

Implementing intersection bounds in Stata

Victor Chernozhukov
Wooyoung Kim
Sokbae Lee
Adam Rosen

The Institute for Fiscal Studies
Department of Economics, UCL

cemmap working paper CWP25/14

IMPLEMENTING INTERSECTION BOUNDS IN STATA

VICTOR CHERNOZHUKOV, WOYOUNG KIM, SOKBAE LEE, AND ADAM M. ROSEN

ABSTRACT. We present the `clrbound`, `clr2bound`, `clr3bound`, and `clrtest` commands for estimation and inference on intersection bounds as developed by [Chernozhukov et al. \(2013\)](#). The intersection bounds framework encompasses situations where a population parameter of interest is partially identified by a collection of consistently estimable upper and lower bounds. The identified set for the parameter is the intersection of regions defined by this collection of bounds. More generally, the methodology can be applied to settings where an estimable *function* of a vector-valued parameter is bounded from above and below, as is the case when the identified set is characterized by conditional moment inequalities.

The commands `clrbound`, `clr2bound`, and `clr3bound` provide bound estimates that can be used directly for estimation or to construct asymptotically valid confidence sets. `clrtest` performs an intersection bound test of the hypothesis that a collection of lower intersection bounds is no greater than zero. The command `clrbound` provides bound estimates for one-sided lower or upper intersection bounds on a parameter, while `clr2bound` and `clr3bound` provide two-sided bound estimates based on both lower and upper intersection bounds. `clr2bound` uses Bonferroni's inequality to construct two-sided bounds that can be used to perform asymptotically valid inference on the identified set or the parameter of interest, whereas `clr3bound` provides a generally tighter confidence interval for the parameter by inverting the hypothesis test performed by `clrtest`. More broadly, inversion of this test can also be used to construct confidence sets based on conditional moment inequalities as described in [Chernozhukov et al. \(2013\)](#). The commands include parametric, series, and local linear estimation procedures, and can be installed from within STATA by typing “`ssc install clrbound`”.

KEY WORDS: `clrbound`, `clr2bound`, `clr3bound`, `clrtest`, bound analysis, conditional moments, partial identification, infinite dimensional constraints, adaptive moment selection.

Date: May 2014.

We thank an anonymous referee for helpful comments and Eleni Aristodemou, Cathy Redmond, Koohyun Kwon, Jungsik Hyun and Hyunmin Park for capable research assistance. Financial support from the UK Economic and Social Research Council through a grant (RES-589-28-0001) to the ESRC Centre for Microdata Methods and Practice (CeMMAP) and through the funding of the “Programme Evaluation for Policy Analysis” node of the UK National Centre for Research Methods, and from the European Research Council (ERC) grant ERC-2009-StG-240910-ROMETA is gratefully acknowledged.

1. INTRODUCTION

In this paper we present the `clrbound`, `clr2bound`, `clr3bound`, and `clrtest` commands for estimation and inference on intersection bounds as developed by Chernozhukov et al. (2013). These commands, summarized in Table 1, enable one to perform hypothesis tests and construct set estimates and asymptotically valid confidence sets for parameters restricted by intersection bounds. The procedures employ parametric, series, and local linear estimators, and can be used to conduct inference on parameters restricted by conditional moment inequalities. The inference method developed by Chernozhukov et al. (2013) is based on sup-norm test statistics. There are a number of related papers in the literature that develop alternative methods for inference with conditional moment inequalities, such as Andrews and Shi (2013, 2014), Armstrong (2011a,b), Armstrong and Chan (2013), Chetverikov (2011), Lee et al. (2013a,b) and others.

Command	Description
<code>clrtest</code>	Test the hypothesis that the maximum of lower intersection bounds is nonpositive.
<code>clrbound</code>	Compute a one-sided bound estimate.
<code>clr2bound</code>	Compute two-sided bound estimates using Bonferroni's inequality.
<code>clr3bound</code>	Compute two-sided bound estimates by inverting <code>clrtest</code> .

TABLE 1. Intersection Bound Commands. Bound estimates can be used to construct asymptotically valid confidence intervals for parameters and identified sets restricted by intersection bounds.

Our software adds to a small but growing set of publicly available software for bound estimation and inference, including Beresteanu and Manski (2000a,b), and Beresteanu et al. (2010). Beresteanu and Manski (2000a,b) implement bound estimation by way of kernel regression for bounds derived in the analysis of treatment response as considered by Manski (1990), Manski (1997), and Manski and Pepper (2000), among others. Our software applies to a broader set of intersection bound problems, and in cases where both apply our software complements theirs by additionally providing parametric and series estimators, as well as methods for bias-correction and asymptotically valid inference. Beresteanu et al. (2010) can be used to replicate the results in Beresteanu and Molinari (2008), and more generally to compute consistent set estimates for best linear prediction (BLP) coefficients with interval-censored outcomes, as well as perform inference on any pair of elements of the BLP coefficient vector.

In Section 2 we recall the underlying framework of the intersection bounds setup from Chernozhukov et al. (2013). In Section 3 we describe the details of how our

STATA program conducts hypothesis tests and constructs bound estimates. In Section 4, we explain how to install our STATA module. In Sections 5, 6, 7, and 8 we describe the `clr2bound`, `clrbound`, `clrtest` and `clr3bound` commands, respectively. We explain how each command is used, what each command does, the available command options, and saved results. In Section 9 we illustrate the use of all four of these commands using data from the National Longitudinal Survey of Youth of 1979 (NLSY79), as in [Carneiro and Lee \(2009\)](#). Specifically, we use these commands to estimate and perform inference on returns to education using monotone treatment response and monotone instrumental variable bounds developed by [Manski and Pepper \(2000\)](#).

2. FRAMEWORK

We begin by considering a parameter of interest θ^* which is bounded above and below by intersection bounds of the form:

$$(1) \quad \max_{j \in \mathcal{J}_l} \sup_{x_j^l \in \mathcal{X}_j^l} \theta_j^l(x_j^l) \leq \theta^* \leq \min_{j \in \mathcal{J}_u} \inf_{x_j^u \in \mathcal{X}_j^u} \theta_j^u(x_j^u),$$

where $\{\theta_j^l(\cdot) : j \in \mathcal{J}_l\}$ and $\{\theta_j^u(\cdot) : j \in \mathcal{J}_u\}$ are consistently estimable lower and upper bounding functions. \mathcal{X}_j^l and \mathcal{X}_j^u are known sets of values for the arguments of these functions, and \mathcal{J}_l and \mathcal{J}_u are index sets with a finite number of positive integers. The interval of all values that lie within the bounds in (1) is the identified set, denoted:

$$(2) \quad \Theta_I \equiv [\theta_0^l, \theta_0^u],$$

where

$$\theta_0^l \equiv \max_{j \in \mathcal{J}_l} \sup_{x_j^l \in \mathcal{X}_j^l} \theta_j^l(x_j^l), \quad \theta_0^u \equiv \min_{j \in \mathcal{J}_u} \inf_{x_j^u \in \mathcal{X}_j^u} \theta_j^u(x_j^u).$$

We focus on the commonly encountered case where the bounding functions $\theta_j^l(\cdot)$ and $\theta_j^u(\cdot)$ are conditional expectation functions such that

$$\theta_j^k(\cdot) \equiv E[Y_j^k | X_j^k = \cdot], \quad k = l, u,$$

where Y_j^k and X_j^k are the dependent variable and explanatory variables of a conditional mean regression for each j and k , respectively. We allow for the possibility that the explanatory variables X_j^k are different or the same across j and k .

Many papers in the recent literature on partial identification feature bounds of the form given in (1) and (2) on a parameter of interest, or on a function of a parameter of interest. Characterization of the asymptotic distribution of plug-in estimators for these bounds is complicated by the fact that they are the infimum and supremum of an estimated function. Moreover, the use of sample analogs for bound estimates is well-known to produce substantial finite sample bias. The inferential methods of [Chernozhukov et al. \(2013\)](#) overcome these problems to produce asymptotically valid confidence sets for θ^* and for Θ_I , and bias-corrected estimates for the upper

and lower bounds of Θ_I . The main idea of our approach is to first form *precision-corrected* estimators for the bounding functions $\theta_j^k(\cdot)$ for each j and k , and then apply the *max*, *sup*, *min*, and *inf* operators to these precision-corrected estimators. The degree of the precision-correction is chosen in order to obtain bias-corrected bound estimates, or bound estimates that achieve asymptotically valid inference at a desired level. Chernozhukov et al. (2013) provide asymptotic theory for formal justification and algorithms for implementation of these methods. The software described in this paper implements these algorithms in STATA.¹

Chernozhukov et al. (2013) provide numerous examples of bound characterizations to which these methods apply. A leading example is given by the nonparametric bounds of Manski (1989, 1990) on mean treatment response and average treatment effects with instrumental variable restrictions. So called “worst-case” bounds on mean treatment response $\theta^* = \theta^*(x) \equiv E[Y(t)|X = x]$ from treatment $t \in \{0, 1\}$ conditional on vector $X = x$ are given by

$$(3) \quad \theta^l(x) \leq \theta^*(x) \leq \theta^u(x),$$

where

$$\theta^l(x) \equiv E[Y \cdot 1\{Z = t\}|X = x], \quad \theta^u(x) \equiv E[Y \cdot 1\{Z = t\} + 1\{Z \neq t\}|X = x].$$

Here $Z \in \{0, 1\}$ denotes the observed treatment and $Y(\cdot)$ maps potential treatments to outcomes, which are normalized to lie on the unit interval, $Y(\cdot) : \{0, 1\} \rightarrow [0, 1]$. The lack of point identification of $E[Y(t)|X = x]$ is a consequence of the fact that the observed outcome is $Y = Y(Z)$, and the potential outcome from the counterfactual treatment, $Y(1 - Z)$, is not observed. The width of the bounds is $P[Z \neq t]$, the probability that observed treatment Z differs from t .

