



AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

THE STATA JOURNAL

Editors

H. JOSEPH NEWTON
Department of Statistics
Texas A&M University
College Station, Texas
editors@stata-journal.com

NICHOLAS J. COX
Department of Geography
Durham University
Durham, UK
editors@stata-journal.com

Associate Editors

CHRISTOPHER F. BAUM, Boston College
NATHANIEL BECK, New York University
RINO BELLOCCO, Karolinska Institutet, Sweden, and
University of Milano-Bicocca, Italy
MAARTEN L. BUIS, WZB, Germany
A. COLIN CAMERON, University of California–Davis
MARIO A. CLEVES, University of Arkansas for
Medical Sciences
WILLIAM D. DUPONT, Vanderbilt University
PHILIP ENDER, University of California–Los Angeles
DAVID EPSTEIN, Columbia University
ALLAN GREGORY, Queen's University
JAMES HARDIN, University of South Carolina
BEN JANN, University of Bern, Switzerland
STEPHEN JENKINS, London School of Economics and
Political Science
ULRICH KOHLER, University of Potsdam, Germany

FRAUKE KREUTER, Univ. of Maryland–College Park
PETER A. LACHENBRUCH, Oregon State University
JENS LAURITSEN, Odense University Hospital
STANLEY LEMESHOW, Ohio State University
J. SCOTT LONG, Indiana University
ROGER NEWSON, Imperial College, London
AUSTIN NICHOLS, Urban Institute, Washington DC
MARCELLO PAGANO, Harvard School of Public Health
SOPHIA RABE-HESKETH, Univ. of California–Berkeley
J. PATRICK ROYSTON, MRC Clinical Trials Unit,
London
PHILIP RYAN, University of Adelaide
MARK E. SCHAFFER, Heriot-Watt Univ., Edinburgh
JEROEN WEESIE, Utrecht University
IAN WHITE, MRC Biostatistics Unit, Cambridge
NICHOLAS J. G. WINTER, University of Virginia
JEFFREY WOOLDRIDGE, Michigan State University

Stata Press Editorial Manager

LISA GILMORE

Stata Press Copy Editors

DAVID CULWELL and DEIRDRE SKAGGS

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go “beyond the Stata manual” in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*, *Scopus*, and *Social Sciences Citation Index*).

For more information on the *Stata Journal*, including information for authors, see the webpage

<http://www.stata-journal.com>

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

<http://www.stata.com/bookstore/sj.html>

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		Printed & electronic	
1-year subscription	\$ 98	1-year subscription	\$138
2-year subscription	\$165	2-year subscription	\$245
3-year subscription	\$225	3-year subscription	\$345
1-year student subscription	\$ 75	1-year student subscription	\$ 99
1-year university library subscription	\$125	1-year university library subscription	\$165
2-year university library subscription	\$215	2-year university library subscription	\$295
3-year university library subscription	\$315	3-year university library subscription	\$435
1-year institutional subscription	\$245	1-year institutional subscription	\$285
2-year institutional subscription	\$445	2-year institutional subscription	\$525
3-year institutional subscription	\$645	3-year institutional subscription	\$765
Electronic only		Electronic only	
1-year subscription	\$ 75	1-year subscription	\$ 75
2-year subscription	\$125	2-year subscription	\$125
3-year subscription	\$165	3-year subscription	\$165
1-year student subscription	\$ 45	1-year student subscription	\$ 45

Back issues of the *Stata Journal* may be ordered online at

<http://www.stata.com/bookstore/sjj.html>

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

<http://www.stata-journal.com/archives.html>

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.



Copyright © 2013 by StataCorp LP

Copyright Statement: The *Stata Journal* and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

The articles appearing in the *Stata Journal* may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the *Stata Journal*, in whole or in part, on publicly accessible websites, file servers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the *Stata Journal* or the supporting files understand that such use is made without warranty of any kind, by either the *Stata Journal*, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the *Stata Journal* is to promote free communication among Stata users.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, **STATA**, Stata Press, Mata, **MATA**, and NetCourse are registered trademarks of StataCorp LP.

Estimation of multivalued treatment effects under conditional independence

Matias D. Cattaneo
Department of Economics
University of Michigan
Ann Arbor, MI
cattaneo@umich.edu

David M. Drukker
StataCorp
College Station, TX
ddrukker@stata.com

Ashley D. Holland
Department of Science and Mathematics
Cedarville University
Cedarville, OH
aholland@cedarville.edu

Abstract. This article discusses the `poparms` command, which implements two semiparametric estimators for multivalued treatment effects discussed in Cattaneo (2010, *Journal of Econometrics* 155: 138–154). The first is a properly reweighted inverse-probability weighted estimator, and the second is an efficient-influence-function estimator, which can be interpreted as having the double-robust property. Our implementation jointly estimates means and quantiles of the potential-outcome distributions, allowing for multiple, discrete treatment levels. These estimators are then used to estimate a variety of multivalued treatment effects. We discuss pre- and postestimation approaches that can be used in conjunction with our main implementation. We illustrate the program and provide a simulation study assessing the finite-sample performance of the inference procedures.

Keywords: `st0303`, `poparms`, `bfit`, inverse-probability weighting, treatment effects, semiparametric estimation, unconfoundedness, generalized propensity score, multivalued treatment effects

1 Introduction

This article introduces the `poparms` (short for *potential-outcome parameters*) command for estimating causal effects of multivalued treatments under ignorability, that is, under the selection-on-observables and common support assumptions. In particular, this command implements the two flexible, semiparametric-efficient estimation procedures proposed in Cattaneo (2010) to conduct joint inference on mean and quantile treatment effects. For recent reviews on the vast literature of treatment effects, see, among others, Heckman and Vytlačil (2007), Imbens and Wooldridge (2009), and Wooldridge (2010) in economics; Morgan and Winship (2007) in sociology; and van der Laan and Robins (2003) and Tsiatis (2006) in biostatistics.

Many multivalued treatment effects are constructed by contrasting the parameters of the distributions that the outcome variable would have had under each level of treat-

ment. These distributions are called the potential-outcome distributions and are identifiable from the observed data under the selection-on-observables or unconfoundedness assumption. Under this assumption, Cattaneo (2010) derives the large-sample properties of inverse-probability weighted (IPW) estimators and efficient-influence-function (EIF) estimators for the means, quantiles, and other features of the potential-outcome distributions when the treatment variable can have multiple distinct values. Using either of these estimators, which are shown to be semiparametric efficient under certain regularity conditions, one can construct a wide variety of treatment-effects estimators as well as valid inference procedures for multivalued treatment effects.

In this article, we describe the new `poparms` command, which implements these IPW and EIF estimators to estimate the means and quantiles of each potential-outcome distribution as well as the associated standard-error estimators. Different contrasts of these estimated parameters are then used to produce semiparametric-efficient estimators with valid standard-error estimators for average and quantile multivalued treatment effects. These procedures require implementing nonparametric series estimators to flexibly approximate certain nonparametric functions. We discuss in detail several pre- and postestimation procedures for the analysis of mean and quantile treatment effects.

The discussed methods crucially rely on the selection-on-observables assumption to identify and estimate the parameters of the potential-outcome distributions. This assumption maintains that after one controls for observed covariates, the potential-outcome distributions are independent of the treatment level, and therefore rules out that some unobservable factor correlated with treatment assignment affects the potential-outcome distributions. This assumption is strong and may not be valid in some applications, although it is popular and frequently used in empirical work. While it is testable in some special cases, this assumption is fundamentally untestable in general. It automatically holds under treatment randomization, in which case the results in this article lead to more efficient estimators than the usual parametric estimators. We further discuss this assumption and its implications below.

In the remainder of the article, we discuss the implemented methods with notation and formality, an example, the syntax of the `poparms` command, and the methods and formulas implemented in the command.

2 Setup, parameters, and estimators

2.1 Model and sampling

We consider a standard cross-sectional setting where we observe a random sample of size n from a large population in which each individual has been assigned one of $J + 1$ possible treatment levels $j = 0, 1, \dots, J$. For each individual $i = 1, 2, \dots, n$, we observe the random vector $\mathbf{z}_i = (y_i, w_i, \mathbf{x}_i')'$, where y_i is the observed outcome variable, w_i denotes the treatment level administered, and \mathbf{x}_i is a $k_x \times 1$ vector of covariates. We also introduce the indicator variables $d_i(j) = \mathbf{1}(w_i = j)$, which take the value 1 if unit i received treatment j and the value 0 otherwise. $\mathbf{1}(\cdot)$ denotes the indicator function, the

observed vectors \mathbf{z}_i , $i = 1, 2, \dots, n$, are independent and identically distributed draws of the vector $\mathbf{z} = (y, w, \mathbf{x})'$, and $d(j) = \mathbf{1}(w = j)$.

To describe the estimands and estimators of interest, we use the classical potential-outcome framework in the context of multivalued treatment effects. This model distinguishes between the observed outcome y_i and the $J + 1$ potential outcomes $y_i(j)$ for each treatment level $j = 0, 1, \dots, J$. In this framework, the observed outcome variable is given by

$$y_i = d_i(0)y_i(0) + d_i(1)y_i(1) + \dots + d_i(J)y_i(J)$$

where $\{y_i(0), y_i(1), \dots, y_i(J)\}'$ is an independent and identically distributed draw from $\{y(0), y(1), \dots, y(J)\}'$ for each individual $i = 1, 2, \dots, n$ in the sample. The distribution of each $y(j)$ is the distribution of the outcome variable that would occur if individuals were given treatment level j ; it is known as the potential-outcome distribution of treatment level j . Many treatment effects of interest reduce to contrasts between parameters of these distributions. Because it is central to parameter interpretation, we highlight the fact that the potential-outcome distributions are marginal distributions with respect to the covariates used in the analysis.

Only one of the $J + 1$ possible potential outcomes can be observed for each individual in the sample because each individual can receive only one treatment level. Holland (1986) termed this situation the fundamental problem of causal inference. From this perspective, estimating the parameters of the potential-outcome distribution is a missing-data problem because we can see only one outcome per individual. The observed y are draws from distribution of $y(j)$ conditional on $w = j$, and hence, we need further assumptions to identify the unconditional distribution of $y(j)$ from the observed data. The following assumption, known as ignorability, is a combination of the selection-on-observables assumption and a no-empty-cell assumption, and it allows us to recover the parameters of the unobserved unconditional distribution from the observed conditional distribution.

Assumption 1. For all $j = 0, 1, \dots, J$:

- (a) (*Selection-on-observables*) $y(j) \perp\!\!\!\perp d(j) | \mathbf{x}$.
- (b) (*No-empty-cell*) $0 < p_{\min} < p_j(\mathbf{x})$ with $p_j(\mathbf{x}) = \mathbb{P}(w = j | \mathbf{x})$.

Assumption 1(a) implies that the distribution of each potential outcome $y(j)$ is independent of the random treatment $d(j)$, conditional on the covariates \mathbf{x} . This assumption has a long history; see, among many others, Heckman, Ichimura, and Todd (1998), Imbens (2004), Heckman and Vytlacil (2007), Imbens and Wooldridge (2009), and Wooldridge (2010). This condition imposes conditional (on observables) random assignment for each treatment level: among individuals with the same observable characteristics, treatment assignment should be independent of the potential outcome. This assumption, although weaker than plain random assignment, is indeed strong because it rules out the presence of observed characteristics that could affect both treatment and outcomes. Nonetheless, in some empirical contexts, this assumption is reasonable and often imposed to estimate treatment effects.

Assumption 1(b) says that for every possible \mathbf{x} in the population, there is a strictly positive probability that someone with that covariate pattern could be assigned to each treatment level. Intuitively, we need to see individuals of each covariate type in each treatment level to recover potential-outcome distribution for individuals of that type. The function $\mathbf{p}(\mathbf{x}) = \{p_0(\mathbf{x}), p_1(\mathbf{x}), \dots, p_J(\mathbf{x})\}'$ is called the generalized propensity score (GPS). Imbens (2000) provides an extensive discussion on identification of multivalued mean treatment effects under ignorability (also see Hirano and Imbens [2004]).

