The Stata Journal (2021)
21, Number 1, pp. 97–122 DOI: 10.1177/1536867X211000006

Evaluating the maximum regret of statistical treatment rules with sample data on treatment response

Valentyn Litvin
Department of Economics
Northwestern University
Evanston, IL
valentynlitvin@u.northwestern.edu

Charles F. Manski
Department of Economics
Northwestern University
Evanston, IL
cfmanski@u.northwestern.edu

Abstract. In this article, we present the `wald_tc` command, which computes the maximum regret (MR) of a user-specified statistical treatment rule that uses sample data on realized treatment response (and optionally an instrumental variable) to determine a treatment choice for a population. Because the outcomes of counterfactual treatments are not observed and treatment selection in the study population may not be random, decision makers may be able only to partially identify average treatment effects. `wald_tc` allows users to compute the MR of a proposed statistical treatment rule under a flexible specification of the data-generating process and determines the state that generates MR.

Keywords: st0629, wald_tc, maximum regret, average treatment effect, instrumental variable, partial identification

1 Introduction

In this article, we present the `wald_tc` command, which computes the maximum regret (MR) of a user-specified statistical treatment rule (STR) that uses sample data on realized treatment response (and optionally an instrumental variable [IV]) to determine a treatment choice for a population. Because the outcomes of counterfactual treatments are not observed and treatment selection in the study population may not be random, decision makers (DMs) may be able only to partially identify average treatment effects (ATEs). `wald_tc` allows the DM to compute the MR of any given STR under a flexible specification of the data-generating process and determines the state that generates MR. `wald_tc` uses MR ex ante to assess the performance of a decision rule over all feasible samples. Because this assessment is done before any actual sample data are drawn, `wald_tc` does not take any data as input. Readers interested in `wald_tc` may also be interested in the `wald_mse` command, which computes the maximum mean squared error of user-specified point predictors of real random variables (Manski and Tabord-Meehan 2017).

Section 2 covers the statistical theory motivating `wald_tc` when a DM observes only treatment and outcome data. Section 3 examines the theory when a DM observes an IV in addition to treatment and outcome data. Section 4 introduces the syntax of `wald_tc`,

explains the available options for specifying the state space of data-generating processes, and documents key implementation details. Section 5 illustrates the command through examples.

2 Theoretical background

2.1 Identification problem with nonrandomized treatment

Take a large population J of observationally identical individuals who face two treatment alternatives, a or b. Suppose that for each $j \in J$, potential treatment outcomes $y(\cdot)$ are binary: $\{y_j(a), y_j(b)\} \in (0,1) \times (0,1)$. A utilitarian DM faces the problem of mandating a single treatment $\delta \in (a,b)$ for the entire population. Before making this decision, the DM samples N individuals in J, indexed by $i \in (1,\ldots,N)$, and observes a (treatment, outcome) pair (t_i, y_i) for each individual where $y_i := y_i(t_i)$. Let $\{y_j(a), y_j(b)\} \sim P$, where P is the population distribution of potential outcomes. Suppose the DM draws a sample $\psi \sim Q$ of N pairs (t_i, y_i), where Q is the sampling distribution over the sample space $\Psi \subseteq \times^N\{(a,b) \times (0,1)\}$. We will restrict attention to the case where (t_i, y_i) are independent and identically distributed so $\psi \sim \times^N \widetilde{Q}, (t,y) \sim \widetilde{Q}$. Then, the DM chooses an STR $\delta : \times^N\{(a,b) \times (0,1)\} \to (a,b)$, yielding expected welfare

$$W(\delta, P, Q) = \mathbb{E}\{y(a)\} \Pr\{\delta(\psi) = a\} + \mathbb{E}\{y(b)\} \Pr\{\delta(\psi) = b\} \tag{1}$$

where $\Pr\{\delta(\psi) = \widetilde{t}\} = \int_\Psi \mathbb{1}\{\delta(\psi) = \widetilde{t}\}\, dQ(\psi)$.

With binary outcomes, $\mathbb{E}\{y(a)\} = \Pr\{y(a) = 1\}$ and $\mathbb{E}\{y(b)\} = \Pr\{y(b) = 1\}$. However, the DM does not observe the counterfactual potential outcome for any individual; that is, $y_i(t_{-i})$. The sampling process reveals $\Pr(t = a)$ [and $\Pr(t = b)$], $\Pr\{y(a) = 1 | t = a\}$, and $\Pr\{y(b) = 1 | t = b\}$, but the DM is interested in

$$\begin{aligned}
\mathbb{E}\{y(a)\} &= \Pr\{y(a) = 1 | t = a\} \Pr(t = a) + \Pr\{y(a) = 1 | t = b\} \Pr(t = b) \\
\mathbb{E}\{y(b)\} &= \Pr\{y(b) = 1 | t = a\} \Pr(t = a) + \Pr\{y(b) = 1 | t = b\} \Pr(t = b)
\end{aligned}$$

If we assume that the DM's sampling process randomly assigns treatment t—that is, $\Pr\left(y | t = \widetilde{t}\right) = \Pr\{y\left(\widetilde{t}\right)\}$, $\widetilde{t} \in (a,b)$—then as $N \to \infty$, the DM can identify $\mathbb{E}\{y(a)\}$ and $\mathbb{E}\{y(b)\}$ and unequivocally choose the best treatment. This assumption is sometimes called unconfoundedness.

However, if we do not make this assumption and allow nonrandomized treatment, then the DM learns nothing about $\Pr\{y(a) | t = b\}$ and $\Pr\{y(b) | t = a\}$. The DM learns only $\Pr(y | t = a)$ and $\Pr(y | t = b)$, which equal $\Pr\{y(a) | t = a\}$ and $\Pr\{y(b) | t = b\}$, respectively. Manski (1990) showed that when $N \to \infty$ and $\Pr(t = a) \in (0,1)$, the DM can partially identify the expected treatment effects

$$\begin{aligned}
\mathbb{E}\{y(\widetilde{t})\} \in [&\Pr\left(y = 1 | t = \widetilde{t}\right) \Pr(t = \widetilde{t}), \Pr\left(y = 1 | t = \widetilde{t}\right) \Pr(t = \widetilde{t}) + \{1 - \Pr(t = \widetilde{t})\}], \\
&\widetilde{t} \in (a,b)
\end{aligned}$$

Even in the limiting case of no statistical imprecision, the DM's optimal decision depends on the unobserved parameters $\Pr\{y(a) = 1|t = b\}$ and $\Pr\{y(b) = 1|t = a\}$.

2.2 Introducing minimum MR

Formalize this decision problem under uncertainty in the manner of Wald (1949), and specify a state space S that contains all populations and sampling processes that the DM considers possible. Each state $s \in S$ is characterized by a feasible pair (P_s, Q_s).

The DM can follow Bayesian decision theory and assign a prior distribution over S. However, as discussed in Manski (2004), the DM may be unable or unwilling to formulate a credible prior. Without a prior, the DM can use the maximin or minimax regret (MMR) criterion to evaluate different STRs. Following Manski (2004), we use the MMR criterion, which tries to choose an STR that yields expected welfare uniformly close to the optimal possible expected welfare across S. For further work related to MMR treatment choice, see Manski (2005, 2007a,b, 2019); Hirano and Porter (2019); Stoye (2009, 2012); Tetenov (2012); Manski and Tetenov (2016, 2019); and Kitagawa and Tetenov (2018).

To define the MMR criterion, we start by defining regret. For any STR δ and state $s \in S$, regret is the difference in ex post facto payoff between expected welfare with δ and expected welfare with the optimal choice

$$\begin{aligned} R(\delta, s) &:= \max_{\delta^*} W(\delta^*, P_s, Q_s) - W(\delta, P_s, Q_s) \\ &= \max[\mathbb{E}\{y(a)\}, \mathbb{E}\{y(b)\}] - W(\delta, P_s, Q_s) \end{aligned}$$

"max" is simply the MR that an STR δ generates over the state space:

$$\max_s R(\delta, s)$$

Finally, the MMR criterion chooses an STR that minimizes max regret

$$\delta^{\text{MMR}} := \arg\min_{\delta} \max_s R(\delta, s)$$

2.3 Asymptotically optimal STRs

When $N \to \infty$, Manski (2020) examines the MR of two STRs: the empirical success (ES) rule and the novel asymptotic MMR (AMMR) rule. As the name suggests, the ES rule chooses $\delta = a$ if

$$\mathbb{E}(y|t = a) \geq \mathbb{E}(y|t = b)$$

and $\delta = b$ if the inequality is reversed. The AMMR chooses $\delta = a$ if

$$2\{\mathbb{E}(y|t = b)\Pr(t = b) - \mathbb{E}(y|t = a)\Pr(t = a)\} + \Pr(t = a) - \Pr(t = b) \leq 0$$

and $\delta = b$ if the inequality is reversed. Manski (2020) shows that the AMMR (asymptotically) minimizes MR for this decision problem and derives conditions when the ES rule does so as well. However, there are no results for the sample analogs of these or any other STRs. To fill this gap, `wald_tc` is designed to provide numerical approximations of MR for STRs under a flexible specification of the state space.

