage estimator (termed IPL-GA), which combines the genetic algorithm to obtain good (even near optimal) solutions that can then be used as starting points with a traditional gradient- or hessian-based search algorithm. The example applications in the paper used an implementation of the BHHH search method (Berndt et al., 1974), but many other algorithms can be used. IPL-GA combines the power of evolutionary algorithms to search the parameter space more "globally", increasing the chances of avoiding local optima (which gradient-based searches are built to find if they are "nearby locally"). It is hoped that the hand-off to the traditional search algorithm with a better starting solution will increase the chance that the estimator converges to the global optimum. While there are no guarantees that this ideal state will ensue in any given application, the combination of methods increases the likelihood of this occurring, particularly when the estimator is embedded in a systematic course of multiple random starting candidate pools.

The use of evolutionary algorithms as a means to generate good starting solutions for optimization routines is far from unique to this research. There is a sizeable and still expanding literature on global optimization methods,[5] which often employ combinations of specific procedures to search for global optima. Historically, an early hybrid algorithm was the combination of a gradient-free method like the Nelder-Mead algorithm (Nelder and Mead 1965) – which uses only objective function values to optimize a function – as the first stage, with a second stage based on a gradient-type method. Other examples of front ends would include taboo searches, simulated annealing, heuristics of many flavors, direct random searches, random direction searches, and multi-resolution genetic algorithms (Holloman et al., 2006).

Genetic algorithms have traditionally been applied to find optimal or near-optimal solutions to optimization problems. In IPL-GA, I have used the GA as a parameter-space exploration procedure to map the topography of the posterior log likelihood function. This gives the method a somewhat computational Bayesian flavor; the literature has recognized this link between genetic algorithms and Markov Chain Monte Carlo methods (see, e.g., Strens 2003, ter Braak, 2004; Holloman et al., 2006). Differently from Bayesian methods, however, there is no requirement for the specification of priors on the parameters.

The Monte Carlo experiment with systematic generation of replicate data sets was used to evaluate IPL-GA's capability to recuperate individual preference parameters, for both uni- and bi-modal uniform preferences. This experiment demonstrated that the IPL-

[5] I direct the reader to the archive at https://arnold-neumaier.at/glopt.html. It purports to cover all things global optimization, and contains references, free code, people to follow, etc.

GA is capable of reliably recuperating preferences, given sufficient choice replications R commensurate with the number of preference dimensions K. (This statement is also true of other estimation methods, of course.) The method is quite effective when operating in conditions equivalent to common Discrete Choice Experiment sizes: $K = 8$–16 dimensions and $R = 8$–16 replications. Naturally, more replications are always beneficial and preferable, particularly as K increases beyond this range; this may not, however, be empirically practical to implement during data collection (though see the scanner panel data example in the prior section of the paper).

In the several applications for which I have used IPL-GA, beyond what is presented in this paper, the second (optimization) phase of the estimator is always helpful in identifying a better solution than obtained by the first (evolutionary) phase. That said, the improvements can be relatively modest (even small) in terms of posterior log likelihood value, depending on the tuning parameters for the GA. But at times the improvement has been substantial. The availability of a maximal posterior log likelihood solution facilitates inferences since maximum likelihood theory applies for hypothesis testing purposes, so the extra effort is worth expending from that perspective. For predictions, the case is not so clear cut for full-blown optimization.

Beyond forecasting within the limitations I discussed earlier, having individual parameter estimates opens up the possibility of reviving an ancient custom of conjoint practitioners, which was the application of clustering techniques to preference estimates to obtain insights into preference regularities and segments in the sample. A related technique is archetypal analysis (Cutler and Breiman 1994), which identifies archetypes (or pure types) that form the basis for characterizing all other decision makers. From a market management and product development perspective, this is rich knowledge indeed, since the clusters and archetypes may form a strong basis for firm decision making and action.

One of the maintained conditions in this research is that individual parameters are considered constant across a set of replications. This restriction can of course be removed, as suggested by Swait et al. (2016) for latent class (mixture) models, and more recently, by Kreuger et al. (2021) for the Mixed Logit model. Both allow for intra-person variation in preference parameters, over and above inter-personal preference heterogeneity. Both employ maximum likelihood methods to estimate parameters, the latter with simulation. Krueger et al. cite other literature that explores alternative approaches to this broader problem using neural networks, regression trees, and so forth. Extensions to models that comport both intra- and inter-personal preference heterogeneity can be explored in future through the combination of evolutionary and optimization methods.

As noted earlier in the manuscript, I implemented the two-stage algorithm described here in Fortran95, compiled with gfortran (a freely available compiler for Windows and Linux, see https://gcc.gnu.org/wiki/GFortranBinaries). This choice was based on prior familiarity with the language and the desire for high computational speed, particularly through the availability of parallel computation capabilities (in this case, through use of the OpenMP library – see www.OpenMP.org – with which gfortran is well integrated). However, Fortran does not have a monopoly on computational speed: C, C++, Gauss, and Julia, among others, are all possible development environments for algorithms such as described here. I originally prototyped the GA described here in Gauss, which has an integrated parallel computation capability, but eventually thought it worthwhile to transfer to Fortran for faster execution to handle larger data sets and multiple starting points. In practice it is my expectation that medium-to large-sized choice modelling problems using evolutionary algorithms will require implementations in languages such as Fortran, C and C++, simply to achieve reasonable execution times.