2.2 Estimands and estimators

Using assumption 1, Cattaneo (2010) proposes two flexible, semiparametric-efficient estimation procedures for a large class of multivalued treatment effects. These estimands are obtained by first estimating the corresponding population parameters for each potential-outcome distribution and then combining these estimates. The general estimators are implicitly defined by a collection of possibly overidentified, nonsmooth moment conditions. We focus on implementing these estimators for the special, important cases of means and quantiles of the potential-outcome distributions. Contrasts between these parameters lead to interesting population parameters in the context of multivalued treatment effects, which extend the usual average and quantile treatment effects from the binary treatment literature.¹

To clarify, we will let $F_{y(j)}(y)$ be the distribution function of the potential outcome $y(j)$, $j = 0, 1, \dots, J$. The $J + 1$ means of the potential-outcome distributions are

$$\boldsymbol{\mu} = (\mu_0, \mu_1, \dots, \mu_J)'$$

where $\mu_j = \mathbb{E}\{y(j)\} = \int y \, dF_{y(j)}(y)$. The τ th quantiles of the $J + 1$ potential outcomes are

$$\mathbf{q}(\tau) = \{q_0(\tau), q_1(\tau), \dots, q_J(\tau)\}'$$

where $q_j(\tau) = F_{y(j)}^{-1}(\tau)$ with $\tau \in (0, 1)$, and $F_{y(j)}(y)$ is assumed to be continuous and strictly increasing in a neighborhood of $q_j(\tau)$. We do not discuss regularity conditions in detail, but we note that they imply that the outcome variable should be continuous. Intuitively, if the potential-outcome distributions $F_{y(j)}(y)$ are identifiable from observed data, then so are the population parameters of interest because they are just the means and quantiles of $F_{y(j)}(y)$ for each j . Assumption 1 implies that $F_{y(j)}(y) = \mathbb{E}\{F_{y(j)|\mathbf{x}}(y|\mathbf{x})\} = \mathbb{E}\{F_y(y|\mathbf{x}, w = j)\}$ for each treatment level j , where $F_{y(j)|\mathbf{x}}(y|\mathbf{x})$ denotes the distribution function of $y(j)|\mathbf{x}$, and $F_y(y|\mathbf{x}, w = j)$ denotes the distribution function of $y|\mathbf{x}, w = j$, the latter distribution being identifiable from the observed data. Thus μ_j and $q_j(\tau)$ can be shown to be identifiable under appropriate regularity conditions.

The implemented methods identify and estimate parameters of the potential-outcome distributions that have been marginalized over the covariate distributions. In

1. Cattaneo (2010) labels the collection of means and quantiles as marginal mean treatment effects and marginal quantile treatment effects, respectively. In this article, however, we will only use the term “treatment effect” to refer to contrasts (pairwise or other) between different means and quantiles to avoid possible confusions.

this sense, they are population-averaged or marginal parameters. The quantiles of the marginal potential-outcome distributions $y(j)$ differ from the means of the conditional quantiles of the potential-outcome distributions: $q_j(\tau) = F_{y(j)}^{-1}(\tau) \neq \mathbb{E}\{q_j(\tau|\mathbf{x})\}$, where the marginal distribution $F_{y(j)}(\tau) = \mathbb{E}\{F_{y(j)|\mathbf{x}}(y|\mathbf{x})\}$ and the conditional quantiles $q_j(\tau|\mathbf{x})$ are defined by $q_j(\tau|\mathbf{x}) = F_{y(j)|\mathbf{x}}^{-1}(\tau)$. In contrast, the mean of the marginal distribution is the mean of the conditional mean distributions, a fact that underlies the popular regression-adjustment estimators for μ_j . The implemented methods identify and estimate quantiles of the marginal potential-outcome distributions $y(j)$.

This identification discussion is associated with the ideas of projection and imputation, which could be used to construct (multivalued) treatment-effect estimators (see, for example, Hahn [1998]; Imbens, Newey, and Ridder [2007]; Chen, Hong, and Tarozzi [2004, 2008]; and Cattaneo and Farrell [2011]). Alternatively, Cattaneo (2010) proposes two Z -estimators, one constructed using an inverse-probability-weighting scheme and the other constructed using the full functional form of the EIF, which are shown to be consistent, asymptotically Gaussian, and semiparametric efficient under appropriate conditions. (Thus the two estimators are asymptotically equivalent to first order.) These estimators are referred to as IPW and EIF, respectively. In the rest of this subsection, we provide some brief intuition for these estimators, but we relegate most of the technical and implementation details to section 7.

IPW estimation follows the work of Hirano, Imbens, and Ridder (2003) and Firpo (2007) for binary mean and quantile treatment effects and extends the idea of inverse-probability weighting to a multivalued treatment context (also see Imbens [2000]). The estimator is motivated by simply noting that for each treatment level j ,

$$\mathbb{E} \left\{ \frac{d(j)(y - \mu_j)}{p_j(\mathbf{x})} \right\} = \mathbb{E} \left[\frac{\mathbb{E}\{d(j)|\mathbf{x}\} \mathbb{E}\{y(j) - \mu_j|\mathbf{x}\}}{p_j(\mathbf{x})} \right] = \mathbb{E}\{y(j) - \mu_j\} = 0$$

and, similarly,

$$\mathbb{E} \left(\frac{d(j) [\mathbf{1}\{y \leq q_j(\tau)\} - \tau]}{p_j(\mathbf{x})} \right) = \mathbb{E}[\mathbf{1}\{y(j) \leq q_j(\tau)\} - \tau] = 0$$

These calculations lead to a collection of moment conditions based on observed data only. For the mean of potential-outcome distribution j , we have

$$\mathbb{E} [\psi_{\text{IPW},j}\{\mathbf{z}_i; \mu_j, p_j(\mathbf{x}_i)\}] = 0 \quad \text{with} \quad \psi_{\text{IPW},j}\{\mathbf{z}_i; \mu_j, p_j(\mathbf{x}_i)\} = \frac{d_i(j)(y_i - \mu_j)}{p_j(\mathbf{x}_i)}$$

Similarly, for each τ th quantile of j th potential-outcome distribution, we have

$$\mathbb{E} [\psi_{\text{IPW},j}\{\mathbf{z}_i; q_j(\tau), p_j(\mathbf{x}_i)\}] = 0 \quad \text{with} \quad \psi_{\text{IPW},j}\{\mathbf{z}_i; q_j(\tau), p_j(\mathbf{x}_i)\} = \frac{d_i(j) [\mathbf{1}\{y_i \leq q_j(\tau)\} - \tau]}{p_j(\mathbf{x}_i)}$$

The only unknown functions for the IPW estimators are the conditional probability functions $p_j(\mathbf{x})$, $j = 0, 1, \dots, J$, forming the GPS $\mathbf{p}(\mathbf{x})$, which can be estimated parametrically or nonparametrically. If we let $\hat{\mathbf{p}}(\mathbf{x}) = \{\hat{p}_0(\mathbf{x}), \hat{p}_1(\mathbf{x}), \dots, \hat{p}_J(\mathbf{x})\}'$ be one such

estimator, a plug-in approach leads to the following estimators discussed in Cattaneo (2010) for the mean and τ th quantile (of the j th potential-outcome distribution), respectively,

$$\widehat{\mu}_{\text{IPW},j} \quad \text{s.th.} \quad \frac{1}{n} \sum_{i=1}^n \psi_{\text{IPW},j} \{ \mathbf{z}_i; \widehat{\mu}_{\text{IPW},j}, \widehat{p}_j(\mathbf{x}_i) \} = 0$$

and

$$\widehat{q}_{\text{IPW},j}(\tau) \quad \text{s.th.} \quad \frac{1}{n} \sum_{i=1}^n \psi_{\text{IPW},j} \{ \mathbf{z}_i; \widehat{q}_{\text{IPW},j}(\tau), \widehat{p}_j(\mathbf{x}_i) \} = 0$$

To gain some intuition, we notice that in the case of the j th mean, the estimator can be expressed in closed form:

$$\widehat{\mu}_{\text{IPW},j} = \left\{ \sum_{i=1}^n \frac{d_i(j)}{\widehat{p}_j(\mathbf{x}_i)} \right\}^{-1} \sum_{i=1}^n \frac{d_i(j)}{\widehat{p}_j(\mathbf{x}_i)} y_i$$

This shows that this approach leads to IPW estimators with proper reweighting. We further discuss this feature in the next subsection.

The moment conditions for the EIF use the complete form of the EIF of the estimands rather than just one portion of it. This approach involves other nonparametric functions that need to be estimated, but it enjoys certain robustness properties that may be appealing from a practical point of view, as we further discuss in section 2.3. To describe these estimators in the special case of means and quantiles, we first introduce the following additional functions,

$$e_j(\mathbf{x}_i; \mu_j) = \mathbb{E}\{y_i(j) - \mu_j | \mathbf{x}_i\} = \mathbb{E}(y_i - \mu_j | \mathbf{x}_i, w_i = j)$$

and

$$e_j\{\mathbf{x}_i; q_j(\tau)\} = \mathbb{E}[\mathbf{1}\{y_i(j) \leq q_j(\tau)\} - \tau | \mathbf{x}_i] = \mathbb{E}[\mathbf{1}\{y_i \leq q_j(\tau)\} - \tau | \mathbf{x}_i, w_i = j]$$

for each treatment level j . These conditional expectations can be estimated from the observed data.

The EIF estimator is then constructed using the following moment conditions for the mean and τ quantile of the j th potential outcome,

$$\mathbb{E}[\psi_{\text{EIF}} \{ \mathbf{z}_i; \mu_j, p_j(\mathbf{x}_i), e_j(\cdot; \mu_j) \}] = 0 \quad \text{with}$$

$$\psi_{\text{EIF}} \{ \mathbf{z}_i; \mu_j, p_j(\mathbf{x}_i), e_j(\cdot; \mu_j) \} = \frac{d_i(j)}{p_j(\mathbf{x}_i)} (y_i - \mu_j) - \frac{e_j(\mathbf{x}_i; \mu_j)}{p_j(\mathbf{x}_i)} \{d_i(j) - p_j(\mathbf{x}_i)\}$$

and

$$\begin{aligned} \mathbb{E}(\psi_{\text{EIF}}[\mathbf{z}_i; q_j(\tau), p_j(\mathbf{x}_i), e_j\{\cdot; q(\tau)\}]) &= 0 \quad \text{with} \\ \psi_{\text{EIF}}[\mathbf{z}_i; q_j(\tau), p_j(\mathbf{x}_i), e_j\{\cdot; q_j(\tau)\}] &= \frac{d_i(j) [\mathbf{1}\{y_i \leq q_j(\tau)\} - \tau]}{p_j(\mathbf{x}_i)} - \frac{e_j\{\mathbf{x}_i; q(\tau)\}}{p_j(\mathbf{x}_i)} \{d_i(j) - p_j(\mathbf{x}_i)\} \end{aligned}$$

As in the case of the IPW estimator, the EIF estimator uses these moment conditions, replacing expectations by sample averages and unknown functions by appropriate (parametric or nonparametric) estimators, leading to the estimates

$$\widehat{\mu}_{\text{EIF},j} \quad \text{s.th.} \quad \frac{1}{n} \sum_{i=1}^n \psi_{\text{EIF}}\{\mathbf{z}_i; \widehat{\mu}_{\text{EIF},j}, \widehat{p}_j(\mathbf{x}_i), \widehat{e}_j(\cdot; \widehat{\mu}_{\text{EIF},j})\} = 0$$

and

$$\widehat{q}_{\text{EIF},j}(\tau) \quad \text{s.th.} \quad \frac{1}{n} \sum_{i=1}^n \psi_{\text{EIF}}[\mathbf{z}_i; \widehat{q}_{\text{EIF},j}(\tau), \tau, \widehat{p}_j(\mathbf{x}_i), \widehat{e}_j\{\cdot; \widehat{q}_{\text{EIF},j}(\tau)\}] = 0$$

for the j th mean and τ quantile, respectively.

There are, of course, several important implementation details surrounding these procedures, including the choice of (non) parametric estimators $\widehat{p}_j(\mathbf{x}_i)$, $\widehat{e}_j(\cdot; \mu_j)$, and $\widehat{e}_j\{\cdot; q_j(\tau)\}$, numerical optimization issues, and standard-error estimators. We address all the details in section 7.

2.3 Some features of the implemented procedures

In this section, we offer some remarks on the estimands and implemented estimators considered in this article.

1. Under standard regularity conditions, the IPW and EIF estimators are consistent, asymptotically normal, and semiparametric efficient when nonparametric estimators are used to approximate the unknown functions introduced above. Thus, from a semiparametric perspective, these estimators are asymptotically equivalent. We discuss these results and how we use them to conduct asymptotically valid joint inference on multivalued mean and quantile treatment effects in section 7. In that section, we detail our variance–covariance matrix estimator (VCE).
2. Because we construct the joint VCE, we can conduct joint inference on the mean and quantile of the potential-outcome distributions, and hence, we can also conduct valid inferences on many other treatment-effect parameters of interest. We illustrate this process in some detail in section 5.

3. Both the IPW and EIF estimators can be used to construct simple inference procedures for joint means and quantiles of the potential-outcome distributions and combinations thereof. While the average treatment effect in binary-treatment contexts, or more generally the difference in means of potential outcomes in the case of a multivalued treatment, is probably the most frequently used measure of a treatment effect, such a central tendency measure is only one of many interesting possibilities. Differences in the quantiles of potential-outcome distributions can uncover effects of a treatment that differ importantly from those measured by the average treatment effect or its analogue in the context of multivalued treatments. Indeed, the treatment effects may differ remarkably at low, middle, and upper quantiles of the potential-outcome distribution; thus conducting inferences on quantile treatment effects allows applied researchers to investigate the existence of such potential differences.