`wald_tc` offers two built-in STRs for applications where only (t, y) are observable: the ES STR and the AMMR STR, which are sample analogs of the rules described above. There are also two built-in STRs for an IVs framework that are explained in section 3. Denoting sample probabilities and expectations with a hat [that is, $\widehat{\mathbb{E}}(\cdot)$], the ES STR chooses $\delta = a$ if $\widehat{E}(y|t = a) \geq \widehat{E}(y|t = b)$ and $\delta = b$ if the inequality is reversed. The AMMR chooses $\delta = a$ if

$$2\{\widehat{\mathbb{E}}(y|t = b)\widehat{\Pr}(t = b) - \widehat{\mathbb{E}}(y|t = a)\widehat{\Pr}(t = a)\} + \widehat{\Pr}(t = a) - \widehat{\Pr}(t = b) \leq 0$$

and $\delta = b$ if the inequality is reversed.

2.4 Computing MR

Now, we go over the algorithm for numerically computing MR for an STR δ over a feasible state space of data-generating processes. Let the s subscript of objects in step 1 denote that these are "primitives" that define state s. There are five such primitives that `wald_tc` allows, listed in step 1 starting with $\Pr_s(t = a)$, which is why the superset $\widetilde{S}$ in step 0 has five dimensions.

The algorithm first defines a five-dimensional cube $\widetilde{S}$, which contains infeasible states and then iterates across each state in $\widetilde{S}$ to check its feasibility according to user-defined constraints C, finally arriving at the set of feasible states S. The reason that the algorithm iterates over nonfeasible states in $\widetilde{S}$ is that the complexity of the constraints C prevents `wald_tc` from being able to explicitly define S as a function of C in one step. Instead, the algorithm must use the two-step process of defining $\widetilde{S}$ and then checking each state in $\widetilde{S}$ against C.

Outline of the algorithm

0. Let the user specify a superset of the state space $\widetilde{S} \subseteq [0, 1]^5$, STR δ, and some set of constraints C that implicitly define the state space $S \subseteq \widetilde{S} = (s \in \widetilde{S} : C \text{ satisfied})$.
1. Fix a state $[\Pr_s(t = a),\ \Pr_s\{y(a) = 1|t = a\},\ \Pr_s\{y(b) = 1|t = b\},\ \Pr_s\{y(b) = 1|t = a\},\ \Pr_s\{y(a) = 1|t = b\}],\ s \in \widetilde{S}$. If these parameters satisfy C, go to step 2. If not, skip this state and proceed to step 5.
2. Given s, draw N observations $(t_i)_{i=1}^N$ from a Bernoulli distribution with $p = \Pr_s(t = a)$. For each $i \in (1, \ldots, N)$, draw y_i from a Bernoulli distribution with $p = \Pr_s\{y(t_i) = 1|t = t_i\}$. Compute $\delta(\cdot)$ using $\{(t_i, y_i)\}_{i=1}^N$. Call this decision δ_k.
3. Repeat step 2 T times, and use the values $(\delta_k)_{k=1}^T$ to approximate $\Pr_s\{\delta(\psi) = a\}$ by its sample average $\widehat{\Pr}\{\delta(\psi) = a\}$.

4. Compute approximate $W(\delta, P_s, Q_s)$ using $\widehat{\Pr}\{\delta(\psi) = a\}$:

$$\widehat{W}(\delta, P_s, Q_s) = \mathbb{E}\{y(a)\}\widehat{\Pr}\{\delta(\psi) = a\} + \mathbb{E}\{y(b)\}[1 - \widehat{\Pr}\{\delta(\psi) = a\}]$$

Compute approximate regret for δ and s:

$$\widehat{R}(\delta, s) = \max[E\{y(a)\}, \mathbb{E}\{y(b)\}] - \widehat{W}(\delta, P_s, Q_s)$$

5. Repeat steps 1–4 for each $s \in S$.

6. Return $\max_{s \in S}\ \widehat{R}(\delta, s)$, the approximate MR.

Although the algorithm described above is simple, searching over a continuous space such as $S = [0, 1]^5$ is intractable. Moreover, the user's constraints C mean that the shape of S may be harder to explicitly describe than $[0, 1]^5$. To enable feasible computation, `wald_tc` specifies the state space as a finite grid over $[0, 1]^5$, subject to a set of flexible constraints available to the user (that are described in section 4.2). For example, the user may specify that $\Pr_s(t = a) \in [0.05, 0.9]$, $\Pr_s\{y(a) = 1|t = a\} \geq \Pr_s\{y(b) = 1|t = a\}$, or $|\mathbb{E}\{y(a)\} - \mathbb{E}\{y(b)\}| \leq 0.5$ based on knowledge about their setting.

3 Extension to the IV setting

3.1 Classic IV (CIV) approach

A common approach to the problem of identifying $\mathbb{E}\{y(a)\}$ and $\mathbb{E}\{y(b)\}$ begins by introducing a linear model for observed y,

$$y = \beta_0 + \beta_1 d_a + \varepsilon$$

where $d_a = \mathbb{1}(t = a)$ is a dummy for treatment a and $\mathbb{E}(\varepsilon) = 0$. β_1 is interpreted as the treatment effect of a versus b, common to all $j \in J$. Ideally, the planner could impose the mean independence assumption $\mathbb{E}(\varepsilon|d_a) = 0$, which implies $\mathbb{E}\{y(\widetilde{t})|t = a\} = \mathbb{E}\{y(\widetilde{t})|t = b\}$ and point-identifies β_1. However, with nonrandomized treatment, this equality may be violated. To overcome this difficulty, the DM could use an IV v, which satisfies the following assumptions:

$$\text{Cov}(v, \varepsilon) = 0 \text{ and } \text{Cov}(v, d_a) \neq 0 \tag{2}$$

It follows that the DM can asymptotically point-identify $\mathbb{E}\{y(a)\} = \beta_0 + \beta_1$ and $\mathbb{E}\{y(b)\} = \beta_0$ with

$$\begin{aligned} \beta_1 &= \frac{\text{Cov}(v, y)}{\text{Cov}(v, d_a)} \\ \beta_0 &= \mathbb{E}(y) - \beta_1 \mathbb{E}(d_a) \end{aligned} \tag{3}$$

The DM would use this knowledge to assign everybody in J to treatment a if $\beta_1 > 0$ and treatment b if $\beta_1 < 0$. See Manski (2007a, chap. 7) for a textbook exposition.

3.2 Partial identification IV (PIIV) approach

In some settings, the DM may find that the assumptions of the CIV approach are not credible. The cases where the CIV approach fails are if treatment effects are heterogeneous across persons or if the instrument v is invalid [that is, $\text{Cov}(v, \varepsilon) \neq 0$].

Even if the CIV approach is unsuitable, the DM can still partially identify $\mathbb{E}\{y(a)\}$ and $\mathbb{E}\{y(b)\}$ using an IV v as shown in Manski (2007a). The missing-data setting used there is applicable here by reinterpreting y as two different observable variables $y(a)$ and $y(b)$ for which data are missing when $t = b$ or $t = a$, respectively. We will refer to this method as the PIIV approach. The PIIV approach will be valid if treatment effects are heterogeneous across persons. The PIIV approach assumes only that $\mathbb{E}\{y(\widetilde{t})\}$ is independent of v [that is, $\mathbb{E}\{y(\widetilde{t})|v\} = \mathbb{E}\{y(\widetilde{t})\}$ for $\widetilde{t} \in (a, b)$]. One example of where the PIIV approach would be preferred over the CIV approach is if the DM does not wish to assume a linear homogeneous model of treatment responses but still assumes that $\mathbb{E}\{y(\widetilde{t})\}$ is independent of v. In this case, ε and therefore (2) are not well defined, while the PIIV approach retains its IV assumption.

Suppose that v can take values in some set K. The PIIV approach begins with an assumption that $\mathbb{E}\{y(\widetilde{t})|v\} = \mathbb{E}\{y(\widetilde{t})\}$. So we can partially identify $\mathbb{E}\{y(\cdot)|v = k\}$ as $N \to \infty$,

$$\begin{aligned}\mathbb{E}\{y(\widetilde{t})|v = k\} \in [\,&\Pr(y = 1|t = \widetilde{t}, v = k)\Pr(t = \widetilde{t}|v = k),\\ &\Pr(y = 1|t = \widetilde{t}, v = k)\Pr(t = \widetilde{t}|v = k) + \{1 - \Pr(t = \widetilde{t}|v = k)\}]\end{aligned}$$

for all $\widetilde{t} \in (a, b),\ k \in K$. Applying the independence assumption, we impose that $\mathbb{E}\{y(\widetilde{t})|v = k\} = \mathbb{E}\{y(\widetilde{t})|v = k'\},\ \widetilde{t} \in (a, b),\ k, k' \in K$, so

$$\begin{aligned}\mathbb{E}\{y(\widetilde{t})\} \in \Big[&\max_{k \in K} \mathbb{E}(y|t = \widetilde{t}, v = k)\Pr(t = \widetilde{t}|v = k),\\ &\min_{k \in K} \mathbb{E}(y|t = \widetilde{t}, v = k)\Pr(t = \widetilde{t}|v = k) + \{1 - \Pr(t = \widetilde{t}|v = k)\}\Big] \qquad (4)\end{aligned}$$

for each $\widetilde{t} \in (a, b)$, where we denote each bound with $[\text{LB}_{\widetilde{t}}, \text{UB}_{\widetilde{t}}]$. Note that if $\text{LB}_a > \text{UB}_a$ or $\text{LB}_b > \text{UB}_b$, then the independence assumption is refuted and another identification strategy is needed. Otherwise, the DM can map these partial identification bounds to a decision. Using the MMR criterion, the DM would assign everybody in J treatment a if the MR from choosing b is higher than that from choosing a:

$$\text{UB}_a - \text{LB}_b \ \geq\ \text{UB}_b - \text{LB}_a$$

If this inequality is reversed, the DM assigns treatment b to everybody in J.

