The Stata Journal (2015)
15, Number 1, pp. 21–44

Implementing intersection bounds in Stata

Victor Chernozhukov
Massachusetts Institute of Technology
Cambridge, MA
vchern@mit.edu

Wooyoung Kim
University of Wisconsin–Madison
Madison, WI
wkim68@wisc.edu

Sokbae Lee
Institute for Fiscal Studies and
Seoul National University
Seoul, Korea
sokbae@snu.ac.kr

Adam M. Rosen
University College London and
Centre for Microdata Methods and Practice
London, UK
adam.rosen@ucl.ac.uk

Abstract. We present the **clrbound**, **clr2bound**, **clr3bound**, and **clrtest** commands for estimation and inference on intersection bounds as developed by Chernozhukov, Lee, and Rosen (2013, *Econometrica* 81: 667–737). 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 using 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, Lee, and Rosen (2013). The commands include parametric, series, and local linear estimation procedures.

Keywords: st0369, clrbound, clr2bound, clr3bound, clrtest, intersection bounds, bound analysis, conditional moments, partial identification, infinite dimensional constraints, adaptive moment selection

1 Introduction

In this article, we present the **clrbound**, **clr2bound**, **clr3bound**, and **clrtest** commands for estimation and inference on intersection bounds as developed by Chernozhukov, Lee, and Rosen (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 use parametric, series, and local linear estimators, and they can be used to conduct inference on parameters restricted by conditional moment inequalities. The inference method developed by Chernozhukov, Lee, and Rosen (2013) uses sup-norm test statistics. There are many related articles in the literature that develop alternative methods for inference with conditional moment inequalities, such as Andrews and Shi (2013, 2014), Armstrong (2015, 2014), Armstrong and Chan (2013), Chetverikov (2011), and Lee, Song, and Whang (2013a,b).

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.

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

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, Molinari, and Steeg Morris (2010). Beresteanu and Manski (2000a,b) implement bound estimation by using kernel regression for bounds derived in the analysis of treatment response, as considered by Manski (1990), Manski (1997), Manski and Pepper (2000), and others. Our software applies to a broader set of intersection bound problems, and it complements existing software by additionally providing parametric and series estimators as well as methods for bias correction and asymptotically valid inference. The software by Beresteanu, Molinari, and Steeg Morris (2010) can be used to replicate the results in the work of Beresteanu and Molinari (2008) and to compute consistent set estimates for best linear prediction coefficients with interval-censored outcomes. It can also perform inference on any pair of elements of the best linear prediction coefficient vector.

In section 2, we recall the underlying framework of the intersection bounds set up by Chernozhukov, Lee, and Rosen (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 command. 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 for each, and the stored 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 (MTR) and monotone instrumental variable (MIV) 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

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

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

$$\Theta_I \equiv (\theta_0^l, \theta_0^u) \tag{2}$$

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 common case where the bounding functions $\theta_j^l(\cdot)$ and $\theta_j^u(\cdot)$ are conditional expectation functions, such that

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

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 articles 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. Characterizing the asymptotic distribution of plug-in estimators for these bounds is complicated because they are the infimum and supremum of an estimated function. Moreover, using sample analogs for bound estimates is known to produce substantial finite sample bias. The inferential methods of Chernozhukov, Lee, and Rosen (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. 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 to obtain bias-corrected bound estimates or bound estimates that achieve asymptotically valid inference at a desired level. Chernozhukov, Lee, and Rosen (2013) provide asymptotic theory for formal justification and algorithms for implementing these methods. The commands described in this article implement these algorithms in Stata.[1]

Chernozhukov, Lee, and Rosen (2013) provide 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

$$\theta^l(x) \le \theta^*(x) \le \theta^u(x) \tag{3}$$

where

$$\theta^l(x) \equiv E\left\{Y \times 1(Z=t)|X=x\right\}, \qquad \theta^u(x) \equiv E\left\{Y \times 1(Z=t) + 1(Z \neq t)|X=x\right\}$$

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, $Y(\cdot) : \{0,1\} \to [0,1]$. We observe outcome $Y = Y(Z)$ but do not observe the potential outcome from the counterfactual treatment $Y(1-Z)$. This causes the lack of point identification of $E\{Y(t)|X = x\}$. The width of the bounds is $P(Z \neq t)$, which is 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 comprise two components $x = (w, v)$ with component v excluded from affecting the conditional mean function, so that

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

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

$$\sup_{v \in \mathcal{V}} \theta^l\{(w,v)\} \le \theta^*(w) \le \inf_{v \in \mathcal{V}} \theta^u\{(w,v)\} \tag{4}$$

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 $\theta^* = \theta^*(w)$. One can apply this reasoning to obtain upper and lower bounds on $\theta^*(w)$ for all values of w. In section 9, we demonstrate our Stata commands with bounds on a conditional expectation similar to those in (4) applied to data from the NLSY79; however, we use a MIV restriction first considered by Manski and Pepper (2000) instead of the instrumental variable restriction used above.

