Generating Random Optimising Choices

Jan Heufer*

25.06.2013

forthcoming in *Computational Economics*

Abstract

We provide an efficient way to generate random choices which are consistent with utility maximisation. They are drawn from an approximate uniform distribution on the admissible region on each budget based on a Markovian Monte Carlo algorithm due to Smith (1984). This can be used to extend Bronars' (1987) method by approximating the power of tests for conditions for which utility maximisation is necessary but not sufficient (e.g., homotheticity, separability, etc.). The approach can also be applied to production analysis.

*Jan Heufer, TU Dortmund, Wirtschafts- und Sozialwissenschaftliche Fakultät, Volkswirtschaftslehre (Mikroökonomie), D-44221 Dortmund, Germany. Tel.: +49 231 755 7223; fax: +49 231 755 3027. E-mail: jan.heufer@tu-dortmund.de. Thanks to Wolfgang Leininger for his support and comments on earlier drafts of the manuscript. Thanks to an anonymous referee for the helpful comments and suggestions.

1 Introduction

Afriat's (1967) Theorem shows that for a set of consumption choices from competitive budgets there exists a continuous, monotonic, and concave utility function if and only if the choices satisfy cyclic consistency. Varian (1982) introduced the *Generalised Axiom of Revealed Preference* (Garp) which is easy to test and equivalent to cyclic consistency.

It is important to know how meaningful this test for utility maximisation is. Therefore, it is helpful to know the probability that random choices violate Garp. This probability can be interpreted as the *power* of the test for utility maximisation against the alternative hypothesis of random choices. Bronars (1987) suggested a Monte Carlo approach to approximate the power: Generate many sets of random choices and tests all of them for Garp. The percentage of sets which violators is the approximate power.

Suppose that the researcher is not only interested in testing Garp but also in testing additional assumption for which Garp is a necessary but not sufficient condition. Such assumptions include homotheticity and different forms of separability (cf. Varian 1983), risk aversion (cf. Heufer 2011), and others. If for example a researcher is interested in testing a set of observations for consistency with homotheticity, using Varian's (1983) *Homothetic Axiom of Revealed Preference* (Harp), then to approximate the power she can use a variant of Bronars' Power by testing random choices for consistency with Harp. But Garp is a necessary condition for Harp, and Bronars' Power for the Garp test might already be close to unity. That means that very few random choice sets satisfy Garp, and therefore cannot satisfy Harp. It would be useful to know the *conditional* probability that a set of random choices violates Harp given that it satisfies Garp.

A simple way to approximate this conditional probability is to use a rejection technique: Draw a set of random choices and reject it if it does not satisfy Garp. However, Bronars' power can be quite literally 100%, that is, even out of thousands of sets of random choices no set satisfies Garp. This is the case in the fifty budget experiments conducted by Fisman et al. (2007) and Choi et al. (2007).

The contribution of this paper is to introduce a Monte Carlo approach to generate sets of random choices which satisfy Garp, using a Markovian method introduced by Smith (1984). He showed that a symmetric mixing algorithm which generates a Markov chain on a bounded region will generate a sequence of points asymptotically uniformly distributed within the region. This algorithm is used to generate sets of random choices on budgets which are uniformly distributed such that the sets satisfy Garp. These Garp-consistent random sets can then be tested for consistency with other conditions such as Harp. The algorithms presented here can also be applied to nonparametric approaches to production analysis (cf. Varian 1984).

2 Preliminaries

2.1 General Definitions and Concepts

A set of observed consumption choices consists of a set of chosen bundles of commodities and the prices and incomes at which these bundles were chosen. Let $\mathbb{R}^L_+$ be the commodity space, where $L \geq 2$ denotes the number of different commodities.[1] The price space is $\mathbb{R}^L_{++}$. Consumers choose bundles $x^i = (x^i_1, \ldots, x^i_L)' \in \mathbb{R}^L_+$ when facing a price vector $p^i = (p^i_1, \ldots, p^i_\ell) \in \mathbb{R}^L_{++}$; a budget is then defined by $B^i = B(p^i) = \{x \in \mathbb{R}^L_+ : p^i x^i \leq 1\}$, and the boundary is $\bar{B}(p^i) = \{x \in \mathbb{R}^L_+ : p^i x^i = 1\}$. That is, prices are normalised such that expenditure always equals 1; we can therefore also identify budgets with their characteristic price vector. The entire set of N observations on a consumer is denoted as $\Omega = \{(x^i, p^i)\}^N_{i=1}$.[2]