3.3 IV STRs

Similarly to the ES and AMMR rules, the finite sample performance of the CIV and PIIV methods in terms of MR is unknown. To address this challenge with numerical

approximation, `wald_tc` allows the user to evaluate the finite sample MR performance of two built-in IV STRs, one for CIV and the other for PIIV. `wald_tc` also allows users to evaluate arbitrary user-defined IV STRs. For computational simplicity, `wald_tc` imposes that v takes values in $(0,1)$ for both the CIV and PIIV approaches. This is done to keep the dimensionality of the state space low and is not a fundamental requirement for the formal problem.

The framework described in sections 2.1 and 2.2 is easily extended to allow STRs that use an IV. The main difference is that there is an additional binary observable variable v, so the DM draws a sample $\psi \sim Q$ of N triples (t_i, y_i, v_i) with sample space $\Psi \subseteq \times^N\{(a,b)\times(0,1)\times(0,1)\}$. We will restrict attention to the case where (t_i, y_i, v_i) are independent and identically distributed so $\psi \sim \times^N \widetilde{Q},\ (t,y,v) \sim \widetilde{Q}$. Then, the DM chooses an STR $\delta : \times^N\{(a,b)\times(0,1)\times(0,1)\} \ \rightarrow (a,b)$, with the same ex post facto payoff as in (1).

Now, we can construct the two built-in IV STRs using the two IV approaches described above and use MR to evaluate their performance. The CIV STR calculates the sample analog of (3),

$$\widehat{\beta}_1 = \frac{\widehat{\mathrm{Cov}}(v,y)}{\widehat{\mathrm{Cov}}(v,d_a)}$$

choosing $\delta = a$ if $\widehat{\beta}_1 \geq 0$ and $\delta = b$ if $\widehat{\beta}_1 < 0$.

For the PIIV STR, take the sample analog of (4) to get

$$(\widehat{\mathrm{LB}}_{\widetilde{t}}, \widehat{\mathrm{UB}}_{\widetilde{t}}) = \Big[\max_{k\in(0,1)} \widehat{\mathbb{E}}(y|t=\widetilde{t}, v=k)\widehat{\Pr}(t=\widetilde{t}|v=k), \\ \min_{k\in(0,1)} \widehat{\mathbb{E}}(y|t=\widetilde{t}, v=k)\widehat{\Pr}(t=\widetilde{t}|v=k) + \{1-\widehat{\Pr}(t=\widetilde{t}|v=k)\}\Big]$$

for each $\widetilde{t} \in (a,b)$. If $\widehat{\mathrm{LB}}_a \leq \widehat{\mathrm{UB}}_a$ and $\widehat{\mathrm{LB}}_b \leq \widehat{\mathrm{UB}}_b$, then the DM assigns $\delta = a$ if $\widehat{\mathrm{UB}}_a - \widehat{\mathrm{LB}}_b \geq \widehat{\mathrm{UB}}_b - \widehat{\mathrm{LB}}_a$ and $\delta = b$ if the inequality is reversed. If $\widehat{\mathrm{LB}}_a > \widehat{\mathrm{UB}}_a$ or $\widehat{\mathrm{LB}}_b > \widehat{\mathrm{UB}}_b$ and the independence assumption is refuted, `wald_tc` specifies that the DM will ignore v and use the AMMR STR described in section 2.3.

As a general result, the MR of an STR can be evaluated on any given state space. Specifically, the MR of the CIV and the PIIV STRs can be found for state spaces containing states that violate the relevant IV assumptions. One feature of `wald_tc` is that it allows users to flexibly specify state spaces and include states that violate the IV assumptions of an IV STR.

At first glance, it may be counterintuitive to examine a DM's choice of an STR when the assumptions that motivate it are violated. Manski (2020) provides some guidance on this matter with the discussion of model choice. Although "the state space should include all states that the planner believes feasible" (Manski 2020), such a high degree of generality may preclude decision making in reality. To simplify the problem enough to allow choice, a DM might select a model as a surrogate simplified reality. A DM "using

a model acts as if the model space is the state space" (Manski 2020). Therefore, a DM who uses the PIIV STR uses a model that takes the relevant IV assumption as true even if the DM suspects this assumption to be wrong in some feasible states of the world. As the statistician George Box wrote, "All models are wrong, but some are useful" (Box 1979). Therefore, `wald_tc` allows the user to explore the settings in which IV STRs may be useful for DMs even if the underlying models used by the DMs are wrong.

3.4 Computing MR in the IV setting

The algorithm for computing MR in the IV setting is similar to the standard algorithm described in section 2.4, except for six additional state parameters and an additional observable variable. The 6 additional state parameters mean that there are 11 state primitives (which are listed in step 1), so $\widetilde{S}$ is an 11-dimensional cube. These changes appear only in steps 0–2:

Modified steps in IV algorithm

0. Let the user specify a superset of the state space $\widetilde{S} \subseteq [0,1]^{11}$, STR δ, and some set of constraints C that implicitly define the state space $S \subseteq \widetilde{S} = (s \in \widetilde{S} : C \text{ satisfied})$.

1. Fix a state $[\Pr_s(t=a),\ \Pr_s\{y(a)=1|t=a\},\ \Pr_s\{y(b)=1|t=b\},\ \Pr_s\{y(b)=1|t=a\}, \Pr_s\{y(a)=1|t=b\},\ \Pr_s\{v=1|t=a,y(a)=1\}, \Pr_s\{v=1|t=a,y(a)=0\}, \Pr_s\{v=1|t=b,y(b)=1\},\ \Pr_s\{v=1|t=b,y(b)=0\} \Pr_s\{y(a)=1|v=1,t=b\},\ \Pr_s\{y(b)=1|v=1,t=a\}],\ s \in \widetilde{S}$. If these parameters satisfy C, go to step 2. If not, skip this state and proceed to step 5.

2. Given s, draw N observations $(t_i)_{i=1}^N$ from a Bernoulli distribution with $p = \Pr_s(t=a)$. For each $i \in (1,\ldots,N)$, draw y_i from a Bernoulli distribution with $p = \mathbb{E}_s\{y(t_i)|t=t_i\}$. For each $i \in (1,\ldots,N)$, draw v_i from a Bernoulli distribution with $p = \Pr_s\{v=1|t=t_i, y(t_i)=y_i\}$. Compute $\delta(\cdot)$ using $\{(t_i,y_i,v_i)\}_{i=1}^N$. Call this decision δ_k.

Qualitatively, the MR algorithm for the IV setting is the same as the algorithm for the standard setting. However, the IV approach is more computationally intensive because the state space now becomes a grid over $[0,1]^{11}$ as opposed to $[0,1]^5$ in the standard problem. Additionally, the user can impose IV-specific constraints such as a restriction on the distribution of v, for example, $\Pr_s\{v=1|t=k,y(a)=l\} \in [0.01, 0.99],\ k \in (a,b),\ l \in [0,1]$.

Note that although the parameters $\Pr_s\{y(a)=1|v=1,t=b\}$ and $\Pr_s\{y(b)=1|v=1,t=a\}$ are not used in the data simulation in step 2, they are used to check whether user-defined restrictions are satisfied in step 1. Specifically, these parameters are used to check the assumption violation restrictions described in the next section.

3.5 Restricting the violation of identifying assumptions

The default grid search includes states that violate either the CIV assumptions of $y = \beta_0 + \beta_1 d_a + \varepsilon$ and (2) or the PIIV independence assumption of $\mathbb{E}\{y(\widetilde{t})|v\} = \mathbb{E}\{y(\widetilde{t})\}$. This allows users to determine the MR penalty of relying on a false IV assumption, including the CIV case where the assumption is not refutable. To allow for a more precise analysis of the effect of using a false assumption, `wald_tc` allows users to set an acceptable limit on the magnitude of violation of these assumptions. Then, the program will search only in states where these assumptions are not violated more than the limit permits.

Under PIIV, an intuitive way to quantify violation is the distance

$$D_{\text{PIIV}}(s) := \max_{\widetilde{t} \in (a,b)} |\Pr\{y(\widetilde{t}) = 1|v = 1\} - \Pr\{y(\widetilde{t}) = 1|v = 0\}| \tag{5}$$

where this distance lies in $[0, 1]$ and is 0 if the independence assumption is satisfied. `wald_tc` allows the user to set an upper bound on this distance, for example, only search over states where $D_{\text{PIIV}} \leq 0.5$.