The estimation problem of Chernozhukov, Lee, and Rosen (2013) is to obtain estimators $\widehat{\theta}^l_{n0}(p)$ and $\widehat{\theta}^u_{n0}(p)$, which provide bias-corrected estimates or the endpoints of

1. All of our commands require the package `moremata` (Jann 2005).

confidence intervals, depending on the chosen value of p; for example, $p = 1/2$ for half-median-unbiased bound estimates, or $p = 1-\alpha$ for confidence intervals. By construction, these estimators satisfy

$$P_n\{\theta_0^l \geq \widehat{\theta}_{n0}^l(p)\} \geq p - o(1), \text{ and } P_n\{\theta_0^u \leq \widehat{\theta}_{n0}^u(p)\} \geq p - o(1) \tag{5}$$

Chernozhukov, Lee, and Rosen (2013), who focus on the upper bound for θ^*, provide further detail on implementation. They explain how 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 $\widetilde{p} = (1 + p)/2$ as an asymptotic level-p confidence set $\{\widehat{\theta}_{n0}^l(\widetilde{p}), \widehat{\theta}_{n0}^u(\widetilde{p})\}$ for Θ_I, which satisfies

$$\liminf_{n\to\infty} P_n\left[\Theta_I \in \left\{\widehat{\theta}_{n0}^l\left(\widetilde{p}\right), \widehat{\theta}_{n0}^u\left(\widetilde{p}\right)\right\}\right] \geq p \tag{6}$$

by (5) 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 asymptotically valid but generally conservative for θ^*.[2] Alternatively, 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

$$T_0(\theta^*) \equiv \max_{k\in\{l,u\}} \max_{j\in\mathcal{J}_k} \sup_{x_j^k\in\mathcal{X}_j^k} T_{jk}\left(x_j^k, \theta^*\right) \leq 0 \tag{7}$$

where

$$T_{ju}\left(x_j^k, \theta^*\right) \equiv \theta^* - \theta_j^u\left(x_j^k\right), \quad T_{jl}\left(x_j^k, \theta^*\right) \equiv \theta_j^l\left(x_j^k\right) - \theta^* \tag{8}$$

For any conjectured value of θ^*, say, θ_{null}, one can apply estimation methods from Chernozhukov, Lee, and Rosen (2013) to perform the hypothesis test

$$H_0 : T_0(\theta_{\text{null}}) \leq 0 \text{ vs. } H_1 : T_0(\theta_{\text{null}}) > 0 \tag{9}$$

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

$$P_n\left\{T_0\left(\theta_{\text{null}}\right) \geq \widehat{T}_{n0}\left(\theta_{\text{null}}, p\right)\right\} \geq p - o\left(1\right) \tag{10}$$

2. Differences 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, Hong, and Tamer (2007), Stoye (2009), and Romano and Shaikh (2010).

which is analogous to the construction of $\widehat{\theta}^l_{n0}(p)$ in (5). The null hypothesis H_0 is then rejected in favor of H_1 at the $1-p$ significance level if $\widehat{T}_{n0}\left(\theta_{\text{null}}, p\right) > 0$. The command `clrtest`, which we describe in section 7, performs such a test. When we invert this test, the set of θ_{null} such that $\widehat{T}_{n0}\left(\theta_{\text{null}}, p\right) \leq 0$ is an asymptotically valid level p confidence set for θ^* because

$$\liminf_{n\to\infty} P_n \left[\theta^* \in \left\{\theta_{\text{null}} : \widehat{T}_{n0}\left(\theta_{\text{null}}, p\right) \leq 0\right\}\right] \geq p \tag{11}$$

by construction. The command `clr3bound`, which we describe in section 8, produces precisely this confidence set.

3 Implementation

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

We let J denote the number of inequalities concerned. Suppose that we have observations $\{(Y_{ji}, X_{ji}) : i = 1, \ldots, n, j = 1, \ldots, J\}$, where n is the sample size. For each $j = 1, \ldots, J$, we let $\mathbf{y}_j$ denote the $n \times 1$ vector whose ith element is Y_{ji}, and we let $\mathbf{X}_j$ denote the $n \times d_j$ matrix whose ith row is X'_{ji}, where d_j is the dimension of X_{ji}. We allow multidimensional $\mathbf{X}_j$ for only 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 $j = 1, \ldots, J$, say, $(\mathbf{x}_1, \ldots, \mathbf{x}_J)$, where $\mathbf{x}_j = (x'_{j1}, \ldots, x'_{jM_j})'$ for some sufficiently large numbers M_j, and $j = 1, \ldots, J$, where each x_{jm} is a $d_j \times 1$ vector. We also let $\mathbf{\Psi}_j$ denote the $M_j \times d_j$ matrix whose mth row is x'_{jm}, where $m = 1, \ldots, M_j$ and $j = 1, \ldots, J$. The number of grid points can be different for different inequalities.

