Online Appendix of "Partial Identification and Inference for Dynamic Models and Counterfactuals"

Myrto Kalouptsidi, Yuichi Kitamura, Lucas Lima, and Eduardo Souza-Rodrigues*

February, 2020

This Online Appendix consists of the following sections: Section E provides several useful examples of linear restrictions that are commonly employed in applied work. Section F describes the implementation of the inferential procedure for the target parameter based on subsampling. Section G shows how to calculate the gradient of the function f when it involves counterfactual average effects based on ergodic distributions of the state variables. Section H discusses practical and computational aspects of calculating the identified set of low-dimensional outcomes of interest; in particular, it presents our proposed stochastic search algorithm to calculate the identified set without analytic gradients.

E Examples of Linear Restrictions

In this section, we provide several useful examples of linear restrictions, $R^{eq}\pi = r^{eq}$ and $R^{iq}\pi \leq r^{iq}$, that are commonly employed in applied work. For ease of exposition, we only consider restrictions on π_J (unless otherwise stated). Recall that $R^{eq} = [R^{eq}_{-J}, R^{eq}_J]$ and $R^{iq} = [R^{iq}_{-J}, R^{iq}_J]$.

Example 1. *(Compact Payoffs) Assume $\delta^l_J \leq \pi_J \leq \delta^u_J$. Then $R^{iq}_{-J} = 0$, $R^{iq}_J = [-I,\ I]'$, $r^{iq} = [-\delta^{l\prime}_J,\ \delta^{u\prime}_J]'$, and the number of inequalities is $m = 2X$.*

Example 2. *(Exclusion Restriction I) Assume $\pi_J(x_1) = \pi_J(x_2)$. Then, $R^{eq}\pi = r$, with $R^{eq}_{-J} = 0$,*

$$R^{eq}_J = \left[\begin{array}{ccccc} 1 & -1 & 0 & \cdots & 0 \end{array}\right],$$

and $r^{eq} = 0$. There is only one equality restriction: $d = 1$.

Example 3. *(Exclusion Restriction II) Suppose we split the state space in $x = (k, w)$, where $k \in \mathcal{K} = \{1, ..., K\}$ and $w \in \mathcal{W} = \{1, ..., W\}$, with K, W finite. Assume π_J does not depend on w, i.e., $\pi_J(k, 1) =$*

*Affiliations: Myrto Kalouptsidi, Harvard University, CEPR and NBER; Yuichi Kitamura, Yale University and Cowles Foundation for Research in Economics; Lucas Lima, Harvard University; Eduardo Souza-Rodrigues, University of Toronto.

$\pi_J(k,2) = ... = \pi_J(k,W)$ *for all* k. *When* $K=2$, $W=3$, *we obtain* $R^{eq}_{-J}=0$,

$$R^{eq}_J = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix},$$

and $r^{eq}=0$. *The number of linear equalities is now* $d = K(W-1) < KW = X$.

Example 4. *(Monotonicity) Without loss, arrange* x *in increasing order. Assume* π_J *increases with* x. *Then* $\pi_J(1) \leq \pi_J(2) \leq ... \leq \pi_J(X)$. *In this case, take* $m = X-1$, $r^{iq}=0$, $R^{iq}_{-J}=0$, *and*

$$R^{iq}_J = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & -1 \end{bmatrix}.$$

Example 5. *(Concavity) Arrange* x *in increasing order, take equidistant points for* x. *Assume* π_J *is concave in* x. *Then* $\pi_J(x_{i-1}) - 2\pi_J(x_i) + \pi_J(x_{i+1}) \leq 0$, *for all* $i = 2, ..., X-1$. *In this case, take* $m = X-2$, $r^{iq}=0$, $R^{iq}_{-J}=0$, *and*

$$R^{iq}_J = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix}.$$

Example 6. *(Smoothness). Suppose* $\pi_J \geq 0$ *(take* $\delta^l_a = 0$*) and assume* $\pi_J(x)$ *is Lipschitz continuous in* x. *Then,* $\pi_J(x_i) - \pi_J(x_{i+1}) \leq L|x_i - x_{i+1}|$, *for some known constant* $L < \infty$, *for all* x. *In this case,* $R^{iq}_{-J}=0$,

$$R^{iq}_J = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & -1 \end{bmatrix},$$

r^{iq} *is the vector with elements* $L|x_i - x_{i+1}|$, *and* $m = X-2$. *Note that we can impose higher order restrictions on the variation of the function* π *as well. This may be important when we discretize a continuous state space and* π *is a smooth function of states.*

Example 7. *(Action-Monotonicity) Take the binary model with actions* $\mathcal{A} = \{a, J\}$, *and assume that*

$\pi_a(x) \geq \pi_J(x)$ *for some* x*. Then* R_a^{iq} *is the vector with* -1 *at position* x *and zeros elsewhere. Similarly,* R_J^{iq} *is the vector with* 1 *at position* x *and zero elsewhere. I.e.,*

$$R^{iq}\pi = \begin{bmatrix} 0 & \cdots & 0 & -1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \pi_a \\ \pi_J \end{bmatrix} \leq 0,$$

where $r^{iq} = 0$ *and* $m = 1$*.*

Example 8. *(Supermodularity) Take the binary model again. Without loss, arrange* x *in increasing order. Assume the increasing differences for* $x_{i+1} \geq x_i$*:*

$$\pi_a(x_{i+1}) - \pi_a(x_i) \geq \pi_J(x_{i+1}) - \pi_J(x_i).$$

Then, take

$$R^{iq} = \left[\begin{array}{cccccc|cccccc} & & & & & & & & & & & \\ 1 & -1 & 0 & 0 & \cdots & 0 & -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 & 0 & -1 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 1 & -1 & 0 & 0 & \cdots & 0 & -1 & 1 \end{array} \right],$$

and $r^{iq} = 0$*, with* $m = X - 1$ *inequalities.*

F Computational Algorithm for Inference

We now describe the implementation of the inferential procedure for the target parameter θ. It builds on the formulation developed in Kitamura and Stoye (2018), where the implications of economic models are expressed in terms of the minimum value of a quadratic form.

Recall that we construct a confidence set by inverting tests of the type $H_0 : \theta = \theta_0$, which are equivalent to testing $H_0' : J(\theta_0) = 0$. We explain how we approximate the distribution of the test statistic $N\widehat{J}_N(\theta_0)$, from which we obtain the critical values $\widehat{c}_{1-\alpha}$. We also emphasize ways in which we can make the procedure computationally faster.