Under the CIV approach, if the model is well specified and the identifying assumptions of (2) are true, then the asymptotic IV estimator β_1 is equal to the ATE of a versus b: $\beta_1 = \mathbb{E}\{y(a) - y(b)\}$. Given a state s, `wald_tc` can always compute the asymptotic value of the CIV estimator β_1, including when $\beta_1 = \pm\infty$ if $\text{Cov}(v, d_a) = 0$. Then, under CIV, an intuitive way to quantify violation of the specification and identify assumptions is the distance

$$D_{\text{CIV}}(s) := |\beta_1 - [\mathbb{E}\{y(a)\} - \mathbb{E}\{y(b)\}]| \tag{6}$$

where this distance lies in $[0, \infty)$ and is 0 if the IV estimator identifies the ATE.

4 The wald_tc command

4.1 Syntax

The syntax for `wald_tc` is

```
wald_tc command_name, samp_size(#) [ta_l(#) ta_r(#) yata(#) ybtb(#)
   yatb(#) ybta(#) eya_l(#) eya_r(#) eyb_l(#) eyb_r(#) d_ey(#)
   d_ya(#) d_yb(#) v_l(#) v_r(#) vtay1(#) vtay0(#) vtby1(#) vtby0(#)
   d_piiv(#) d_civ(#) mon unconf rand_tb grid(#) mc_iter(#) user_def
   iv_mode]
```

command_name specifies the name of the STR to be used. It must be either a built-in STR (see section 4.3 for details) or any user-specified STR that is written as an e-class command that returns 0 or 1 to `e(b)` and takes a two- or three-column vector according to whether an IV is included (see section 4.4 for details).

4.2 Options

`samp_size(#)` specifies the size of the sample (that is, N in sections 2 and 3). `samp_size()` is required.

Options to specify restrictions to the parameter space:

`ta_l(#)` specifies the lower bound on $\Pr(t = a)$, the probability that the DM observes treatment a. The default is `ta_l(0)`.

`ta_r(#)` specifies the upper bound on $\Pr(t = a)$. The default is `ta_r(1)`.

`yata(#)` specifies a point value for $\Pr\{y(a) = 1|t = a\}$, that is, the probability of observing $y = 1$ conditional on $t = a$. This value can be in $[0, 1]$ or the default `yata(-1)`, which specifies that $\Pr\{y(a) = 1|t = a\}$ is not restricted to a point.

`ybtb(#)` specifies a point value for $\Pr\{y(b) = 1|t = b\}$. This value can be in $[0, 1]$ or the default `ybtb(-1)`, which specifies that $\Pr\{y(b) = 1|t = b\}$ is not restricted to a point.

`yatb(#)` specifies a point value for $\Pr\{y(a) = 1|t = b\}$. This value can be in $[0, 1]$ or the default `yatb(-1)`, which specifies that $\Pr\{y(a) = 1|t = b\}$ is not restricted to a point.

`ybta(#)` specifies a point value for $\Pr\{y(b) = 1|t = a\}$. This value can be in $[0, 1]$ or the default `ybta(-1)`, which specifies that $\Pr\{y(b) = 1|t = a\}$ is not restricted to a point.

`eya_l(#)` specifies the lower bound on $\mathbb{E}\{y(a)\}$, the expected treatment effect of a. The default is `eya_l(0)`.

`eya_r(#)` specifies the upper bound on $\mathbb{E}\{y(a)\}$. The default is `eya_r(1)`.

`eyb_l(#)` specifies the lower bound on $\mathbb{E}\{y(b)\}$. The default is `eyb_l(0)`.

`eyb_r(#)` specifies the upper bound on $\mathbb{E}\{y(b)\}$. The default is `eyb_r(1)`.

`d_ey(#)` specifies the upper bound on the distance $|\mathbb{E}\{y(a)\} - \mathbb{E}\{y(b)\}|$ between the expected treatment effects of the two treatments. The default is `d_ey(1)`.

`d_ya(#)` specifies the upper bound on the distance $|\mathbb{E}\{y(a)|t = a\} - \mathbb{E}\{y(a)|t = b\}|$, the distance between the observed and the missing conditional expectations of $\mathbb{E}\{y(a)\}$. The default is `d_ya(1)`. For example, choosing `d_ya(0)` imposes that $y(a)$ is independent of observed treatment t.

`d_yb(#)` specifies the upper bound on the distance $|\mathbb{E}\{y(b)|t = a\} - \mathbb{E}\{y(b)|t = b\}|$. The default is `d_yb(1)`.

`v_l(#)` specifies the lower bound on each of the four probabilities in

$\Pr\{v = 1|t = \widetilde{t}, y(\widetilde{t}) = k\},\ \widetilde{t} \in (a, b),\ k \in (0, 1)$. The default is `v_l(0)`.

`v_r(#)` specifies the upper bound on each of the four probabilities in

$\Pr\{v = 1|t = \widetilde{t}, y(\widetilde{t}) = k\},\ \widetilde{t} \in (a, b),\ k \in (0, 1)$. The default is `v_r(1)`.

`vtay1(#)` specifies a point value for $\Pr(v = 1|t = a, y = 1)$. This value can be in $[0, 1]$ or the default `vtay1(-1)`, which specifies that $\Pr(v = 1|t = a, y = 1)$ is not restricted to a point. `vtay1()` overrides limits set in `v_l()` and `v_r()` for $\Pr(v = 1|t = a, y = 1)$.

`vtay0(#)` specifies a point value for $\Pr(v = 1|t = a, y = 0)$. This value can be in $[0, 1]$ or the default `vtay0(-1)`, which specifies that $\Pr(v = 1|t = a, y = 0)$ is not restricted to a point. `vtay0()` overrides limits set in `v_l()` and `v_r()` for $\Pr(v = 1|t = a, y = 0)$.

`vtby1(#)` specifies a point value for $\Pr(v = 1|t = b, y = 1)$. This value can be in $[0, 1]$ or the default `vtby1(-1)`, which specifies that $\Pr(v = 1|t = b, y = 1)$ is not restricted to a point. `vtby1()` overrides limits set in `v_l()` and `v_r()` for $\Pr(v = 1|t = b, y = 1)$.

`vtby0(#)` specifies a point value for $\Pr(v = 1|t = b, y = 0)$. This value can be in $[0, 1]$ or the default `vtby0(-1)`, which specifies that $\Pr(v = 1|t = b, y = 0)$ is not restricted to a point. `vtby0()` overrides limits set in `v_l()` and `v_r()` for $\Pr(v = 1|t = b, y = 0)$.

`d_piiv(#)` specifies the upper bound on the distance D_{PIIV} as defined in (5). The default is `d_piiv(1)`. Setting the option to 0 imposes the IV assumption that motivates the use of the PIIV STR.

`d_civ(#)` specifies the upper bound on the distance D_{CIV} as defined in (6). The distance is in $[0, \infty]$, and the user can set `d_civ(#)` to any finite nonnegative real value. The default is `d_civ_spec(-1)`, which represents an "allowable" upper bound of ∞.

`mon` imposes the monotone treatment selection assumption that

$$\Pr\{y(a) = 1|t = a\} - \Pr\{y(b) = 1|t = a\} \geq 0$$
$$\Pr\{y(b) = 1|t = b\} - \Pr\{y(a) = 1|t = b\} \geq 0$$

This could represent a setting where each sampled individual chooses t to maximize his or her own outcome. See Manski (2007a) for a discussion of this assumption, which weakens unconfoundedness.

`unconf` imposes the unconfoundedness assumption that

$$\Pr\{y(a) = 1|t = a\} = \Pr\{y(a) = 1|t = b\}$$
$$\Pr\{y(b) = 1|t = a\} = \Pr\{y(b) = 1|t = b\}$$

This could represent a setting where treatment is randomized, as discussed in section 2.1. Note that specifying `unconf` is equivalent to, and a shortcut for, specifying `d_ya(0) d_yb(0)`.

`rand_tb` imposes a randomized tiebreak rule for the chosen non-IV STR, which chooses $\delta = a$ with probability 0.5 in cases where there is no uniquely prescribed decision. By default, `wald_tc` deterministically chooses $\delta = a$ in these tiebreak cases. More specifically, the chosen tiebreak rule is applied in the following four cases: 1) with the

ES rule, when only one treatment is observed in a sample; 2) with the AMMR rule, when only one treatment $\widehat{t}$ is observed in a sample and $\widehat{E}(y|t=\widehat{t})=0.5$; 3) with the ES rule, when both treatments are observed in a sample and $\widehat{\mathbb{E}}(y|t=a)\geq\widehat{\mathbb{E}}(y|t=b)$; and 4) with the AMMR rule, when both treatments are observed in a sample and $2\{\widehat{\mathbb{E}}(y|t=b)\widehat{\Pr}(t=b)-\widehat{\mathbb{E}}(y|t=a)\widehat{\Pr}(t=a)\}+\widehat{\Pr}(t=a)-\widehat{\Pr}(t=b)=0$.