Importantly, and as it is well known, differences in quantiles need not always correspond to quantile treatment effects. Specifically, $q_j(\tau) - q_l(\tau)$, for some pair of distinct treatment levels j and l , is usually understood as a measure of how the τ th quantile of the distribution of the outcome variable would change if everyone in the population were given treatment j instead of treatment l , even though the quantile of the differences need not coincide with the differences in the quantiles.

4. While the IPW and EIF estimators are semiparametric efficient and asymptotically equivalent, we recommend using the EIF estimator because it enjoys the so-called double-robust property when viewed from a (flexible) parametric implementation perspective (the IPW estimator does not have this property). See, for example, van der Laan and Robins (2003), Imbens and Wooldridge (2009), and Wooldridge (2010) for reviews. The EIF estimator can be interpreted as a nonparametric version of the doubly robust estimators: although we interpret the estimators $\hat{p}_j(\cdot)$, $\hat{e}_j(\cdot; \mu_j)$, and $\hat{e}_j\{\cdot; q_j(\tau)\}$ as consistent nonparametric estimators of their population counterparts, from a more (flexible) parametric perspective, the EIF estimators require only that either i) $\hat{p}_j(\cdot)$ or ii) $\hat{e}_j(\cdot; \mu_j)$ and $\hat{e}_j\{\cdot; q_j(\tau)\}$ be “correctly specified”. Thus, from this perspective, the EIF estimator could be argued to possibly dominate the IPW estimator. In `poparms`, the EIF estimator is the default. (For further discussion on the double-robust property, also see Kang and Schafer [2007] and the accompanying comments and rejoinder.)
5. While the IPW estimator may be preferred over the EIF because of its simplicity, it is important to note that the VCE for the IPW estimator requires implementing all the ingredients of the EIF estimator. Thus the IPW estimator is simpler only as a point estimator. See section 7 for more details.
6. The IPW estimator is well known and has a long tradition in the literature of missing data, treatment effects, and measurement error and survey, at least since the work of Horvitz and Thompson (1952). The implementation considered here is slightly different from the standard one because the resulting estimators use a different weighting scheme: the weights associated with the propensity score sum to 1. Busso, DiNardo, and McCrary (2013) report Monte Carlo evidence

suggesting that IPW estimators that divide by the sum of the weights perform better than IPW estimators that divide the number of observations when some of the predicted propensity scores are very small; also see Millimet and Tchernis (2009). Thus the estimators implemented in `poparms`, which are motivated from a theoretical Z -estimation perspective, are expected to exhibit a good performance in applications because they divide by the sum of the weights.

7. It may be useful to extend the methods in Cattaneo (2010) to include population parameters for subpopulations of interest such as treated and control groups. These estimands would be useful to generalize the average treatment effect on the treated and the quantile treatment effect on the treated from the binary treatment effect literature to the case of multivalued treatment effects. We plan to address these extensions in future work.

3 The `poparms` command

This section describes the syntax of the `poparms` command to conduct point estimation and inference across and between mean and quantiles of the different potential-outcome distributions.

3.1 Syntax

```
poparms (treatvar gpsvars) (depvar cvars) [if] [in] [, quantiles(numlist)
      vce(vcemethod [, vceoptions]) ipw]
```

treatvar is a categorical variable indicating treatment.

gpsvars are the covariates in the equations for the GPS.

depvar is the outcome variable.

cvars are the covariates on which conditional moments are calculated in the EIF estimator.

gpsvars and *cvars* may contain factor variables.

3.2 Description

`poparms` estimates the means and quantiles of the potential-outcome distributions of *depvar* corresponding to each level of the treatment variable *treatvar*.

You must specify both the polynomial in the covariates for the GPS in *gpsvars* and the polynomial in the covariates for the conditional mean in *cvars*; see section 5 for an example.

We discuss how to use the `bfit` command to select the variables for *gpsvars* and *cvars* in section 5. We discuss the syntax of `bfit` in section 4.

3.3 Options

`quantiles(numlist)` specifies the quantiles of the potential-outcome distributions that are to be estimated jointly with the means. By default, only the means are estimated. The values in the number list must be greater than 0 and less than 1. By default, method `vce(bootstrap)` is used when `quantiles()` is specified. We strongly recommend not using `vce(analytic)` when `quantiles()` is specified.

`vce(vcemethod [, vceoptions])` specifies the method for estimating the variance–covariance of the estimator. The three available methods are `bootstrap`, `analytic`, and `none`. With method `bootstrap`, the *vceoption* `reps(#)` specifies the number of bootstrap repetitions to perform. With method `analytic`, the *vceoptions* are `bwscale(#)`, `bwidths(matname)`, and `densities(matname)`. These suboptions are mutually exclusive. By default, `poparms` uses an analytic estimator when only means are estimated but uses a bootstrap estimator when quantiles are estimated. The analytic method for quantiles requires estimating the density of each potential outcome evaluated at each (estimated) quantile level. We implement this estimator using a nonparametric kernel-based density estimator, which requires a choice of bandwidth. See section 7 for details. In our Monte Carlo simulations (section 6), the resulting analytic standard-error estimator performed poorly, exhibiting great sensitivity to the bandwidth choice; therefore, we cannot recommend using this analytic method when quantiles are specified.

With method `bootstrap`, you may change the number of repetitions from the default 2,000 by specifying `vce(bootstrap, reps(#))`. The specified number of repetitions must be an integer greater than 49.

With method `analytic`, you may rescale the ad hoc rule-of-thumb (ROT) bandwidths used to estimate the densities by specifying `vce(analytic, bwscale(#))`. The specified number must be in the interval [0.1, 10].

With method `analytic`, you may specify the bandwidths used to estimate the densities by specifying `vce(analytic, bwidths(matname))`, where *matname* specifies a Stata row vector with the number of columns equal to the number of quantiles times the number of treatment levels.

With method `analytic`, you may specify the value of each density at each quantile level used by specifying `vce(analytic, densities(matname))`, where *matname* specifies a Stata row vector with the number of columns equal to the number of quantiles times the number of treatment levels.

`ipw` specifies that `poparms` use the IPW estimator instead of the default EIF estimator.

4 The `bfit` command

4.1 Syntax

```
bfit regress depvar indepvars [if] [in] [, corder(#) sort(bic|aic)
      options]
```

```
bfit logit depvar indepvars [if] [in] [, corder(#) sort(bic|aic) options]
```

```
bfit poisson depvar indepvars [if] [in] [, corder(#) sort(bic|aic)
      options]
```

4.2 Description

Each `bfit` subcommand sorts a set of fitted candidate regression models by an information criterion, puts the best-fitting model in `ereturn`, and displays a table showing the ranking of the models. The Bayesian information criterion (BIC) is the default, and the Akaike information criterion (AIC) may optionally be specified.

`bfit regress` fits the candidate linear-regression models by ordinary least squares (OLS). `bfit logit` fits the candidate multinomial logit models (MLMs) by maximum likelihood. `bfit poisson` fits the candidate Poisson regression models by maximum likelihood.

For each `bfit` subcommand, the candidate models are a series of polynomials in *indepvars*. The smallest of the candidate models includes only the first variable specified in *indepvars*. The largest of the candidate models is a fully interacted polynomial of the order specified in `corder()`. See 7.1 for details on the set of candidate models.

4.3 Options

`corder(#)` specifies the maximum order of the covariate polynomial. The default is `corder(2)`, which specifies a fully interacted second-order polynomial.

`sort(bic|aic)` specifies the information criterion by which the candidate models are to be sorted. The default is `sort(bic)`, which sorts the fitted candidate models by the BIC. `sort(aic)` sorts the fitted candidate models by the AIC.

options are passed to the estimation command. The allowable options depend on the estimation command invoked by the subcommand. For example, `base()` may be specified only with the `logit` subcommand. See [R] `regress`, [R] `logit`, and [R] `poisson` for the allowable command options.

5 A numerical example

In `spmdata`, we have some simulated data from a fictitious study that allows us to illustrate how to make some of the modeling choices and how to interpret the parameters. The goal of this imaginary study was to estimate the population effects of harder tests on student performance. Suppose that at the beginning of the year, each student is assigned to one of three possible class types: a class with normal tests ($w = 0$), a class whose tests included some hard questions ($w = 1$), or a class whose tests included even harder questions ($w = 2$). At the end of the year, student performance was measured by tests and interviews, the results of which were combined into a normalized index, `spmeasure`. The end-of-year tests were the same over all three class types. In this fictional study, we want to know whether some hard questions ($w = 1$) or even harder questions ($w = 2$) increase student performance relative to the control level of $w = 0$.

We assume that the potential outcomes of `spmeasure` were independent of the assignment of students to classes after controlling for the continuous variables parental status, `pindex`, and the student's environment, `eindex`.

As mentioned above, the estimators implemented in `poparms` require that we specify a model for the GPS and for the conditional mean. We first discuss and select a model for the GPS and then for the conditional mean.

The estimators in `poparms` use an MLM to predict the probability of each treatment level as a function of covariates. Cattaneo (2010) treats this estimator as nonparametric; the purpose is to predict the treatment probabilities, while the coefficients in the MLM are of no interest in themselves. From a theoretical perspective, it is assumed that a sufficient number of terms in a polynomial of the covariates are included in the MLM. The practical implication of this approach is that we must find an MLM for treatment probabilities before using `poparms`. We use our `bfit` command to automate this procedure.

In section 4, we discuss the syntax and implementation of `bfit`. At this point, we provide only the details required for our example. We select the MLM that minimizes the AIC from a set of candidate models. Our set of models are MLMs with increasingly rich polynomials in the covariates.

5.1 Preestimation analysis

In the output below, we specify the `logit` subcommand with the `bfit` command to cause `bfit` to search among candidate MLMs. The covariates `pindex` and `eindex` follow the outcome variable `w`. The option `order(3)` specifies that `bfit` should fit a sequence of models that builds up to a fully interacted, third-order polynomial in the two covariates. We specify the option `base(0)` so that $w=0$ will be the base category for the candidate MLMs. Finally, we specify option `aic` to select the model that minimizes the AIC instead of using the default BIC.

```
. use spmdata
. bfit logit w pindex eindex, corder(3) base(0) sort(aic)
bfit logit results sorted by aic
```

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
_bfit_5	5000	-5492.867	-5445.067	12	10914.13	10992.34
_bfit_6	5000	-5492.867	-5443.451	14	10914.9	11006.14
_bfit_9	5000	-5492.867	-5438.61	20	10917.22	11047.56
_bfit_8	5000	-5492.867	-5440.768	18	10917.54	11034.85
_bfit_7	5000	-5492.867	-5443.339	16	10918.68	11022.95
_bfit_3	5000	-5492.867	-5458.089	8	10932.18	10984.32
_bfit_4	5000	-5492.867	-5456.19	10	10932.38	10997.55
_bfit_2	5000	-5492.867	-5478.629	6	10969.26	11008.36
_bfit_1	5000	-5492.867	-5488.85	4	10985.7	11011.77

Note: N= used in calculating BIC
(results _bfit_5 are active now)

The above output shows that `bfit` estimated the parameters of nine different MLMs and that the fifth model, stored in memory under the name `_bfit_5`, minimizes the AIC. The output also indicates that the results from model `_bfit_5` are the results in `ereturn`.

To see these results, we use the `replay` feature of `mlogit` in the output below.

```
. mlogit
Multinomial logistic regression
```

	Number of obs	=	5000	
	LR chi2(10)	=	95.60	
	Prob > chi2	=	0.0000	
	Pseudo R2	=	0.0087	

```
Log likelihood = -5445.067
```

		Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
0		(base outcome)				
1						
	pindex	.119614	.0516415	2.32	0.021	.0183986 .2208295
	eindex	.1648585	.0515396	3.20	0.001	.0638428 .2658742
	c.pindex#					
	c.pindex	.2932857	.0551735	5.32	0.000	.1851476 .4014238
	c.pindex#					
	c.eindex	.0730155	.0774321	0.94	0.346	-.0787487 .2247797
	c.eindex#					
	c.eindex	.1875037	.0537868	3.49	0.000	.0820835 .2929238
	_cons	-.2040544	.0497818	-4.10	0.000	-.3016248 -.106484
2						
	pindex	.1629716	.0517606	3.15	0.002	.0615227 .2644205
	eindex	.242823	.0514467	4.72	0.000	.1419893 .3436567
	c.pindex#					
	c.pindex	.3263785	.0551118	5.92	0.000	.2183614 .4343957
	c.pindex#					
	c.eindex	.1426304	.0771243	1.85	0.064	-.0085305 .2937913
	c.eindex#					
	c.eindex	.2385979	.0534097	4.47	0.000	.1339169 .3432789
	_cons	-.2727777	.0503034	-5.42	0.000	-.3713705 -.1741848

We see that `bfit` selected a fully interacted, second-order polynomial. We consider this our working model for the GPSs, despite the fact that the 2 interaction terms are not significantly different from 0 at the 5% level.