We assume that $\widehat{\mathfrak{p}}_N$ is a frequency estimator for $\mathfrak{p}$ (collecting the frequency estimators for p and F), and that the researcher has R replications $\widehat{\mathfrak{p}}^{*r}$, $r = 1, ..., R$. Recall that, for a fixed θ_0, our test statistic is

$$\widehat{J}_N(\theta_0) := \min_{\substack{(\widetilde{p},\pi)\in\widetilde{\mathbf{P}}\times\mathbb{R}^{(A+1)X} : R^{iq}\pi \leq r^{iq}, \\ \mathcal{R}(\theta_0,\pi,\widetilde{p};\widehat{\mathfrak{p}}_N)=0}} [b_{-J}(\widehat{\mathfrak{p}}_N) - \widehat{\mathbf{M}}_N\pi]' \, \Omega \, [b_{-J}(\widehat{\mathfrak{p}}_N) - \widehat{\mathbf{M}}_N\pi]. \tag{F1}$$

The corresponding test statistic being simulated is

$$\widehat{J}^{*r}_{h_N}(\theta_0) := \min_{\substack{(\widetilde{p},\pi)\in\widetilde{\mathbf{P}}\times\mathbb{R}^{(A+1)X}\,:\,R^{iq}\pi\leq r^{iq},\\ \mathcal{R}\left(\theta_0,\pi,\widetilde{p};\widehat{\mathfrak{p}}^{*r}_{h_N}\right)=0}} [\widehat{b}^{*r}_{-J} - \widehat{\mathbf{M}}^{*r}_{h_N}\pi]'\Omega[\widehat{b}^{*r}_{-J} - \widehat{\mathbf{M}}^{*r}_{h_N}\pi], \tag{F2}$$

for $r = 1, ..., R$, where $\widehat{\mathfrak{p}}^{*r}_{h_N}$, $\widehat{b}^{*r}_{-J}$ and $\widehat{\mathbf{M}}^{*r}_{h_N}$ are the r-th subsampling replication of the estimator for $\mathfrak{p}$, b_{-J} and $\mathbf{M}$; see Section 5 in the main paper for the definitions of these elements.

Recall the testing procedure: We use the empirical distribution of $h_N\widehat{J}^{*}_{h_N}(\theta_0)$ to obtain the critical value $\widehat{c}_{1-\alpha}$. When the value of the test statistic is smaller than the critical value, $N\widehat{J}_N(\theta_0) \leq \widehat{c}_{1-\alpha}$, we do not reject the null $H_0' : J(\theta_0) = 0$, otherwise we reject it. The $1-\alpha$ confidence set is the collection of θ_0's for which these tests do not reject the null.

We now turn to several remarks that make this procedure operational and computationally fast (or at least faster than approaches that ignore these remarks).

Grid-Search. When θ is a scalar, we can implement a limited grid-search on θ. Specifically, let $\widehat{\theta}^L$ and $\widehat{\theta}^U$ be the estimated lower and upper bounds of the identified set for θ, obtained by solving the appropriate constrained minimization and maximization problems in the full data. Clearly, $\widehat{J}_N(\theta_0) = 0$ for all $\theta_0 \in [\widehat{\theta}^L, \widehat{\theta}^U]$, so no point in that interval would be rejected by the data. We therefore start the grid-search at points slightly below $\widehat{\theta}^L$ and slightly above $\widehat{\theta}^U$. To simplify, consider the points above $\widehat{\theta}^U$. We start with the point, say, $\theta_0 = \widehat{\theta}^U + 0.01$, and test the null $H_0' : J(\theta_0) = 0$ using the subsampling procedure described above. If we fail to reject the null, we then move to the next point, say, $\theta_0 = \widehat{\theta}^U + 0.02$ and test the null for that new point. We keep doing so until we reject the null for the first time; we stop the grid-search when we first reject the null because all points to the right will be rejected by the data as well. We adopt a similar procedure for the lower end $\widehat{\theta}^L$, by taking $\widehat{\theta}^L - 0.01$, then $\widehat{\theta}^L - 0.02$, etc.

One benefit to this approach is that we just need to search over a limited region of the real line $\mathbb{R}$: right below $\widehat{\theta}^L$ and right above $\widehat{\theta}^U$. If we reject the null for the first time at the points θ^l and θ^u, then the asymptotically uniformly valid $1-\alpha$ confidence set for the true θ is the interval $[\theta^l, \theta^u]$.

Exploiting Continuity. We can solve a sequence of minimization problems (F1) exploiting the fact that the model is smooth and well-behaved. Specifically, by changing θ_0 sequentially and incrementally, we always obtain good initial guesses for the next minimization problem, reducing the total computational costs. More explicitly, suppose we have already solved the maximization problem (20)–(21) in the full sample, and obtained the estimated maximum $\widehat{\theta}^U$. Denote the solution by $(\widetilde{p}^s, \pi^s)$, i.e., $\widehat{\theta}^U = f(\widetilde{p}^s, \pi^s; \widehat{\mathfrak{p}})$. This implies that the solution to the problem (F1) with $\theta_0 = \widehat{\theta}^U$ is $(\widetilde{p}^s, \pi^s)$, and that $\widehat{J}_N(\widehat{\theta}^U) = 0$ (by construction). Then, by continuity, the solution to (F1) for $\theta_0 = \widehat{\theta}^U + \eta$ should be close to $(\widetilde{p}^s, \pi^s)$ when η is small (e.g., $\eta = 0.01$), making $(\widetilde{p}^s, \pi^s)$ an excellent initial value for the numerical calculations. (That

also implies that $\widehat{J}_N(\theta_0)$ for $\theta_0 = \widehat{\theta}^U + \eta$ should be a small strictly positive number – a fact that we exploit further below.) Solving a series of well-behaved minimization problems with good initial values (by changing η incrementally) speeds up the computation of our test statistic in the full sample for various θ_0's.

The same idea can be extended to the subsampling version (F2): We can solve the minimization (F2) in the r-th replication for a fixed θ_0 that equals $\widehat{\theta}^U + \eta$ by using the full data solution of (F1) as initial guess. For larger values of η, we can use the solution to (F2) obtained for smaller η's as initial guesses (rather than always using the full data solution as initial guess).

Computing $J_N(\theta_0)$ in Practice. When f is costly to evaluate, it is difficult to solve the minimization problems (F1) and (F2) in practice. The reason is that it is difficult to search over $(\widetilde{p}, \pi)$ to minimize $J(\theta_0)$ when the constraint $\theta_0 = f(\widetilde{p}, \pi; \mathfrak{p})$ must be satisfied for a fixed θ_0. Putting differently, finding particular values for $(\widetilde{p}, \pi)$ that satisfy $\theta_0 = f(\widetilde{p}, \pi; \mathfrak{p})$ can be computationally costly.

One way to bypass this difficulty is to take advantage of the relationship between the optimization problems (20)–(21) and (F1) (and the subsampling version (F2)). We have mentioned briefly how these optimization problems relate to each other in a previous paragraph. Now, we discuss it in more detail.