An observation x^i is *directly revealed preferred* to x, written $x^i \, \mathrm{R}^0 \, x$, if $p^i x^i \geq p^i x$; it is *revealed preferred* to x if $x^i \, \mathrm{R} \, x$, where R is the transitive closure of R^0; it is *strictly directly revealed preferred* to x, written $x^i \, \mathrm{P}^0 \, x$, if $p^i x^i > p^i x$.

A utility function $u : \mathbb{R}^L_+ \to \mathbb{R}$ *rationalises* Ω if $u(x^i) \geq u(x)$ whenever $x^i \, \mathrm{R} \, x$. The set Ω satisfies the *Generalised Axiom of Revealed Preference* (Garp) if $x^i \, \mathrm{R} \, x^j$ implies $\big[\text{not } x^j \, \mathrm{P}^0 \, x^i\big]$. It can then be shown (Afriat 1967; Diewert 1973; Varian 1982) that there exists a continuous, monotonic, and concave utility function that rationalises Ω if and only if Ω satisfies Garp.

2.2 Supporting Bundles and Forecasting Choices

Following Varian (1982), we define the set of bundles which *support* a price vector p^0 not previously observed as

$$S(p^0 | \{(x^i, p^i)\}^N_{i=1}) = \{x^0 \in \mathbb{R}^L_+ : \{(x^i, p^i)\}^N_{i=0} \text{ satisfies Garp and } p^0 x^0 = 1\}. \tag{1}$$

That is, $S(p^0|\Omega)$ is the set of all bundles which can be chosen on $B(p^0)$ without violating Garp when combined with the previous observations. Any utility maximising consumer for whom we have observed Ω will choose a bundle in $S(p^0)$ when facing the new budget $B(p^0)$. See Figure 1 for an illustration.

The set $S(p^0|\Omega)$ is important as it will be the set from which we want to draw a random element x^0, such that (x^0, p^0) is an observation that is consistent with Ω (i.e., $\Omega \cup \{(x^0, p^0)\}$ satisfies Garp). Varian (1982) shows that the set $S(p^0|\Omega)$ is characterised by a system of linear inequalities. To set up this system, we need *indirect* preferences on budgets (cf. Sakai 1977, Little 1979). We use the definitions and terminology introduced in Varian (1982), but it should be emphasised that the following relations are indeed indirect preferences. A

[1] The following notation is used: For all $x, y \in \mathbb{R}^L$ we write $x \geq y$ for $x_i \geq y_i$ for all i, $x > y$ for $x_i \geq y_i$ and $x \neq y$ for all i, and $x \gg y$ for $x_i > y_i$ for all i. We denote $\mathbb{R}^L_+ = \{x \in \mathbb{R}^L : x \geq 0\}$ and $\mathbb{R}^L_{++} = \{x \in \mathbb{R}^L : x \gg 0\}$.

[2] Strictly speaking, we observe budgets as a pair $(q^i, w^i) \in \mathbb{R}^L_{++} \times \mathbb{R}_{++}$ and then set $p^i = q^i / w^i$. The implicit assumption here is that demand is homogeneous.