Researchers are often willing to invoke instrumental variable restrictions, or level-set restrictions as in Manski (1990), that limit the degree to which the conditional expectation $E[Y(t)|X = x]$ varies with x . For instance, x may be comprised of two components $x = (w, v)$ with component v *excluded* from affecting the conditional mean function, so that

$$(4) \quad \forall v \in \mathcal{V}, \quad E[Y(t)|X = (w, v)] = E[Y(t)|W = w],$$

where \mathcal{V} denotes the support of V . Then, letting $\theta^*(w) := E[Y(t)|W = w]$, by virtue of (3) holding for $x = (w, v)$ for any fixed w and all $v \in \mathcal{V}$, it follows that

$$(5) \quad \sup_{v \in \mathcal{V}} \theta^l((w, v)) \leq \theta^*(w) \leq \inf_{v \in \mathcal{V}} \theta^u((w, v)),$$

¹The software is “CLRBOUND: STATA module to perform estimation and inference on intersection bounds” by Victor Chernozhukov, Wooyoung Kim, Sokbae Lee, and Adam M. Rosen, 2013, and is available at <http://econpapers.repec.org/software/bocbocode/s457674.htm>. All of our commands require the package *moremata* (Jann (2005)), available at <http://econpapers.repec.org/software/bocbocode/s455001.htm>.

which is precisely the form of (1) with singleton (and thus omitted) sets \mathcal{J}_l and \mathcal{J}_u , $\mathcal{X}^l = \mathcal{X}^u = \mathcal{V}$, and $\theta^* = \theta^*(w)$. This reasoning can be applied to obtain upper and lower bounds on $\theta^*(w)$ for all values of w . In Section 9 we illustrate the use of our STATA commands with bounds on a conditional expectation similar to those in (5) applied to data from the 1979 National Longitudinal Survey of Youth, but where a monotone instrumental variable restriction first considered by Manski and Pepper (2000) is employed instead of the instrumental variable restriction used above.

The estimation problem of Chernozhukov et al. (2013) is to obtain estimators $\hat{\theta}_{n0}^l(p)$ and $\hat{\theta}_{n0}^u(p)$ that provide bias-corrected estimates or the endpoints of confidence intervals depending on the chosen value of p , e.g. $p = 1/2$ for half-median-unbiased bound estimates or $p = 1 - \alpha$ for confidence intervals. By construction, these estimators satisfy

$$(6) \quad P_n \{ \theta_0^l \geq \hat{\theta}_{n0}^l(p) \} \geq p - o(1), \text{ and } P_n \{ \theta_0^u \leq \hat{\theta}_{n0}^u(p) \} \geq p - o(1).$$

Implementation details can be found in Chernozhukov et al. (2013), who focus on the upper bound for θ^* . As explained there, the estimation procedure can be easily adapted for the lower bound for θ^* . The command `clrbound` presented below gives estimators for these one-sided intersection bounds.

If one wishes to perform inference on the identified set, then one can use the intersection of upper and lower one-sided intervals each based on $\tilde{p} = (1 + p)/2$ as an asymptotic level- p confidence set $[\hat{\theta}_{n0}^l(\tilde{p}), \hat{\theta}_{n0}^u(\tilde{p})]$ for Θ_I satisfying

$$(7) \quad \liminf_{n \rightarrow \infty} P_n \{ \Theta_I \in [\hat{\theta}_{n0}^l(\tilde{p}), \hat{\theta}_{n0}^u(\tilde{p})] \} \geq p,$$

by (6) and Bonferroni's inequality. For example, to obtain a 95% confidence set for Θ_I , one can use upper and lower one-sided intervals each with 97.5% nominal coverage probability. The command `clr2bound` described in Section 5 provides this type of confidence interval.

Because $\theta^* \in \Theta_I$, such confidence intervals are also asymptotically valid but generally conservative for θ^* .² As an alternative, one may consider inference on θ^* by first transforming the collection of lower and upper bounds in (1) into a collection of only one-sided bounds on a function of θ^* . Specifically, the inequalities in (1) are equivalent to

$$(8) \quad T_0(\theta^*) \equiv \max_{k \in \{l, u\}} \max_{j \in \mathcal{J}_k} \sup_{x_j^k \in \mathcal{X}_j^k} T_{jk}(x_j^k, \theta^*) \leq 0,$$

where

$$(9) \quad T_{ju}(x_j^k, \theta^*) \equiv \theta^* - \theta_j^u(x_j^k), \quad T_{jl}(x_j^k, \theta^*) \equiv \theta_j^l(x_j^k) - \theta^*.$$

²Differences between confidence regions for an identified set Θ_I and a single point θ^* within that set have been well-studied in the prior literature. See for instance Imbens and Manski (2004), Chernozhukov et al. (2007), Stoye (2009), and Romano and Shaikh (2010).

For any conjectured value of θ^* , say θ_{null} , one can apply estimation methods from [Chernozhukov et al. \(2013\)](#) to perform the hypothesis test

$$(10) \quad H_0 : T_0(\theta_{null}) \leq 0 \text{ vs. } H_1 : T_0(\theta_{null}) > 0,$$

This is carried out by placing $T_0(\theta_{null})$ in the role of the bounding function $\theta_0^l(\cdot)$ in (1) to produce an estimator $\hat{T}_{n0}(\theta_{null}, p)$ such that

$$(11) \quad P_n \left\{ T_0(\theta_{null}) \geq \hat{T}_{n0}(\theta_{null}, p) \right\} \geq p - o(1),$$

analogously to the construction of $\hat{\theta}_{n0}^l(p)$ in (6). The null hypothesis H_0 is then rejected in favor of H_1 at the $1-p$ significance level if $\hat{T}_{n0}(\theta_{null}, p) > 0$. The command `clrtest` described in Section 7 performs such a test. By inverting this test, the set of θ_{null} such that $\hat{T}_{n0}(\theta_{null}, p) \leq 0$ is an asymptotically valid level p confidence set for θ^* since

$$(12) \quad \liminf_{n \rightarrow \infty} P_n \left\{ \theta^* \in \{ \theta_{null} : \hat{T}_{n0}(\theta_{null}, p) \leq 0 \} \right\} \geq p,$$

by construction. The command `clr3bound` described in Section 8 produces precisely this confidence set.

3. IMPLEMENTATION

In this section, we describe the details of our implementation for estimation of one-sided bounds. We focus on the lower intersection bounds and drop the l superscript to simplify notation.

Let J denote the number of inequalities concerned. Suppose that we have observations $\{(Y_{ji}, X_{ji}) : i = 1, \dots, n, j = 1, \dots, J\}$, where n is the sample size. For each $j = 1, \dots, J$, let \mathbf{y}_j denote the $n \times 1$ vector whose i th element is Y_{ji} and \mathbf{X}_j the $n \times d_j$ matrix whose i th row is X'_{ji} , where d_j is the dimension of X_{ji} . We allow multidimensional \mathbf{X}_j only for parametric estimation. We set $d_j = 1$ for series and local linear estimation.

To evaluate the supremum in (1) numerically, we set a dense set of grid points for each $j = 1, \dots, J$, say $\{\mathbf{x}_1, \dots, \mathbf{x}_J\}$, where $\mathbf{x}_j = (x'_{j1}, \dots, x'_{jM_j})'$ for some sufficiently large numbers M_j , $j = 1, \dots, J$, where each x_{jm} is a $d_j \times 1$ vector. Also, let Ψ_j denote the $M_j \times d_j$ matrix whose m th row is x'_{jm} , where $m = 1, \dots, M_j$ and $j = 1, \dots, J$. Note that the number of grid points can be different for different inequalities.