Options to specify the computational details:

`grid(#)` specifies the number of grid points used to search over each parameter, for nonsingleton parameters. The default is `grid(10)`. For example, suppose the user sets `grid(5)` with `ta_l(0.4)` and `ta_r(0.6)`. This specifies that `wald_tc` will set $\Pr(t=a)$ to each point in the set $(0.4, 0.45, 0.5, 0.55, 0.6)$ in its search over the parameter space. On the other hand, if the user sets `ta_l(0.4)` and `ta_r(0.4)`, then `wald_tc` will set only $\Pr(t=a)=0.4$ regardless of `grid(#)`.

`mc_iter(#)` specifies the number of Monte Carlo (MC) iterations used to compute the regret of an STR in a state. This number corresponds to T in step 3 of the `wald_tc` algorithm described in section 2.4. The default is `mc_iter(1000)`.

`user_def` specifies that the STR being used is user defined (not one of the built-in STRs described in section 4.3).

`iv_mode` specifies that the IV framework applies. If `iv_mode` is enabled, the STR takes in data with three variables: treatment, outcome, and IV [that is, (t, y, v) in section 3]. If `iv_mode` is not enabled, the STR takes in data with the standard two variables of treatment and outcome [that is, (t, y) in section 2].

4.3 Built-in STRs

There are two built-in STRs for the standard framework where the STR takes data in the form (t, y) and two built-in STRs for the IV framework where the STR takes data in the form (t, y, v). `wald_tc` uses the standard framework by default and switches to the IV framework if the user specifies `iv_mode`. The built-in standard STRs are derived in section 2.3, and the built-in IV STRs are derived in section 3.3. Finally, the hats over parameters indicate sample values used by the STR.

Standard STRs:

`ES` is the Empirical Success rule, which chooses treatment $\delta=a$ when $\widehat{\mathbb{E}}(y|t=a)\geq\widehat{\mathbb{E}}(y|t=b)$. In the edge case where the ES rule observes a sample with only one treatment, it will choose $\delta=a$.

`AMMR` is the Asymptotic Minimum Maximum Regret rule, which chooses treatment $\delta=a$ when

$$2\{\widehat{\mathbb{E}}(y|t=b)\widehat{\Pr}(t=b)-\widehat{\mathbb{E}}(y|t=a)\widehat{\Pr}(t=a)\}+\widehat{\Pr}(t=a)-\widehat{\Pr}(t=b)\leq 0$$

Note that when the AMMR rule observes only one treatment $\widehat{t}$, it will choose $\delta = \widehat{t}$ if $\widehat{\mathbb{E}}(y|t=\widehat{t}) > 0.5$, breaking ties at 0.5 in favor of $\delta = a$.

IV STRs:

`CIV` is the Classic IV rule, which chooses treatment $\delta = a$ when

$$0 \leq \frac{\widehat{\text{Cov}}(v,y)}{\widehat{\text{Cov}}\{v, \mathbb{1}(t=a)\}}$$

`PIIV` is the Partial Identification IV rule, which computes the bounds

$$(\widehat{\text{LB}}_{\widetilde{t}}, \widehat{\text{UB}}_{\widetilde{t}}) = \Big[\max_{k\in(0,1)} \widehat{\mathbb{E}}(y|t=\widetilde{t}, v=k)\widehat{\Pr}(t=\widetilde{t}|v=k), \min_{k\in(0,1)} \widehat{\mathbb{E}}(y|t=\widetilde{t}, v=k)\widehat{\Pr}(t=\widetilde{t}|v=k) + \{1 - \widehat{\Pr}(t=\widetilde{t}|v=k)\}\Big]$$

for each $\widetilde{t} \in (a,b)$. If $\widehat{\text{LB}}_a \leq \widehat{\text{UB}}_a$ and $\widehat{\text{LB}}_b \leq \widehat{\text{UB}}_b$, then PIIV chooses treatment $\delta = a$ when $\widehat{\text{UB}}_a - \widehat{\text{LB}}_b \geq \widehat{\text{UB}}_b - \widehat{\text{LB}}_a$.

If instead $\widehat{\text{LB}}_a > \widehat{\text{UB}}_a$ or $\widehat{\text{LB}}_b > \widehat{\text{UB}}_b$, `PIIV` chooses treatment $\delta = a$ when

$$2\{\widehat{\mathbb{E}}(y|t=b)\widehat{\Pr}(t=b) - \widehat{\mathbb{E}}(y|t=a)\widehat{\Pr}(t=a)\} + \widehat{\Pr}(t=a) - \widehat{\Pr}(t=b) \leq 0$$

4.4 Implementation details

Details on user-defined STRs

`wald_tc` allows use of a built-in STR (see section 4.3) or of any user-defined STR if it is an e-class Stata command that returns to `e(b)`. This command should return 1 if the STR chooses treatment a and 0 if the STR chooses treatment b. If `iv_mode` is not specified, then the command should take two data columns, where the first column is the indicator for treatment a: $d_a := \mathbb{1}(t=a)$, and the second column is the observed treatment outcome $y := y(t)$. If `iv_mode` is specified, then the command should take three data columns, where the first is d_a, the second is y, and the third is v, the IV. Each data point in the input matrix will be either 0 and 1, and there should be no missing data (`.`). However, note that the missing-data problem that motivates `wald_tc` is still present, with $y(t_{-i})$ remaining unobserved for each individual.

Details on option roles

The options of `wald_tc` can be separated into three groups: algorithm settings, simple parameter restrictions, and compound parameter restrictions. Algorithm settings alter the base problem and algorithm implementation and are defined by `samp_size()`, `grid()`, `mc_iter()`, `user_def`, and `iv_mode`. Simple parameter restrictions bound individual primitive elements of the state space as described in step 1 of the algorithm

overviews in sections 2.3 and 3.4. The simple restrictions are `ta_l()`, `ta_r()`, `yata()`, `ybtb()`, `yatb()`, `ybta()`, `v_l()`, `v_r()`, `vtay1()`, `vtay0()`, `vtby1()`, and `vtby0()`. The remaining options are compound parameter restrictions that are more complicated and impose logical constraints on combinations of primitive state-space parameters, for example, `d_ya(0.2)` imposing that $|\Pr_s\{y(a) = 1|t = a\} - \Pr_s\{y(a) = 1|t = b\}| \leq 0.2$. Following the algorithm notation in sections 2.3 and 2.4, the simple parameter restrictions determine $\widetilde{S} \subseteq [0,1]^5$ or $\widetilde{S} \subseteq [0,1]^{11}$ (depending on `iv_mode`), while the compound parameter restrictions C implicitly define the state space $S \subseteq \widetilde{S}$.

Details on grid search

By default, `wald_tc` searches over a grid of size `grid()` for each parameter, with each iteration looking at every unique 5-tuple or 11-tuple of parameters on the grid $\widetilde{S}$. However, if a simple (parameter) restriction bounds a parameter [like $\Pr_s(t = a)$] to a single point, then to improve speed, `wald_tc` will "search" over a one-point grid for this parameter. For example, the user may fix $\Pr_s(t = a) = 0.7$ with `ta_l(0.7) ta_r(0.7)`, which will improve execution speed.

In addition, because the set of compound restrictions C imposes $S \subseteq \widetilde{S}$ implicitly, for each unique 5-tuple or 11-tuple on the grid $\widetilde{S}$, `wald_tc` checks C. As described in step 1 in sections 2.4 and 3.4, if a restriction in C is violated, `wald_tc` skips the state and continues searching over $\widetilde{S}$. If C is satisfied, `wald_tc` proceeds with the MC simulation and regret approximation.

To accommodate the Mata command `rbinomial()`, `wald_tc` cannot simulate data using parameters that are exactly 0 or 1. Thus, any parameter bounds that are specified as 0 or 1 will be transformed into 10^{-8} and $1 - 10^{-8}$. For example, if a user imposes `grid(3)` and leaves the default `ta_l(0)` and `ta_r(1)`, `wald_tc` will search over the grid $(10^{-8}, 0.5, 1 - 10^{-8})$ for $\Pr_s(t = a)$.

As a caveat, the use of grid search causes `wald_tc` to have poor accuracy in reporting the true MR when there are no identification issues [that is, $\mathbb{E}\{y(\widetilde{t})\} \approx \Pr(y = 1|t = \widetilde{t})$] and when sample sizes are large (that is, > 500 for a grid size of 10). The limitation of grid search for this case is that the region of the parameter space on which regret is substantially higher than zero shrinks with sample size, so a crude grid may include only parameter values where regret is almost zero. This issue should not arise in problems with partial identification, because the region of the parameter space with nonnegligible regret does not shrink with sample size.

Details on execution speed

In general, STRs that take in (t, y) will be much faster than IV STRs that take in (t, y, v), where the former do not have `iv_mode` specified and the latter do have `iv_mode` specified. The reason is that without `iv_mode`, the state space is some 5-dimensional subset of $[0,1]^5$, while with `iv_mode`, the state space is some 11-dimensional subset of $[0,1]^{11}$. Note that if `d_piiv(1)`, which it is by default, then `wald_tc` does not

calculate $D_{\text{PIIV}}(s)$. Then, `wald_tc` does not search over $\Pr_s\{y(a) = 1 | v = 1, t = b\}$ or $\Pr_s\{y(b) = 1 | v = 1, t = a\}$, reducing the state-space superset to $[0, 1]^9$.