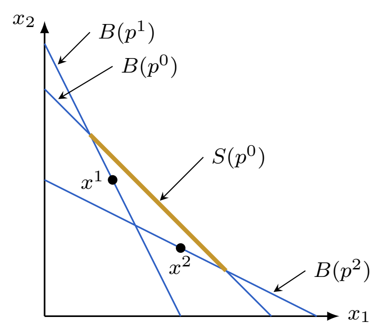


Figure 1: The set $S(p^0)$.

budget $B(p)$ is *directly revealed preferred* to a budget $B(p^i)$, written $p\,\mathrm{R}_B^0\,p^i$ if $px^i \leq 1$, where x^i is the observed choice on $B(p^i)$; it is *revealed preferred* to $B(p^i)$ if $p\,\mathrm{R}_B\,p^i$, where R_B is the transitive closure of R_B^0; it is *strictly directly revealed preferred* to $B(p^i)$ if $p\,\mathrm{P}_B^0\,p^i$ if $px^i < 1$; it is *strictly revealed preferred* to $B(p^i)$, written $p\,\mathrm{P}_B\,p^i$ if there exist observed budgets $B(p^j)$ and $B(p^k)$ such that $p\,\mathrm{R}_B\,p^j$, $p^j\,\mathrm{P}_B\,p^k$, and $p^k\,\mathrm{R}_B\,p^i$.

Fact 1 *A bundle x^0 is in $S(p^0|\Omega)$ if and only if it satisfies the conditions*
(1) $p^0x^0 = 1$,
(2) $p^ix^i \leq p^ix^0$ *for all* p^i *such that* $p^0\,\mathrm{R}_B\,p^i$,
(3) $p^ix^i < p^ix^0$ *for all* p^i *such that* $p^0\,\mathrm{P}_B\,p^i$.

Fact 1 can be found in Varian (1982, Fact 8).

We also need an open subset $T(p^0|\Omega)$ of $S(p^0|\Omega)$, defined as the set of all $x^0 \in \mathbb{R}^L_{++}$ such that
(1) $x^0 \in S(p^0|\Omega)$,
(2) $p^ix^i < p^ix^0$ for all p^i such that $p^0\,\mathrm{R}_B\,p^i$ and $p^0 \neq p^i$.
We need the set T for technical reasons explained in the next section. It is straightforward to show that the set $T(p^0|\Omega)$ is the relative interior of $S(p^0|\Omega)$; that is, the interior of $S(p^0|\Omega)$ within the subspace defined by the budget hyperplane $\bar{B}(p^0)$.[3] The following fact is then straightforward.

Fact 2 *A bundle $x^0 \in \mathbb{R}^L_+$ is in $T(p^0|\Omega)$ if and only if it satisfies the conditions*
(1) $x^0_j > 0$ *for all* $j = 1, \ldots, L$,
(2) $p^0x^0 = 1$,
(3) $p^ix^i < p^ix^0$ *for all* p^i *such that* $p^0\,\mathrm{R}_B\,p^i$ *and* $p^0 \neq p^i$.

Thus, when trying to find a point on T, we can pick any x^0 which satisfies all the linear inequalities in Fact 2.

[3] More precisely, T is the interior of S within the affine hull of S. As the set $S(p^0|\Omega)$ is completely contained in the hyperplane $\bar{B}(p^0)$, $\bar{B}(p^0)$ is the affine hull of S in $\mathbb{R}^L_+$.

3 Algorithms

3.1 Preliminary Algorithms

Algorithm 1, the *simplex point picking algorithm*, returns a random point uniformly distributed on the unit simplex (see for example Tempo et al. 2005, p. 245).

Algorithm 1 *Input: An integer $L \geq 2$.*
Output: A random point X uniformly distributed on the $L-1$ unit simplex.

1. *Generate L independent random variables $Y = (Y_1, \ldots, Y_L)$ from the Gamma distribution with parameters $\alpha = \beta = 1$.*
2. *Set $X = Y/(\sum_{i=1}^{L} Y_i)$ and return X.*

The simplex point picking algorithm can then be used to generate a random choice on a budget from a uniform distribution. Algorithm 2 does this for each budget in a set of N budgets; it can be used to compute Bronars' power.

Algorithm 2 *Input: A set of N normalised price vectors $\{p^i\}_{i=1}^N$.*
Output: A set of random choices on each $B(p^i)$ uniformly distributed on $\bar{B}(p^i)$.