3.1. Parametric Estimation. Define

$$\mathbf{X} := \begin{pmatrix} \mathbf{X}_1 & \cdots & \mathbf{0} \\ & \vdots & \\ \mathbf{0} & \cdots & \mathbf{X}_J \end{pmatrix}, \quad \mathbf{y} := \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_J \end{pmatrix}, \quad \text{and} \quad \Psi := \begin{pmatrix} \Psi_1 & \cdots & \mathbf{0} \\ & \vdots & \\ \mathbf{0} & \cdots & \Psi_J \end{pmatrix}.$$

Let $\boldsymbol{\theta}_j(\mathbf{x}_j) \equiv (\theta_j(x_{j1}), \dots, \theta_j(x_{jM_j}))'$ and $\boldsymbol{\theta} \equiv (\boldsymbol{\theta}_1(\mathbf{x}_1)', \dots, \boldsymbol{\theta}_J(\mathbf{x}_J)')$. Then the estimator of $\boldsymbol{\theta}$ is $\widehat{\boldsymbol{\theta}} \equiv \boldsymbol{\Psi}\widehat{\boldsymbol{\beta}}$, where $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. Also, the heteroskedasticity-robust standard error of $\widehat{\boldsymbol{\theta}}$, say $\widehat{\mathbf{s}}$, can be computed as

$$\widehat{\mathbf{s}} \equiv \sqrt{\text{diag}_{\text{vec}}(\mathbf{V})},$$

where

$$\boldsymbol{\Omega} = [\text{diag}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})]^2, \quad \mathbf{V} = \boldsymbol{\Psi}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Omega}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\boldsymbol{\Psi}',$$

$\text{diag}(\mathbf{a})$ is the diagonal matrix whose diagonal terms are elements of the vector \mathbf{a} , and $\text{diag}_{\text{vec}}(\mathbf{A})$ is the vector whose elements are diagonal elements of the matrix \mathbf{A} .

A precision-corrected estimate is obtained by maximizing the precision-corrected curve, which is given by the function estimate minus critical value times the standard error. To compute the critical value, say $k(p)$, define

$$\widehat{\boldsymbol{\Sigma}} := [\text{diag}(\widehat{\mathbf{s}})]^{-1}\mathbf{V}[\text{diag}(\widehat{\mathbf{s}})]^{-1}.$$

Let $\text{chol}(\mathbf{A})$ denote the Cholesky decomposition of the matrix \mathbf{A} such that

$$\mathbf{A} = \text{chol}(\mathbf{A})\text{chol}(\mathbf{A})'.$$

We simulate pseudo random numbers from the $N(0, 1)$ distribution and construct a $\text{dim}(\widehat{\boldsymbol{\Sigma}}) \times R$ -dimensional matrix, say \mathbf{Z}_R . Then the critical value is selected as

$$(13) \quad k(p) = \text{the } p\text{th quantile of } \max_{\text{col.}}[\text{chol}(\widehat{\boldsymbol{\Sigma}})\mathbf{Z}_R],$$

where $\max_{\text{col.}}(\mathbf{B})$ is a set of maximum values in each column of the matrix \mathbf{B} . Then our bias-corrected estimator $\widehat{\theta}_{n0}(p)$ for $\max_{j \in \mathcal{J}_l} \sup_{x_j^l \in \mathcal{X}_j^l} \theta_j^l(x_j^l)$ is

$$(14) \quad \widehat{\theta}_{n0}(p) = \max_{\text{col.}}[\boldsymbol{\Psi}\widehat{\boldsymbol{\beta}} - k(p)\widehat{\mathbf{s}}].$$

The critical value in (14) is obtained under the least favorable case. To improve the estimator, we carry out the following adaptive inequality selection (AIS) procedure:

(Step 1) Set $\tilde{\gamma}_n \equiv 1 - .1/\log n$. Let ψ'_k denote the k th row of $\boldsymbol{\Psi}$, where $k = 1, \dots, \sum_{j=1}^J M_j$. Keep each row ψ'_k of $\boldsymbol{\Psi}$ if and only if

$$\psi'_k \widehat{\boldsymbol{\beta}} \geq \widehat{\theta}_{n0}(\tilde{\gamma}_n) - 2k(\tilde{\gamma}_n)\widehat{s}_k,$$

where \widehat{s}_k is the k th element of $\widehat{\mathbf{s}}$.

(Step 2) Replace $\boldsymbol{\Psi}$ with the kept rows of $\boldsymbol{\Psi}$ in Step 1. Then recompute \mathbf{V} and $\widehat{\boldsymbol{\Sigma}}$ to update the critical value in (13), and obtain the final estimate $\widehat{\theta}_{n0}(p)$ in (14) with the updated critical value.

3.2. Series Estimation. The implementation of series estimation is similar to parametric estimation. For each $j = 1, \dots, J$, let $p_{nj}(x) \equiv (p_{n,1}(x), \dots, p_{n,\kappa_j}(x))'$ denote the κ_j -dimensional vector of approximating functions by cubic B-splines. Here, the number of series terms κ_j can be different from one inequality to another. Let $\tilde{\mathbf{X}}_j$ denote the $n \times \kappa_j$ matrix whose i th row is $p_{nj}(X_{ji})'$ and $\tilde{\Psi}_j$ the $M_j \times \kappa_j$ matrix whose m th row is $p_{nj}(x_{jm})'$. Then the same procedure as described in Section 3.1 can be carried out, substituting $\tilde{\mathbf{X}}_j$ and $\tilde{\Psi}_j$ for \mathbf{X}_j and Ψ_j , respectively.

In this implementation, the dimension d_j of X_{ji} is one and the approximating functions are cubic B-splines. However, it is possible to implement high dimensional X_{ji} and other possible basis functions by programming a suitable design matrix manually and running our commands with an option of parametric estimation. This is basically equivalent to modifying $\tilde{\mathbf{X}}_j$ and $\tilde{\Psi}_j$ in series estimation. See Section 4.2 of Chernozhukov et al. (2013) for details.

3.3. Local Linear Estimation. For any vector \mathbf{v} , let $\hat{\rho}_j(\mathbf{v})$ denote the vector whose k th element is the local linear regression estimate of \mathbf{y}_j on \mathbf{X}_j at the k th element of \mathbf{v} . In detail, the k th element of $\hat{\rho}_j(\mathbf{v})$, say $\hat{\rho}_j(v_k)$, is defined as follows:

$$\hat{\rho}_j(v_k) \equiv \mathbf{e}'_1 (\mathbf{X}'_{v_k} \mathbf{W}_j \mathbf{X}_{v_k})^{-1} \mathbf{X}'_{v_k} \mathbf{W}_j \mathbf{y}_j,$$

where $\mathbf{e}_1 \equiv (1, 0)'$,

$$\mathbf{X}_{v_k} \equiv \begin{pmatrix} 1 & (X_{j1} - v_k) \\ \vdots & \vdots \\ 1 & (X_{jn} - v_k) \end{pmatrix}, \quad \mathbf{W}_j \equiv \text{diag} \left(K \left(\frac{X_{j1} - v_k}{h_j} \right), \dots, K \left(\frac{X_{jn} - v_k}{h_j} \right) \right),$$

$K(\cdot)$ is a kernel function, and h_j is the bandwidth for inequality j . Recall that the dimension d_j of X_{ji} is one in local linear estimation. In our implementation, we used the following kernel function:

$$K(s) = \frac{15}{16} (1 - s^2)^2 \mathbf{1}(|s| \leq 1).$$

Then the estimator of $\boldsymbol{\theta} \equiv (\boldsymbol{\theta}_1(\mathbf{x}_1)', \dots, \boldsymbol{\theta}_J(\mathbf{x}_J)')$ is $\hat{\boldsymbol{\theta}} \equiv (\hat{\rho}_1(\boldsymbol{\psi}_1)', \dots, \hat{\rho}_J(\boldsymbol{\psi}_J)')$, where $\boldsymbol{\psi}_j$ denotes the $M_j \times 1$ vector whose m th element is x_{jm} .

Now let \hat{s}_j denote the $M_j \times 1$ vector whose m th element is $\sqrt{\overline{g^2}_{jm}(\mathbf{y}_j, \mathbf{X}_j)}/nh_j$, where

$$\begin{aligned} \overline{g^2}_{jm}(\mathbf{y}_j, \mathbf{X}_j) &= n^{-1} \sum_{i=1}^n \hat{g}_{ji}(Y_{ji}, X_{ji}, x_{jm})^2, \\ \hat{g}_{ji}(Y_{ji}, X_{ji}, x_{jm}) &= \frac{Y_{ji} - \hat{\rho}_j(X_{ji})}{\sqrt{h_j \hat{f}_j(x_{jm})}} K \left(\frac{x_{jm} - X_{ji}}{h_j} \right), \end{aligned}$$

$\hat{f}_j(x_{jm})$ is the kernel estimate of the density of the covariate for the j th inequality, evaluated at x_{jm} . Then, $\hat{\mathbf{s}}$ can be computed as $\hat{\mathbf{s}} = (\hat{s}'_1, \dots, \hat{s}'_J)'$.