Before we proceed, we need to check that all the predicted probabilities are sufficiently greater than 0 and less than 1. If some predicted probabilities from the selected MLM are too close to either 0 or 1, the parameters may not be identifiable. Khan and Tamer (2010) give a theoretical discussion on the potential pitfalls of small propensity scores, and Busso, DiNardo, and McCrary (2013) provide simulation evidence. Drukker and Wiggins (2004) argue that even for identifiable estimands, a consistent estimator may not work well in all possible samples, which in the present context suggests that when the predicted probabilities are too small, the combination of estimator and sample may lead to an ill-performing inference procedure.

Following Busso, DiNardo, and McCrary (2013), we discuss how to use overlap plots to look for potential problematic cases. For each treatment level j , an overlap plot depicts the estimated density of the predicted probabilities for treatment level j conditional on each possible treatment level. If any estimated density displays a sufficient mass near 0 or 1, the predicted probabilities are too close to 0 or 1, and the semiparametric estimators will probably not perform well in finite samples, even when assumption 1(b) holds.

In the output below, we compute the predicted probabilities for each treatment level and summarize each predicted probability conditional on each treatment level.

```
. predict double (phat0 phat1 phat2), pr
. sort w
. by w: summarize phat0 phat1 phat2
```

```
-> w = 0
```

Variable	Obs	Mean	Std. Dev.	Min	Max
phat0	1666	.3441123	.0504371	.0926236	.4011283
phat1	1666	.3339166	.0174793	.3133332	.4420854
phat2	1666	.3219711	.0349155	.2852155	.5207362

```
-> w = 1
```

Variable	Obs	Mean	Std. Dev.	Min	Max
phat0	1685	.3302699	.0628426	.0480128	.4012017
phat1	1685	.3379785	.0200443	.3133551	.4608249
phat2	1685	.3317516	.0460445	.2852104	.6110826

```
-> w = 2
```

Variable	Obs	Mean	Std. Dev.	Min	Max
phat0	1649	.3251693	.0661943	.0321425	.4012229
phat1	1649	.3391154	.0207652	.3133495	.4424814
phat2	1649	.3357154	.0488648	.2852087	.652054

While the above summaries would catch flagrant cases, overlap plots are also helpful. For instance, do the small minimums for `phat0` likely indicate a problem? Below we estimate the conditional densities of the predicted probability of treatment level 0 conditional on each treatment level observed. We use option `nograph` to suppress the graph and use the `generate()` option to store the abscissa and density values because we want to plot the three densities on a single graph. The `n(5000)` option causes the density estimates to be based on all the observations, and we specify `kernel(triangle)` because this kernel has finite support.

```

. kdensity phat0 if w==0, generate(xp00 den00) nograph n(5000) kernel(triangle)
. kdensity phat0 if w==1, generate(xp01 den01) nograph n(5000) kernel(triangle)
. kdensity phat0 if w==2, generate(xp02 den02) nograph n(5000) kernel(triangle)
. twoway line den00 xp00 || line den01 xp01 || line den02 xp02,
> legend(label(1 "w==0") label(2 "w==1") label(3 "w==2"))
> title("Conditional densities for probability of treatment level 0")
> name(pw0)

```

After computing the predicted densities, the `twoway` command plots the three densities on a single graph, shown in figure 1.

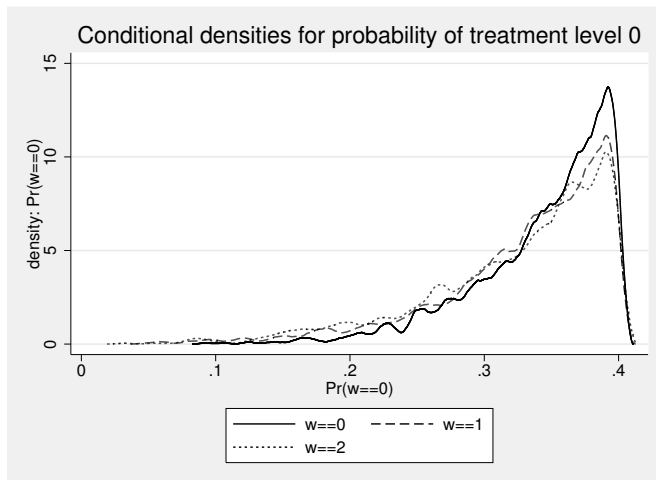


Figure 1. Densities for treatment level 0

This graph presents no evidence that there is any mass of observations with predicted probabilities close to either 0 or 1. None of the predicted densities show any mass for values too close to 0 or 1.

In the output below, we perform the analogous calculations to estimate the conditional densities for treatment levels 1 and 2.

```

. kdensity phat1 if w==0, generate(xp10 den10) nograph n(5000) kernel(triangle)
. kdensity phat1 if w==1, generate(xp11 den11) nograph n(5000) kernel(triangle)
. kdensity phat1 if w==2, generate(xp12 den12) nograph n(5000) kernel(triangle)
. kdensity phat2 if w==0, generate(xp20 den20) nograph n(5000) kernel(triangle)
. kdensity phat2 if w==1, generate(xp21 den21) nograph n(5000) kernel(triangle)
. kdensity phat2 if w==2, generate(xp22 den22) nograph n(5000) kernel(triangle)

```

Now we use `twoway` to plot the estimated densities in two graphs, shown in figures 2 and 3.

```
. twoway line den10 xp10 || line den11 xp11 || line den12 xp12,
> legend(label(1 "w==0") label(2 "w==1") label(3 "w==2"))
> title("Conditional densities for probability of treatment level 1")
> name(pw1)

. twoway line den20 xp20 || line den21 xp21 || line den22 xp22,
> legend(label(1 "w==0") label(2 "w==1") label(3 "w==2"))
> title("Conditional densities for probability of treatment level 2")
> name(pw2)
```

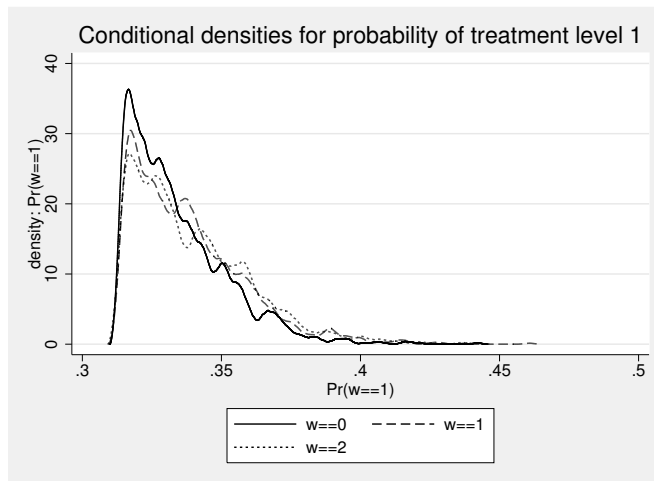


Figure 2. Densities for treatment level 1

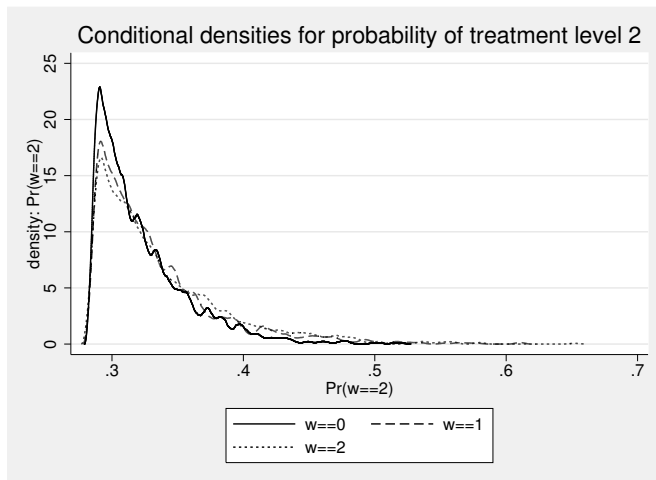


Figure 3. Densities for treatment level 2

Neither graph shows any mass too close to 0 or 1 in our artificial dataset. Given that the overlap plots of the predicted probabilities from our selected model show no cause for concern, we proceed with `_bfit_5` as our model for the predicted probabilities.

Having selected a model for the predicted probabilities, we then select a model for the conditional means. In the output below, we specify the `regress` subcommand with the `bfit` command to cause `bfit` to search among candidate models whose parameters can be estimated by OLS. The covariates `pindex` and `eindex` follow the outcome variable `spmeasure`. The option `corder(3)` specifies that `bfit` should fit a sequence of models that builds up to a fully interacted, third-order polynomial in the two covariates. These models specify nonlinear regression functions that are linear in the parameters permitting the use of the OLS estimator.

```
. bfit regress spmeasure pindex eindex, corder(3)
```

```
bfit regress results sorted by bic
```

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
_bfit_5	5000	-9459.868	-7572.222	6	15156.44	15195.55
_bfit_6	5000	-9459.868	-7568.105	7	15150.21	15195.83
_bfit_7	5000	-9459.868	-7565.506	8	15147.01	15199.15
_bfit_9	5000	-9459.868	-7560.053	10	15140.11	15205.28
_bfit_8	5000	-9459.868	-7564.766	9	15147.53	15206.19
_bfit_4	5000	-9459.868	-7885.959	5	15781.92	15814.5
_bfit_3	5000	-9459.868	-8036.736	4	16081.47	16107.54
_bfit_2	5000	-9459.868	-8253.596	3	16513.19	16532.74
_bfit_1	5000	-9459.868	-8933.951	2	17871.9	17884.94

Note: N= used in calculating BIC
(results `_bfit_5` are active now)

The above output shows that `bfit` estimated the parameters of nine different regression models by OLS and that the fifth model, stored in memory under the name `_bfit_5`, minimizes the BIC. The output also indicates that the results from model `_bfit_5` are the results in `ereturn`.

To see these results, we use the `replay` feature of `regress` in the output below.

```
. regress
```

Source	SS	df	MS			
Model	6825.47129	5	1365.09426	Number of obs =	5000	
Residual	6052.36854	4994	1.21192802	F(5, 4994) =	1126.38	
				Prob > F =	0.0000	
				R-squared =	0.5300	
				Adj R-squared =	0.5295	
				Root MSE =	1.1009	
Total	12877.8398	4999	2.57608318			

spmeasure	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
pindex	.9996322	.0221043	45.22	0.000	.956298	1.042966
eindex	1.026162	.0222179	46.19	0.000	.9826049	1.069719
c.pindex#						
c.pindex	.5426225	.0224409	24.18	0.000	.4986285	.5866166
c.pindex#						
c.eindex	.5762576	.0319718	18.02	0.000	.5135788	.6389365
c.eindex#						
c.eindex	.5781937	.0223753	25.84	0.000	.5343283	.6220591
_cons	.0932159	.0221566	4.21	0.000	.0497793	.1366526

We see that `bfit` selected a fully interacted, second-order polynomial. Because the output presents no indication of any problems, we accept this model as our model for the conditional mean. Note that for simplicity, we use the same flexible specification for both regression functions.

For easy access, `bfit` stores the varlist of the selected model in the local macro `r(bvlist)`. Below we show how to replicate the above `regress` results to illustrate this feature; see [U] 18.3.1 Local macros if you are not familiar with local macros.

```
. quietly bfit regress spmeasure pindex eindex, corder(3)
. regress spmeasure `r(bvlist)'
```

Source	SS	df	MS	Number of obs = 5000		
Model	6825.47129	5	1365.09426	F(5, 4994) = 1126.38		
Residual	6052.36854	4994	1.21192802	Prob > F = 0.0000		
Total	12877.8398	4999	2.57608318	R-squared = 0.5300		
				Adj R-squared = 0.5295		
				Root MSE = 1.1009		

spmeasure	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
pindex	.9996322	.0221043	45.22	0.000	.956298	1.042966
eindex	1.026162	.0222179	46.19	0.000	.9826049	1.069719
c.pindex#						
c.pindex	.5426225	.0224409	24.18	0.000	.4986285	.5866166
c.pindex#						
c.eindex	.5762576	.0319718	18.02	0.000	.5135788	.6389365
c.eindex#						
c.eindex	.5781937	.0223753	25.84	0.000	.5343283	.6220591
_cons	.0932159	.0221566	4.21	0.000	.0497793	.1366526

5.2 Mean effects

Having selected models for both the predicted probabilities and the conditional mean, we now use `poparms` to estimate the means of the potential-outcome distributions for each value of `w` in the sample. While we discuss the complete syntax of `poparms` in section 3.1, the output below illustrates that `poparms` takes two equations. The first equation specifies the model for the predicted probabilities; the second equation specifies the model for the conditional expectations (for the means and quantiles).

```
. poparms (w c.(pindex eindex)##c.(pindex eindex))
> (spmeasure c.(pindex eindex)##c.(pindex eindex))
Treatment Mean and Quantiles Estimation          Number of obs   =       5000
(efficient influence function)
```

spmeasure	Coef.	analytic Std. Err.	z	P> z	[95% Conf. Interval]	
w						
0	.1983929	.0114294	17.36	0.000	.1759916	.2207942
1	.5277476	.022959	22.99	0.000	.4827488	.5727464
2	.9149628	.0336986	27.15	0.000	.8489147	.981011

Below we use `contrast` to estimate the average treatment effects of changing the treatment level from 0 to 1 and from 0 to 2.

```
. contrast r.w, nowald
Warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins      : asbalanced
```

	Contrast	Std. Err.	[95% Conf. Interval]	
^w (1 vs 0)	.3293547	.025636	.2791091	.3796003
(2 vs 0)	.71657	.0355852	.6468241	.7863158

These results show the estimated average treatment effects of treatment-level 1 versus the control level of 0 and the estimated average treatment effects of treatment-level 2 versus the control level of 0. The results for 1 versus 0 are the same as above. The output indicates that the estimated average effect of changing the treatment level from 0 to 2 is 0.7166 and that it is statistically different from 0. The nonoverlapping confidence intervals suggest also that we will reject the null hypothesis that these two treatment effects have the same value, as discussed below.