Abstracting from sampling issues, consider the relaxed version of the maximization (20)–(21):

$$\theta^U(\epsilon) \equiv \max_{(\widetilde{p},\pi)\in\widetilde{\mathbf{P}}\times\mathbb{R}^{(A+1)X}} f\left(\widetilde{p}, \pi;\, \mathfrak{p}\right) \tag{F3}$$

subject to

$$\begin{aligned} \left\|\mathbf{M}(\mathfrak{p})\,\pi - b_{-J}(\mathfrak{p})\right\|_\Omega &\leq \epsilon, \\ R^{eq}\,\pi &= r^{eq}, \\ R^{iq}\,\pi &\leq r^{iq}, \\ (\widetilde{\mathbf{M}}(\mathfrak{p})\,\mathcal{H})\,\pi &= \widetilde{b}_{-J}\left(\widetilde{p}, \mathfrak{p}\right) - \widetilde{\mathbf{M}}(\mathfrak{p})\,g. \end{aligned} \tag{F4}$$

where $\|.\|_\Omega$ is the matrix norm defined as $\|x\|_\Omega = x'\Omega x$ for $x \in \mathbb{R}^{AX}$, and $\epsilon \geq 0$. We also consider the relaxed minimization problem by replacing the max operator in (F3) by the min operator. We index the minimum and the maximum of the relaxed problems by ϵ, so that we have $\theta^L(\epsilon)$ and $\theta^U(\epsilon)$, respectively.

The difference between the original problem (20)–(21) and its relaxed version (F3)–(F4) is the inequality constraint $\left\|\mathbf{M}(\mathfrak{p})\,\pi - b_{-J}(\mathfrak{p})\right\|_\Omega \leq \epsilon$. Evidently, the problems coincide when $\epsilon = 0$. Furthermore, $\theta^L(\epsilon) \leq \theta^L(0) \equiv \theta^L$ and $\theta^U(\epsilon) \geq \theta^U(0) \equiv \theta^U$. I.e., the true identified set $[\theta^L, \theta^U]$ is contained in the interval $[\theta^L(\epsilon), \theta^U(\epsilon)]$ when $\epsilon > 0$.

Importantly, while $J(\theta_0) = 0$ for all points θ_0 in the identified set $[\theta^L, \theta^U]$, we have $J(\theta_0) \leq \epsilon$ for all points θ_0 in the wider interval $[\theta^L(\epsilon), \theta^U(\epsilon)]$ by construction. This implies that $0 < J(\theta_0) \leq \epsilon$ for all points

belonging to the wider interval, $\theta_0 \in [\theta^L(\epsilon), \theta^U(\epsilon)]$, but not to the smaller set, $\theta_0 \notin [\theta^L, \theta^U]$. Take such a point, and denote it by $\theta_0^*(\epsilon)$. If we solve the following problem for $\theta_0^*(\epsilon)$,

$$J(\theta_0^*(\epsilon)) := \min_{\substack{(\widetilde{p},\pi)\in\widetilde{\mathbf{P}}\times\mathbb{R}^{(A+1)X}\,:\,R^{iq}\pi\leq r^{iq},\\ \mathcal{R}(\theta_0^*(\epsilon),\pi,\widetilde{p};\mathfrak{p})=0}} [b_{-J}(\mathfrak{p}) - \mathbf{M}(\mathfrak{p})\pi]'\,\Omega\,[b_{-J}(\mathfrak{p}) - \mathbf{M}(\mathfrak{p})\pi], \tag{F5}$$

then the minimum $J(\theta_0^*(\epsilon))$ must be strictly greater than 0 and (weakly) smaller than ϵ. Note that by taking a sequence of $\epsilon' s$, $0 = \epsilon_0 < \epsilon_1 < ... < \epsilon_k < ... < \epsilon_K$, and solving the corresponding relaxed problem (F3)–(F4) for all ϵ_k, we obtain the sequence of increasing intervals

$$[\theta^L(0), \theta^U(0)] \subseteq ... \subseteq [\theta^L(\epsilon_k), \theta^U(\epsilon_k)] \subseteq ... \subseteq [\theta^L(\epsilon_K), \theta^U(\epsilon_K)].$$

We also obtain a sequence of J's such that: (a) $J(\theta_0) = 0$ if $\theta_0 \in [\theta^L(0), \theta^U(0)]$, and (b) $\epsilon_{k-1} < J(\theta_0) \leq \epsilon_k$ if $\theta_0 \in [\theta^L(\epsilon_k), \theta^U(\epsilon_k)]$ and $\theta_0 \notin [\theta^L(\epsilon_{k-1}), \theta^U(\epsilon_{k-1})]$. This means that, by taking a fine grid of ϵ's and solving their corresponding relaxed problems (F3)–(F4), we obtain good approximations to the minimum value of the problem (F5) for various θ_0's.

Note that this approach can exploit the continuity of the relaxed problem (F3)–(F4) with respect to ϵ. That is, once we solve (F3)–(F4) for some ϵ_k, it is computationally cheap to solve the problem for a close $\epsilon_{k+1} > \epsilon_k$, by using the solution to the ϵ_k–relaxed problem as initial guess to the ϵ_{k+1}–relaxed problem.

Summarizing. We want to test the null $H_0 : \theta = \theta_0$ for various θ_0's. We estimate $\widehat{\theta}^L$ and $\widehat{\theta}^U$ by solving the minimization and maximization problems (20)–(21). We then start with the point $\theta_0 = \widehat{\theta}^U + \eta$, as explained previously. For that point, solve the relaxed problem (F3)–(F4) for various ϵ_k, $0 = \epsilon_0 < \epsilon_1 < ... < \epsilon_2 < ... < \epsilon_K$, *both* in the full sample and in *all* R replicated samples (taking advantage of the continuity of the solutions with respect to ϵ). If this particular θ_0 lies the interval $[\widehat{\theta}^L(\epsilon_k), \widehat{\theta}^U(\epsilon_k)]$ but does not lie in $[\widehat{\theta}^L(\epsilon_{k-1}), \widehat{\theta}^U(\epsilon_{k-1})]$, then $\epsilon_{k-1} < \widehat{J}_N(\theta_0) \leq \epsilon_k$. Assuming the difference between ϵ_{k-1} and ϵ_k is small enough, we approximate the test statistic by $N\widehat{J}_N(\theta_0) \simeq N \times \epsilon_k$. Applying the same reasoning in each simulated sample r, we obtain approximated values for $\widehat{J}^{*r}_{h_N}(\theta_0)$, $r = 1, ..., R$, from which we can approximate $h_N \widehat{J}^{*r}_{h_N}(\theta_0)$. Inverting the empirical distribution of these values we can construct the critical value $\widehat{c}_{1-\alpha}$ and test the null for our particular $\theta_0 = \widehat{\theta}^U + \eta$. Then, we repeat the procedure by changing η incrementally until the null is rejected for the first time, providing the estimated endpoints of the confidence interval.

In essence, to improve the performance of the subsampling procedure, we can exploit (i) the relationship between the optimizations (20)–(21) and (F1) (or yet (F3)–(F4) and (F5)); (ii) the continuity of the solutions to the optimization problems with respect to *both* ϵ and η; as well as (iii) a grid-search for θ performed over a *limited* region of the real line. In doing so, we can compute an asymptotically uniformly valid $1 - \alpha$ confidence set for the true outcome of interest θ in a tractable way.

G Gradient of f involving Ergodic Distribution

In this section, we show how to calculate the gradient of the function f when it involves counterfactual average effects based on ergodic distributions of the state variables.