To compute the critical value, $k(p)$, let Φ_j denote the $M_j \times n$ matrix whose m th row is $(\hat{g}_{j1}(Y_{j1}, X_{j1}, x_{jm}), \dots, \hat{g}_{jn}(Y_{jn}, X_{jn}, x_{jm})) / \sqrt{nh_j g^2_{jm}(\mathbf{y}_j, \mathbf{X}_j)}$. Define

$$\Phi \equiv \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_J \end{pmatrix}.$$

We simulate pseudo random numbers from the $N(0, 1)$ distribution and construct a $n \times R$ matrix, \mathbf{Z}_R . Then, the critical value is selected as

$$(15) \quad k(p) = \text{the } p\text{th quantile of } \max_{\text{col.}}[\Phi \mathbf{Z}_R].$$

The calculation of the bias-corrected estimator $\hat{\theta}_{n0}(p)$ is almost the same as that of parametric estimation. That is,

$$(16) \quad \hat{\theta}_{n0}(p) = \max_{\text{col.}}[\hat{\boldsymbol{\theta}} - k(p)\hat{\mathbf{s}}].$$

However, the AIS procedure is slightly different since we do not use Ψ in local linear estimation.

(Step 1) Set $\tilde{\gamma}_n \equiv 1 - .1/\log n$. Keep the m th row of each Φ_j , $j = 1, \dots, J$, if and only if

$$\hat{\rho}_j(x_{jm}) \geq \hat{\theta}_{n0}(\tilde{\gamma}_n) - 2k(\tilde{\gamma}_n)\hat{s}_{jm},$$

where \hat{s}_{jm} is the m th element of $\hat{\mathbf{s}}_j$.

(Step 2) For $j = 1, \dots, J$, replace Φ_j with the kept rows of Φ_j in Step 1. Then recompute the critical value in (15), and obtain the final estimate $\hat{\theta}_{n0}(p)$ with the updated critical value.

4. INSTALLATION OF THE CLRBOUND PACKAGE

All STATA commands below are available at the Statistical Software Components (SSC) archive. Our STATA module called `clrbound` can be installed from within STATA by typing “`ssc install clrbound`”.³ All of our commands require the package `moremata` (Jann (2005))⁴, which can also be installed by typing “`ssc install moremata, replace`” in the STATA command window.

³<http://econpapers.repec.org/software/bocbocode/s457674.htm>.

⁴<http://econpapers.repec.org/software/bocbocode/s455001.htm>.

5. THE CLR2BOUND COMMAND

5.1. **Syntax.** The syntax of `clr2bound` is as follows:

```
clr2bound (( lowerdepvar1 indepvars1 range1) ( lowerdepvar2 indepvars2 range2) ... (
  lowerdepvarN indepvarsN rangeN)) (( upperdepvarN+1 indepvarsN+1 rangeN+1) (
  upperdepvarN+2 indepvarsN+2 rangeN+2) ... ( upperdepvarN+M indepvarsN+M
  rangeN+M)) [ if ] [ in ] [ , method("series"|"local") notest null(real) level(numlist
) noais minsmooth(#) maxsmooth(#) noundersmooth bandwidth(numlist) rnd(#)
norseed seed(#) ]
```

5.2. **Description.** `clr2bound` estimates a two-sided confidence interval $[\hat{\theta}_{n0}^l(\tilde{p}), \hat{\theta}_{n0}^u(\tilde{p})]$ where $\tilde{p} = \frac{\text{level}+1}{2}$. By (6) and Bonferroni's inequality this interval contains the identified set Θ_I with probability at least `level` asymptotically, i.e. such that (7) holds with $p = \text{level}$. The variables $lowerdepvar1 \sim lowerdepvarN$ are the dependent variables (Y_j^l 's) for the lower bounding functions and the $upperdepvarN+1 \sim upperdepvarN+M$ are the dependent variables (Y_j^u 's) for the upper bounding functions, respectively. The variables $indepvars1 \sim indepvarsN+M$ are explanatory variables for the corresponding dependent variables. Recall that `clr2bound` allows for multidimensional *indepvars* for parametric estimation, but only for a one dimensional independent variable for series and local linear estimation.

The variables $range1 \sim rangeN+M$ are sets of grid points over which the bounding function is estimated, corresponding to the sets \mathcal{X}_j^l and \mathcal{X}_j^u in (1). The number of observations for the *range* is not necessarily the same as the number of observations for the *depvar* and *indepvars*. The latter is the sample size, whereas the former is the number of grid points to evaluate the maximum or minimum values of the bounding functions.

It should be noted that the parentheses must be used properly. Variables for lower bounds and upper bounds must be put in additional parentheses separately. For example, if there are two variable sets, $(ldepvar1 indepvars1 range1)$ and $(ldepvar2 indepvars2 range2)$, for the lower bounds estimation and one variable set, $(udepvar1 indepvars3 range3)$, for the upper bounds estimation, the right syntax for two-sided intersection bounds estimation is $((ldepvar1 indepvars1 range1)(ldepvar2 indepvars2 range2))((udepvar1 indepvars3 range3))$.

In addition, `clr2bound` provides a test result for the null hypothesis that the specified value is in the intersection bounds for each confidence level. If the value is left unspecified, the null hypothesis is that the parameter of interest is 0. This test uses (11) which is a more stringent requirement than simply checking whether the value lies within the confidence set reported by `clr2bound`, which is based on Bonferroni's inequality. Therefore, this test may reject some values in the reported confidence set at the same confidence level.

- 5.3. Options.** `method(string)` specifies the method of estimation. By default, `clr2bound` will conduct parametric estimation. Specifying `method("series")` or `method("local")`, it will conduct series estimation with cubic B-splines or local linear estimation, respectively.
- `notest` determines whether `clr2bound` conducts a test or not. `clr2bound` provides a test for the null hypothesis that the specified value is in the intersection bounds at the confidence levels specified in the `level` option below. By default, `clr2bound` conducts the test. Specifying this option causes `clr2bound` to output Bonferroni bounds only.
- `null(real)` specifies the value for θ^* under the null hypothesis of the test we described above. The default value is `null(0)`.
- `level(numlist)` specifies confidence levels. `numlist` has to be filled with real numbers between 0 and 1. In particular, if this option is specified as `level(0.5)`, the result is the half-median-unbiased estimator of the parameter of interest. The default is `level(0.5 0.9 0.95 0.99)`.
- `noais` determines whether adaptive inequality selection is used. Adaptive inequality selection (AIS) helps to get sharper bounds by using a problem-dependent cutoff to drop irrelevant grid points of the `range`. The default is to use AIS.
- `minsmooth(#)` and `maxsmooth(#)` specify the minimum and maximum possible numbers of approximating functions considered in the cross validation procedure for B-splines. Specifically, the number of approximating functions \hat{K}_{cv} is set to the minimizer of the leave-one-out least squares cross validation score within this range. For example, if a user inputs `minsmooth(5)` and `maxsmooth(9)`, \hat{K}_{cv} is chosen from the set $\{5,6,7,8,9\}$. The procedure calculates this number separately for each inequality. The default is `minsmooth(5)` and `maxsmooth(20)`. If under-smoothing is performed, the number of approximating functions K ultimately used will be given by the largest integer smaller than \hat{K}_{cv} times the under-smoothing factor $n^{-1/5} \times n^{2/7}$, see option `nundersmooth` below. This option is only available for series estimation.
- `bandwidth(#)` specifies the value of the bandwidth used in local linear estimation. By default, `clr2bound` calculates a bandwidth for each inequality. With under-smoothing, we use the rule of thumb bandwidth $h = \hat{h}_{ROT} \times \hat{s}_v \times n^{1/5} \times n^{-2/7}$ where \hat{s}_v is the square root of the sample variance of V , and \hat{h}_{ROT} is the rule-of-thumb bandwidth for estimation of $\theta(v)$ with studentized V . See [Chernozhukov et al. \(2013\)](#) for the exact form of \hat{h}_{ROT} . When the `bandwidth(#)` is specified, `clr2bound` uses the given bandwidth as the global bandwidth for every inequality. This option is only available for local linear estimation.
- `nundersmooth` determines whether under-smoothing is carried out, with the default being to under-smooth. In series estimation, under-smoothing is implemented by first computing \hat{K}_{cv} as the minimizer of the leave-one-out least squares cross validation score. We then set the number of approximating functions to K , given

by the largest integer which is smaller than or equal to $\hat{K} := \hat{K}_{cv} \times n^{-1/5} \times n^{2/7}$. The `nundersmooth` option simply uses \hat{K}_{cv} . For local linear estimation under-smoothing is done by setting the bandwidth to $h = \hat{h}_{ROT} \times \hat{s}_v \times n^{1/5} \times n^{-2/7}$, where \hat{h}_{ROT} , is the rule-of-thumb bandwidth used in Chernozhukov et al. (2013). The `nundersmooth` option instead uses $\hat{h}_{ROT} \times \hat{s}_v$. This option is only available for series and local linear estimation.