3.1 Parametric estimation

To define

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

we let $\boldsymbol{\theta}_j(\mathbf{x}_j) \equiv \{\theta_j(x_{j1}), \cdots, \theta_j(x_{jM_j})\}'$ and $\boldsymbol{\theta} \equiv \{\boldsymbol{\theta}_1(\mathbf{x}_1)', \ldots, \boldsymbol{\theta}_J(\mathbf{x}_J)'\}'$. Then the estimator of $\boldsymbol{\theta}$ is $\widehat{\boldsymbol{\theta}} \equiv \mathbf{\Psi}\widehat{\boldsymbol{\beta}}$, 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

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

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 $\mathbf{A}$.

To obtain a precision-corrected estimate, we maximize the precision-corrected curve, which we get by multiplying the function estimate minus the critical value by the standard error. To compute the critical value, say, $k(p)$, define

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

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

$$k(p) = \text{the } p\text{th quantile of } \max_{\text{col.}} \left\{ \text{chol}\left(\widehat{\mathbf{\Sigma}}\right) \mathbf{Z}_R \right\} \tag{12}$$

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

$$\widehat{\theta}_{n0}(p) = \max_{\text{col.}} \left\{ \mathbf{\Psi}\widehat{\boldsymbol{\beta}} - k(p)\widehat{\mathbf{s}} \right\} \tag{13}$$

The critical value in (13) is obtained under the least favorable case. To improve the estimator, we carry out the following adaptive inequality selection (AIS) procedure:

1. Set $\widetilde{\gamma}_n \equiv 1 - 0.1/\log n$. Let ψ_k' denote the kth row of $\mathbf{\Psi}$, where $k = 1, \ldots, \sum_{j=1}^J M_j$. Keep each row ψ_k' of $\mathbf{\Psi}$ if and only if

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

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

2. Replace $\mathbf{\Psi}$ with the kept rows of $\mathbf{\Psi}$ in step 1. Then, recompute $\mathbf{V}$ and $\widehat{\mathbf{\Sigma}}$ to update the critical value in (12) and obtain the final estimate $\widehat{\theta}_{n0}(p)$ in (13) with the updated critical value.

3.2 Series estimation

The implementation of series estimation is similar to that of parametric estimation. For each $j = 1, \ldots, J$, we let $p_{nj}(x) \equiv \{p_{n,1}(x), \ldots, p_{n,\kappa_j}(x)\}'$, and we denote the κ_j-dimensional vector of approximating functions by cubic B-splines. Here the number of series terms κ_j can be different for each inequality. We let $\widetilde{\mathbf{X}}_j$ denote the $n \times \kappa_j$ matrix whose ith row is $p_{nj}(X_{ji})'$ and $\widetilde{\mathbf{\Psi}}_j$ denote the $M_j \times \kappa_j$ matrix whose mth row is $p_{nj}(x_{jm})'$. We can then complete the same procedure described in section 3.1, substituting $\widetilde{\mathbf{X}}_j$ and $\widetilde{\mathbf{\Psi}}_j$ for $\mathbf{X}_j$ and $\mathbf{\Psi}_j$, respectively.

In this implementation, the dimension d_j of X_{ji} is 1, 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 $\widetilde{\mathbf{X}}_j$ and $\widetilde{\boldsymbol{\Psi}}_j$ in series estimation. See section 4.2 of Chernozhukov, Lee, and Rosen (2013) for details.

3.3 Local linear estimation

For any vector $\mathbf{v}$, we let $\widehat{\boldsymbol{\rho}}_j(\mathbf{v})$ denote the vector whose kth element is the local linear regression estimate of $\mathbf{y}_j$ on $\mathbf{X}_j$ at the kth element of $\mathbf{v}$. In detail, the kth element of $\widehat{\boldsymbol{\rho}}_j(\mathbf{v})$, say, $\widehat{\rho}_j(v_k)$, is defined as follows,

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

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

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

$K(\cdot)$ 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:

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

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

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

$$\overline{g^2}_{jm}(\mathbf{y}_j,\mathbf{X}_j) = n^{-1}\sum_{i=1}^{n}\widehat{g}_{ji}(Y_{ji},X_{ji},x_{jm})^2$$

$$\widehat{g}_{ji}(Y_{ji},X_{ji},x_{jm}) = \frac{Y_{ji}-\widehat{\rho}_j(X_{ji})}{\sqrt{h_j}\widehat{f}_j(x_{jm})}K\left(\frac{x_{jm}-X_{ji}}{h_j}\right)$$

$\widehat{f}_j(x_{jm})$ is the kernel estimate of the density of the covariate for the jth inequality, evaluated at x_{jm}. Then we can compute $\widehat{\boldsymbol{s}}$ as $\widehat{\boldsymbol{s}} = (\widehat{s}_1',\ldots,\widehat{s}_J')'$.