Although it may not be the goal of our fictional study, another common task is to estimate all pairwise comparisons. Below we use `margins` to compute the estimates and use `marginsplot` to plot the three estimated effects and their confidence intervals.

```
. margins i.w, pwcompare
Warning: cannot perform check for estimable functions.
Pairwise comparisons of adjusted predictions
Model VCE      : analytic
Expression     : Linear prediction, predict()
```

	Delta-method		Unadjusted	
	Contrast	Std. Err.	[95% Conf. Interval]	
^w 1 vs 0	.3293547	.025636	.2791091	.3796003
2 vs 0	.71657	.0355852	.6468241	.7863158
2 vs 1	.3872153	.0300574	.3283037	.4461268

```
. marginsplot, unique plotopts(connect(none))
Variables that uniquely identify margins: _pw
    _pw enumerates all pairwise comparisons; _pw0 enumerates the reference
    categories; _pw1 enumerates the comparison categories.
```

Figure 4 shows the graph produced by `marginsplot`.

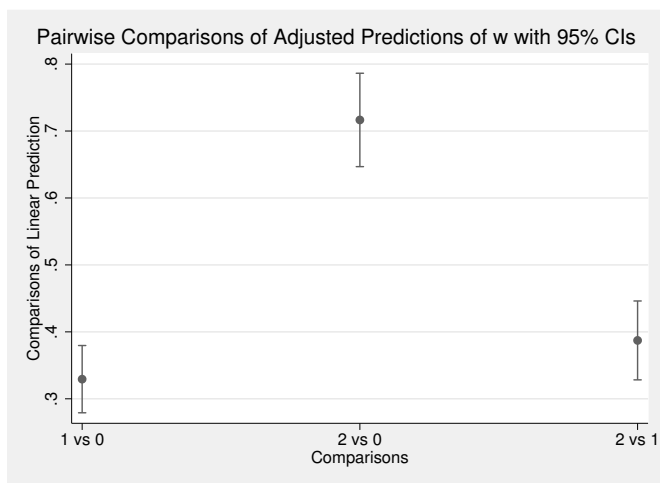


Figure 4. Estimated effects and confidence intervals

Although this example is trivial, the `marginsplot` command becomes increasingly useful as the number of treatment levels increases.

5.3 Quantile effects

`poparms` estimates not only the means but also the quantiles of the potential-outcome distributions. As discussed in section 2.3, we need the rank-preservation assumption to interpret the differences in the quantiles estimated by `poparms` as quantiles of treatment effects. Here rank preservation means that a student who would score at the, say, 50th percentile if all students received treatment 1 would also score at the 50th percentile if all students received treatment 2 or 3 (and similarly for all other students). This estimation is indeed done jointly, as we discuss in the next subsection.

In the output below, we use the `quantile()` option to jointly estimate the means, 0.25 quantiles, and the 0.75 quantiles of the 3 potential-outcome distributions.

```
. set seed 12345671
. poparms (w c.(pindex eindex)##c.(pindex eindex))
> (spmeasure c.(pindex eindex)##c.(pindex eindex)),
> quantile(.25 .5 .75)
Treatment Mean and Quantiles Estimation          Number of obs   =       5000
(efficient influence function)
```

spmeasure	Coef.	bootstrap Std. Err.	z	P> z	[95% Conf. Interval]	
mean						
w						
0	.1983929	.011701	16.96	0.000	.1754594	.2213263
1	.5277476	.0226869	23.26	0.000	.483282	.5722132
2	.9149628	.0344245	26.58	0.000	.847492	.9824337
q25						
w						
0	.0078533	.0008762	8.96	0.000	.006136	.0095707
1	-.1401748	.0123894	-11.31	0.000	-.1644575	-.1158921
2	-.4078067	.0281641	-14.48	0.000	-.4630073	-.3526061
q50						
w						
0	.0481364	.0029476	16.33	0.000	.0423592	.0539136
1	.1251076	.0157332	7.95	0.000	.0942711	.1559441
2	.2726521	.0384106	7.10	0.000	.1973687	.3479355
q75						
w						
0	.1930453	.0118169	16.34	0.000	.1698846	.2162059
1	.7353868	.0445843	16.49	0.000	.6480031	.8227705
2	1.567837	.0638119	24.57	0.000	1.442768	1.692907

The above output shows that the estimated 0.25 quantiles of the potential-outcome distributions corresponding to treatment levels 0, 1, and 2 are 0.0079, -0.1402 , and -0.4078 , respectively. Similarly, the output shows that the estimated 0.75 quantiles of the potential-outcome distributions corresponding to treatment levels 0, 1, and 2 are 0.1930, 0.7354, and 1.5678, respectively.

As stated in the output, the VCE was estimated by a bootstrap. We used the default number of 2,000 repetitions. We recommend using the bootstrap method to estimate the VCE when estimating quantiles, as discussed in section 7.3.

As in the case of means, our interest lies in differences of a given quantile across the potential-outcome distributions. The following questions describe some examples of why these comparisons may be of interest: Is the 0.25 quantile of the potential-outcome distribution at treatment level 1 the same as the 0.25 quantile of the potential-outcome distribution at treatment level 0? Is the 0.25 quantile of the potential-outcome distribution at treatment level 2 the same as the 0.25 quantile of the potential-outcome distribution at treatment level 0? The same questions can be recast for other quantiles.

In the output below, we use the `margins` command to formally estimate the differences of interest for the 0.25 quantile. This command uses the `predict(equation(#2))` option to specify the second equation (that is, the 25th quantile) in the output obtained above from `poparms`.

```
. margins i.w, pwcompare predict(equation(#2))
Warning: cannot perform check for estimable functions.
Pairwise comparisons of adjusted predictions
Model VCE      : bootstrap
Expression    : Linear prediction, predict(equation(#2))
```

		Delta-method		Unadjusted	
	Contrast	Std. Err.	[95% Conf. Interval]		
	^w				
	1 vs 0	-.1480281	.0124135	-.1723582	-.123698
	2 vs 0	-.41566	.0281952	-.4709216	-.3603984
	2 vs 1	-.2676319	.0287559	-.3239924	-.2112714

The output shows that the estimated difference between the 0.25 quantile of the population potential-outcome distribution when everyone gets class-type 1 and the 0.25 quantile of the population potential-outcome distribution when everyone gets class-type 0 is -0.1480 . The estimated difference between the 0.25 quantile of the population potential-outcome distribution when everyone gets class-type 2 and the 0.25 quantile of the population potential-outcome distribution when everyone gets class-type 0 is -0.4157 .

Next we look at the median, which is the third equation in the output from `poparms` above. Thus we obtain

```
. margins i.w, pwcompare predict(equation(#3))
Warning: cannot perform check for estimable functions.
Pairwise comparisons of adjusted predictions
Model VCE      : bootstrap
Expression    : Linear prediction, predict(equation(#3))
```

		Delta-method		Unadjusted	
	Contrast	Std. Err.	[95% Conf. Interval]		
	^w				
	1 vs 0	.0769712	.0159921	.0456273	.1083151
	2 vs 0	.2245157	.038555	.1489493	.300082
	2 vs 1	.1475445	.0365091	.0759879	.219101

5.4 Joint mean and quantile effects

`poparms` also allows for hypothesis testing within and across treatment levels and parameters of interest. To provide some illustrative examples, we first consider figure 5, which depicts the estimated means and quantiles for each treatment level with their corresponding 95% confidence intervals. The do-file that produces this figure can be downloaded from the Statistical Software Components; type `findit poparms` for a link.

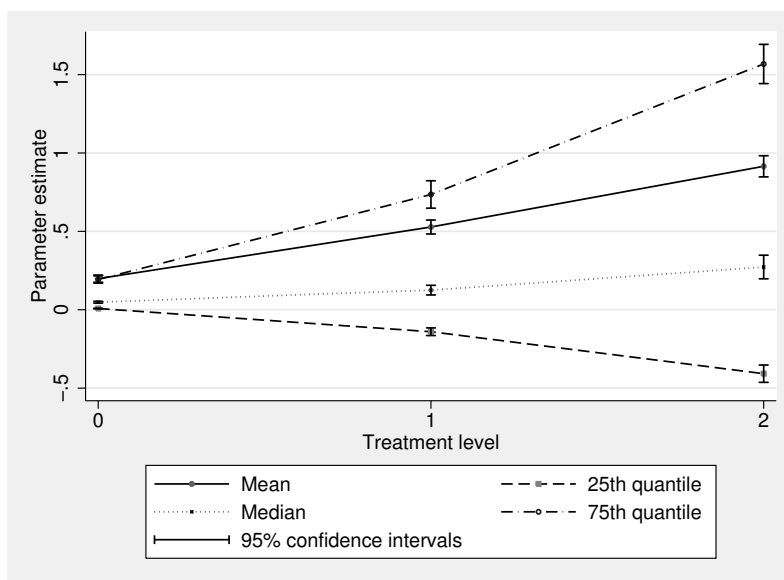


Figure 5. Estimated means and quantiles for treatment levels 0, 1, and 2

This figure shows that the parameters of the potential-outcome distributions have different trajectories over treatment levels. The means and medians increase almost linearly, the 25th quantiles decrease nonlinearly, and the 75th quantiles increase nonlinearly. These patterns imply that the spread of the distribution is growing as, in this case, the treatment level increases.

Next, motivated by figure 5, we illustrate how to perform three joint hypothesis tests within and across treatment levels. First, consider a hypothesis associated with zero-incremental treatment effects across means and quantiles. For example, consider the hypothesis that none of the parameters change over treatment level:

$$\begin{aligned}
 H_0 : \quad & \mu_0 = \mu_1 = \mu_2 \\
 & q_0(0.25) = q_1(0.25) = q_2(0.25) \\
 & q_0(0.50) = q_1(0.50) = q_2(0.50) \\
 & q_0(0.75) = q_1(0.75) = q_2(0.75)
 \end{aligned}$$

This joint hypothesis is implemented using the `test` command as follows:

```
. test (_b[mean:0.w] = _b[mean:1.w] = _b[mean:2.w])
> (_b[q25:0.w] = _b[q25:1.w] = _b[q25:2.w])
> (_b[q50:0.w] = _b[q50:1.w] = _b[q50:2.w])
> (_b[q75:0.w] = _b[q75:1.w] = _b[q75:2.w])

( 1)  [mean]0bn.w - [mean]1.w = 0
( 2)  [mean]0bn.w - [mean]2.w = 0
( 3)  [q25]0bn.w - [q25]1.w = 0
( 4)  [q25]0bn.w - [q25]2.w = 0
( 5)  [q50]0bn.w - [q50]1.w = 0
( 6)  [q50]0bn.w - [q50]2.w = 0
( 7)  [q75]0bn.w - [q75]1.w = 0
( 8)  [q75]0bn.w - [q75]2.w = 0

      chi2( 8) = 1246.68
      Prob > chi2 = 0.0000
```

As indicated by figure 5, we strongly reject this hypothesis.