Assume the function f is given by

$$f(\widetilde{p},\pi;p,F)=\sum_{x\in\widetilde{\mathcal{X}}}\widetilde{Y}(x;\pi)\widetilde{f}^{*}(x)-\sum_{x\in\mathcal{X}}Y(x;\pi)f^{*}(x),$$

where $\widetilde{Y}(x;\pi)$ and $Y(x;\pi)$ are outcome variables of interest in the counterfactual and baseline scenarios and that may depend on baseline payoffs π (e.g., consumer surplus, or the firm value). Note that all examples presented in the Supplemental Material (Section B) are of this type; in the empirical application of Section 6 in the main text, we take the ratio of objects of this type.

The term $\widetilde{f}^{*}(x)$ is the ergodic distribution of the (endogenous) Markovian process for the state variables

$$\widetilde{F}(x'|x)=\sum_{a\in\widetilde{\mathcal{A}}}\widetilde{p}(a|x)\widetilde{F}(x'|a,x).$$

(A similar expression holds for $f^{*}(x)$.) In matrix notation, we have

$$f(\widetilde{p},\pi;p,F)=\widetilde{\mathbf{Y}}'\,\widetilde{f}^{*}-\mathbf{Y}'\,f^{*},$$

where $\widetilde{\mathbf{Y}}$ and $\mathbf{Y}$ are vectors of the outcome variables in the counterfactual and baseline; and $\widetilde{f}^{*}$ is the vector of the ergodic distribution satisfying the steady-state condition

$$\widetilde{f}^{*\prime}=\widetilde{f}^{*\prime}\,\widetilde{\mathbf{F}},\tag{G1}$$

where

$$\widetilde{\mathbf{F}}=\sum_{a\in\widetilde{\mathcal{A}}}\widetilde{P}_{a}\widetilde{F}_{a},\tag{G2}$$

and $\widetilde{P}_{a}$ is a diagonal matrix with $\widetilde{p}_{a}$ in its diagonal, and $\widetilde{F}_{a}$ is the counterfactual transition matrix conditional on the choice a. (Again, a similar expression holds for f^{*}.) Importantly, the ergodic distribution $\widetilde{f}^{*}$ depends directly on $\widetilde{p}$ (through equations (G1)–(G2)), and indirectly on the baseline payoff π, since $\widetilde{p}$ depends on π through equation (15) presented in the main text.

We want to know the derivative of f with respect to π, holding all other arguments of f constant (e.g., the baseline CCP p and the state transitions F). Clearly, we have

$$\frac{\partial f}{\partial\pi'}=\left(\widetilde{f}^{*\prime}\,\frac{\partial\widetilde{\mathbf{Y}}}{\partial\pi'}\right)-\left(f^{*\prime}\,\frac{\partial\mathbf{Y}}{\partial\pi'}\right)+\left(\widetilde{\mathbf{Y}}'\,\frac{\partial\widetilde{f}^{*}}{\partial\pi'}\right).$$

The derivatives $\frac{\partial \widetilde{\mathbf{Y}}}{\partial \pi'}$ and $\frac{\partial \mathbf{Y}}{\partial \pi'}$ depend on the specific outcome of interest. Here, we focus on the third term of the right-hand-side, $\frac{\partial \widetilde{f}^*}{\partial \pi'}$. By the chain rule, we have

$$\frac{\partial \widetilde{f}^*}{\partial \pi'} = \frac{\partial \widetilde{f}^*}{\partial \widetilde{p}'} \frac{\partial \widetilde{p}}{\partial \pi'}.$$

By equation (15), we know that

$$\frac{\partial \widetilde{p}}{\partial \pi'} = \left(\frac{\partial \widetilde{b}_{-J}}{\partial \widetilde{p}'} \right)^{-1} \widetilde{\mathbf{M}} \mathcal{H}.$$

We now derive the remaining term $\frac{\partial \widetilde{f}^*}{\partial \widetilde{p}'}$. Let $x, x', \tilde{x}$ be arbitrary states and $\widetilde{a} \neq J$. Then (G1) pointwise becomes

$$\widetilde{f}^*\left(x'\right) = \sum_x \widetilde{f}^*\left(x\right) \sum_a \widetilde{p}_a\left(x\right) \widetilde{F}\left(x'|x,a\right).$$

Therefore,

$$\frac{\partial \widetilde{f}^*\left(x'\right)}{\partial \widetilde{p}_{\widetilde{a}}\left(\tilde{x}\right)} = \sum_x \frac{\partial \widetilde{f}^*\left(x\right)}{\partial \widetilde{p}_{\widetilde{a}}\left(\tilde{x}\right)} \widetilde{F}\left(x'|x\right) + \widetilde{f}^*\left(\tilde{x}\right) \left[\widetilde{F}\left(x'|\tilde{x},\widetilde{a}\right) - \widetilde{F}\left(x'|\tilde{x},J\right) \right].$$

This is written compactly in matrix as,

$$\frac{\partial \widetilde{f}^*}{\partial \widetilde{p}'_a} = -(\widetilde{\mathbf{F}}' - I)^+ (\widetilde{F}'_a - \widetilde{F}'_J) \, \widetilde{\mathbf{f}}^*, \tag{G3}$$

where $(\widetilde{\mathbf{F}}' - I)^+$ is the pseudo-inverse of $(\widetilde{\mathbf{F}}' - I)$, and $\widetilde{\mathbf{f}}^*$ is a diagonal matrix with $\widetilde{f}^*$ in its diagonal.

H A Proposed Algorithm Based on Stochastic Search

In this section, we discuss practical and computational aspects of calculating the identified set of low-dimensional outcomes of interest θ. As explained in the main text, in our experience, standard solvers are highly efficient in solving (20)–(21) when the researcher can provide the gradient of f. However, when numerical (or analytical) gradients are costly to evaluate in practice, standard solvers can be slow in converging to the maximum (again, in our experience). For such cases, we propose a stochastic algorithm that exploits the structure of the problem and combines the strengths of alternative stochastic search procedures, as we explain below.

Our proposed algorithm builds upon a couple of observations. First, while a search over π to maximize f is feasible, it is computationally costly and (in our experience) takes a long time to converge when calculating the gradient of f numerically is expensive. (In high-dimensional problems, this may become impractical.) This procedure searches over the admissible values that π can take, and, for each candidate, it finds the corresponding counterfactual CCP by solving the nonlinear equation (15), and then it evaluates f – and its (numerical) derivative, to obtain updated directions for π – until reaching the maximum value for θ. Although finding admissible values for π is not difficult in high-dimensional problems (as it only

depends on linear constraints), and solving the nonlinear equation (15) once is not computationally costly (as standard quasi-Newton methods can be used to find $\widetilde{p}$), solving (15) too many times and calculating the gradient of f numerically can be demanding. Unless the econometrician imposes a sufficient number of assumptions to make π effectively a low dimensional vector (e.g., 3-dimensional or smaller), this method takes a long time to converge, as it requires too many evaluations before we can increase θ substantially in the direction of its maximum.