To compute the critical value $k(p)$, we let $\boldsymbol{\Phi}_j$ denote the $M_j \times n$ matrix whose mth row is $\{\widehat{g}_{j1}(Y_{j1},X_{j1},x_{jm}),\ldots,\widehat{g}_{jn}(Y_{jn},X_{jn},x_{jm})\}/\sqrt{nh_j\overline{g^2}_{jm}(\mathbf{y}_j,\mathbf{X}_j)}$. We define

$$\boldsymbol{\Phi} \equiv \begin{pmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_J \end{pmatrix}$$

We simulate pseudorandom numbers from the $N(0,1)$ distribution and construct an $n \times R$ matrix, $\mathbf{Z_R}$. We then select the critical value as

$$k(p) = \text{the } p\text{th quantile of } \max_{\text{col.}} (\mathbf{\Phi Z_R}) \tag{14}$$

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

$$\widehat{\theta}_{n0}(p) = \max_{\text{col.}} \left\{ \widehat{\boldsymbol{\theta}} - k(p)\widehat{\mathbf{s}} \right\}$$

However, the AIS procedure is slightly different because we do not use $\mathbf{\Psi}$ in local linear estimation.

1. Set $\widetilde{\gamma}_n \equiv 1 - .1/\log n$. Keep the mth row of each $\mathbf{\Phi}_j$, $j = 1, \ldots, J$, if and only if

$$\widehat{\rho}_j\left(x_{jm}\right) \geq \widehat{\theta}_{n0}\left(\widetilde{\gamma}_n\right) - 2k\left(\widetilde{\gamma}_n\right)\widehat{s}_{jm}$$

   where $\widehat{s}_{jm}$ is the mth element of $\widehat{\mathbf{s}}_j$.

2. For $j = 1, \ldots, J$, replace $\mathbf{\Phi}_j$ with the kept rows of $\mathbf{\Phi}_j$ in step 1. Then, recompute the critical value in (14), and obtain the final estimate $\widehat{\theta}_{n0}(p)$ with the updated critical value.

4 Installation of the clrbound package

All of our commands require the package `moremata` (Jann 2005), which can be installed by typing `ssc install moremata, replace` in the Stata Command window.

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(#) rnd(#)`
  `norseed seed(#)`]

5.2 Description

The command `clr2bound` estimates a two-sided confidence interval $[\widehat{\theta}^l_{n0}(\widetilde{p}), \widehat{\theta}^u_{n0}(\widetilde{p})]$, where $\widetilde{p} = (\texttt{level}() + 1)/2$. By (5) and Bonferroni's inequality, this interval contains the identified set Θ_I with probability at least `level()` asymptotically, that is, such that (6) holds with $p = \texttt{level}()$. The variables *lowerdepvar1*, ..., *lowerdepvarN* are the dependent variables (Y^l_j's) for the lower-bounding functions, and the variables *upperdepvarN+1*, ..., *upperdepvarN+M* are the dependent variables (Y^u_j's) for the upper-bounding functions. The variables *indepvars1*, ..., *indepvarN+M* are explanatory variables for the corresponding dependent variables. `clr2bound` allows for multidimensional *indepvars* for parametric estimation but for only a one-dimensional independent variable for series and local linear estimation.

The variables *range1*, ..., *rangeN+M* are sets of grid points over which the bounding function is estimated, corresponding to the sets $\mathcal{X}^l_j$ and $\mathcal{X}^u_j$ 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.

Note 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, say, `(`*ldepvar1 indepvars1 range1*`)` and `(`*ldepvar2 indepvars2 range2*`)` for the lower-bounds estimation and one variable set, say, `(`*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 unspecified, the null hypothesis is that the parameter of interest is 0. This test uses (10), 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(series|local)` specifies the method of estimation. By default, `clr2bound` will conduct parametric estimation. If `method(series)` is specified, `clr2bound` will conduct series estimation with cubic B-splines. If `method(local)` is specified, `clr2bound` will conduct local linear estimation.

`notest` determines whether `clr2bound` conducts a test. `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 only Bonferroni bounds.

`null(`*real*`)` specifies the value for θ^* under the null hypothesis of the test we described above. The default is `null(0)`.

`level(`*numlist*`)` specifies confidence levels. *numlist* must contain only real numbers between 0 and 1. 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 AIS should be applied. 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 $\widehat{K}_{\mathrm{cv}}$ 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)`, $\widehat{K}_{\mathrm{cv}}$ 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 undersmoothing is performed, the number of approximating functions K ultimately used will be given by the largest integer smaller than $\widehat{K}_{\mathrm{cv}}$ multiplied by the undersmoothing factor $n^{-1/5} \times n^{2/7}$; see option `noundersmooth` below. This option is available for only series estimation.