Now consider the hypothesis that the increments, as opposed to the levels, across consecutive treatment levels are constant. The null hypothesis takes the form

$$\begin{aligned}
 H_0 : \quad & \mu_1 - \mu_0 = \mu_2 - \mu_1 \\
 & q_1(0.25) - q_0(0.25) = q_2(0.25) - q_1(0.25) \\
 & q_1(0.50) - q_0(0.50) = q_2(0.50) - q_1(0.50) \\
 & q_1(0.75) - q_0(0.75) = q_2(0.75) - q_1(0.75)
 \end{aligned}$$

which is implemented in this case as follows:

```
. test (_b[mean:1.w] - _b[mean:0.w] = _b[mean:2.w] - _b[mean:1.w] )
> (_b[q25:1.w] - _b[q25:0.w] = _b[q25:2.w] - _b[q25:1.w] )
> (_b[q50:1.w] - _b[q50:0.w] = _b[q50:2.w] - _b[q50:1.w] )
> (_b[q75:1.w] - _b[q75:0.w] = _b[q75:2.w] - _b[q75:1.w] )

( 1)  - [mean]0bn.w + 2*[mean]1.w - [mean]2.w = 0
( 2)  - [q25]0bn.w + 2*[q25]1.w - [q25]2.w = 0
( 3)  - [q50]0bn.w + 2*[q50]1.w - [q50]2.w = 0
( 4)  - [q75]0bn.w + 2*[q75]1.w - [q75]2.w = 0

      chi2( 4) = 24.91
      Prob > chi2 = 0.0001
```

We also strongly reject the null hypothesis of constant increments. This result is consistent with the nonlinear patterns in figure 5.

Figure 5 also suggests constant increments for the means and medians, so we formally test these hypotheses. For the means, we obtain

```
. test _b[mean:1.w] - _b[mean:0.w] = _b[mean:2.w] - _b[mean:1.w]
( 1)  - [mean]0bn.w + 2*[mean]1.w - [mean]2.w = 0

      chi2( 1) = 1.83
      Prob > chi2 = 0.1763
```

and fail to reject the null hypothesis of constant increments. For the medians, we obtain

```
. test _b[q50:1.w] - _b[q50:0.w] = _b[q50:2.w] - _b[q50:1.w]
( 1) - [q50]0bn.w + 2*[q50]1.w - [q50]2.w = 0
      chi2( 1) =    2.95
      Prob > chi2 =    0.0861
```

and only fail to reject the null hypothesis of constant increments at the 5% and 1% levels but not at the 10% level.

Finally, we conduct a joint hypothesis within each treatment level. In particular, we test whether the mean and the median are equal for all three potential-outcome distributions. That is,

$$\begin{aligned} H_0 : \quad \mu_0 &= q_0(0.50) \\ \mu_1 &= q_1(0.50) \\ \mu_2 &= q_2(0.50) \end{aligned}$$

This test yields

```
. test (_b[mean:0.w] = _b[q50:0.w]) (_b[mean:1.w] = _b[q50:1.w])
> (_b[mean:2.w] = _b[q50:2.w])
( 1) [mean]0bn.w - [q50]0bn.w = 0
( 2) [mean]1.w - [q50]1.w = 0
( 3) [mean]2.w - [q50]2.w = 0
      chi2( 3) =  770.76
      Prob > chi2 =    0.0000
```

which shows that these two parameters are indeed different across the three distributions, as expected from figure 5.

Many other point estimators and hypothesis tests can be easily constructed using the output of `poparms`. This covers a large class of potentially interesting treatment-effects estimands based on means and quantiles of the potential-outcome distributions.

6 Simulations

In this section, we discuss a series of Monte Carlo simulations that illustrate the finite-sample performance of the implemented methods. Overall, we found that the point estimates and the estimated VCE for the mean parameter estimators performed very well. The point estimates for the quantile parameters also performed very well, but the analytic estimator of the VCE, discussed in section 7.2, did not perform well because we have not yet found a bandwidth selection method that provides a sufficiently reliable estimator for the density used to estimate the VCE. A bootstrap estimator for the VCE with mean and quantile parameter estimators performed very well. Thus we recommend using the bootstrap estimator for the VCE of the quantile parameter estimator.

We describe two types of Monte Carlo experiments. The first uses an analytic estimator of the VCE, while the second uses a bootstrap estimator of the VCE. They share the same basic design. Next we describe the design and analytic results followed by the bootstrap results.

6.1 Basic simulation design

We consider four cases: i) the known functional form of nuisance functions $p_j(\cdot)$, $e_j(\cdot; \mu_j)$, and $e_j\{\cdot; q_j(\tau)\}$; ii) the known functional form of the GPS only; iii) the known functional form of the regression functions $e_j(\cdot; \mu_j)$ and $e_j\{\cdot; q_j(\tau)\}$ only; and iv) the unknown functional form of all nuisance functions. In each case, we drew 10,000 replications from the data-generating process (DGP); each replication had a sample size of 2,000. In each replication, we performed estimation and inference for 9 parameters: means, 0.25 quantiles, and 0.75 quantiles of 3 treatment levels ($j = 1, 2, 3$). For each parameter, repetition, and case, we recorded the EIF point estimate, the EIF standard error, a binary indicator of whether we reject the null hypothesis that the parameter equals its true value using the EIF point estimate and standard error, the IPW point estimate, the IPW standard error, and a binary indicator of whether we reject the null hypothesis that the parameter equals its true value using the IPW point estimate and standard error.

6.2 DGPs

We drew from four DGPs. After discussing the common features of all four, we discuss how they differ.

In all four DGPs, the GPSs are generated from a multinomial logit, and the outcome variable y comes from a Weibull distribution conditional on the treatment level w and the two covariates x_1 and x_2 . Each of the two covariates comes from a uniform distribution over $(-0.5, 0.5)$.

We chose a multinomial logit for the treatment levels $w \in \{1, 2, 3\}$ because we are interested in assessing what happens when we know the distribution from which the treatments are generated but not the function of the covariates.

We chose a Weibull distribution for y conditional on \mathbf{x} because it is unsymmetric and specifies the mean and quantiles are nonlinear functions of the parameters of the distribution. We used the Weibull distribution with scale parameter η and shape parameter θ , which has mean $\eta\Gamma\{(\theta + 1)/\theta\}$ and τ th quantile $\eta[\ln\{1/(1 - \tau)\}]^{(1/\theta)}$. By specifying functional forms for the distribution parameters $\eta(\mathbf{x}, w)$ and $\theta(w)$, we obtained a class of models for nonsymmetric distributions with analytic conditional means and quantiles. We also note that models are conditionally heteroskedastic with variance $\eta(\mathbf{x}, w)^2 (\Gamma\{[\theta(w) + 2]/\theta(w)\} - \{\Gamma\{[\theta(w) + 2]/\theta(w)\}\}^2)$.

In DGP 1, the functional forms for both the GPS and the conditional mean are known. In DGP 2, the functional form for the GPS is known, but the functional form for the

conditional mean is unknown. In DGP 3, the functional form for the GPS is unknown, but the functional form for the conditional mean is known. In DGP 4, the functional forms for both the GPS and the conditional mean are unknown.

Here we discuss the functional forms used in each case. Below we discuss how the estimation was performed. We use different functional forms for the cases of known and unknown forms because we want the unknown forms to be outside the set of forms that can be exactly represented.

We begin by describing how we generated the data on the treatment levels. Because there are three treatment levels ($w \in \{1, 2, 3\}$) and the true propensity score is a multinomial logit (with treatment level 1 as base level),

$$\mathbb{P}(w_i = 1) = \frac{1}{q_i}, \quad \mathbb{P}(w_i = 2) = \frac{ex_{2i}}{q_i}, \quad \mathbb{P}(w_i = 3) = \frac{ex_{3i}}{q_i}$$

where ex_{2i} is the functional form for the covariates for treatment level 2 at observation i , ex_{3i} is the functional form for the covariates for treatment level 3 at observation i , and $q_i = 1 + ex_{2i} + ex_{3i}$. Given the probabilities and the $(0, 1)$ uniform variate u_{wi} ,

$$w_i = \begin{cases} 1 & \text{if } u_{wi} \leq \mathbb{P}(w_i = 1) \\ 2 & \text{if } \mathbb{P}(w_i = 1) < u_{wi} \leq \mathbb{P}(w_i = 1) + \mathbb{P}(w_i = 2) \\ 3 & \text{otherwise} \end{cases}$$

When the functional form for the GPS is assumed known, we use

$$\begin{aligned} ex_{2i} &= \exp\{1.5(-.2 + x_{1i} + x_{2i})\} \\ ex_{3i} &= \exp\{1.2(-.1 + x_{1i} + x_{2i})\} \end{aligned}$$

If we use a standard MLM, the functional form for the GPS function in the known case is a polynomial in x_1 and x_2 . If we use a standard MLM, the functional form for the GPS function in the unknown case can only be approximated by a polynomial in x_1 and x_2 .

When the functional form for GPS is assumed unknown, we use

$$\begin{aligned} ex_{2i} &= \exp[0.1\{-0.8 + x_{1i} + x_{2i} + 3 \exp(x_{1i} + x_{2i})\}] \\ ex_{3i} &= \exp[0.2\{-0.8 + x_{1i} + x_{2i} + 2.5 \exp(x_{1i} + x_{2i})\}] \end{aligned}$$

We now describe how we generated y_i conditional on \mathbf{x}_i and w_i . In all cases, we set $\theta_i = w_i$. When the functional form for the conditional mean function is assumed known, we used $\eta_i = (w_i/3)(2 + x_{1i} + x_2 + x_{1i}^2 + x_{2i}^2 + x_{1i}x_{2i})$. When the functional form for the conditional mean function is assumed unknown, we used $\eta_i = (w_i/3)[2 + x_{1i} + x_{2i} + \exp\{\sqrt{w}(1 + x_{1i} + x_{2i})\}]$.

The functional form for the conditional mean in the known case is a polynomial in x_{1i} and x_{2i} . The functional form for the conditional mean in the unknown case can only be approximated by a polynomial in x_{1i} and x_{2i} .

6.3 Estimation procedures

In this section, we discuss how we performed the estimation and inference for each repetition over the four cases.

For case 1, in which the functional forms for GPS and the conditional mean are known, we specified these functional forms to the `poparms` command to obtain the EIF and IPW parameter estimates. We used the `poparms` estimation results to perform the Wald tests against the true null hypotheses.

For case 2, in which the functional form for GPS is known and the conditional mean is unknown, we specified the known functional form for the GPS and the functional form selected by `bfit` for the conditional mean to the `poparms` command to obtain the EIF and IPW parameter estimates. We used the `poparms` estimation results to perform the Wald tests against the true null hypotheses.

For case 3, in which the functional form for GPS is unknown and the conditional mean is known, we specified the functional form selected by `bfit` for the GPS and the known functional form for the conditional mean to the `poparms` command to obtain the EIF and IPW parameter estimates. We used the `poparms` estimation results to perform the Wald tests against the true null hypotheses.

For case 4, in which the functional forms for both GPS and the conditional mean are unknown, we specified the functional forms selected by `bfit` for the GPS and for the conditional mean to the `poparms` command to obtain the EIF and IPW parameter estimates. We used the `poparms` estimation results to perform the Wald tests against the true null hypotheses.

6.4 Results with analytic estimator for VCE

The detailed results are in tables 1–9.

Each table contains the results for a specific parameter. Each number in each table is computed over 10,000 repetitions. In each table, the first column specifies the functional-form case, the second column specifies the estimator, the third column gives the true value for the parameter, the fourth column gives the mean of the point estimates over the 10,000 repetitions, the fifth column gives the standard deviation of the point estimates over the 10,000 repetitions, the sixth column gives the mean of the estimated standard errors over the 10,000 repetitions, and the seventh column gives the mean of the rejection indicators over the 10,000 repetitions.

Ideally, the mean of the point estimates should be very close to the true value, the standard deviation of the point estimates should be very close to the mean of the standard errors, and the mean of the rejection indicators should be 0.05. Differences from these ideal relationships indicate that the finite-sample behavior of the estimator differs from the large-sample behavior.

Both the EIF and the IPW estimators performed very well for all cases in estimating the point estimates and the standard errors for the three mean parameters.

Both the EIF and the IPW estimators performed very well for all cases in estimating the quantile parameters, but the analytic estimator for the VCE performed poorly.

As discussed in section 7.2, the analytic estimator for the VCE for the EIF and IPW estimators of the quantile parameters requires a density estimator of the potential-outcome variables at specific (estimated) points. We implemented an IPW-based nonparametric kernel density estimator to construct these analytic quantile standard-error estimators. These estimators require a choice of bandwidth for their implementation. Following standard methods, we experimented with an ad hoc ROT bandwidth selector to construct the weighted kernel density estimator at the estimated quantiles. This ROT choice of bandwidth is ad hoc because it is constructed on the basis of the (asymptotic) mean-square error of a kernel density estimator using the potential outcomes rather than the observed outcomes (and using inverse-probability weighting). See section 7.2 for further details.

Tables 4–9 present the simulation results using the plug-in bandwidth discussed in section 7.2 to construct an analytic VCE estimator. These results show that this analytic approach performs poorly in some cases. We found in our simulations that the results are highly sensitive to the specific choice of bandwidth, but the overall performance of the procedures improves as the sample size increases. (We do not report additional simulation results for different bandwidth choices and sample sizes to conserve space.) To verify that estimating the density was the source of the poor performance of the analytic VCE estimator, we reran the simulations, replacing the kernel density estimator with the population value of the density implied by the DGP, and found that the analytic VCE estimator using these infeasible density values performs very well in all the sample sizes considered. Further research on bandwidth selection for quantile treatment effects is underway. In the meantime, we recommend using the nonparametric bootstrap VCE estimator discussed in section 7.3. In the next subsection, we report some simulation results for this bootstrap estimator.