Second, it is possible to perform a search over $\widetilde{p}$, instead of over π, to calculate θ^U. For any given $\widetilde{p}$, existence of a π satisfying linear constraints is computationally cheap (for example, existence can be easily checked as a solution to a quadratic programming problem). If there is no such π satisfying all restrictions, we discard $\widetilde{p}$, since it does not belong to the identified set $\widetilde{\mathbf{P}}^I$. If there exists *some* π satisfying the restrictions, we keep $\widetilde{p}$, and compute the corresponding θ. This approach may be particularly useful when f is not a direct function of π, in which case it is not necessary to find a particular π to calculate θ – existence of *some* π suffices. The difficulty here is that, while an exhaustive grid search over $\widetilde{p}$ can be used to find the maximum θ^U, grid search is unfeasible for empirically-relevant high-dimensional problems. An alternative would be to perform a stochastic search (to find good directions for $\widetilde{p}$).[1] Yet, and more importantly, the random search must be performed on the $\widetilde{A}\widetilde{X}$–space $\widetilde{\mathbf{P}}$, while the identified set $\widetilde{\mathbf{P}}^I$ can be of much smaller dimension: $X-d$, or smaller (depending on the rank of $C(I-P_Q)$; see Proposition 3). In other words, $\widetilde{\mathbf{P}}^I$ may be a "thin" set in $\widetilde{\mathbf{P}}$. The combination of a "thin" set with an unknown shape (recall that $\widetilde{p}$ is a nonlinear function of π – see equation (15)) makes it is difficult to find points within that set randomly. Further, it is easy for perturbation methods to "exit" the set, increasing the costs of finding the maximum θ. Note that searching over π to maximize θ does not suffer from this problem because finding admissible values (and updated directions) for π are computationally easier.

These trade-offs led us to consider an algorithm that exploits the structure of the problem and combines the strengths of these alternative search procedures. Intuitively, we move in the "$\widetilde{p}$-world" (to avoid solving the nonlinear equation (15) repeatedly), but we keep a close eye on the "π-world" (to keep track of the model restrictions and search in relevant directions). Searching in relevant directions without solving (15) and computing the numerical gradient of f in every step improves substantially how fast θ moves on each iteration to the maximum.

[1] For instance, one possibility is to perturb $\widetilde{p}$ completely randomly ($\widetilde{p}+\varepsilon$) and check whether the perturbed vector lies in the identified set $\widetilde{\mathbf{P}}^I$ (or within a tolerance level) – where checking this amounts to checking existence of π satisfying linear restrictions, as mentioned above. We then keep the perturbed $\widetilde{p}'s$ that deliver large values for θ (and perturb them further), and discard those with small values of θ. We iterate until θ cannot be increased any longer. (This is similar to genetic algorithm, or to stochastic search methods more generally.)

H.1 The Proposed Stochastic Search Algorithm

We now present our proposed algorithm. In order to not worry about $\widetilde{p}$ being positive and adding up to one, we work with the transformation

$$\widetilde{\delta} = \ln \widetilde{p}_{-J} - \ln \widetilde{p}_J,$$

where $\ln \widetilde{p}_a$ is the $\widetilde{X} \times 1$ vector with elements $\ln \widetilde{p}_a(x)$, for all $x \in \widetilde{\mathcal{X}}$, and $\ln \widetilde{p}_{-J}$ stacks $\ln \widetilde{p}_a$ for all $a \neq J$. The functions of $\widetilde{p}$, namely $\widetilde{b}_{-J}$ and f, are adjusted accordingly. To simplify notation, we drop the subscript of the function $\widetilde{b}_{-J}$, as well as the arguments (p, F) of the function f. The algorithm proceed as follows (we provide detailed discussions of the most important steps below):

The Proposed Stochastic Search Algorithm:

1. Initialize $k = 0 \in \mathbb{N}$.

   Set π^k satisfying $R^{eq}\pi^k = r^{eq}$ and $R^{iq}\pi^k \leq r^{iq}$. Find $\widetilde{\delta}^k$ by solving (15) with π^k. Calculate $\theta^k = f(\widetilde{\delta}^k, \pi^k)$.

2. Increment k.

3. Set (perturbed) direction $\Delta\pi^k$. Given $\Delta\pi^k$, set direction for $\widetilde{\delta}^k$,

$$\Delta\widetilde{\delta}^k = \left(\frac{\partial \widetilde{b}}{\partial \widetilde{\delta}}\right)^{-1} \widetilde{\mathbf{M}}\,\mathcal{H}\,\Delta\pi^k,$$

   where $\left(\frac{\partial \widetilde{b}}{\partial \widetilde{\delta}}\right)$ is the derivative of $\widetilde{b}$ with respect to $\widetilde{\delta}$ evaluated at $\widetilde{\delta}^k$.

4. Solve for $\alpha \in \mathbb{R}$:

$$\alpha^* = \underset{\alpha}{\operatorname{argmax}} \; f(\widetilde{\delta}^k + \alpha\Delta\widetilde{\delta}^k, \pi^k + \alpha\Delta\pi^k),$$

   subject to the constraints (21), allowing (15) to be violated at most by a tolerance level, $\mathrm{tol} > 0$.

5. Set $\widetilde{\delta}^* = \widetilde{\delta}^k + \alpha^*\Delta\widetilde{\delta}^k$ and $\pi^* = \pi^k + \alpha^*\Delta\pi^k$.

6. Update $\widetilde{\delta}^k$:

$$\widetilde{\delta}^{k+1} = \widetilde{\delta}^* - \left(\frac{\partial \widetilde{b}^*}{\partial \widetilde{\delta}}\right)^{-1} \left(\widetilde{b}^* - \widetilde{\mathbf{M}}g - \widetilde{\mathbf{M}}\mathcal{H}\pi^*\right),$$

   where $\left(\frac{\partial \widetilde{b}^*}{\partial \widetilde{\delta}}\right)$ is the derivative of $\widetilde{b}$ with respect to $\widetilde{\delta}$ evaluated at $\widetilde{\delta}^*$, and $\widetilde{b}^* = \widetilde{b}(\widetilde{\delta}^*)$. Set $\pi^{k+1} = \pi^*$.

7. Calculate $\theta^{k+1} = f(\widetilde{\delta}^{k+1}, \pi^{k+1})$.

   If $\left\|\theta^{k+1} - \theta^k\right\| \leq \epsilon$ go to 8; otherwise go to 2.

8. Set $\pi = \pi^{k+1}$. Solve (15) exactly for π, and get $\widetilde{\delta}$. Return $\theta^U = f(\widetilde{\delta}, \pi)$.

We now discuss the rationale for each step. In Subsection H.2, we provide further details for the implementation of each step, as well as a discussion about the overall cost of the algorithm.

Step 1. The first step requires finding a π that satisfies the model restrictions (7) and (8) so that we obtain an initial $\widetilde{p}$ (or $\widetilde{\delta}$) that lies inside the (potentially "thin") set $\widetilde{\mathbf{P}}^I$ by construction, and a corresponding θ in the identified set $\mathbf{\Theta}^I$. Such initial π can be obtained as any solution to the following quadratic programming problem

$$\min_{\pi} \ \left(R^{eq}\pi - r^{eq}\right)' \left(R^{eq}\pi - r^{eq}\right) + \left(R^{iq}\pi - r^{iq}\right)'_{+} \left(R^{iq}\pi - r^{iq}\right)_{+}, \tag{H1}$$

where $(x)_+ = \max\{x, 0\}$. Another option is to start with a few points and project them into the identified set for π, which can also be done easily. (Of note, if the minimum of (H1) is strictly greater than zero, then there is no π that satisfies all the constraints.) Given π, we can solve (15) numerically using some quasi-Newton method.