`noundersmooth` determines whether undersmoothing is carried out, with the default being to undersmooth. In series estimation, undersmoothing is implemented by first computing $\widehat{K}_{\mathrm{cv}}$ as the minimizer of the leave-one-out least-squares cross-validation score. Without this option, the number of approximating functions is then set to K, which is given by the largest integer that is less than or equal to $\widehat{K} := \widehat{K}_{\mathrm{cv}} \times n^{-1/5} \times n^{2/7}$. The `noundersmooth` option uses $\widehat{K}_{\mathrm{cv}}$. For local linear estimation, undersmoothing is done by setting the bandwidth to $h = \widehat{h}_{\mathrm{ROT}} \times \widehat{s}_v \times n^{1/5} \times n^{-2/7}$, where $\widehat{h}_{\mathrm{ROT}}$ is the "rule-of-thumb" bandwidth that is used in Chernozhukov, Lee, and Rosen (2013). The `noundersmooth` option instead uses $\widehat{h}_{\mathrm{ROT}} \times \widehat{s}_v$. This option is available for only series and local linear estimation.

`bandwidth(`*#*`)` specifies the value of the bandwidth used in local linear estimation. By default, `clr2bound` calculates a bandwidth for each inequality. With undersmoothing, we use the "rule-of-thumb" bandwidth $h = \widehat{h}_{\mathrm{ROT}} \times \widehat{s}_v \times n^{1/5} \times n^{-2/7}$, where $\widehat{s}_v$ is the square root of the sample variance of V, and $\widehat{h}_{\mathrm{ROT}}$ is the "rule-of-thumb" bandwidth for estimation of $\theta(v)$ with Studentized V. See Chernozhukov, Lee, and Rosen (2013) for the exact form of $\widehat{h}_{\mathrm{ROT}}$. When the `bandwidth(`*#*`)` is specified, `clr2bound` uses the given bandwidth as the global bandwidth for every inequality. This option is available for only local linear estimation.

`rnd(`*#*`)` specifies the number of columns of the random matrix generated from the standard normal distribution. This matrix is used to compute critical values. For example, if the number is 10,000 and the level is 0.95, we choose the 0.95 quantile from 10,000 randomly generated elements. The default is `rnd(10000)`.

`norseed` determines whether to reset the seed number for the simulation used in the calculation. 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 initially conducts `set seed #`. The default is `seed(0)`.

5.4 Stored results

In the following, "l.b.e." stands for lower-bound estimation, "u.b.e." for upper-bound estimation, and "ineq" stands for inequality. (i) denotes the ith 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 available for only series or local linear estimation.

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

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

See Chernozhukov, Lee, and Rosen (2013) for details on $\widehat{\theta}_n(v)$, $s_n(v)$, and $\widehat{\Omega}_n$.

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(#)` `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 $\widehat{\theta}^l_{n0}(p)$ and upper-bound estimates $\widehat{\theta}^u_{n0}(p)$ are constructed to satisfy (5) for p set equal to `level()`. The variables are defined similarly as for `clr2bound`.

6.3 Options

`lower` and `upper` specify whether the estimation is for the lower bound or the upper bound. By default, `clrbound` will return the upper-intersection bound. If `lower` is specified, `clrbound` will return the lower-intersection bound.

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

6.4 Stored results

In the following, we use the same abbreviations as in section 5.4. `clrbound` stores the following in `e()`:

Scalars

`e(N)`	number of observations	`e(n_ineq)`	# of inequality
`e(grid(i))`	# of grids points (i)	`e(nf_x(i))`	# of approx. functions in (i)
`e(bd(lev))`	results of estimation	`e(cl(lev))`	critical value
`e(bdwh(i))`	bandwidth for (i)		

Macros

`e(cmd)`	`clrbound`	`e(level)`	confidence levels
`e(depvar)`	dependent variables	`e(smoothing)`	`(not) Undersmoothed`
`e(indep(i))`	indep. variables in (i)	`e(range(i))`	range in (i)
`e(title)`	title in estimation output		

Matrices

`e(omega)`	$\widehat{\Omega}_n$	`e(theta(i))`	$\widehat{\theta}_n(v)$ for each v
`e(se(i))`	$s_n(v)$ for each v	`e(ais(i))`	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*] [, {<u>low</u>er|<u>upp</u>er}
<u>met</u>hod(series|local) <u>lev</u>el(*numlist*) noais <u>mins</u>mooth(#) <u>maxs</u>mooth(#)
<u>noundersm</u>ooth <u>band</u>width(#) rnd(#) norseed <u>se</u>ed(#)]

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 (9) by using the given *depvar*s and *indepvar*s 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, and X^u are coded as `yl`, `yu`, `xl`, and `xu`, respectively. To test this hypothesis, one first creates the variables `yl_test` = `yl` − 0.59 and `yu_test` = 0.59 − `yu` and then executes the command `clrtest (yl_test xl vl) (yu_test xu vu), level(0.95)`, where `vl` and `vu` are grid points. The `level(0.95)` corresponds to the value of p used for the intersection bound estimate described in (10) that was required to perform the test (9) at the $1-p$ significance level. We illustrate the use of this command in section 9.3.