1. *Set $k = 1$.*
2. *Generate a point X on the $L-1$ simplex using Algorithm 1. Set $x^k = (X_1/p_1^k, \ldots, X_L/p_L^k)$. Set $k = k+1$.*
3. *If $k = N$, stop and return $\{x^i\}_{i=1}^N$. Otherwise, go to Step 2.*

For any set $\{p^i\}_{i=1}^N$, we can execute Algorithm 2 many times and test all the generated choice sets for GARP. Bronars' power is then the percentage of choice sets which do not satisfy GARP.

Let $D = \{d \in \mathbb{R}^L : \|d\| = 1\}$ be the unit sphere. Selecting a random element $d \in D$ from a uniform distribution on D is equivalent to selecting a random direction in $\mathbb{R}^L$. Algorithm 3, the *random direction algorithm*, generates such a random direction (see for example Knuth 1998 [1969], p. 135).

Algorithm 3 *Input: An integer $L \geq 2$.*
Output: A random point $d \in D$ uniformly distributed on D.

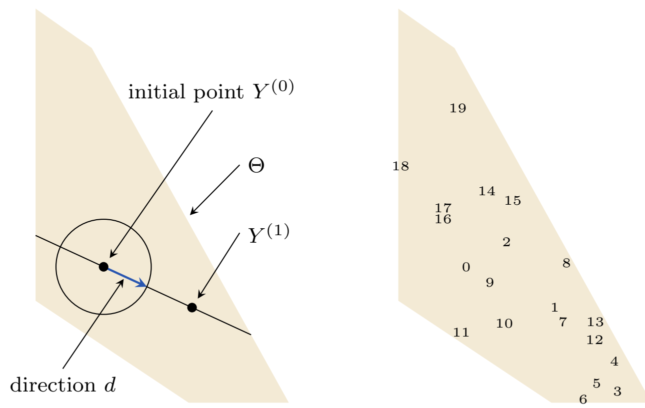
1. *Generate L independent normally distributed random variables, $\delta = (\delta_1, \ldots, \delta_L)$.*
2. *Set $d = \delta / \|\delta\|$ and return d.*

Algorithm 4, the *mixing algorithm*, can be found in Smith (1984).

Algorithm 4 *Input: An integer $M \geq 1$ and a set $\Theta \subseteq \mathbb{R}^L$ with $L \geq 2$.*
Output: A Markov chain of points $Y^{(0)}, \ldots, Y^{(M)}$ in Θ.

1. *Set $k = 0$. Choose an initial point $Y^{(0)} \in \Theta$.*
2. *Generate a random direction $d \in \mathbb{R}^L$ using Algorithm 3.*
3. *Set $\mathcal{L} = \Theta \cap \{x \in \mathbb{R}^L : x = Y^{(k)} + \lambda d\}$, where $\lambda \in \mathbb{R}$.*
4. *Generate a random point $Y^{(k+1)}$ uniformly distributed on $\mathcal{L}$.*
5. *If $k = M$, stop and return $Y^{(0)}, \ldots, Y^{(M)}$. Otherwise, set $k = k+1$ and go to Step 2.*

Algorithm 4 generates a Markov chain of points in Θ. Smith (1984) showed that under certain conditions the generated points approach a uniform distribution over the region for any starting point $Y^{(0)}$. In particular, the assumptions are satisfied if Θ is an open and bounded subset of $\mathbb{R}^L$ with $L \geq 2$ and Θ is itself L-dimensional (i.e., the affine hull of Θ is of dimension L). See Figure 2 for an illustration of the algorithm.



(a) The first step of Algorithm 4. (b) Markov chain generated by Algorithm 4.

Figure 2: An illustration of the first step of Algorithm 4 and the first twenty points generated by it.

In Section 3.3, we consider commodity spaces with $L \geq 3$, and we want to draw a random point on the set T from an approximate uniform distribution. This is accomplished with Algorithm 6 below. In this case, the set T is of dimension $L - 1 \geq 2$ (more precisely, the dimension of the affine hull of T is $L - 1$), which is why we need Algorithm 4 as part of Algorithm 6.