Step 3. After we have our starting point π (and corresponding $\widetilde{\delta}$), we need to obtain an updated direction $\Delta\pi$ (and $\Delta\widetilde{\delta}$). Overall, the idea of providing first a direction and only then optimize (as we do here) is a standard way to solve complex optimization problems. Ideally, we would use the gradient of f, but calculating this gradient can be expensive in some cases, as mentioned previously. An alternative (that we use) is either to get a completely random direction for $\Delta\pi$ (e.g., $\Delta\pi = \eta$, where η is a random vector drawn from, say, a multivariate standard normal distribution), or a random direction weighted by states that are more important (e.g., in terms of the ergodic distribution of the state variables).[2]

It is also important to not let an updated point to get too close to the boundary of the inequality constraints (8). We follow the insights of interior-point methods to help the algorithm to not get stuck early on a boundary. Specifically, we add a term to $\Delta\pi$ that moves it way from the most binding ones. Formally,

$$\Delta\pi = \eta - \lambda \left(\frac{1}{r^{iq} - R^{iq}\pi}\right)' R^{iq},$$

where $\lambda = \frac{\lambda_0}{N}$, with $\lambda_0 > 0$ and N = the number of iterations; and $\left(\frac{1}{r^{iq} - R^{iq}\pi}\right)$ denotes the $m \times 1$ vector with the reciprocal elements of the vector $r^{iq} - R^{iq}\pi$ (recall that m is the number of inequality restrictions, so that R^{iq} is $m \times X$ and r^{iq} is $m \times 1$). The adjustment term $\lambda \left(\frac{1}{r^{iq} - R^{iq}\pi}\right)' R^{iq}$ is a common

[2] In practice, to weight the random direction η by states that are more important in terms of the steady-state distribution, we draw η from a normal distribution with zero mean and a diagonal variance-covariance matrix with a diagonal that equals the probabilities of the state variables under the ergodic distribution. The ergodic distribution is based on the latest updated $\widetilde{p}$.

way to handle inequality constraints. This is a simple implementation of a interior-point method, and helps the algorithm to not get stuck early on a boundary.[3]

We link the direction $\Delta\widetilde{\delta}$ with $\Delta\pi$ based on equation (15). We do so because completely random directions on $\widetilde{p}$ (or more precisely, on $\widetilde{\delta}$) will likely push $\widetilde{p}$ outside of the "thin" set $\widetilde{\mathbf{P}}^I$. The direction $\Delta\widetilde{\delta}$ is obtained by differentiating the inverse function $\widetilde{b}^{-1}$ with respect to π in the direction $\Delta\pi$.

Step 4. Given $\Delta\widetilde{\delta}$, we now find how far in that direction we should go without moving away too much from the identified set $\widetilde{\mathbf{P}}^I$. To that end, we allow for small violations in equation (15) when searching for α^*. Specifically, we replace the restriction (15) by $\left\|\widetilde{b} - \widetilde{\mathbf{M}}g - \widetilde{\mathbf{M}}\mathcal{H}\pi\right\| \leq \text{tol}$, where $\|.\|$ is some matrix norm and $\text{tol} > 0$ is a tolerance level. Here, the optimization is one-dimensional (line-search). We use a simple golden rule search, but even more crude approaches work.

Step 5. We now update both $\widetilde{\delta}$ and π in their respective directions $\alpha^*\Delta\widetilde{\delta}$ and $\alpha^*\Delta\pi_J$, where α^* is obtained in step 4.

Step 6. This step is important because at the end of step 5 it is common that the intermediary $\widetilde{\delta}^*$ violates the nonlinear system (15) by the maximum tolerance tol. So this step insures that we move $\widetilde{\delta}$ back to the set that violates (15) by strictly less than tol. Not doing so would constraint the directions that $\Delta\widetilde{\delta}$ can move in the next iteration and slow down the algorithm considerably.

Step 7. The $\epsilon > 0$ in step 7 specifies the convergence tolerance. We focus on convergence on θ because verifying a "derivative equals zero" condition for convergence is difficult given the high-dimensionality of the problem and the complexity of computing derivatives of f (analytically or numerically).

Step 8. After convergence, we solve the nonlinear system (15) exactly to guarantee that $\widetilde{p}$ lies in the identified set $\widetilde{\mathbf{P}}^I$, and so that the computed θ^U belongs to $\mathbf{\Theta}^I$.

One of the main computational cost of this algorithm is to calculate the inverse matrix $\left(\frac{\partial\widetilde{b}}{\partial\widetilde{\delta}}\right)^{-1}$, used in steps 3 and 6. In the next subsection we discuss under which conditions calculating $\left(\frac{\partial\widetilde{b}}{\partial\widetilde{\delta}}\right)^{-1}$ is not extremely costly.

[3]Intuitively, to maximize $f(x)$ subject to $g(x) \leq 0$, an interior-point method can make use of the logarithmic "barrier function" $B(x,\lambda) = f(x) - \lambda\sum_{i=1}^{n}\log\left(g_i(x)\right)$, where n is the dimension of g. The gradient of B is $\frac{\partial f}{\partial x} - \lambda\sum_{i=1}^{n}\frac{1}{g(x)}\frac{\partial g}{\partial x}$. The idea is that when some element $g_i(x)$ is close to zero for some trial x, the barrier function "explodes" to minus infinity, so that the algorithm does not get stuck on a boundary. However, because the solution may indeed lie on the boundary, it is necessary to allow for the possibility that $g_i(x) = 0$ at the optimum. To do so, λ must converge to zero as the number of iterations grows larger. In the present case, we take $\lambda = \frac{\lambda_0}{N} \to 0$ (as $N \to \infty$). The term $\left(\frac{1}{r^{iq} - R^{iq}\pi}\right)' R^{iq}$ is the derivative of the sum of the logs of $(r^{iq} - R^{iq}\pi)$ with respect to π (i.e., the derivative of $\lambda\sum log(r^{iq} - R^{iq}\pi)$, where the summation runs from 1 to m).

H.2 Further Comments on Implementation

We now comment on the computational costs of the algorithm.

1. The matrix M_a equals $(I - \beta F_a)(I - \beta F_J)^{-1}$, which involves the inversion of an $X \times X$ matrix. The computational cost of inverting a matrix is of the order of $O(X^3)$ in general. There are ways to reduce these costs, however. When action J is renewal or terminal, the matrix simplifies to $M_a = I + \beta(F_J - F_a)$, for all $a \in \mathcal{A}$, which can be calculated fast since it involves no matrix inversion.[4] When all actions are neither renewal nor terminal, computing M_a requires calculating the inverse of $(I - \beta F_J)$. Because F_J is a transition matrix, we can approximate that inverse based on the geometric series:

$$(I - \beta F_J)^{-1} = \sum_{\tau=0}^{\infty} \beta^\tau F_J^\tau.$$

   By truncating the series, we can reduce the computational costs and obtain a reasonable approximation (see more on that below). Note that both M_a and $\widetilde{M}_a$ can be precomputed, so they do not add costs to the iterated procedure.