`rnd(#)` specifies the number of columns of the random matrix generated from the standard normal distribution. This matrix is used for computation of critical values. For example, if the number is 10000 and the level is 0.95, we choose the 0.95 quantile from 10000 randomly generated elements. The default is `rnd(10000)`.

`norseed` determines whether the seed number for the simulation used in the calculation will be reset. For example, if a user wants to use this command for simulations carried out as part of a Monte Carlo study, this command can be used to prevent resetting the seed number in each Monte Carlo iteration. The default is to reset the seed number.

`seed(#)` specifies the seed number for the random number generation described above. To prevent the estimation result from changing one particular value to another randomly, `clr2bound` always conducts `set seed #` initially. The default is `seed(0)`.

5.4. Saved results. In the following, “l.b.e.” stands for lower bound estimation, “u.b.e.” for upper bound estimation, and “ineq” means an inequality. (i) denotes the i-th inequality. (lev) means the confidence level’s decimal part. For example, when the confidence level is 97.5% or 0.975, (lev) is 975. The number of elements in (lev) is equal to the number of confidence levels specified by the `level` option. Some results are only available for series or local linear estimation.

`clr2bound` saves the following in `e()`. Note that for this and all other commands, in the saved AIS results 1 is used to denote values that were kept in the index set, and 0 values that were dropped.

Scalars			
<code>e(N)</code>	#of observations	<code>e(null)</code>	The null hypothesis
<code>e(l_ineq)</code>	#of ineq's in l.b.e.	<code>e(u_ineq)</code>	#of ineq's in r.b.e.
<code>e(l_grid(i))</code>	#of grid points in (i) of l.b.e.	<code>e(u_grid(i))</code>	#of grid points in (i) of r.b.e.
<code>e(l_nf_x(i))</code>	#of approx. functions for l.b.e. at $x(i)$	<code>e(u_nf_x(i))</code>	#of approx. functions for u.b.e. at $x(i)$
<code>e(l_bdwh(i))</code>	bandwidth for (i) of l.b.e.	<code>e(l_bdwh(i))</code>	bandwidth for (i) of u.b.e.
<code>e(lbd(lev))</code>	est. results of l.b.e.	<code>e(ubd(lev))</code>	est. results of u.b.e.
<code>e(lcl(lev))</code>	critical value of l.b.e.	<code>e(uc1(lev))</code>	critical value of l.b.e.
<code>e(t_det(lev))</code>	1 : in the bound 0 : not	<code>e(t_cv1(lev))</code>	critical value of test
<code>e(t_bd(lev))</code>	est. results of test	<code>e(t_nf_x(i))</code>	#of approx. functions in test
Macros			
<code>e(cmd)</code>	"clr2bound"	<code>e(title)</code>	"CLR Intersection Bounds (method)"
<code>e(ldepvar)</code>	dep. var. in l.b.e.	<code>e(udepvar)</code>	dep. var. in r.b.e.
<code>e(level)</code>	confidence levels	<code>e(smoothing)</code>	"(NOT) Undersmoothed"
<code>e(l_indep(i))</code>	indep. var. in (i) of l.b.e.	<code>e(u_indep(i))</code>	indep. var. in (i) of u.b.e.
<code>e(l_range(i))</code>	range in (i) of l.b.e.	<code>e(u_range(i))</code>	range in (i) of u.b.e.
Matrices			
<code>e(l_omega)</code>	$\hat{\Omega}_n$ for l.b.e.	<code>e(u_omega)</code>	$\hat{\Omega}_n$ for u.b.e.
<code>e(l_theta(i))</code>	$\hat{\theta}_n(v)$ for each v in l.b.e.	<code>e(u_theta(i))</code>	$\hat{\theta}_n(v)$ for each v in u.b.e.
<code>e(l_se(i))</code>	$s_n(v)$ for each v in l.b.e.	<code>e(u_se(i))</code>	$s_n(v)$ for each v in u.b.e.
<code>e(l_ais(i))</code>	AIS result for each v in l.b.e.	<code>e(u_ais(i))</code>	AIS result for each v in u.b.e.
<code>e(t_omega)</code>	$\hat{\Omega}_n$ for test	<code>e(t_theta(i))</code>	$\hat{\theta}_n(v)$ for each v in test
<code>e(t_se(i))</code>	$s_n(v)$ for each v in test	<code>e(t_ais(i))</code>	AIS result for each v in test

See [Chernozhukov et al. \(2013\)](#) for details on $\hat{\theta}_n(v)$, $s_n(v)$, and $\hat{\Omega}_n$.

6. THE CLRBOUND COMMAND

6.1. **Syntax.** The syntax of `clrbound` is as follows:

```

clrbound ( depvar1 indepvars1 range1 ) ( depvar2 indepvars2 range2 ) ... ( depvarN
indepvarsN rangeN ) [ if ] [ in ] [ , lower | upper method("series"|"local")
level(numlist) noais minsmooth(#) maxsmooth(#) noundersmooth bandwidth(numlist)
rnd(#) norseed seed(#) ]

```

6.2. **Description.** `clrbound` estimates one-sided lower or upper intersection bounds on parameter θ^* , as specified by the user. Lower bound estimates $\hat{\theta}_{n0}^l(p)$ and upper bound estimates $\hat{\theta}_{n0}^u(p)$ are constructed to satisfy (6) for p set equal to `level`. The variables are defined similarly as for `clr2bound`.

6.3. **Options.** `lower` specifies whether the estimation is for the lower bound or the upper bound. By default, it will return the upper intersection bound. Specifying `lower`, `clrbound` will return the lower intersection bound.

Other options of the `clrbound` are the same as those of the `clr2bound`. However, the `clrbound` command does not have the `notest` and `null` options because it does not explicitly conduct a test.

6.4. **Saved results.** In the following, we use the same abbreviations as in Section 5.4. The `clrbound` saves the following in `e()`:

Scalars			
<code>e(N)</code>	#of observations	<code>e(n_ineq)</code>	#of inequality
<code>e(grid(i))</code>	#of grids points (i)	<code>e(nf_x(i))</code>	#of approx. functions in (i)
<code>e(bd(lev))</code>	results of estimation	<code>e(cl(lev))</code>	critical value
<code>e(bdwh(i))</code>	bandwidth for (i)		
Macros			
<code>e(cmd)</code>	"clrbound"	<code>e(title)</code>	"CLR Intersection (upper/lower) Bounds (<i>method</i>)"
<code>e(depvar)</code>	dependent variables	<code>e(level)</code>	confidence levels
<code>e(smoothing)</code>	"(NOT) Undersmoothed"	<code>e(indep(i))</code>	indep. variables in (i)
<code>e(range(i))</code>	range in (i)		
Matrices			
<code>e(omega)</code>	$\hat{\Omega}_n$	<code>e(theta(i))</code>	$\hat{\theta}_n(v)$ for each v
<code>e(se(i))</code>	$s_n(v)$ for each v	<code>e(ais(i))</code>	AIS result for each v

7. THE CLRTEST COMMAND

7.1. **Syntax.** The syntax of `clrtest` is as follows:

```
clrtest ( depvar1 indepvars1 range1 ) ( depvar2 indepvars2 range2 ) ... ( depvarN
  indepvarsN rangeN ) [ if ] [ in ] [ , method("series"|"local") level(numlist) noais
  minsmooth(#) maxsmooth(#) nundersmooth bandwidth(numlist) rnd(#) norseed
  seed(#) ]
```

7.2. **Description.** Variables are defined similarly as for the `clr2bound` command, but `clrtest` offers a more refined testing procedure. It performs the lower intersection bound test described in (10) using the given *depvars* and *indepvars* as dependent and independent variables, respectively. For example, suppose that one wants to test the null hypothesis that 0.59 is in the interval $[\theta_0^l, \theta_0^u]$ at the 5% level, where $\theta_0^l \equiv \sup_{x^l \in \mathcal{X}^l} E[Y^l | X^l = x]$ and $\theta_0^u \equiv \inf_{x^u \in \mathcal{X}^u} E[Y^u | X^u = x]$.