7.3 Options

Because the options for `clrtest` are the same as those for `clrbound`, the explanation of options is omitted.

7.4 Stored results

Other stored results are the same as those of `clrbound`, except for the following:

Scalars
`e(det(lev))` 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*] [`, stepsize(#) method(series|local) level(#) noais minsmooth(#) maxsmooth(#) noundersmooth bandwidth(#) rnd(#) norseed seed(#)`]

8.2 Description

`clr3bound` estimates a two-sided confidence interval for the parameter θ^* by inverting the test (9) performed by the `clrtest` command. The result is a collection of values of θ_{null} that estimate a confidence set for θ^* with asymptotic coverage `level()`, as described by (11). Note that when only one-sided intersection bounds are used, there is no need to implement the pointwise test using `clr3bound`.

Because this command is relevant for only 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 uses Bonferroni's inequality to produce confidence sets for Θ_I. Unlike the previous commands, `clr3bound` can deal with only one confidence level at a time. It takes longer to compute bounds using the `clr3bound` command than it does using the `clr2bound` command, because `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 equispaced grid and implements the `clrtest` command for each grid point to determine a possible tighter bound. The default is `stepsize(0.01)`.

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

Other options are the same as those of `clr2bound`. However, the `clr3bound` command does not have the `notest` and `null()` options, because it does not explicitly conduct a test.

8.4 Stored results

`clr3bound` stores the following in `e()`:

Scalars

`e(N)`	number of observations	`e(u_nf_x(i))`	# of approx. functions in (i) of u.b.e.
`e(step)`	step size		
`e(level)`	confidence level	`e(l_bdwh(i))`	bandwidth for (i) of l.b.e
`e(l_ineq)`	# of ineq's in l.b.e.	`e(u_bdwh(i))`	bandwidth for (i) of u.b.e.
`e(u_ineq)`	# of ineq's in u.b.e.	`e(lbd)`	est. results of l.b.e.
`e(l_grid(i))`	# of grids points for l.b.e. at observation (i)	`e(ubd)`	est. results of u.b.e.
		`e(lbd(lev))`	Bonferroni results of l.b.e.
`e(u_grid(i))`	# of grids points for u.b.e. at observation (i)	`e(ubd(lev))`	Bonferroni results of u.b.e.
		`e(lcl(lev))`	critical value of l.b.e.
`e(l_nf_x(i))`	# of approx. functions in (i) of l.b.e.	`e(ucl(lev))`	critical value of l.b.e.

Macros

`e(cmd)`	`clr3bound`	`e(smoothing)`	`(not) Undersmoothed`
`e(title)`	title in estimation output	`e(l_indep(i))`	indep. var. in (i) of l.b.e.
`e(ldepvar)`	dep. var. in l.b.e.	`e(u_indep(i))`	indep. var. in (i) of u.b.e.
`e(udepvar)`	dep. var. in u.b.e.	`e(l_range(i))`	range in (i) of l.b.e.
`e(method)`	estimation method	`e(u_range(i))`	range in (i) of u.b.e.

Matrices

`e(l_omega)`	$\widehat{\Omega}_n$ for l.b.e.	`e(l_se(i))`	$s_n(v)$ for each v in l.b.e.
`e(u_omega)`	$\widehat{\Omega}_n$ for u.b.e.	`e(u_se(i))`	$s_n(v)$ for each v in u.b.e.
`e(l_theta(i))`	$\widehat{\theta}_n(v)$ for each v in l.b.e.	`e(l_ais(i))`	AIS result for each v in l.b.e.
`e(u_theta(i))`	$\widehat{\theta}_n(v)$ for each v in u.b.e.	`e(u_ais(i))`	AIS result for each v in u.b.e.

9 Examples

To illustrate the use of `clrbound`, `clr2bound`, `clr3bound`, and `clrtest`, we present some examples using the joint MIV and MTR bounds of Manski and Pepper (2000, proposition 2), as in Chernozhukov, Lee, and Rosen (2013), to study log wages as a function of years of schooling. We use the same data extract from the NLSY79 as Carneiro and Lee (2009). See also Carneiro, Heckman, and Vytlacil (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 (AFQT) score (`afqt`).[3] 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 each individual (i). V_i is the AFQT score, a measure of cognitive ability, Studentized to have mean zero and variance one in the NLSY79 population. We let Z_i denote the realized

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

treatment, which here is the realized years of schooling and is possibly self-selected by individuals. The source of the identification problem is the same as that of the example considered in section 2—namely, for each individual i, we observe only $Y_i \equiv Y_i(Z_i)$ along with (Z_i, V_i), but not $Y_i(t)$ with $t \neq Z_i$.

The MIV assumption introduced by Manski and Pepper (2000) asserts that for all treatment levels t, the conditional expectation $E\{Y_i(t)|V_i = v\}$ weakly increases 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 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 or instrumental variable restrictions (stronger than the MIV restriction), expected returns to schooling are generally not point identified, but they can be bounded. To illustrate, we condition on the average AFQT score $V_i = 0$, but one could do an identical analysis by conditioning on other values.