With respect to tapping into available evolutionary algorithm implementations, some simple web searching yields at least three "black box" options: MATLAB has a toolbox of evolutionary algorithms which might be adaptable for this purpose, as does the julia language. R users should examine package rgenoud, which implements genetic search with a FOM optimizer using derivative information (Mebane and Sekhon 2011). I cite these as examples, but I have not used any of them myself or evaluated their suitability for inclusion in a parameter estimation algorithm with high dimensionality. In my view, there is no shortage of evolutionary algorithm ideations and implementations in the literature, so adoption of these evolutionary first stages in estimation algorithms should not be overly difficult. But, as ever, *caveat emptor*!

In bringing this paper to a close, I should note that I have also implemented both Differential Evolution (DE) and Particle Swarm (PS) evolutionary algorithms for the problem to hand. They have both performed similarly to the GA on the data sets I have tried them on, but my initial thought is that they seem better suited to work with a solution strategy seeking an explicit optimal solution (like SOLUTION=BEST) rather than a summary solution strategy (like SOLUTION = AVERAGE). My overall impression is that the GA works better for choice data than these other evolutionary exemplars, but this opinion is based on selective and idiosyncratic experience that has emphasized GA over the others, so it should be taken with a grain of salt. Each of these algorithms has parameters which need to be "tuned" to the data, so any of them might perform better in a given setting than the others.

Author contributions

Single-authored paper, Joffre Swait responsible for all aspects of paper development.

Declaration of competing interest

None.

Data availability

The authors do not have permission to share data.

References

Banzhaf, Wolfgang, Nordin, Peter, Keller, Robert, Frank, Francone, 1998. Genetic Programming – an Introduction. Morgan Kaufmann, San Francisco, CA.

Ben-Akiva, Moshe, Lerman, Steven, 1985. Discrete Choice Analysis: Theory and Application to Travel Demand. The MIT Press, Cambridge, MA.

Berndt, E., Hall, B., Hall, R., Hausman, J., 1974. Estimation and inference in nonlinear structural models. Ann. Econ. Soc. Meas. 3 (4), 653–665.

Brownstone, David, Bunch, David, Train, Kenneth, 2000. Joint mixed logit models of stated and revealed preferences for alternative-fuel vehicles. Transp. Res. Part B Methodol. 34 (5), 315–338.

Brownstone, David, Train, Kenneth, 1999. Forecasting new product penetration with flexible substitution patterns. J. Econom. 89 (1–2), 109–129.

Chaterjee, Sangit, Laudato, Matthew, Lynch, Lucy, 1996. Genetic algorithms and their statistical applications: an introduction. Comput. Stat. Data Anal. 22, 633–651.

Cutler, Adele, Breiman, Leo, 1994. Archetypal analysis. Technometrics 36 (4), 338–347.

Daly, Andrew, Hess, Stephane, Train, Kenneth, 2012. Assuring finite moments for Willingness to Pay in random coefficient models. Transportation 9, 19–31. https://doi.org/10.1007/s11116-011-9331-3.

de Jong, K.A., 1975. An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Unpublished PhD Dissertation. University of Michigan, Department of Computer Science.

Dennis, J.E., Schnabel, Robert, 1983. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Englewood Cliffs, NJ.

Dorsey, Robert, Mayer, Walter, 1995. Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features. J. Bus. Econ. Stat. 13 (1), 53–66.

Drake, Adrian, Marks, Robert, 2002. Genetic algorithms in economics and finance: forecasting stock market prices and Foreign exchange — a review. In: Chen, S.H. (Ed.), Genetic Algorithms and Genetic Programming in Computational Finance. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-0835-9_2.

Efron, Bradley, Tibshirani, Robert, 1993. An Introduction to the Bootstrap. In: Monographs on Statistics and Applied Probability, vol. 57. Chapman & Hall, New York.

Fader, Peter, Hardie, Bruce, 1996. Modeling consumer choice among SKUs. J. Market. Res. 33 (November), 442–452.

Fogel, David, 2006. Evolutionary Computation: toward a New Philosophy of Machine Intelligence, third ed. IEEE Press, Piscataway, NJ.

Frischknecht, Bart D., Eckert, Christine, Geweke, John, Louviere, Jordan J., 2014. A simple method for estimating preference parameters for individuals. Int. J. Res. Market. 31, 35–48. https://doi.org/10.1016/j.ijresmar.2013.07.005.

Gill, Philip, Murray, Walter, Wright, Margaret, 1981. Practical Optimization. Academic Press, New York.

Gilli, Manfred, Winker, Peter, 2008. A review of heuristic optimization methods in econometrics, research paper series 08-12. Swiss Finance Institute.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Wesley, Reading, MA.