Therefore, taking the default `grid(10)`, IV STRs will search over $10,000$ times more states as non-IV STRs if `d_piiv(1)`. If `d_piiv()` is set to less than 1, this ratio goes to $1,000,000$. Setting `d_piiv()` to less than 1 shrinks the state space but does so in a manner that makes the structure of the state space more complex. As `wald_tc` iterates over the state-space superset, it must check whether each state is feasible according to `d_piiv()`, requiring more computation time. Because there are `mc_iter()` samples in the MC simulation for each state, it is clear to see how `iv_mode` can affect speed.

To improve the speed of both IV and non-IV STRs, the user can decrease `samp_size()`, `grid()`, and `mc_iter()`. The user can also fix state parameters [for example, with `yata()`] to decrease the number of dimensions that `wald_tc` must search over.

Also, user-defined STRs will generally be much slower than built-in STRs because a user-defined STR forces `wald_tc` to call Stata in each MC iteration (that is, for each observed sample), whereas with the built-in STRs, `wald_tc` can generate all T samples concurrently in Mata. To improve speed, the user can decrease the number of MC iterations [by setting a lower `mc_iter()`] or decrease the parameter grid fineness [by decreasing `grid()`].

Advanced users can also improve the speed of custom STRs by adding them directly to the `wald_tc` code alongside the built-in STRs, allowing `wald_tc` to implement the STR directly in Mata.

Details on failure to find a solution

`wald_tc` starts its grid search with an initial value of 0 for MR. This is the value of regret in a state where the STR chooses the optimal decision. In the presence of uncertainty about the unobserved parameters [for example, $\Pr_s\{y(a) = 1 | t = b\}$], the MR of any STR will be above 0.

If the final result is an MR of 0, this means that the algorithm failed to approximate the real MR. `wald_tc` will alert the user of an ineffective run by throwing an error. This result can occur if user-set constraints are so restrictive that every state in $\widetilde{S}$ violates C so `wald_tc` skips every state in $\widetilde{S}$, so that $S = \emptyset$ and MR are not evaluated at all. Additionally, this can occur if $S \neq \emptyset$, but user-set constraints are so restrictive that each $s \in S$ leaves no uncertainty for the DM, leading to a trivial problem where MR is 0.

Either of these causes may be due to the constraints alone or due to the algorithm implementation. The culprit is the constraints alone if, even abstracting from the grid search or MC simulation, the "real" state space is empty or the "real" MR is 0. In this case, the user should loosen the simple or compound parameter restrictions.

If the culprit is the algorithm interpretation, then the "real" state space may be nonempty, and the "real" MR may be larger than 0, but `wald_tc` does not reach these states or approximates them poorly. In this case, the user should increase `grid()`, loosen the simple or compound parameter restrictions, or increase `mc_iter()`.

A note on sample size

Users of `wald_tc` may notice that for STRs that do not minimize MR, MR may increase with sample size. Although counterintuitive, this may be because for some non-MMR STRs, larger sample sizes lead to more statistical precision and hence more uniformly incorrect decisions, for example, choosing $\delta = a$ in 95% instead of 90% of samples. This will yield a larger MR if $\delta = a$ is the inferior choice in the state that generates MR. An STR that minimizes MR (at some sample size N) cannot have a lower MR for a lower sample size. To see why, suppose that the opposite is true, and notice that the same STR that ignores the last observation (that is, forcing a sample size of $N-1$) would have a lower MR for the same sample size N, meaning the original STR does not minimize MR.

4.5 Stored results

`wald_tc` stores the following in `r()`. For convenience, denote $s^* \in S$ as the state at which `wald_tc` generates the MR for the STR.

Scalars

`r(MR)`	MR of the STR
`r(treata_val)`	fraction of MC runs at s^* where STR chose $\delta = a$
`r(ta_val)`	$\Pr_{s^*}(t = a)$ at s^*
`r(yata_val)`	$\Pr_{s^*}\{y(a) = 1 \vert t = a\}$ at s^*
`r(ybtb_val)`	$\Pr_{s^*}\{y(b) = 1 \vert t = b\}$ at s^*
`r(yatb_val)`	$\Pr_{s^*}\{y(a) = 1 \vert t = b\}$ at s^*
`r(ybta_val)`	$\Pr_{s^*}\{y(b) = 1 \vert t = a\}$ at s^*
`r(eya_val)`	$\mathbb{E}\{y(a)\}$ at s^*
`r(eyb_val)`	$\mathbb{E}\{y(b)\}$ at s^*

IV scalars (only shown if `iv_mode` specified)

`r(vtay1_val)`	$\Pr_{s^*}(v = 1 \vert t = a, y = 1)$ at s^*
`r(vtay0_val)`	$\Pr_{s^*}(v = 1 \vert t = a, y = 0)$ at s^*
`r(vtby1_val)`	$\Pr_{s^*}(v = 1 \vert t = b, y = 1)$ at s^*
`r(vtby0_val)`	$\Pr_{s^*}(v = 1 \vert t = b, y = 0)$ at s^*
`r(beta1_val)`	asymptotic β_1 estimator at s^*; only shown if `CIV` specified as the STR

Macros

`r(cmd)`	`wald_tc`
`r(STR)`	the specified STR

5 Examples and empirical illustration

In this section, we illustrate the functionality of `wald_tc` through a series of examples. We will show just a few of the many ways users can use the `wald_tc` options to evaluate the effect of different environments or STRs on the DM's MR. We begin in section 5.1

with a few short introductory examples showing some commonly used assumptions. We proceed in section 5.2 with a canonical setting where the DM must choose between a status quo treatment or an innovation. We finish in section 5.3 with a modified setting of Bhattacharya, Shaikh, and Vytlacil (2012) where the DM must set a universal policy on Swan–Ganz catheterization use after observing data on treatment, outcomes, and an IV.

Note that after a successful execution, `wald_tc` displays the time elapsed in seconds. The examples below were run on a 4-core 1.30 GHz CPU, and the time elapsed will reflect this processing power.

5.1 Short introductory examples

Unrestricted state-space example

To run `wald_tc` without any restrictions to the state space, we just need to specify the STR (and `iv_mode` if it takes IV data) and a sample size. For example, take the following run of the ES STR with a sample size of 50 and no restrictions on the state space:

```
. wald_tc ES, samp_size(50)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .99999998
Time elapsed (sec): 4.732
```

Without strong restrictions to the state space, the ES rule will yield an MR value close to 1, which is the higher MR possible given the normalized outcomes used in `wald_tc`.

Note that the parameter space specified above contains states where $\Pr(t = a) \approx 0$ and $\Pr(t = a) \approx 1$. In this case, samples are likely to contain only $t = a$ or $t = b$ observations, which might be unrealistic and force the ES rule to use deterministic tiebreaking rules. To avoid this case, users can bound $\Pr(t = a)$ away from 0 and 1 using `ta_l()` and `ta_r()`. For example, the following run bounds $\Pr(t = a) \in (0.01, 0.99)$:

```
. wald_tc ES, samp_size(50) ta_l(0.01) ta_r(0.99)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .98999998
Time elapsed (sec): 4.704
```

Notice that MR has not fallen much. However, suppose we use randomized tiebreaks with option `rand_tb` to remove the impact of the deterministic tiebreaking rules in edge cases:

```
. wald_tc ES, samp_size(50) ta_l(0.01) ta_r(0.99) rand_tb
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .780860231
Time elapsed (sec): 4.75
```

Now, we see that MR is still high but far lower than before. In contrast, the AMMR rule performs better in an unrestricted state space, even with deterministic tiebreaks:

```
. wald_tc AMMR, samp_size(50)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .380518508
Time elapsed (sec): 4.701
```

Unconfoundedness example

Suppose we want to impose unconfoundedness, that is, the assumption that the ATE is the same across groups. We can impose this by specifying `unconf`, either alone or with other restrictions. One setting that satisfies this assumption is one with randomized assignment of treatment and perfect compliance. Take the following run:

```
. wald_tc AMMR, samp_size(50) unconf
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .371999993
Time elapsed (sec): 4.765
```

The imposition of unconfoundedness has slightly decreased the MR of the AMMR STR in this problem from 0.372 to 0.381.

Valid PIIV assumption example

Suppose we wish to impose the validity of the identifying assumption for the partial identification approach to IV data. We can do so by setting `d_piiv(0)`, which restricts the state space to states where $D_{\text{PIIV}} = 0$, where D_{PIIV} is defined in (5). Take the following run, which imposes this assumption with the PIIV STR, as well as imposes $\Pr(t = a) = 0.5$, $\Pr\{y(a) = 1|t = a\} = 0.5$, $\Pr\{y(a) = 1|t = b\} = 0.5$, $\Pr\{y(b) = 1|t = a\} = 0.6$:

```
. wald_tc PIIV, samp_size(50) d_piiv(0) ta_l(0.5) ta_r(0.5) iv_mode
> yata(0.5) yatb(0.5) ybtb(0.6)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .151799996
Time elapsed (sec): 293.636
```

Monotone treatment selection example

Suppose we would like to impose monotone treatment selection, which is relevant in a setting where individuals choose whichever treatment has a higher individual outcome. We can impose this assumption by specifying `mon`. For example, take the following run with `mon` specified:

```
. wald_tc AMMR, samp_size(50) mon
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .377679001
Time elapsed (sec): 4.825
```

The imposition of monotone selection leaves the MR roughly the same.