From Manski and Pepper's (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,

$$\sup_{u \le v} E\left(Y_i^l | V_i = u\right) \le E\{Y_i(t)|V_i = v\} \le \inf_{s \ge v} E\left(Y_i^u | V_i = s\right) \tag{15}$$

where

$$Y_i^l \equiv Y_i \times 1\left(t \ge Z_i\right) + y_0 \times 1\left(t < Z_i\right), \qquad Y_i^u \equiv Y_i \times 1\left(t \le Z_i\right) + y_1 \times 1\left(t > Z_i\right)$$

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, we take the parameter of interest to be

$$\theta^* = \mathrm{P}\{Y_i(t) > y | V_i = v\}$$

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 1 more year of schooling than high school graduates). Thus our goal will be to perform inference on θ^*, which is the probability that the hourly wage obtained by a college attendee ($t = 13$) is greater than \$16, conditional on having an AFQT score at the average level in the NLSY79 population.

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

$$\theta^* \le \inf_{u \ge v} E\left\{1(Y_i > y) \times 1(t \le Z_i) + 1(t > Z_i)|V_i = u\right\} \tag{16}$$

and the lower bound is

$$\theta^* \ge \sup_{u \le v} E\left\{1(Y_i > y) \times 1(t \ge Z_i)|V_i = u\right\} \tag{17}$$

Indeed, the derivation of these bounds is identical to that of the conditional expectation bound (15), with the indicator function $1\{Y_i(t) > y\}$ in place of $Y_i(t)$ in the conditional expectation $E\{Y_i(t)|V_i = v\}$. To illustrate, we focus on the threshold $y = \log(16)$, but such bounds can be studied for any level of log wages y of interest, and they can be conditional on any value of v and any desired level of schooling t.[4]

When studying the joint MIV-MTR bounds, we must be careful when setting the *range* variable, which we described in section 5.2. This variable provides grids of values that represent the sets $\mathcal{X}^u$ and $\mathcal{X}^l$ that appear in (1).[5] In this example, these sets differ from one another because, as in (16) and (17) above, $\mathcal{X}^u$ 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. Because we focus on the value of $\theta^* = E\{Y_i(t)|V_i = v\}$ at $v = 0$, a new variable that contains grid points larger (smaller) than 0 should be used for upper- (lower-) bound estimation. To obtain bounds conditional on other values of v, we must change the *range* accordingly. As *range* variables for our NLSY79 dataset, we used `vl_afqt` for the lower bound and `vu_afqt` for the upper bound. Each contains 101 grid points from -2 to 0 and 0 to 2, respectively. We used the following commands to make the *range* variables:

```
. use nlsy.dta
. 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 we calculate the MIV-MTR upper bound, we must define the dependent variable as $Y_i^u = 1(Y_i > y) \times 1(t \le Z_i) + 1(t > Z_i)$. In our example, we let `yl` denote the dependent variable for lower-bound estimation and `yu` for the upper-bound estimation. We use the following commands to construct these variables:

```
. generate yl = (lnwage > log(16)) * (eduyr <= 13)
. generate yu = (lnwage > log(16)) * (eduyr >= 13) + (eduyr < 13)
```

4. Under the stronger assumption that the distribution of $Y_i(t)|V_i = v$ is stochastically increasing in v so that $\mathrm{P}[\{Y_i(t) > y\}|V_i = v]$ 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 $V_i = v$.

5. In this example, the sets $\mathcal{J}_u$ and $\mathcal{J}_l$ in (1) are both singletons. The subscript j on these sets is thus superfluous and has been dropped.

Here we show how to use the three estimation methods (parametric, local linear, and series estimation). We also include the test result for whether 0.1 is in the two-sided intersection bounds when using series estimation. The results are as follows:

```
. 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 an average-level 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. When 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 (6), these confidence intervals are constructed using 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 the following:

```
. clrbound (yl afqt vl_afqt), lower
CLR Intersection Lower Bounds (Parametric)              Number of obs : 2044
Inequality #1 : yl (# 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 it effectively conveys the same information by providing the half-median-unbiased estimator for the bound. The half-median-unbiased estimator is precisely $\widehat{\theta}^l_{n0}(p)$, appearing in (5) with $p = 1/2$ so that

$$P_n\{\theta^l_0 \geq \widehat{\theta}^l_{n0}(p)\} \geq \frac{1}{2} - o(1)$$

It follows that the interval $[\widehat{\theta}^l_{n0}(1/2), \infty)$ is a 50% confidence interval.

The one-sided confidence intervals are all of the form $[\widehat{\theta}^l_{n0}(p), \infty)$ for $p = 0.9$, 0.95, and 0.99, with $\widehat{\theta}^l_{n0}(p)$ constructed to satisfy (5). This guarantees that

$$\liminf_{n\to\infty} P_n\left\{\theta^* \in \left[\widehat{\theta}^l_{n0}(p), \infty\right)\right\} \geq p$$

9.3 clrtest

We now test the null hypothesis that 0.59 is in the identified set by using a parametric estimator. We use the construction described in (7) and (8) to test whether both the lower bound minus 0.59 and 0.59 minus the upper bound are less than or equal to 0. Thus we are carrying out a test of the form (9). To implement this, we must construct new dependent variables before we implement the test. The commands and results are as follows:

```
. generate yl_test = yl - 0.59
. generate 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.
```

We can see that 0.59 is not in the 95% confidence interval. This means that we reject the hypothesis H_0 in (9) 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 done by `clrtest` to construct a confidence interval for θ^* of the form given in (11). 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` provides only 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, because this command uses a grid search to invert `clrtest` to construct the reported confidence interval, it takes longer than `clr2bound`.

10 Acknowledgments

We thank an anonymous referee for helpful comments, and we thank 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 and through the funding of the "Programme Evaluation for Policy Analysis" node of the UK National Centre for Research Methods is gratefully acknowledged, as is financial support from the European Research Council grant ERC-2009-StG-240910-ROMETA. This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government(NRF-2013S1A2A1A01033960).

11 References

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

———. 2014. Nonparametric inference based on conditional moment inequalities. *Journal of Econometrics* 179: 31–45.

Armstrong, T. B. 2014. Weighted KS statistics for inference on conditional moment inequalities. *Journal of Econometrics* 181: 92–116.

———. 2015. Asymptotically exact inference in conditional moment inequality models. *Journal of Econometrics* 186: 51–65.

Armstrong, T. B., and H. P. Chan. 2013. Multiscale adaptive inference on conditional moment inequalities. Discussion Paper No. 1885, Cowles Foundation. http://cowles.econ.yale.edu/P/cd/d18b/d1885.pdf.

Beresteanu, A., and C. F. Manski. 2000a. Bounds for MatLab: Draft version 1.0. Software at http://faculty.wcas.northwestern.edu/%7ecfm754/bounds_matlab.zip, documentation at http://faculty.wcas.northwestern.edu/%7ecfm754/bounds_matlab.pdf.

———. 2000b. Bounds for Stata: Draft version 1.0. Software at http://faculty.wcas.northwestern.edu/%7ecfm754/bounds_stata.zip, documentation at http://faculty.wcas.northwestern.edu/%7ecfm754/bounds_stata.pdf.

Beresteanu, A., and F. Molinari. 2008. Asymptotic properties for a class of partially identified models. *Econometrica* 76: 763–814.

Beresteanu, A., F. Molinari, and D. Steeg Morris. 2010. Asymptotics for partially identified models in Stata. https://molinari.economics.cornell.edu/programs/Stata_SetBLP.zip.

Carneiro, P., J. J. Heckman, and E. J. Vytlacil. 2011. Estimating marginal returns to education. *American Economic Review* 101: 2754–2781.

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. Estimation and confidence regions for parameter sets in econometric models. *Econometrica* 75: 1243–1284.

Chernozhukov, V., S. Lee, and A. M. Rosen. 2013. Intersection bounds: Estimation and inference. *Econometrica* 81: 667–737.

Chetverikov, D. 2011. Adaptive test of conditional moment inequalities. Working Paper, MIT.

Imbens, G. W., and C. F. Manski. 2004. Confidence intervals for partially identified parameters. *Econometrica* 72: 1845–1857.

Jann, B. 2005. moremata: Stata module (Mata) to provide various functions. Statistical Software Components S455001, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s455001.html.

Lee, S., K. Song, and Y.-J. Whang. 2013a. Testing for a general class of functional inequalities. ArXiv Working Paper No. arXiv:1311.1595. http://arxiv.org/abs/1311.1595.

———. 2013b. Testing functional inequalities. *Journal of Econometrics* 172: 14–32.

Manski, C. F. 1989. Anatomy of the selection problem. *Journal of Human Resources* 24: 343–360.

———. 1990. Nonparametric bounds on treatment effects. *American Economic Review* 80: 319–323.

———. 1997. Monotone treatment response. *Econometrica* 65: 1311–1334.

Manski, C. F., and J. V. Pepper. 2000. Monotone instrumental variables: With an application to the returns to schooling. *Econometrica* 68: 997–1010.

Romano, J. P., and A. M. Shaikh. 2010. Inference for the identified set in partially identified econometric models. *Econometrica* 78: 169–211.

Stoye, J. 2009. More on confidence intervals for partially identified parameters. *Econometrica* 77: 1299–1315.

About the authors

Victor Chernozhukov is a professor of economics in the Department of Economics at Massachusetts Institute of Technology.

Wooyoung Kim is a doctoral student in the Department of Economics at the University of Wisconsin–Madison.

Sokbae Lee is a professor of economics in the Department of Economics at Seoul National University.

Adam Rosen is a reader in economics in the Department of Economics at University College London.