Suppose the variables Y^l, Y^u, X^l, X^u are coded as `y1`, `yu`, `x1`, and `xu`, respectively. To test this hypothesis one first creates the variables `y1_test = y1 - 0.59` and `yu_test = 0.59 - yu`, and then executes the command

```
clrtest (y1_test x1) (yu_test xu), level(0.95).
```

The `level` 0.95 corresponds to the value of p used for the intersection bound estimate described in (11) required to perform the test (10) at the $1 - p$ significance level. We illustrate the use of this command in Section 9.3.

7.3. **Options.** Since the options for `clrtest` are the same as those for `clrbound`, the explanation of options is omitted.

7.4. **Saved results.** Other saved results are the same as those of `clrbound` except the following:

Scalars	
<code>e(det(lev))</code>	rejected : 0, not rejected : 1

8. THE CLR3BOUND COMMAND

8.1. **Syntax.** The syntax of `clr3bound` is as follows:

```
clr3bound ((lowerdepvar1 indepvars1 range1) (lowerdepvar2 indepvars2 range2) ...
(lowerdepvarN indepvarsN rangeN)) ((upperdepvarN+1 indepvarsN+1 rangeN+1)
(upperdepvarN+2 indepvarsN+2 rangeN+2) ... (upperdepvarN+M indepvarsN+M
rangeN+M)) [if] [in] [, start(#) end(#) grid(#) method("series"|"local")
level(#) noais minsmooth(#) maxsmooth(#) nundersmooth bandwidth(#) rnd(#)
norseed seed(#) ]
```

8.2. **Description.** `clr3bound` estimates a two-sided confidence interval for the parameter θ^* by inverting the test (10) performed by the `clrtest` command. The end result is a collection of values of θ_{null} that estimate a confidence set for θ^* with asymptotic coverage `level` as described by (12). Note that when only one-sided intersection bounds are used, there is no need to implement the pointwise test carried out by `clr3bound`.

Since this command is only relevant for two-sided intersection bounds, users should input variables for both lower and upper bounds to calculate the bound. The variables are defined similarly as for `clr2bound`. This command generally provides tighter bounds than those provided by `clr2bound`, which employs Bonferroni's inequality to produce confidence sets for Θ_I . Unlike the previous commands, `clr3bound` can only deal with one confidence level at a time. It takes longer to compute bounds using the `clr3bound` command than the `clr2bound` command since `clr3bound` is implemented by repeating the `clrtest` command on a grid. In practice, we recommend using `clr2bound` to obtain initial bound estimates and confidence sets, and then using `clr3bound` to produce tighter bound estimates for the desired confidence level.

8.3. **Options.** `stepsize(#)` specifies the distance between two consecutive grid points. The procedure divides the Bonferroni-based confidence set produced by `clr2bound` into an equi-spaced grid and implements the `clrtest` command for each grid point to determine a possible tighter bound. The default is 0.01.

`level(#)` specifies the confidence level of the estimation. In contrast to previous commands, `clr3bound` can only deal with one confidence level at a time. The default is 0.95.

Other options are exactly the same as those of `clr2bound`.

8.4. **Saved results.** `clr3bound` saves the following in `e()`:

Scalars			
<code>e(N)</code>	#of observations	<code>e(step)</code>	step size
<code>e(level)</code>	confidence level		
<code>e(l_ineq)</code>	#of ineq's in l.b.e.	<code>e(u_ineq)</code>	#of ineq's in r.b.e.
<code>e(l_grid(i))</code>	#of grids points for l.b.e. observation (i)	<code>ate(u_grid(i))</code>	#of grids points for r.b.e. at observation (i)
<code>e(l_nf_x(i))</code>	#of approx. functions in (i) l.b.e.	<code>ofe(u_nf_x(i))</code>	#of approx. functions in (i) of u.b.e.
<code>e(l_bdwh(i))</code>	bandwidth for (i) of l.b.e.	<code>e(l_bdwh(i))</code>	bandwidth for (i) of u.b.e.
<code>e(lbd)</code>	est. results of l.b.e.	<code>e(ubd)</code>	est. results of u.b.e.
<code>e(lbd(lev))</code>	Bonferroni results of l.b.e.	<code>e(ubd(lev))</code>	Bonferroni results of u.b.e.
<code>e(lcl(lev))</code>	critical value of l.b.e.	<code>e(ucl(lev))</code>	critical value of u.b.e.
Macros			
<code>e(cmd)</code>	" <code>clr3bound</code> "	<code>e(title)</code>	"CLR Intersection Bounds: inverting test bounds"
<code>e(ldepvar)</code>	dep. var. in l.b.e.	<code>e(udepvar)</code>	dep. var. in r.b.e.
<code>e(method)</code>	Estimation method	<code>e(smoothing)</code>	"(NOT) Undersmoothed"
<code>e(l_indep(i))</code>	indep. var. in (i) of l.b.e.	<code>e(u_indep(i))</code>	indep. var. in (i) of u.b.e.
<code>e(l_range(i))</code>	range in (i) of l.b.e.	<code>e(u_range(i))</code>	range in (i) of u.b.e.
Matrices			
<code>e(l_omega)</code>	$\hat{\Omega}_n$ for l.b.e.	<code>e(u_omega)</code>	$\hat{\Omega}_n$ for u.b.e.
<code>e(l_theta(i))</code>	$\hat{\theta}_n(v)$ for each v in l.b.e.	<code>e(u_theta(i))</code>	$\hat{\theta}_n(v)$ for each v in u.b.e.
<code>e(l_se(i))</code>	$s_n(v)$ for each v in l.b.e.	<code>e(u_se(i))</code>	$s_n(v)$ for each v in u.b.e.
<code>e(l_ais(i))</code>	AIS result for each v in l.b.e.	<code>e(u_ais(i))</code>	AIS result for each v in u.b.e.

9. EXAMPLES

To illustrate the use of `clrbound`, `clr2bound`, `clr3bound`, and `clrtest`, we consider some examples based on joint monotone instrumental variable and monotone treatment response (MIV-MTR) bounds of Manski and Pepper (2000, Proposition 2), as in Chernozhukov et al. (2013) to study log wages as a function of years of schooling. We use the same data extract as Carneiro and Lee (2009) from the National Longitudinal Survey of Youth of 1979 (NLSY79).⁵ See also Carneiro et al. (2011) for the dataset and recent advances in estimating returns to schooling.

The data constitute a random sample of observations of white males born between 1957 and 1964. For each individual i we observe hourly wages in U.S. dollars in 1994, years of schooling (`eduyr`), and Armed Forces Qualifying Test score (`afqt`).⁶ We focus attention on potential outcome $Y_i(t)$, which denotes the logarithm of hourly wages (`lnwage`) in U.S. dollars in 1994 as a function of years of schooling t for individual i . V_i is the AFQT score, a measure of cognitive ability, studentized to have mean zero and variance one in the NLSY population. Let Z_i denote the realized treatment, here realized years of schooling, possibly self-selected by individuals. The source of the identification problem is the same as that of the example considered in Section 2, namely that for each individual i , we only observe $Y_i \equiv Y_i(Z_i)$ along with (Z_i, V_i) , but not $Y_i(t)$ with $t \neq Z_i$.

⁵Accompanying STATA `.dta`, `.do`, and `.log` files for these examples are available at [http://www.homepages.ucl.ac.uk/~sim\\$uctpar/IboundsFiles/NLSY.dta](http://www.homepages.ucl.ac.uk/~sim$uctpar/IboundsFiles/NLSY.dta), [http://www.homepages.ucl.ac.uk/~sim\\$uctpar/IboundsFiles/example.do](http://www.homepages.ucl.ac.uk/~sim$uctpar/IboundsFiles/example.do), and [http://www.homepages.ucl.ac.uk/~sim\\$uctpar/IboundsFiles/example.log](http://www.homepages.ucl.ac.uk/~sim$uctpar/IboundsFiles/example.log), respectively.

⁶See Carneiro and Lee (2009) for further details about the data.

The monotone instrumental variable (MIV) assumption introduced by [Manski and Pepper \(2000\)](#), asserts that for all treatment levels t , the conditional expectation $E[Y_i(t)|V_i = v]$ is weakly increasing in v . Thus, expected wages conditional on AFQT score are assumed to be increasing in the score, a reasonable assumption given the interpretation of the AFQT score as a measure of cognitive ability. The monotone treatment response (MTR) assumption asserts that each individual's log wage function $Y_i(t)$ is increasing in the level of schooling t . Without further restrictions, such as a parametric functional form for log wages $Y_i(t)$, or instrumental variable restrictions (stronger than the MIV restriction), expected returns to schooling are in general not point-identified, but can be bounded. For the purpose of illustration we condition on the average AFQT score $V_i = 0$, but identical analysis can be carried out conditioning on other values.