5.2 Canonical status quo versus innovation example

Here we examine a setting where the DM must choose between a status quo treatment a or an innovation b for the population after recording a sample of $N = 50$ treatments and outcomes in an observational study. We will examine a setting where we (as the user) know that the expected treatment effect of a is 0.5. This maps to `yata(0.5) yatb(0.5)`. Also, suppose we want to impose that (asymptotically) the sampling process will assign at least 10% of the population to each treatment, giving $\Pr(t = a) \in (0.1, 0.9)$. This maps to `ta_l(0.1) ta_r(0.9)`. Suppose we were interested in a setting where the ATE of b can be far worse but only somewhat better than the ATE of a. Specifically, suppose we bound $\mathbb{E}\{y(b)\} \in (0, 0.6)$, which maps to `eyb_r(0.6)`. Now, we can use `wald_tc` to compare STRs. To evaluate the ES rule, we run

```
. wald_tc ES, samp_size(50) yata(0.5) yatb(0.5) ta_l(0.1) ta_r(0.9) eyb_r(0.6)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .397999992
Time elapsed (sec): .515
```

We get an MR of 0.398. Suppose we are interested in the state that generates this regret. We run

```
. return list
scalars:
                  r(MR) =  .39799999204
          r(treata_val) =  .005
              r(ta_val) =  .9
            r(yata_val) =  .5
            r(ybtb_val) =  .9999999899999999
            r(yatb_val) =  .5
            r(ybta_val) =  1.00000000000e-08
             r(eya_val) =  .5
            r(eyb_val) =  .100000008
macros:
                 r(cmd) : "wald_tc"
                 r(STR) : "ES"
```

Although $\mathbb{E}\{y(a)\} = 0.5$ and $\mathbb{E}\{y(b)\} = 0.1$, `r(treata_val)=0.005` tells us that out of 1,000 MC runs, $\delta = b$ in 995 runs! Why does the ES rule assuredly choose the worse treatment? The answer lies in `r(ybtb_val)` and `r(ybta_val)`. The former shows $\Pr_{s^*}\{y(b)|t = b\} \approx 1$, and the latter shows $\Pr_{s^*}\{y(b)|t = a\} \approx 0$. Because the former is the sampling distribution for the DM's observed y conditional on $t = b$ and the DM never observes $y(b)$ conditional on $t = a$, this is a state that punishes the ES rule the most for ignoring unobserved counterfactual outcomes.

Now, let us evaluate the AMMR STR:

```
. wald_tc AMMR, samp_size(50) yata(0.5) yatb(0.5) ta_l(0.1) ta_r(0.9) eyb_r(0.6)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .297599994
Time elapsed (sec): .531
```

We see that AMMR performs better by yielding a smaller MR of approximately 0.298. Examining `r()`, we see that s^* appears to be the same, with $\mathbb{E}\{y(a)\} = 0.5$, $\mathbb{E}\{y(b)\} = 0.1$, $\Pr_{s^*}(t = a) = 0.9$, $\Pr_{s^*}\{y(b)|t = b\} \approx 1$, and $\Pr_{s^*}\{y(b)|t = a\} \approx 0$. However, `r(treata_val)=0.255`, meaning that in 255 out of 1,000 MC runs, the AMMR STR made the optimal choice $\delta = a$. This is because AMMR pays attention to missing data and may choose $\delta = a$ even when $\widehat{\Pr}\{y(b) = 1|t = b\} > \widehat{\Pr}\{y(a) = 1|t = a\}$. Is this a feature of a small number of MC simulations? We can run 100,000 MC simulations per state:

```
. wald_tc AMMR, samp_size(50) yata(0.5) yatb(0.5) ta_l(0.1) ta_r(0.9) eyb_r(0.6)
> mc_iter(100000)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .291723994
Time elapsed (sec): 56.486
```

We get a similar MR of 0.292 and `r(treata_val)=0.27069`, showing the robustness of this result.

Suppose that the result of $\Pr_{s^*}\{y(b)|t = b\} \approx 1$ and that $\Pr_{s^*}\{y(b)|t = a\} \approx 0$ seems extreme to us. We can impose that these values differ by no more than 0.5, which maps to `d_yb(0.5)`. This restriction limits the magnitude by which missing data can hurt a DM. This is especially true for the ES STR, which does not account for missing data:

```
. wald_tc ES, samp_size(50) yata(0.5) yatb(0.5) ta_l(0.1) ta_r(0.9) eyb_r(0.6)
> d_yb(0.5)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .200122218
Time elapsed (sec): .683
```

Now, we get a lower MR than before. As for AMMR:

```
. wald_tc AMMR, samp_size(50) yata(0.5) yatb(0.5) ta_l(0.1) ta_r(0.9) eyb_r(0.6)
> d_yb(0.5)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .185866663
Time elapsed (sec): .706
```

We also get a lower MR than before, but the gap between the MR of ES and AMMR has narrowed to 0.014 from 0.1. This is due to the `d_yb(0.5)` restriction, which limits the size of the mistake the ES rule can make from ignoring missing data. This is just one example of how `wald_tc` allows the user to flexibly adjust the data-generating process to quantify the effect of different restrictions on STR performance.

Demonstrating a user-defined function

Suppose we want to evaluate the performance of a DM who uses a canonical one-sided t test with $\alpha = 0.05$ to choose a treatment. Because this STR is not a built-in function, the user would have to specify a user-defined function to `wald_tc`, which represents this STR. To demonstrate this process, we have written `htest.ado` and included it as a stand-alone ado-file in the `wald_tc` package.

The t test STR takes the following null and alternate hypotheses:

$$H_0 : \Pr\{y(a) = 1 | t = a\} = \Pr\{y(b) = 1 | t = b\}$$
$$H_1 : \Pr\{y(a) = 1 | t = a\} < \Pr\{y(b) = 1 | t = b\}$$

If the STR rejects the null at a $\alpha = 0.05$ level, it will choose $\delta = b$; otherwise, it will choose $\delta = a$. Running this STR with the same parameters as before, we obtain

```
. wald_tc htest, samp_size(50) yata(0.5) yatb(0.5) ta_l(0.1) ta_r(0.9)
> eyb_r(0.6) user_def
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .383999992
Time elapsed (sec): 460.055
```

We see that the MR of this STR is only marginally better than that of the ES STR. Examining `return list`, we find the reason: although the MR-generating state s^* is the same, `r(treata_val)=0.04`, meaning that the t test was more conservative than the ES rule and sometimes chose $\delta = a$.

In terms of implementation, this run is around 500 times slower than our previous runs. This is due to the issue of having to run the STR in Stata for each MC sample as opposed to keeping all the computation in Mata. For example, when we modify the `wald_tc` code to have a built-in t test STR that runs in Mata, we get the following:

```
. wald_tc_modified htest, samp_size(50) yata(0.5) yatb(0.5) ta_l(0.1) ta_r(0.9)
> eyb_r(0.6)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .387999992
Time elapsed (sec): 1.015
```

This run is clearly much faster and is comparable with the speed of other built-in STRs. Moreover, `return list` shows us that it returns the same s^* with a similar MR.

5.3 Swan–Ganz catheterization example

In this section, we will demonstrate the functionality of `wald_tc` in settings with IVs. We will start by briefly describing the setting of Bhattacharya, Shaikh, and Vytlacil (2012) as motivation. The authors examine a procedure called Swan–Ganz catheterization, which is commonly applied to severely ill patients admitted to intensive care units. Although catheterization is associated with higher mortality, the authors find that "this

result is due to profound differences between the catheterized and noncatheterized patients: the former are much more severely ill than the latter" (Bhattacharya, Shaikh, and Vytlacil 2012). However, the authors do not find conclusive evidence regarding the net benefit of catheterizations (page 237):

> Our primary substantive finding is that catheterization improves mortality outcomes only in the short run, if at all, and in most cases we cannot rule out that it increases mortality in the long run.

To reach their findings, Bhattacharya, Shaikh, and Vytlacil (2012) use data that record whether a patient receives a catheter ($t = b$) or not ($t = a$), whether a patient dies within a specified time after admission ($y = 0$) or not ($y = 1$), and an IV indicating whether the patient is admitted on a weekend or a weekday. The authors claim that this is a valid instrument because mortality is uncorrelated with the day of admission, while patients admitted on a weekday are about four to eight percentage points more likely to be catheterized (Bhattacharya, Shaikh, and Vytlacil 2012).