Table 1. Mean, treatment 1

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	0.7222	0.7223	0.03017	0.02997	0.0536
ps known cm known	IPW	0.7222	0.7223	0.03031	0.02997	0.0548
ps known cm unknown	EIF	1.317	1.317	0.06333	0.06177	0.0586
ps known cm unknown	IPW	1.317	1.317	0.06381	0.06177	0.0627
ps unknown cm known	EIF	0.7222	0.7226	0.03386	0.03321	0.0602
ps unknown cm known	IPW	0.7222	0.721	0.03356	0.03321	0.0592
ps unknown cm unknown	EIF	1.317	1.318	0.06751	0.06627	0.0559
ps unknown cm unknown	IPW	1.317	1.313	0.06623	0.06627	0.0542

Table 2. Mean, treatment 2

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	1.28	1.28	0.02909	0.02884	0.0534
ps known cm known	IPW	1.28	1.28	0.02905	0.02884	0.0523
ps known cm unknown	EIF	3.454	3.454	0.09047	0.08976	0.0514
ps known cm unknown	IPW	3.454	3.454	0.09013	0.08976	0.0499
ps unknown cm known	EIF	1.28	1.28	0.02695	0.02682	0.0518
ps unknown cm known	IPW	1.28	1.28	0.02701	0.02682	0.0528
ps unknown cm unknown	EIF	3.454	3.453	0.09084	0.09012	0.0541
ps unknown cm unknown	IPW	3.454	3.453	0.09191	0.09012	0.0571

Table 3. Mean, treatment 3

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	1.935	1.935	0.02917	0.02875	0.0543
ps known cm known	IPW	1.935	1.935	0.02915	0.02875	0.0547
ps known cm unknown	EIF	7.335	7.333	0.1633	0.163	0.0531
ps known cm unknown	IPW	7.335	7.338	0.1661	0.163	0.0559
ps unknown cm known	EIF	1.935	1.935	0.0265	0.02663	0.0497
ps unknown cm known	IPW	1.935	1.936	0.02661	0.02663	0.051
ps unknown cm unknown	EIF	7.335	7.332	0.157	0.1565	0.0502
ps unknown cm unknown	IPW	7.335	7.352	0.1613	0.1565	0.0567

Table 4. Quantile 0.25, treatment 1

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	0.2016	0.2016	0.01512	0.01788	0.0208
ps known cm known	IPW	0.2016	0.2016	0.01514	0.01788	0.0216
ps known cm unknown	EIF	0.326	0.3258	0.02409	0.03012	0.0152
ps known cm unknown	IPW	0.326	0.3257	0.02399	0.03012	0.0152
ps unknown cm known	EIF	0.2016	0.2017	0.01765	0.02149	0.0181
ps unknown cm known	IPW	0.2016	0.2013	0.0176	0.02148	0.0173
ps unknown cm unknown	EIF	0.326	0.326	0.02823	0.03659	0.0121
ps unknown cm unknown	IPW	0.326	0.3259	0.02803	0.03659	0.0109

Table 5. Quantile 0.25, treatment 2

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	0.7429	0.742	0.03394	0.03559	0.0419
ps known cm known	IPW	0.7429	0.742	0.03391	0.0356	0.0414
ps known cm unknown	EIF	1.484	1.481	0.0773	0.08657	0.0279
ps known cm unknown	IPW	1.484	1.482	0.0817	0.08659	0.0371
ps unknown cm known	EIF	0.7429	0.742	0.02895	0.03035	0.042
ps unknown cm known	IPW	0.7429	0.742	0.02892	0.03035	0.0414
ps unknown cm unknown	EIF	1.484	1.483	0.06531	0.07275	0.0292
ps unknown cm unknown	IPW	1.484	1.483	0.06591	0.07275	0.0308

Table 6. Quantile 0.25, treatment 3

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	1.357	1.355	0.03725	0.0386	0.0443
ps known cm known	IPW	1.357	1.355	0.03723	0.0386	0.0449
ps known cm unknown	EIF	3.172	3.165	0.1231	0.1399	0.026
ps known cm unknown	IPW	3.172	3.167	0.134	0.14	0.0418
ps unknown cm known	EIF	1.357	1.355	0.0338	0.0351	0.0432
ps unknown cm known	IPW	1.357	1.355	0.03382	0.0351	0.044
ps unknown cm unknown	EIF	3.172	3.166	0.1144	0.1291	0.0248
ps unknown cm unknown	IPW	3.172	3.149	0.1187	0.129	0.0334

Table 7. Quantile 0.75, treatment 1

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	0.9892	0.988	0.04911	0.04751	0.0611
ps known cm known	IPW	0.9892	0.988	0.04915	0.04751	0.0629
ps known cm unknown	EIF	1.746	1.743	0.09621	0.09143	0.0677
ps known cm unknown	IPW	1.746	1.743	0.09669	0.09144	0.0712
ps unknown cm known	EIF	0.9892	0.9873	0.05577	0.05358	0.0664
ps unknown cm known	IPW	0.9892	0.9853	0.05533	0.05347	0.0668
ps unknown cm unknown	EIF	1.746	1.744	0.1054	0.1012	0.0665
ps unknown cm unknown	IPW	1.746	1.74	0.104	0.1009	0.0657

Table 8. Quantile 0.75, treatment 2

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	1.689	1.687	0.04718	0.04647	0.0578
ps known cm known	IPW	1.689	1.687	0.04705	0.04648	0.0555
ps known cm unknown	EIF	4.562	4.556	0.1644	0.1571	0.0646
ps known cm unknown	IPW	4.562	4.556	0.1646	0.157	0.0657
ps unknown cm known	EIF	1.689	1.688	0.04324	0.04292	0.0542
ps unknown cm known	IPW	1.689	1.688	0.0433	0.04292	0.0531
ps unknown cm unknown	EIF	4.562	4.555	0.161	0.1541	0.0638
ps unknown cm unknown	IPW	4.562	4.556	0.1627	0.1542	0.0664

Table 9. Quantile 0.75, treatment 3

Case	Estimator	True value	Mean estimates	Standard deviation estimates	Mean standard error	Rejection rate
ps known cm known	EIF	2.413	2.411	0.04449	0.04374	0.0552
ps known cm known	IPW	2.413	2.411	0.04444	0.04375	0.056
ps known cm unknown	EIF	9.547	9.546	0.3049	0.2963	0.0604
ps known cm unknown	IPW	9.547	9.538	0.3102	0.296	0.0654
ps unknown cm known	EIF	2.413	2.411	0.04072	0.04069	0.0528
ps unknown cm known	IPW	2.413	2.412	0.04087	0.04071	0.0534
ps unknown cm unknown	EIF	9.547	9.543	0.2898	0.2827	0.058
ps unknown cm unknown	IPW	9.547	9.566	0.2985	0.2835	0.0628

6.5 Bootstrap VCE results

We recommend using at least 2,000 repetitions when using the bootstrap estimator of the VCE discussed in this section. As a result, each repetition in our simulation study takes a lot of time, and considering all possible designs becomes very time consuming. To make the simulations feasible, we report only results for the EIF estimator in the case of “ps known cm known”. (These simulations required more than seven days to complete.)

Table 10 presents the results for this case. We found that the bootstrap VCE estimator performed well, leading to confidence intervals with good empirical coverage rates in all cases. For example, for the quantile 0.25 in treatment 1 (table 4), a 5% nominal test exhibited an empirical rejection rate of 2.08% when using the analytic VCE estimator, but the empirical rejection rate was 5.50% when using the bootstrap VCE estimator.

Table 10. Bootstrap results

Parameter	True	Mean	Standard deviation	Mean standard error	Rejection rate
m_1	0.7222	0.7213	0.03043	0.03005	0.058
m_2	1.28	1.281	0.0285	0.02907	0.044
m_3	1.935	1.935	0.02861	0.02887	0.048
$q25_1$	0.2016	0.2013	0.0147	0.01541	0.055
$q25_2$	0.7429	0.7422	0.03283	0.03445	0.046
$q25_3$	1.357	1.355	0.03651	0.03779	0.0585
$q75_1$	0.9892	0.9861	0.04931	0.04943	0.0605
$q75_2$	1.689	1.69	0.04676	0.04783	0.06
$q75_3$	2.413	2.411	0.04413	0.04424	0.067

Given the good performance of the EIF point estimator in the other cases and the similar performance of the IPW estimator, we expect these results to be representative for the other cases discussed in the previous section.

7 Details on implementation

This section discusses the details of implementing the IPW and EIF estimators, the associated VCE, and the pre- and postestimation procedures discussed in the previous sections.

7.1 `bfit`

`bfit` creates the set of candidate models for a given set of *indepvars*. The method is the same for all subcommands.

1. `bfit` partitions the *indepvars* into discrete variables *dvarlist* and continuous variables *cvarlist*.
2. `bfit` uses factor-variable notation to define the fully interacted polynomial of the specified order of the continuous variables.

For example, for continuous variables `x1`, `x2`, and `corder(3)`, this step produces `c.(x1 x2)##c.(x1 x2)##c.(x1 x2)`.

3. `bfit` uses `fvexpand` to expand the factor-variable notation version of the fully interacted polynomial of the specified order of the continuous variables, which we denote by `fvclist`.

For example, `c.(x1 x2)##c.(x1 x2)##c.(x1 x2)` expands to `x1 x2 c.x1#c.x1 c.x1#c.x2 c.x2#c.x2 c.x1#c.x1#c.x1 c.x1#c.x1#c.x2 c.x1#c.x2#c.x2 c.x2#c.x2#c.x2`.

4. `bfit` loops over the terms in `fvclist`, progressively building up the varlist `clist`. The first time through the loop, `clist` contains only the first term in `fvclist`. The second time through the loop, `clist` contains the first two terms in `fvclist`. The k th time through the loop, `clist` contains the first k terms in `fvclist`.

For each step in the process of building up `clist` to be the same as `fvclist`, `bfit` creates the following candidate models.

- a. `bfit` defines a candidate model with the current variables in `clist`.
- b. In a process analogous to the one used for the terms in `fvclist`, `bfit` progressively builds up `dlist` from the list `dvarlist`.

For each version of `dlist`, `bfit` does the following steps.

- i. `bfit` creates a candidate model with `dlist` included as additive factors. For example, for given `dlist` and `clist`, the candidate model is `i.(dlist) clist`.
- ii. `bfit` creates a candidate model with `dlist` fully interacted with `clist`. For example, for given `dlist` and `clist`, the candidate model is `i.(dlist)##(clist)`.

7.2 poparms

In this section, we discuss the implementation details underlying the `poparms` command.

First, we are interested in conducting joint inference on the means and on the quantiles of the $(J + 1)$ potential-outcome distributions, so we need notation for the full parameter vector. As can be seen in the `poparms` output presented above, we nest treatment levels within parameter type, which yields the parameter vector

$$\beta = \{\boldsymbol{\mu}', \mathbf{q}(\tau_1)', \mathbf{q}(\tau_2)', \dots, \mathbf{q}(\tau_{k_\tau})'\}'$$

with the $J + 1$ means in $\boldsymbol{\mu} = (\mu_0, \mu_1, \dots, \mu_J)'$ and the $J + 1$ τ_ℓ quantiles in $\mathbf{q}(\tau_\ell) = \{q_0(\tau_\ell), q_1(\tau_\ell), \dots, q_J(\tau_\ell)\}'$ for each $\tau_1, \tau_2, \dots, \tau_{k_\tau}$ with $k_\tau \geq 0$. Note that $k_\tau = 0$ means that the quantiles are not considered. We have a total of $(J + 1) \times (1 + k_\tau)$ parameters, and hence, β is $1 \times \{(J + 1)(1 + k_\tau)\}$.