From [Manski and Pepper \(2000\)](#) Proposition 2, the MIV-MTR assumptions imply the following bounds on expected log wage at a given level of schooling t and conditional on AFQT score v :

$$(17) \quad \sup_{u \leq v} E[Y_i^l | V_i = u] \leq E[Y_i(t) | V_i = v] \leq \inf_{u \geq v} E[Y_i^u | V_i = u],$$

where

$$(18) \quad Y_i^l \equiv Y_i \cdot 1\{t \geq Z_i\} + y_0 \cdot 1\{t < Z_i\}, \quad Y_i^u \equiv Y_i \cdot 1\{t \leq Z_i\} + y_1 \cdot 1\{t > Z_i\},$$

and where $[y_0, y_1]$ is the support of Y_i . Thus we have the bounds of (1) with bound-generating functions $\theta^l(v) = E[Y_i^l | V_i = v]$ and $\theta^u(v) = E[Y_i^u | V_i = v]$ with intersection sets $\mathcal{V}^l = (-\infty, v]$ for the lower bound and $\mathcal{V}^u = [v, \infty)$ for the upper bound.

The MIV-MTR bounds are uninformative if the support of Y is unbounded. To avoid this issue, for the sake of illustration we take the parameter of interest to be

$$\theta^* = P[Y_i(t) > y | V_i = v],$$

at $y = \log(16)$, where \$16 is approximately the 70th percentile of hourly wages in the data, $v = 0$ and $t = 13$ (college attendees with one more year of schooling than high school graduates). Thus our goal will be to perform inference on θ^* , the probability that the hourly wage obtained by a college attendee ($t = 13$) is greater than 16 dollars conditional on having an AFQT score at the average level in the NLSY population.

Under the MIV restriction that $P[\{Y_i(t) > y\} | V_i = v]$ is weakly increasing in v and the same MTR assumption as above, the MIV-MTR upper bound is

$$(19) \quad \theta^* \leq \inf_{u \geq v} E[1\{Y_i > y\} \cdot 1\{t \leq Z_i\} + 1\{t > Z_i\} | V_i = u],$$

and the lower bound is

$$(20) \quad \theta^* \geq \sup_{u \leq v} E[1\{Y_i > y\} \cdot 1\{t \geq Z_i\} | V_i = u].$$

Indeed, the derivation of these bounds is identical to that of the conditional expectation bound (17) with the indicator function $1\{Y_i(t) > y\}$ in place of $Y_i(t)$ in the

conditional expectation $E[Y_i(t)|V_i = v]$. We focus on the threshold $y = \log(16)$ for the sake of illustration, but such bounds can be studied for any level of log wages y of interest, or indeed conditional on any value of v and any desired level of schooling t .⁷

Note that for the study of the joint MIV-MTR bounds, care must be exercised when setting the *range* variable, which was described in Section 5.2. This variable provides grids of values representing the sets \mathcal{X}^u and \mathcal{X}^l appearing in (1).⁸ In this example these sets differ from one another, as it follows from (19) and (20) above that \mathcal{X}^u are all possible values of V_i of at least v and \mathcal{X}^l are all possible values of V_i no more than v . Since we focus on the value of $\theta^* = E[Y_i(t)|V_i = v]$ at $v = 0$, a new variable which contains grid points larger (smaller) than 0 should be used for upper (lower) bound estimation. To obtain bounds conditional on other values of v , the *range* would need to be changed accordingly. As *range* variables for our NLSY79 dataset, we used `vl_afqt` for the lower bound, and `vu_afqt` for the upper bound, which each contain 101 grid points from -2 to 0 and 0 to 2, respectively. The commands we used for making the *range* variables were:

```
. egen vl_afqt = fill("-2 -1.98")
. replace vl_afqt = . if vl_afqt > 0
(1943 real changes made, 1943 to missing)
. egen vu_afqt = fill("0 0.02")
. replace vu_afqt = . if vu_afqt > 2
(1943 real changes made, 1943 to missing)
```

9.1. clr2bound. The first step is to create the dependent variables. For example, when calculating the MIV-MTR upper bound, we need to define the dependent variable as $Y_i^u = 1\{Y_i > y\} \cdot 1\{t \leq Z_i\} + 1\{t > Z_i\}$. In our example, we let `y1` denote the dependent variable for lower bound estimation and `yu` for the upper bound. The commands for constructing these variables were:

```
. gen y1 = (lnwage > log(16)) * (eduyr <= 13)
. gen yu = (lnwage > log(16)) * (eduyr >= 13) + (eduyr < 13)
```

Here we show how to use the three estimation methods (parametric, local linear, and series estimation). For the sake of illustration we also include the test result for whether or not 0.1 is in the two-sided intersection bounds using series estimation. The results were:

⁷Under the stronger assumption that the distribution of $Y_i(t)|V_i = v$ is stochastically increasing in v so that $P[\{Y_i(t) > y\}|V_i = v]$ is weakly increasing in v for all y , the MIV-MTR bounds can be applied at every y to bound the entire conditional distribution of $Y_i(t)$ given $V_i = v$.

⁸Note that in this example the sets \mathcal{J}_u and \mathcal{J}_l appearing in (1) are both singletons. The subscript j on these sets is thus superfluous and has been dropped.

```
. use NLSY, clear

. clr2bound ((yl afqt vl_afqt))((yu afqt vu_afqt)), notest

CLR Intersection Bounds (Parametric)                Number of obs : 2044
```

```
< Lower Side >
Inequality #1 : yl (# of Grid Points : 101, Independent Variables : afqt )
< Upper Side >
Inequality #1 : yu (# of Grid Points : 101, Independent Variables : afqt )
```

AIS(adaptive inequality selection) is applied

Bonferroni Bounds	Value
50% two-sided confidence interval	[0.1282908, 0.5663461]
90% two-sided confidence interval	[0.1142149, 0.5879820]
95% two-sided confidence interval	[0.1099835, 0.5947234]
99% two-sided confidence interval	[0.1008918, 0.6064604]

```
. clr2bound ((yl afqt vl_afqt))((yu afqt vu_afqt)), notest met("local")
```

```
CLR Intersection Bounds (Local Linear)                Number of obs : 2044
```

```
< Lower Side >
Inequality #1 : yl (# of Grid Points : 101, Independent Variables : afqt )
< Upper Side >
Inequality #1 : yu (# of Grid Points : 101, Independent Variables : afqt )
```

AIS(adaptive inequality selection) is applied
Bandwidths are undersmoothed

Bonferroni Bounds	Value
50% two-sided confidence interval	[0.1324061, 0.6406517]
90% two-sided confidence interval	[0.1182558, 0.6593739]
95% two-sided confidence interval	[0.1135008, 0.6656595]
99% two-sided confidence interval	[0.1043056, 0.6782933]

```
. clr2bound ((yl afqt vl_afqt))((yu afqt vu_afqt)), notest met("series")
```

```
CLR Intersection Bounds (Series)                Number of obs : 2044
Estimation Method : Cubic B-Spline (Undersmoothed)
```

```
< Lower Side >
-----
Inequality #1 : yl (# of Grid Points : 101, Independent Variables : afqt )
Numbers of Approximating Functions : 21
< Upper Side >
-----
```

```
Inequality #1 : yu (# of Grid Points : 101, Independent Variables : afqt )
Numbers of Approximating Functions : 9
```

AIS(adaptive inequality selection) is applied

Bonferroni bounds	Value
50% two-sided confidence interval	[0.1267539, 0.6261939]
90% two-sided confidence interval	[0.1041585, 0.6455886]
95% two-sided confidence interval	[0.0965073, 0.6515738]
99% two-sided confidence interval	[0.0811739, 0.6647557]

The results show that the parametric bound is the narrowest. The parametric 95% confidence interval for the counterfactual probability that a college attendee with the average level of the AFQT score earns more than \$16 per hour is from roughly 0.11 to 0.59. We can interpret results from series and local linear estimation similarly. Notice that, using local linear and series estimation, the output also contains information about bandwidths and the number of approximating functions, respectively. Also, if one does not specify `level`, the procedure automatically provides four different confidence levels: 50%, 90%, 95%, and 99%, by default. As indicated in (7), these confidence intervals are constructed with the use of Bonferroni’s inequality, such that they contain the entire identified set Θ_I with at least the given nominal level asymptotically. The label “Bonferroni Bounds” underscores this point.