2. When we find the direction $\Delta\widetilde{\delta}$ implied by $\Delta\pi$ we need to solve the linear system

$$\Delta\widetilde{\delta} = \left(\frac{\partial \widetilde{b}}{\partial \widetilde{\delta}}\right)^{-1} \widetilde{\mathbf{M}}\,\mathcal{H}\Delta\pi$$

   Usually this would cost $O(A^3X^3)$. However, we can take advantage of the structure of the function $\widetilde{b}$. Recall that $\widetilde{b}_a(\widetilde{p}) = \widetilde{M}_a\psi_J(\widetilde{p}) - \psi_a(\widetilde{p})$. To simplify notation, take $\widetilde{\psi}_J = \psi_J(\widetilde{p})$ and $\widetilde{\psi}_a = \psi_a(\widetilde{p})$, let $\widetilde{\psi}_{-J}$ stack $\widetilde{\psi}_a$ for all actions $a \neq J$. For expositional convenience, consider the three actions case with reference choice $J = 3$:

$$\widetilde{b} = \widetilde{M}_{-J}\widetilde{\psi}_J - \widetilde{\psi}_{-J} = \widetilde{\delta} - \left(\begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix} - \widetilde{M}_{-J}\right)\log\left(1 + \sum_{j=1}^{J-1}\exp(\widetilde{\delta}_j)\right),$$

   where $\mathbf{I}$ is the identity matrix. So

$$\frac{\partial \widetilde{b}}{\partial \widetilde{\delta}} = \mathbf{I} - \left(\begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix} - \widetilde{M}_{-J}\right)\begin{bmatrix}\widetilde{P}_1 & \widetilde{P}_2\end{bmatrix},$$

   where $\widetilde{P}_j$ is an $X \times X$ diagonal matrix with $\widetilde{p}_j$ as its entries.

[4] Formally, when action J is either a renewal or a terminal action, then for all a, $j \in \mathcal{A}$, $F_aF_J = F_jF_J$; see Kalouptsidi, Lima, and Souza-Rodrigues (2019).

Now we need its inverse

$$\left(\frac{\partial\widetilde{b}}{\partial\widetilde{\delta}}\right)^{-1} \overset{(1)}{=} \mathbf{I} + \left(\begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix} - \widetilde{M}_{-J}\right)\left(\mathbf{I} - \begin{bmatrix}\widetilde{P}_1 & \widetilde{P}_2\end{bmatrix}\left(\begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix} - \widetilde{M}_{-J}\right)\right)^{-1}\begin{bmatrix}\widetilde{P}_1 & \widetilde{P}_2\end{bmatrix} =$$

$$= \mathbf{I} + \left(\begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix} - \widetilde{M}_{-J}\right)\left(\mathbf{I} - \widetilde{P}_1 - \widetilde{P}_2 + \widetilde{P}_1\widetilde{M}_1 + \widetilde{P}_2\widetilde{M}_2\right)^{-1}\begin{bmatrix}\widetilde{P}_1 & \widetilde{P}_2\end{bmatrix},$$

where (1) follows from the Woodbury formula $(\mathbf{I} - DB)^{-1} = \mathbf{I} + D(\mathbf{I} - BD)^{-1}B$.

Now notice that $\widetilde{P}_J = \mathbf{I} - \widetilde{P}_1 - \widetilde{P}_2$ and that $\widetilde{M}_j = (\mathbf{I} - \beta\widetilde{F}_j)(\mathbf{I} - \beta\widetilde{F}_J)^{-1}$. So

$$\left(\frac{\partial\widetilde{b}}{\partial\widetilde{\delta}}\right)^{-1} = \mathbf{I} + \left(\begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix} - \widetilde{M}_{-J}\right)(\mathbf{I} - \beta\widetilde{F}_J)\left(\widetilde{P}_J(\mathbf{I} - \beta\widetilde{F}_J) + \widetilde{P}_1(\mathbf{I} - \beta\widetilde{F}_1) + \widetilde{P}_2(\mathbf{I} - \beta\widetilde{F}_2)\right)^{-1}\begin{bmatrix}\widetilde{P}_1 & \widetilde{P}_2\end{bmatrix}$$

$$= \mathbf{I} + \beta\begin{bmatrix}\widetilde{F}_1 - \widetilde{F}_J\\ \widetilde{F}_2 - \widetilde{F}_J\end{bmatrix}\left(\mathbf{I} - \beta\left(\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2\right)\right)^{-1}\begin{bmatrix}\widetilde{P}_1 & \widetilde{P}_2\end{bmatrix}.$$

This reduces the cost to $O(X^3)$ because the matrix

$$\left(\mathbf{I} - \beta\left(\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2\right)\right)^{-1}$$

has dimension $X \times X$.

But we can improve on that by noticing that for a given vector v,

$$\left(\mathbf{I} - \beta\left(\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2\right)\right)^{-1} v = \sum_{\tau=0}^{\infty}\beta^\tau\left(\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2\right)^\tau v.$$

Because $\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2$ is a transition matrix we know that

$$\left(\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2\right)^\tau v \to v^*$$

for some v^* as τ goes to infinity.[5] Therefore, we can approximate

$$\left(\mathbf{I} - \beta\left(\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2\right)\right)^{-1} v \approx \sum_{\tau=0}^{K-1} \beta^\tau \left(\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2\right)^\tau v + \\ + \frac{\beta^K}{1-\beta}\left(\widetilde{P}_J\widetilde{F}_J + \widetilde{P}_1\widetilde{F}_1 + \widetilde{P}_2\widetilde{F}_2\right)^K v,$$

which can be computed in $O(KX^2)$ operations. K can be taken small because we only need a reasonable approximation (and, as long as the exogenous states are not too persistent, it should mix fast.)

H.3 A Simple Example

We illustrate how the method performs in practice in a simple example. We consider the firm toy model in the absence of exogenous shocks w. In the numerical exercise, we take $s = 2, vp - fc = 1, ec = 3$, and $\beta = 0.95$. For these values, the baseline CCP is

$$p = \begin{pmatrix} \Pr\left(a = 1|k = 0\right) \\ \Pr\left(a = 1|k = 1\right) \end{pmatrix} = \begin{pmatrix} 0.65 \\ 0.83 \end{pmatrix}.$$

We assume the researcher correctly imposes the inequalities $0 \leq s \leq 5$, and the equality $\overline{\pi}_0(0) = 0$. So, the number of inequality restrictions is $m = 2$, while the number of equality restrictions is $d = 1$. Given these assumptions, the identified set Π^I is a one-dimensional set (since $X - d = 1$).