We do not need Algorithm 4 for the two-dimensional case considered in Section 3.2. The reason is that the set T is only a line segment (see Figure 1) and therefore of dimension 1. For a line segment, it is always sufficient to draw a point from a uniform distribution on a 1-simplex with Algorithm 1 and translate the point accordingly. Therefore, no Markovian algorithm is needed, and the task can be easily accomplished with Algorithm 5.

3.2 The Two-Dimensional Case

The method based on the mixing algorithm only works for commodity spaces with $L \geq 3$, as in the two-dimensional case the set $S(p^0)$ is only a line segment, and we do not need Algorithm 4. As many induced budget experiments only consider two goods, the two-dimensional case is quite relevant, which is why we treat it separately here. Algorithm 5 generates a set of choices, $\{x^i\}_{i=1}^N$, on a set of budgets with price vectors $\{p^i\}_{i=1}^N$, such that $\{(x^i, p^i)\}_{i=1}^N$ satisfies GARP.

Algorithm 5 *Input: A set of $N \geq 2$ normalised price vectors $\{p^i\}_{i=1}^N$ with $p^i \in \mathbb{R}^2_{++}$ for $i = 1, \ldots, N$.*
Output: A set of choices $\{x^i\}_{i=1}^N$ drawn from a uniform distribution on $\{\bar{B}(p^i)\}_{i=1}^N$ such that $\{(x^i, p^i)\}_{i=1}^N$ satisfies GARP.

1. *Set $k = 1$. Generate the first choice x^k on $B(p^k)$ using Algorithm 2.*
2. *Set*

$$\begin{aligned}
\tilde{x}^{\min} &= \arg\min_{x^0 \in S\left(p^{k+1} | \{(x^i, p^i)\}_{i=1}^k\right)} x_2^0 \\
\tilde{x}^{\max} &= \arg\max_{x^0 \in S\left(p^{k+1} | \{(x^i, p^i)\}_{i=1}^k\right)} x_2^0 \\
\phi &= (\tilde{x}_1^{\max}, \tilde{x}_2^{\min}) \\
\tilde{p} &= ([\tilde{x}^{\min} - \phi_1]^{-1}, [\tilde{x}^{\max} - \phi_2]^{-1})
\end{aligned}$$

3. *Set $k = k + 1$. Generate a point X on the $L - 1$ simplex using Algorithm 1. Set $x^k = (X_1/\tilde{p}_1, \ldots, X_L/\tilde{p}_L) + \phi$.*
4. *If $k = N$, stop and return $\{x^i\}_{i=1}^N$. Otherwise, go to Step 2.*

We omit the proof that Algorithm 5 generates a set of GARP-consistent observations, as it is rather straightforward. Step 2 computes ϕ, which is used as a new "origin"; then $\tilde{p}$ describes a new budget which if translated by ϕ equals $S(p^{k+1} | \{(x^i, p^i)\}_{i=1}^k)$. Step 3 generates a random choice on $\tilde{p}$ and then translates it to obtain a choice on $S(p^{k+1} | \{(x^i, p^i)\}_{i=1}^k)$. Most importantly, the algorithm does not simply compute a set of GARP-consistent choices, but does so by drawing random choices from a uniform distribution over the admissible region given by the set of supporting bundles, conditional on the previously generated choices. Algorithm 5 can be executed many times, using random permutations of the budgets such that each budget is equally likely to be the kth one. This will provide many sets of random choices which satisfy GARP and can then be tested for additional assumptions.

3.3 The Higher-Dimensional Case

We will now present an algorithm which accomplishes the same as Algorithm 5 for the higher dimensional case. The set $S(p^0|\Omega)$ (and therefore T) for the commodity space $\mathbb{R}^L_+$ is a subset of the budget hyperplane $\bar{B}(p^0)$ and thus only of dimension $L - 1$. Therefore, we cannot directly use T as an input for Algorithm 4. Instead, we use a slightly different way to compute the line on which a random point is chosen; this is done in Algorithm 6, which is explained below. In particular, Steps 3 to 6 are adopted from Algorithm 4 and slightly modified. Equivalently, one could embed the set T in the space $\mathbb{R}^{L-1}$, use this as input for Algorithm 4, and then embed the point returned by Algorithm 4 back into $\mathbb{R}^L$.