9.2. clrbound. In this section, we show how estimation of one-sided intersection bounds works. The result for parametric estimation of the lower bound was:

```
. clrbound (y1 afqt vl_afqt), lower

CLR Intersection Lower Bounds (Parametric)          Number of obs : 2044
Inequality #1 : y1 (# of Grid Points : 101, Independent Variables : afqt )

AIS(adaptive inequality selection) is applied
```

	Value
half-median-unbiased est.	0.1380992
90% one-sided confidence interval	[0.1191487, inf)
95% one-sided confidence interval	[0.1142149, inf)
99% one-sided confidence interval	[0.1047138, inf)

Unlike the two-sided bounds provided by `clr2bound`, this procedure does not explicitly report a 50% confidence interval but effectively conveys the same information by providing the half-median-unbiased estimator for the bound. The half-median-unbiased estimator is precisely $\hat{\theta}_{n0}^l(p)$ appearing in (6) with $p = \frac{1}{2}$ so that

$$P_n\{\theta_0^l \geq \hat{\theta}_{n0}^l(p)\} \geq \frac{1}{2} - o(1).$$

It follows that the interval $[\hat{\theta}_{n0}^l(\frac{1}{2}), \infty)$ is a 50% confidence interval.

The one-sided confidence intervals are all of the form $[\hat{\theta}_{n0}^l(p), \infty)$, for $p = 0.9, 0.95, 0.99$, respectively, with $\hat{\theta}_{n0}^l(p)$ constructed in order to satisfy (6). This guarantees that

$$(21) \quad \liminf_{n \rightarrow \infty} P_n\{\theta^* \in [\hat{\theta}_{n0}^l(p), \infty)\} \geq p.$$

9.3. clrtest. We now test the null hypothesis that 0.59 is in the identified set using a parametric estimator. We use the construction described in (8) and (9) to test whether both the lower bound minus 0.59, and 0.59 minus the upper bound are less

than or equal to zero. Thus, we are carrying out a test of the form (10). To implement this we must first construct new dependent variables before implementing the test. The commands and results were:

```
. gen yl_test = yl - 0.59
. gen yu_test = 0.59 - yu
. clrtest (yl_test afqt vl_afqt)(yu_test afqt vu_afqt), level(0.95)

CLR Intersection Bounds (Test)                                Number of obs : 2044
Inequality #1 : yl_test (# of Grid Points : 101, Independent Variables : afqt )
Inequality #2 : yu_test (# of Grid Points : 101, Independent Variables : afqt )

AIS(adaptive inequality selection) is applied

< Testing Result >
The testing value is NOT in the 95% confidence interval.
In other words, the null hypothesis is rejected at the 5% level.
```

It can be seen that 0.59 is not in the 95% confidence interval. This means that we reject the hypothesis H_0 in (10) in favor of the alternative H_1 . That is, we reject the null hypothesis that the counterfactual probability of earning more than \$16 per hour at schooling level $t = 13$ conditional on having the mean AFQT score is equal to 0.59 at the 5% level.

9.4. clr3bound. This command can obtain a tighter confidence interval than the one given by `clr2bound`, which uses Bonferroni's inequality. Instead of using Bonferroni's inequality, `clr3bound` inverts the test carried out by `clrtest` to construct a confidence interval for θ^* of the form given in (12). The confidence interval given by `clr2bound` is valid for both the point θ^* and the set Θ_I , but the tighter confidence interval provided by `clr3bound` only provides asymptotically valid coverage of the point θ^* . The confidence interval obtained from `clr3bound` was obtained as follows:

```
. clr3bound ((yl afqt vl_afqt)) ((yu afqt vu_afqt))

CLR Intersection Bounds: Test inversion bounds                Number of obs : 2044
Method : Parametric estimation                               Step size : .01
AIS(adaptive inequality selection) is applied

95% Bonferroni bounds:      (0.1097781 , 0.5945532)
95% Test inversion bounds: (0.1299823 , 0.5747234)
```

The last two lines of the results show the Bonferroni bounds delivered by `clr2bound`, as well as the test inversion bounds computed by `clr3bound`. Indeed, we see that the confidence interval obtained by using `clr3bound` is tighter than the one obtained by `clr2bound`. However, since this command uses a grid search to invert `clrtest` to construct the reported confidence interval, it takes longer than `clr2bound`.

REFERENCES

Andrews, D. W. K. and X. Shi (2013). Inference based on conditional moment inequalities. *Econometrica* 81(2), 609–666.

- Andrews, D. W. K. and X. Shi (2014). Nonparametric inference based on conditional moment inequalities. *Journal of Econometrics* 179(1), 31–45.
- Armstrong, T. B. (2011a). Asymptotically exact inference in conditional moment inequality models. Working Paper, Stanford University.
- Armstrong, T. B. (2011b). Weighted KS statistics for inference on conditional moment inequalities. Working Paper, Stanford University.
- Armstrong, T. B. and H. P. Chan (2013). Multiscale adaptive inference on conditional moment inequalities. Discussion Paper 1885, Cowles Foundation.
- Beresteanu, A. and C. F. Manski (2000a). Bounds for matlab. Software at http://faculty.wcas.northwestern.edu/~cfm/bounds_matlab.zip, documentation at http://faculty.wcas.northwestern.edu/~cfm/bounds_matlab.pdf.
- Beresteanu, A. and C. F. Manski (2000b). Bounds for stata. Software at http://faculty.wcas.northwestern.edu/~cfm/bounds_stata.zip, documentation at http://faculty.wcas.northwestern.edu/~cfm/bounds_stata.pdf.
- Beresteanu, A. and F. Molinari (2008). Asymptotic properties for a class of partially identified models. *Econometrica* 76(4), 763–814.
- Beresteanu, A., F. Molinari, and D. Steeg Morris (2010). Asymptotics for partially identified models in stata. Software at http://economics.cornell.edu/fmolinari/Stata_SetBLP.zip.
- Carneiro, P., J. J. Heckman, and E. J. Vytlacil (2011). Estimating marginal returns to education. *American Economic Review* 101(6), 2754–81.
- Carneiro, P. and S. Lee (2009). Estimating distributions of potential outcomes using local instrumental variables with an application to changes in college enrollment and wage inequality. *Journal of Econometrics* 149, 191–208.
- Chernozhukov, V., H. Hong, and E. Tamer (2007, September). Estimation and confidence regions for parameter sets in econometric models. *Econometrica* 75(5), 1243–1284.
- Chernozhukov, V., S. Lee, and A. Rosen (2013). Intersection bounds: Estimation and inference. *Econometrica* 81(2), 667–737.
- Chetverikov, D. (2011). Adaptive rate optimal test of conditional moment inequalities. working paper, MIT.
- Imbens, G. and C. F. Manski (2004, November). Confidence intervals for partially identified parameters. *Econometrica* 72(6), 1845–1857.
- Jann, B. (2005). MOREMATA: Stata module (mata) to provide various functions. Boston College Statistical Software Components S455001.
- Lee, S., K. Song, and Y.-J. Whang (2013a). Testing a general class of functional inequalities. arXiv working paper, arXiv:1311.1595.
- Lee, S., K. Song, and Y.-J. Whang (2013b). Testing functional inequalities. *Journal of Econometrics* 172(1), 14–32.
- Manski, C. F. (1989, Summer). Anatomy of the selection problem. *The Journal of Human Resources* 24(3), 343–360.

- Manski, C. F. (1990, May). Nonparametric bounds on treatment effects. *American Economic Review* 80(2), 319–323.
- Manski, C. F. (1997, November). Monotone treatment response. *Econometrica* 65(6), 1311–1334.
- Manski, C. F. and J. V. Pepper (2000, July). Monotone instrumental variables: With an application to the returns to schooling. *Econometrica* 68(4), 997–1010.
- Romano, J. P. and A. M. Shaikh (2010, February). Inference for the identified set in partially identified econometric models. *Econometrica* 78(1), 169–211.
- Stoye, J. (2009). More on confidence regions for partially identified parameters. *Econometrica* 77(4), 1299–1315.

DEPARTMENT OF ECONOMICS, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MA, USA.

E-mail address: vchern@mit.edu

DEPARTMENT OF ECONOMICS, SEOUL NATIONAL UNIVERSITY, 1 GWANAK-RO, GWANAK-GU, SEOUL, 151-742, REPUBLIC OF KOREA.

E-mail address: burnfire83@gmail.com

DEPARTMENT OF ECONOMICS, SEOUL NATIONAL UNIVERSITY, 1 GWANAK-RO, GWANAK-GU, SEOUL, 151-742, REPUBLIC OF KOREA, AND CENTRE FOR MICRODATA METHODS AND PRACTICE, INSTITUTE FOR FISCAL STUDIES, 7 RIDGMOUNT STREET, LONDON, WC1E 7AE, UK.

E-mail address: sokbae@gmail.com

DEPARTMENT OF ECONOMICS, UNIVERSITY COLLEGE LONDON, GOWER STREET, LONDON, WC1E 6BT, UK, AND CENTRE FOR MICRODATA METHODS AND PRACTICE, INSTITUTE FOR FISCAL STUDIES, 7 RIDGMOUNT STREET, LONDON, WC1E 7AE, UK.

E-mail address: adam.rosen@ucl.ac.uk