Haghani, Milad, Bliemer, Michiel C.J., Hensher, David A., 2021. The landscape of econometric discrete choice modelling research. J. Choice Modelling. https://doi.org/10.1016/j.jocm.2021.100303.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor.

Holloman, Christopher, Lee, Herbert, Dave, Higdon, 2006. Multiresolution genetic algorithms and Markov chain Monte Carlo. J. Comput. Graph Stat. 15 (4), 861–879. https://doi.org/10.1198/106186006X157423.

Johnson, Eric J., Meyer, Robert J., 1984. Compensatory choice models of noncompensatory Processes: the effect of varying context. J. Consum. Res. 11 (1), 528–541.

Kim, Dong-Sun, Lee, Sang-Seol, 2012. Improved mutation method for providing high genetic diversity of genetic algorithm processor. IEICE Electron. Express 9 (9), 822–827. https://doi.org/10.1587/elex.9.822.

Kreuger, Rico, Bierlaire, Michel, Daziano, Ricardo, Taha, Rashidi, Bansal, Prateek, 2021. Evaluating the predictive abilities of mixed logit models with unobserved inter- and intra-individual heterogeneity. J. Choice Modelling 41. https://doi.org/10.1016/j.jocm.2021.100323.

McFadden, Daniel, 2022. Instability in mixed logit demand models. J. Choice Modelling 43, 100353. https://doi.org/10.1016/j.jocm.2022.100353.

McFadden, Daniel, Train, Kenneth, 2000. Mixed MNL models for discrete response. J. Appl. Econom. 15, 447–470.

Mebane Jr., W.R., Sekhon, J.S., 2011. Genetic optimization using derivatives: the rgenoud Package for R. J. Stat. Software 42 (11), 1–26. https://www.jstatsoft.org/v42/i11/.

Nelder, John, Mead, R., 1965. A simplex method for function minimization. Comput. J. 7 (4), 308–313. https://doi.org/10.1093/comjnl/7.4.308.

Padmanabhan, Balaji, Barfar, Arash, 2021. Learning individual preferences from aggregate data: a genetic algorithm for discovering baskets of television shows with affinities to political and social interests. Expert Syst. Appl. 168, 114184.

Revelt, David, Train, Kenneth, 1998. Mixed logit with repeated choices: households' choices of appliance efficiency level. Rev. Econ. Stat. 80 (4), 647–657.

Schmitt, Lothar, 2001. Theory of genetic algorithms. Theor. Comput. Sci. 259 (1–2), 1–61. https://doi.org/10.1016/S0304-3975(00)00406-0.

Soekhai, Vikas, de Bekker-Grob, Esther W., Ellis, Alan, Vass, Caroline, 2019. Discrete choice experiments in health economics: past, present and future. Pharmacoeconomics 37, 201–226. https://doi.org/10.1007/s40273-018-0734-2.

Spall, James, 2005. Monte Carlo computation of the Fisher information matrix in nonstandard settings. J. Comput. Graph Stat. 14 (4), 889–909. https://doi.org/10.1198/106186005X78800.

Storn, Rainer, Price, Kenneth, 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11, 341–359.

Strens, Malcolm, 2003. Evolutionary MCMC sampling and optimization in discrete spaces. In: Proceedings of the Twentieth International Conference on Machine Learning. ICML-2003), Washington DC.

Swait, Joffre, 2009. Choice models based on mixed discrete/continuous PDFs. Transport. Res. Part B 43, 766–783.

Swait, Joffre, Adamowicz, Wiktor, 2001a. The influence of task complexity on consumer choice: a latent class model of decision strategy switching. J. Consum. Res. 28, 135–148. June.

Swait, Joffre, Adamowicz, Wiktor, 2001b. Incorporating the effect of choice environment and complexity into random utility models. Organ. Behav. Hum. Decis. Process. 86 (2), 141–167.

Swait, Joffre, Erdem, Tulin, 2002. The effects of temporal consistency of sales promotions and availability on consumer choice behavior. J. Market. Res. 39, 304–320.

Swait, Joffre, Popa, Monica, Wang, Luming, 2016. Capturing context-sensitive information usage in choice models via mixtures of information archetypes. J. Market. Res. 53 (5), 646–664. https://doi.org/10.1509/jmr.12.0518.

ter Braak, Cajo, J.F., 2004. Genetic Algorithms and Markov Chain Monte Carlo: Differential Evolution Markov Chain Makes Bayesian Computing Easy. Report from Centre for Biometry of Plant Research International and the Department of Mathematical and Statistical Methods, Wageningen University, The Netherlands.

Train, Kenneth, 1998. Recreation demand models with taste differences over people. Land Econ. 74 (2), 230–239.

Train, Kenneth, 2009. Discrete Choice Methods with Simulation, second ed. Cambridge University Press, Cambridge, UK.

Train, Kenneth, 2016. Mixed logit with a flexible mixing distribution. J. Choice Modelling 19, 40–53.