Algorithm 6 *Input: An integer $M \geq 1$ and set of $N \geq 2$ normalised price vectors $\{p^i\}_{i=1}^N$ with $p^i \in \mathbb{R}^L_{++}$ with $L \geq 3$ for $i = 1, \ldots, N$.*

Output: A set of choices $\{x^i\}_{i=1}^N$ *drawn from a uniform distribution on* $\{\bar{B}(p^i)\}_{i=1}^N$ *such that* $\{(x^i, p^i)\}_{i=1}^N$ *satisfies* Garp.

1. *Set* $k = 1$. *Generate the first choice* x^k *on* $B(p^k)$ *using Algorithm 2.*
2. *Set* $k = k + 1$. *Set* $\mathcal{T} = T(p^k | \{(x^i, p^i)\}_{i=1}^{k-1})$.
3. *Set* $\ell = 0$. *Choose an initial point* $Y^{(\ell)} \in \mathcal{T}$.
4. *Set* $\ell = \ell + 1$. *Generate a random direction* $d \in \mathbb{R}^{L-1}$ *using Algorithm 3. Set*

   $$\mathcal{T}' = \{x \in \mathbb{R}^L_{++} : x_j = Y_j^{(\ell)} + \lambda d_j \text{ for } j = 1, \ldots, L-1 \text{ and } (x_1, \ldots, x_L) \in \bar{B}(p^k)\},$$

   where $\lambda \in \mathbb{R}$. *Set* $\mathcal{L} = \mathcal{T} \cap \mathcal{T}'$.
5. *Generate a random point* $Y^{(\ell)}$ *uniformly distributed on* $\mathcal{L}$.
6. *If* $\ell = M$, *set* $x^k = Y^{(\ell)}$ *and go to Step 7. Otherwise, go to Step 4.*
7. *If* $k = N$, *stop and return* $\{x^i\}_{i=1}^N$. *Otherwise go to Step 2.*

The first three steps are straightforward. Note that we use the set T instead of S. The set S is closed (within the budget hyperplane), whereas T is open; thus, using T assures that Smith's (1984) conditions are satisfied. This is not a relevant limitation as the probability of drawing a point on the boundary is zero.

The first time Step 4 is reached, it computes a line $\mathcal{T}$ in the budget hyperplane $\bar{B}(p^2)$ through the initial point $Y^{(0)}$ using a random direction in $\mathbb{R}^{L-1}$. Given λ and d and the fact that $\mathcal{T}' \subset \bar{B}(p^2)$, the value of x_L is unique. Step 5 then generates a point uniformly distributed on the intersection of the line $\mathcal{T}'$ with the admissible set $\mathcal{T}$ such that the generated points can be used as choices which satisfy Garp. Algorithm 6 therefore generates a Markov chain in the $L-1$ dimensional subspace defined by the budget hyperplanes.

Generating a random point on $\mathcal{L}$ in Step 5 is particularly easy because S and T are convex sets; this follows from Fact 1. Thus, we can simply compute the minimal and maximal λ such that $\mathcal{L} \neq \emptyset$ and then draw λ from a uniform distribution on that interval. More details on the practical implementation can be found in Section 3.4.

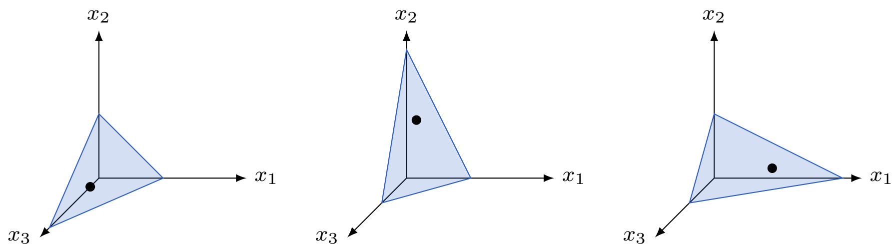
Again, we omit the full proof that the set generated by the algorithm satisfies Garp, as it is straightforward. See Figure 3 for an illustration. Note that the points generated with $M = 15$ already appear to be uniformly distributed on T; there is little difference to the distribution of points generated with $M = 40$.