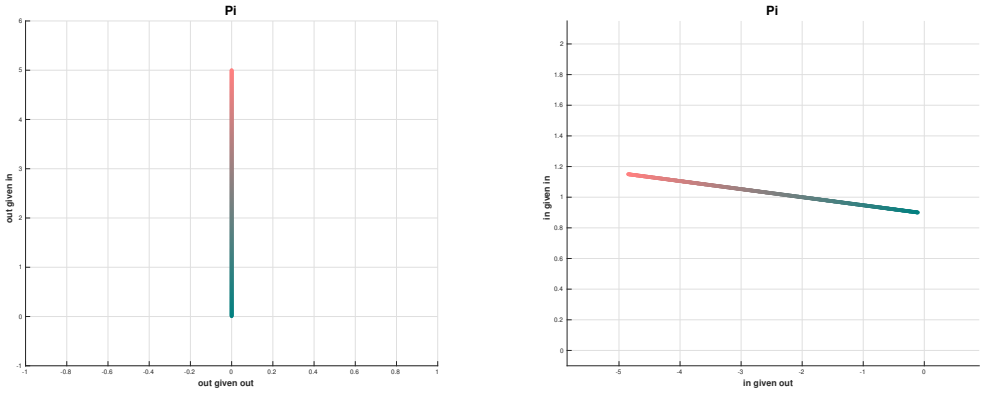
Figure H1 shows the identified set for π. Figure H1.a presents the set for π_0. (This set satisfies the equality and inequality constraints by construction.) Figure H1.b shows the set for π_1. The admissible values for π_1 are obtained from equation (2) in the main text (taking $J = 0$), and varies as scrap values range from 0 to 5 (corresponding to the colors in the figure going from dark green to red). The true π is the point $\pi_0 = (0, 2)'$ and $\pi_1 = (-2, 1)'$.

We consider a counterfactual entry subsidy that decreases entry cost e by 50%. The true counterfactual CCP is $\widetilde{p} = (0.74, 0.64)'$. Figure H2 depicts the identified set $\widetilde{\mathbf{P}}^I$. The counterfactual probability of entry in the identified set ranges from 0.68 to 0.84; and the counterfactual probability of staying in the market ranges from 0.42 to 0.78.[6]

Given the low dimension of this problem, we calculated the identified sets Π^I and $\widetilde{\mathbf{P}}^I$ using a grid search. Next, we explain how we solve the maximization problem (20)–(21) using the proposed algorithm described in Section H.1.

[5] Under the ℓ_1 norm, this convergence is a contraction and the contraction coefficient is known as Dobrushin ergodic coefficient.

[6] These numbers vary as scrap values go from 0 to 5 (corresponding, as before, to the colors in the figure going from dark green to red). As expected, when scrap values increase, the probability of staying in the market (in the y-axis) decreases, since it becomes more profitable to exit, while the probability of entry increases (in the x-axis), given that the firm anticipates greater earnings when exiting the market in the future.



(a) π_0 **(b)** π_1

Figure H1: Identified Set for the Payoffs, Π^I.

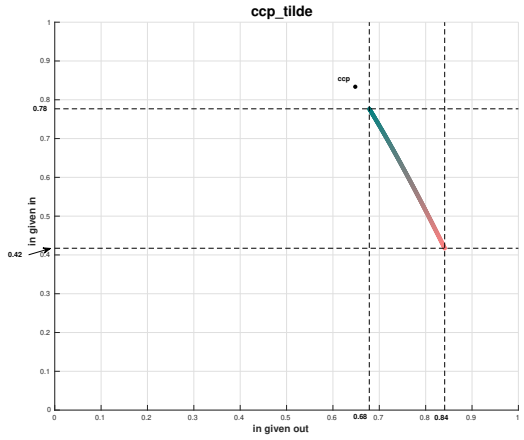


Figure H2: Identified Set for the Counterfactual CCP, $\widetilde{\mathbf{P}}^I$.

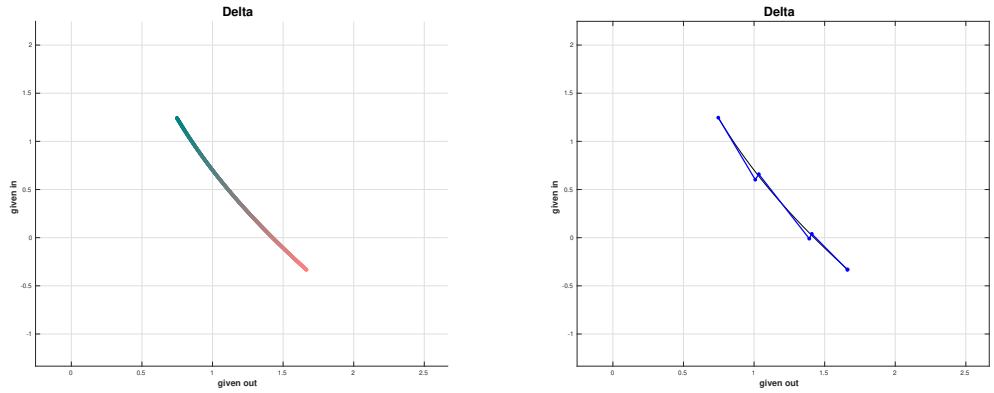
Consider the outcome of interest to be the counterfactual probability of entry, $\theta = \widetilde{\Pr}(a = 1|k = 0)$. We now use our proposed algorithm to compute the maximum probability of entry θ^U, which equals 0.84 in this example. Recall that we convert $\widetilde{p}$ into $\widetilde{\delta}$. The identified set for $\widetilde{\delta}$ is depicted in Figure H3.a. The (rescaled) parameter of interest is on the horizontal axis. The steps of our algorithm can be seen in Figure H3.b. Note that the direction

$$\Delta\widetilde{\delta} = \left(\frac{\partial\tilde{b}}{\partial\widetilde{\delta}}\right)^{-1} \widetilde{\mathbf{M}}\mathcal{H}\Delta\pi$$

is tangent to the identified set for $\widetilde{\delta}$. From an initial point, we move as much as we can in the direction $\Delta\widetilde{\delta}$ until we are too far from the identified set (that is until we are violating the nonlinear dynamic system (15) by more than tol). At this point, we do one step of the Newton-method for the nonlinear system, which moves $\widetilde{\delta}$ closer to the identified set. This is repeated until we reach the maximum.

This is more efficient than searching in the two-dimensional space for $\widetilde{p}$ and also more efficient than solving the nonlinear dynamic system (15) for each different test point for π (which would guarantee that

we never leave the blue line in Figure H3.b, but would likely cost more).



(a) Identified Set for $\widetilde{\delta}$ **(b)** Directions for $\widetilde{\delta}$

Figure H3: Identified Set for $\widetilde{\delta}$, and Directions of our Proposed Algorithm.

References

Kalouptsidi, M., L. Lima, and E. Souza-Rodrigues (2019): "On Estimating Counterfactuals Directly in Dynamic Models," Discussion paper, University of Toronto.

Kitamura, Y., and J. Stoye (2018): "Nonparametric Analysis of Random Utility Models," *Econometrica*, 86(6), 1883–1909.