Suppose that in this setting, a DM uses some another instrument v to try to find a more definitive result regarding the treatment effect of catheterization for a population. Suppose that the DM observes a sample ($N = 100$) and must choose between requiring that all relevant patients receive catheters ($\delta = a$) or none do ($\delta = b$). We will now demonstrate how `wald_tc` can help us understand the properties of these estimators.

CIV STR

Suppose we know that in our population, catheterization is somewhat common but associated with worse observable outcomes, with the following parameters:

$$\Pr(t = a) = 0.7$$
$$\Pr\{y(a) = 1 | t = a\} = 0.4$$
$$\Pr\{y(b) = 1 | t = b\} = 0.3$$

Suppose that we also believe our instrument v has a nonzero population variance [conditional on (t, y)]. To specify this concretely, we can tighten `v_l()` and `v_r()`, suppose to `v_l(0.1)` and `v_r(0.9)`. Then, we get

```
. wald_tc CIV, samp_size(100) yata(0.4) ybtb(0.3) ta_l(0.7) ta_r(0.7) iv_mode
> v_l(0.1) v_r(0.9)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .298349994
Time elapsed (sec): 474.567
```

Then, analyzing the state, we get

```
. return list
scalars:
                  r(MR) =  .2983499941499999
          r(treata_val) =  .585
              r(ta_val) =  .7
            r(yata_val) =  .4
            r(ybtb_val) =  .3
            r(yatb_val) =  1.00000000000e-08
            r(ybta_val) =  .9999999899999999
             r(eya_val) =  .280000003
             r(eyb_val) =  .7899999929999999
           r(vtay1_val) =  .5444444444444445
           r(vtay0_val) =  .1888888888888889
           r(vtby1_val) =  .6333333333333333
           r(vtby0_val) =  .1
           r(beta1_val) =  6.349999999999982
macros:
                r(cmd) : "wald_tc"
                r(STR) : "CIV"
```

Therefore, at s^*, we have that v is weakly correlated with t but has higher correlation with y, giving a β_1 that is much higher than the true treatment effect! Such a state is more likely to yield samples that would call the validity and strength of v into question. Thus, we might want to impose an upper bound on D_{CIV}, such as 0.1. Adding this bound, we get

```
. wald_tc CIV, samp_size(100) yata(0.4) ybtb(0.3) ta_l(0.7) ta_r(0.7) iv_mode
> v_l(0.1) v_r(0.9) d_civ(0.1)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .294779994
Time elapsed (sec): 376.014
. return list
scalars:
                  r(MR) =  .29477999422
          r(treata_val) =  .578
              r(ta_val) =  .7
            r(yata_val) =  .4
            r(ybtb_val) =  .3
            r(yatb_val) =  1.00000000000e-08
            r(ybta_val) =  .9999999899999999
             r(eya_val) =  .280000003
             r(eyb_val) =  .7899999929999999
           r(vtay1_val) =  .8111111111111111
           r(vtay0_val) =  .9
           r(vtby1_val) =  .5444444444444445
           r(vtby0_val) =  .7222222222222222
           r(beta1_val) =  -.5363636363636357
macros:
                r(cmd) : "wald_tc"
                r(STR) : "CIV"
```

Compared with the original run, MR is negligibly lower, `r(treata_val)` has decreased slightly from 0.585 to 0.578, and `r(beta1_val)` shows that β_1 has also been moved down

to fit within the upper bound of $D_{\text{PIIV}} \leq 0.1$. In this case, imposing that asymptotic β_1 is close to the true mean treatment response (which is around $0.28 - 0.79 = -0.51$) does little to improve the MR of the CIV STR.

However, there are other interesting assumptions that can change the performance of STRs, such as a monotone selection assumption. In addition to reporting that catheterized patients tend to be more severely ill than noncatheterized patients, Bhattacharya, Shaikh, and Vytlacil (2012) suggest that their findings may be due to a selection story: catheterization can be lifesaving for the most severely ill patients in the short term (that is, seven days) but cannot overcome their latent health problems in the long term.

PIIV STR

Now, let us see what happens when we run the original parameterization with the PIIV STR:

```
. wald_tc PIIV, samp_size(100) yata(0.4) ybtb(0.3) ta_l(0.7) ta_r(0.7) iv_mode
> v_l(0.1) v_r(0.9)
Percent done.....10.....20.....30.....40.....50.....60.....70.....80.....90.....
> 100
Maximum Regret: .299389994
Time elapsed (sec): 508.515
. return list
scalars:
                  r(MR) =  .29938999389
          r(treata_val) =  .389
              r(ta_val) =  .7
            r(yata_val) =  .4
            r(ybtb_val) =  .3
            r(yatb_val) =  .9999999899999999
            r(ybta_val) =  1.00000000000e-08
             r(eya_val) =  .579999997
             r(eyb_val) =  .090000007
           r(vtay1_val) =  .3666666666666667
           r(vtay0_val) =  .8111111111111111
           r(vtby1_val) =  .7222222222222222
           r(vtby0_val) =  .2777777777777778
macros:
                r(cmd) : "wald_tc"
                r(STR) : "PIIV"
```

We have a similar regret to the CIV STR, although the state s^* is somewhat different and PIIV has a lower probability of choosing $\delta = a$ in its s^*.

All of these examples shown above demonstrate a small fraction of the flexible options that the user has in exploring the performance of STRs under different settings of nonrandomized treatment.

6 Acknowledgments

We are grateful to Aleksey Tetenov and Eduardo Campillo-Betancourt for their comments.

7 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 21-1
. net install st0629     (to install program files, if available)
. net get st0629         (to install ancillary files, if available)
```

8 References

Bhattacharya, J., A. M. Shaikh, and E. Vytlacil. 2012. Treatment effect bounds: An application to Swan-Ganz catheterization. *Journal of Econometrics* 168: 223–243. https://doi.org/10.1016/j.jeconom.2012.01.001.

Box, G. E. P. 1979. Robustness in the strategy of scientific model building. In *Robustness in Statistics*, ed. R. L. Launer and G. N. Wilkinson, 201–236. New York: Academic Press. https://doi.org/10.1016/B978-0-12-438150-6.50018-2.

Hirano, K., and J. R. Porter. 2019. Asymptotics for statistical treatment rules. *Econometrica* 77: 1683–1701. https://doi.org/10.3982/ECTA6630.

Kitagawa, T., and A. Tetenov. 2018. Who should be treated? Empirical welfare maximization methods for treatment choice. *Econometrica* 86: 591–616. https://doi.org/10.3982/ECTA13288.

Manski, C. F. 1990. Nonparametric bounds on treatment effects. *American Economic Review* 80: 319–323.

———. 2004. Statistical treatment rules for heterogeneous populations. *Econometrica* 72: 1221–1246. https://doi.org/10.1111/j.1468-0262.2004.00530.x.

———. 2005. Partial identification with missing data: Concepts and findings. *International Journal of Approximate Reasoning* 39: 151–165. https://doi.org/10.1016/j.ijar.2004.10.006.

———. 2007a. *Identification for Prediction and Decision*. Cambridge, MA: Harvard University Press.

———. 2007b. Minimax-regret treatment choice with missing outcome data. *Journal of Econometrics* 139: 105–115. https://doi.org/10.1016/j.jeconom.2006.06.006.

———. 2019. Treatment choice with trial data: Statistical decision theory should supplant hypothesis testing. *American Statistician* 73: 296–304. https://doi.org/10.1080/00031305.2018.1513377.

———. 2020. Econometrics for decision making: Building foundations sketched by Haavelmo and Wald. Haavelmo Lecture notes, University of Oslo, Oslo, Norway. https://faculty.wcas.northwestern.edu/~cfm754/econometrics_for_decision_making.pdf.

Manski, C. F., and M. Tabord-Meehan. 2017. Evaluating the maximum MSE of mean estimators with missing data. *Stata Journal* 17: 723–735. https://doi.org/10.1177/1536867X1701700311.

Manski, C. F., and A. Tetenov. 2016. Sufficient trial size to inform clinical practice. *Proceedings of the National Academy of Sciences of the United States of America* 113: 10518–10523. https://doi.org/10.1073/pnas.1612174113.

———. 2019. Trial size for near-optimal choice between surveillance and aggressive treatment: Reconsidering MSLT-II. *American Statistician* 73: 305–311. https://doi.org/10.1080/00031305.2018.1543617.

Stoye, J. 2009. Minimax regret treatment choice with finite samples. *Journal of Econometrics* 151: 70–81. https://doi.org/10.1016/j.jeconom.2009.02.013.

———. 2012. Minimax regret treatment choice with covariates or with limited validity of experiments. *Journal of Econometrics* 166: 138–156. https://doi.org/10.1016/j.jeconom.2011.06.012.

Tetenov, A. 2012. Statistical treatment choice based on asymmetric minimax regret criteria. *Journal of Econometrics* 166: 157–165. https://doi.org/10.1016/j.jeconom.2011.06.013.

Wald, A. 1949. Statistical Decision Functions. *Annals of Mathematical Statistics* 20: 165–205. https://doi.org/10.1214/aoms/1177730030.

About the authors

Charles F. Manski is the Board of Trustees Professor in Economics at Northwestern University.

Valentyn Litvin is a graduate student in Economics at Northwestern University.