3.4 Practical Implementation of the Higher Dimensional Case

An implementation of the method with examples is available from the author upon request. The algorithms have been implemented in Wolfram Mathematica® 8. Parts of the Mathematica code are implementations of standard revealed preference computations (see, for example, Varian 1996). This section gives a brief summary of the implementation of Algorithm 6.

Step 1 of Algorithm 6 is straightforward.

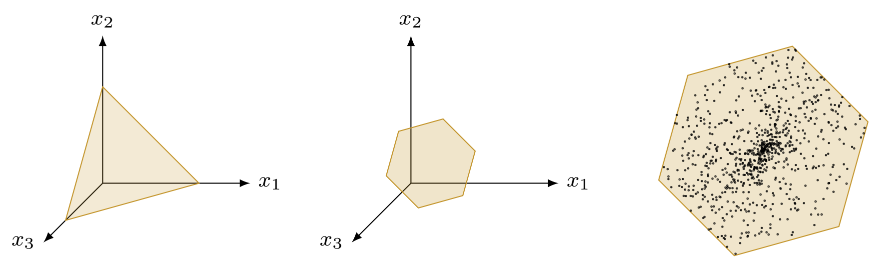
For step 2, we need a characterisation of the set $T(p^k | \{(x^i, p^i)\}_{i=1}^{k-1})$ to be used for computations. We first need to find all budgets to which p^k is revealed



(a) Budget hyperplane $\bar{B}(p^1)$ and choice x^1.

(b) Budget hyperplane $\bar{B}(p^2)$ and choice x^2.

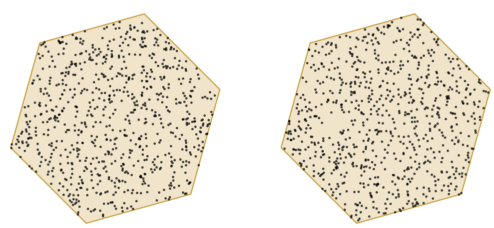
(c) Budget hyperplane $\bar{B}(p^3)$ and choice x^3.



(d) Budget hyperlane $\bar{B}(p^4)$.

(e) The set $S(p^4|\{(x^i, p^i)\}_{i=1}^3$ and its relative interior.

(f) 800 points with $M = 2$.



(g) 800 points with $M = 15$.

(h) 800 points with $M = 40$.

Figure 3: The first three budgets and the choices are shown in (a) - (c). (d) shows the fourth budget. (e) shows the admissible region on the fourth budget given the three previous choices. (f) - (h) show 800 points generated using parts of Algorithm 6, with different M.

preferred. These are all p^i, $i < k$, such that $p^k\, \mathrm{R}_B^0\, p^i$, and then all p^j, $j < k$, such that $p^i\, \mathrm{R}_B\, p^j$, where R_B is based on the set $\{(x^i, p^i)\}_{i=1}^{k-1}$. The set of inequalities in Fact 2 characterises the set $\mathcal{T}$. These inequalities need to be stored to be used as constrains in a linear programming problem. In Mathematica, one can simply set a variable to the value `p[[k]].x0 == 1 && p[[1]].x[[1]] < p[[1]].x0 && ...`.

For step 3, we have to find a point which satisfies the inequalities which characterise $\mathcal{T}$. In the Mathematica package, the function `FindInstance` is used.

For step 4, we use Algorithm 3 to draw a random direction d in $L - 1$ space. The set $\mathcal{T}'$ is characterised by a set of equations, which again need to be stored. The set $\mathcal{L}$ is characterised by the union of the inequalities for the set $\mathcal{T}$ and the equations for the set $\mathcal{T}'$.

For step 5, we need to find the minimal and maximal value of λ, that is, the interval for λ such that the equations and inequalities for $\mathcal{L}$ hold. Again, this is a linear programming problem. In Mathematica, the functions `NMinimize` and `NMaximize` are used. We then draw a random λ from a uniform distribution over the interval found in step 5. We can then compute $x_j = Y_j^{(0)} + \lambda d_j$ for $j = 1, \ldots, L-1$, and solving $x \in \bar{B}(p^k) \Leftrightarrow p^k x = 1$ for x_L completes finding the point $Y^{(1)} = (x_1, \ldots, x_L)$.

Steps 4 to 6 are then repeated until point $Y^{(M)}$ is found, which is set as the value for the random choice x^k on p^k. Steps 2 to 6 are then repeated until all x^i, $i = 1, \ldots, N$, are found.

The computational complexity of the method is driven by the number of constrained in the optimisation problems in steps 3 and 5. The set $\mathcal{L}$ in step 5 will consist of $N + 2L$ constraints at most, so the method is computable in polynomial time. Algorithm 6 has been tried with randomly generated price vectors for $N = 50$ budgets in $L = 20$ dimensions without problems.

4 Generalisations and Extensions

A possible generalisation is to generate sets of Garp-consistent choices and compute efficiency indices for the additional condition, such as homothetic efficiency (cf. Heufer 2013) or stochastic-dominance efficiency (cf. Heufer 2011). The approach can be further extended by generating choice sets which satisfy other conditions, such as Harp, and then to approximate the conditional probability that it also satisfies another condition, such as separability. The procedure can also be applied to testing assumptions on cost functions when we observe input and output data instead of consumption choices.

References

Afriat, S. N. (1967): The Construction of Utility Functions From Expenditure Data, *International Economic Review*, 8(1), 67-77.

Bronars, S. G. (1987): The Power of Nonparametric Tests of Preference Maximization, *Econometrica*, 55(3), 693-698.

Choi, S., R. Fisman, D. Gale, and S. Kariv (2007): Consistency and Heterogeneity of Individual Behavior under Uncertainty, *American Economic Review*, 97(5), 1921-1938.

Diewert, W. E. (1973): Afriat and Revealed Preference Theory, *Review of Economic Studies*, 40(3), 419-425.

Fisman, R., S. Kariv, and D. Markovits (2007): Individual Preferences for Giving, *American Economic Review*, 97(5), 1858-1876.

Heufer, J. (2011): Stochastic Dominance and Nonparametric Comparative Revealed Risk Aversion, *Ruhr Economic Papers*, #289, TU Dortmund University, Discussion Paper.

——— (2013): Testing Revealed Preferences for Homotheticity with Two-Good Experiments, *Experimental Economics*, 16(1), 114-124.

Knuth, D. E. (1998 [1969]): *The Art of Computer Programming*, volume 2, Addison-Wesley, 3rd edition.

Little, J. T. (1979): Indirect Preferences, *Journal of Economic Theory*, 20(2), 182–193.

Sakai, Y. (1977): Revealed Favorability, Indirect Utility, and Direct Utility, *Journal of Economic Theory*, 14(1), 113–129.

Smith, R. L. (1984): Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed over Bounded Regions, *Operations Research*, 32(6), 1296-1308.

Tempo, R., G. Calafiore, and F. Dabbene (2005): *Randomized Algorithms for Analysis and Control of Uncertain Systems*, London: Springer.

Varian, H. R. (1982): The Nonparametric Approach to Demand Analysis, *Econometrica*, 50(4), 945-972.

——— (1983): Non-parametric Tests of Consumer Behaviour, *Review of Economic Studies*, 50(1), 99-110.

——— (1984): The Nonparametric Approach to Production Analysis, *Econometrica*, 52(3), 579-597.

——— (1996): *Computational Economics and Finance*, chapter Efficiency in Production and Consumption, Springer-Verlag New York, 131–142.