Second, using this notation, we define the stacked version of the observation-level contributions to the moment conditions characterizing the asymptotic behavior of these estimators. As mentioned above, these definitions are not needed to construct the point

estimators but are essential to characterize the joint distribution of the estimators, thus permitting joint inference within and across treatment levels. We define

$$\boldsymbol{\psi}_{\text{EIF}}\{\mathbf{z}_i; \boldsymbol{\beta}, \mathbf{p}(\cdot), \mathbf{e}(\cdot; \boldsymbol{\beta})\} = \begin{bmatrix} \boldsymbol{\psi}_{\text{EIF}}\{\mathbf{z}_i; \boldsymbol{\mu}, \mathbf{p}(\mathbf{x}_i), \mathbf{e}(\mathbf{x}_i; \boldsymbol{\mu})\} \\ \boldsymbol{\psi}_{\text{EIF}}[\mathbf{z}_i; \mathbf{q}(\tau_1), \mathbf{p}(\mathbf{x}_i), \mathbf{e}\{\mathbf{x}_i; \mathbf{q}(\tau_1)\}] \\ \vdots \\ \boldsymbol{\psi}_{\text{EIF}}[\mathbf{z}_i; \mathbf{q}(\tau_{k_\tau}), \mathbf{p}(\mathbf{x}_i), \mathbf{e}\{\mathbf{x}_i; \mathbf{q}(\tau_{k_\tau})\}] \end{bmatrix}$$

where $\boldsymbol{\psi}_{\text{EIF}_i}\{\mathbf{z}_i; \boldsymbol{\beta}, \mathbf{p}(\mathbf{x}_i), \mathbf{e}(\mathbf{x}_i; \boldsymbol{\beta})\}$ is a $(1 + k_\tau)(J + 1) \times 1$ column vector,

$$\boldsymbol{\psi}_{\text{EIF}}\{\mathbf{z}_i; \boldsymbol{\mu}, \mathbf{p}(\cdot), \mathbf{e}(\cdot; \boldsymbol{\mu})\} = \begin{bmatrix} \boldsymbol{\psi}_{\text{EIF}}\{\mathbf{z}_i; \mu_0, p_0(\cdot), e_j(\cdot; \mu_0)\} \\ \boldsymbol{\psi}_{\text{EIF}}\{\mathbf{z}_i; \mu_1, p_1(\cdot), e_j(\cdot; \mu_1)\} \\ \vdots \\ \boldsymbol{\psi}_{\text{EIF}}\{\mathbf{z}_i; \mu_J, p_J(\cdot), e_j(\cdot; \mu_J)\} \end{bmatrix}$$

is a $(J + 1) \times 1$ column vector, and

$$\boldsymbol{\psi}_{\text{EIF}}(\mathbf{z}_i; \mathbf{q}(\tau_\ell), \mathbf{p}[\mathbf{x}_i], \mathbf{e}\{\mathbf{x}_i; \mathbf{q}(\tau_\ell)\}) = \begin{bmatrix} \boldsymbol{\psi}_{\text{EIF}}[\mathbf{z}_i; q_0(\tau_\ell), p_0(\cdot), e_0\{\cdot; q_1(\tau_\ell)\}] \\ \boldsymbol{\psi}_{\text{EIF}}[\mathbf{z}_i; q_1(\tau_\ell), p_1(\cdot), e_1\{\cdot; q_2(\tau_\ell)\}] \\ \vdots \\ \boldsymbol{\psi}_{\text{EIF}}[\mathbf{z}_i; q_J(\tau_\ell), p_J(\cdot), e_J\{\cdot; q_{J+1}(\tau_\ell)\}] \end{bmatrix}$$

is a $(J + 1) \times 1$ column vector, for $\ell = 1, 2, \dots, k_\tau$, with

$$\begin{aligned} \mathbf{p}(\mathbf{x}) &= \{p_0(\mathbf{x}), p_1(\mathbf{x}), \dots, p_J(\mathbf{x})\}' \\ \mathbf{e}(\mathbf{x}, \boldsymbol{\mu}) &= \{e_0(\mathbf{x}, \mu_0), e_1(\mathbf{x}, \mu_1), \dots, e_J(\mathbf{x}, \mu_J)\}' \\ \mathbf{e}\{\mathbf{x}, \mathbf{q}(\tau)\} &= [e_j\{\mathbf{x}; q_0(\tau)\}, e_j\{\mathbf{x}; q_1(\tau)\}, \dots, e_j\{\mathbf{x}; q_J(\tau)\}]' \end{aligned}$$

Recall that $p_j(\mathbf{x}) = \mathbb{P}(w = j|\mathbf{x})$,

$$e_j(\mathbf{x}; \mu_j) = \mathbb{E}(y - \mu_j | \mathbf{x}, w = j), \quad e_j\{\mathbf{x}; q_j(\tau)\} = \mathbb{E}[\mathbf{1}\{y \leq q_j(\tau)\} - \tau | \mathbf{x}, w = j]$$

for each treatment level j .

Third, the semiparametric IPW and EIF estimators considered in `poparms` use polynomial-regression series estimators to approximate the unknown functions $\mathbf{p}(\mathbf{x})$, $\mathbf{e}(\mathbf{x}, \boldsymbol{\mu})$, and $\mathbf{e}\{\mathbf{x}, \mathbf{q}(\tau)\}$. Thus we denote $\mathbf{z}_p(\mathbf{x})$ and $\mathbf{z}_e(\mathbf{x})$ as the polynomial basis in \mathbf{x} of a given order used to approximate, respectively, the function $\mathbf{p}(\mathbf{x})$ and the two functions $\mathbf{e}(\mathbf{x}, \boldsymbol{\mu})$ and $\mathbf{e}\{\mathbf{x}, \mathbf{q}(\tau)\}$. We use the same approximating basis for the latter two functions for simplicity. Note that in the syntax diagram in section 3.1, the variables in $\mathbf{z}_p(\mathbf{x})$ are specified to `poparms` as `gpsvars` and that the variables in $\mathbf{z}_e(\mathbf{x}_i)$ are specified to `poparms` as `cvars`. Thus the `poparms` command allows for any basis of approximation, although our implementation based on `bfit` focuses on polynomial regression, the terms of which are selected in a preliminary step, as discussed above.

To approximate the GPS $\mathbf{p}(\mathbf{x})$, we follow Cattaneo (2010) and use a nonlinear multinomial logit sieve estimation approach. That is, the variables specified in *gpvars*, denoted here by $\mathbf{z}_p(\mathbf{x})$, are assumed to be a sufficiently flexible polynomial in the conditioning variables so that we can consistently estimate (or approximate) the GPS by multinomial logit. Thus, given the $\mathbf{z}_p(\mathbf{x})$, we estimate the multinomial logit parameters by maximum pseudolikelihood: with the standard normalization that $\gamma_0 = \mathbf{0}$, the $j = \{1, \dots, J\}$ vectors of multinomial logit parameters γ_j , we solve

$$\hat{\gamma}_j = \arg \max_{\gamma} \sum_{i=1}^n \sum_{j=0}^J d_i(j) \ln \left[\frac{\exp\{\mathbf{z}_p(\mathbf{x}_i)' \gamma_j\}}{\sum_{j=0}^J \exp\{\mathbf{z}_p(\mathbf{x}_i)' \gamma_j\}} \right]$$

where $\gamma = (0', \gamma_1', \gamma_2', \dots, \gamma_J')'$. Given these parameter estimates, each element of the estimated GPS is

$$\hat{p}_j(\mathbf{x}) = \left[\frac{\exp\{\mathbf{z}_p(\mathbf{x})' \hat{\gamma}_j\}}{1 + \sum_{j=1}^J \exp\{\mathbf{z}_p(\mathbf{x})' \hat{\gamma}_j\}} \right], \quad j = 0, 1, \dots, J$$

In the case of the conditional expectations $\mathbf{e}(\mathbf{x}, \boldsymbol{\mu})$ and $\mathbf{e}\{\mathbf{x}, \mathbf{q}(\tau)\}$, for each candidate value of $\boldsymbol{\mu}$ and $\mathbf{q}(\tau)$, we approximate each component of the vector by using a linear sieve based on the covariates provided in *cvars*, which we denote $\mathbf{z}_e(\mathbf{x})$. If *bf* is used in a preliminary step, then the basis functions in $\mathbf{z}_e(\mathbf{x})$ take the form of polynomials up to the order selected. Thus, for each treatment level, we solve the problems

$$\hat{\gamma}_j(\mu_j) = \arg \max_{\gamma_j} \sum_{i=1, w_i=j}^n \{y_i - \mu_j - \mathbf{z}_e(\mathbf{x}_i)' \gamma_j\}^2$$

and

$$\hat{\gamma}_j\{q_j(\tau)\} = \arg \max_{\gamma_j} \sum_{i=1, w_i=j}^n [\mathbf{1}\{y \leq q_j(\tau)\} - \tau - \mathbf{z}_e(\mathbf{x}_i)' \gamma_j]^2$$

which gives the estimators, respectively,

$$\hat{e}_j(\mathbf{x}; \mu_j) = \mathbf{z}_e(\mathbf{x})' \hat{\gamma}_j(\mu_j), \quad e_j\{\mathbf{x}; q_j(\tau)\} = \mathbf{z}_e(\mathbf{x})' \hat{\gamma}_j\{q_j(\tau)\}$$

Once the nonparametric estimators have been constructed, the IPW and EIF procedures described above will lead to consistent, asymptotically normal, and semiparametric-efficient estimators of β under appropriate regularity conditions. Because the generalized method-of-moments problem we consider is just identified, each point estimator can be constructed separately, even though we will consider them all together to discuss joint semiparametric inference. Following the notation and discussion above, we denote the IPW estimators as $\hat{\beta}_{\text{IPW}}$ and the EIF estimators as $\hat{\beta}_{\text{EIF}}$. In particular, for each j , the analytic solution for $\hat{\mu}_{\text{EIF},j}$ is

$$\hat{\mu}_{\text{EIF},j} = \frac{1}{n} \sum_{i=1}^n \left[\frac{d_i(j)}{p_j(\mathbf{x}_i)} y_i - \left\{ \frac{d_i(j)}{p_j(\mathbf{x}_i)} - 1 \right\} \hat{y}_i(j) \right]$$

where $\widehat{y}_i(j)$ are the predicted values from regressing y_i on \mathbf{x}_i for those observations with $d_i(j) = 1$.

Under appropriate regularity conditions, it can be shown that

$$\sqrt{n} \left(\widehat{\boldsymbol{\beta}}_{\text{IPW}} - \boldsymbol{\beta} \right) \rightarrow_d \mathcal{N} \left(0, \mathbf{V}_{\text{SPEB}} \right) \quad \text{and} \quad \sqrt{n} \left(\widehat{\boldsymbol{\beta}}_{\text{EIF}} - \boldsymbol{\beta} \right) \rightarrow_d \mathcal{N} \left(0, \mathbf{V}_{\text{SPEB}} \right)$$

where $\mathbf{V}_{\text{SPEB}} = \boldsymbol{\Gamma}^{-1} \mathbf{V}_{\text{EIF}} \boldsymbol{\Gamma}^{-1}$ is the semiparametric efficiency bound for regular estimators of $\boldsymbol{\beta}$, and

$$\begin{aligned} \mathbf{V}_{\text{EIF}} &= \mathbb{E} \left[\boldsymbol{\psi}_{\text{EIF}} \{ \mathbf{z}_i; \boldsymbol{\beta}, \mathbf{p}(\cdot), \mathbf{e}(\cdot; \boldsymbol{\beta}) \} \boldsymbol{\psi}_{\text{EIF}} \{ \mathbf{z}_i; \boldsymbol{\beta}, \mathbf{p}(\cdot), \mathbf{e}(\cdot; \boldsymbol{\beta}) \}' \right] \\ \boldsymbol{\Gamma} &= \begin{bmatrix} \mathbf{I}(J+1) & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Gamma}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\Gamma}_{k_\tau} \end{bmatrix}, \quad \mathbf{I}(J+1) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{[(J+1) \times (J+1)]} \\ \boldsymbol{\Gamma}_\ell &= \begin{bmatrix} f_{y(0)} \{ q_0(\tau_\ell) \} & 0 & \cdots & 0 \\ 0 & f_{y(1)} \{ q_1(\tau_\ell) \} & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f_{y(J)} \{ q_J(\tau_\ell) \} \end{bmatrix} \end{aligned}$$

for $\ell = 1, \dots, k_\tau$, where $f_{y(j)}(y) = dF_{y(j)}(y)/dy$ is the density of the potential outcome $y(j)$ for all $j = 0, 1, \dots, J$.

It follows from the results above that under the appropriate regularity conditions, the IPW and EIF estimators are asymptotically equivalent to first order in the sense that $\sqrt{n}(\widehat{\boldsymbol{\beta}}_{\text{IPW}} - \widehat{\boldsymbol{\beta}}_{\text{EIF}}) = o_p(1)$. Furthermore, these asymptotic results show that the same standard-error estimator could be used for both estimators, which is given by

$$\widehat{\mathbf{V}}_{\text{SPEB}}/n = \widehat{\boldsymbol{\Gamma}}^{-1} \widehat{\mathbf{V}}_{\text{EIF}} \widehat{\boldsymbol{\Gamma}}^{-1}/n$$

where

$$\widehat{\mathbf{V}}_{\text{EIF}} = \frac{1}{n} \sum_{i=1}^n \boldsymbol{\psi}_{\text{EIF}_i} \left\{ \mathbf{z}_i; \widehat{\boldsymbol{\beta}}, \widehat{\mathbf{p}}(\cdot), \widehat{\mathbf{e}}(\cdot; \widehat{\boldsymbol{\beta}}) \right\} \boldsymbol{\psi}_{\text{EIF}_i} \left\{ \mathbf{z}_i; \widehat{\boldsymbol{\beta}}, \widehat{\mathbf{p}}(\cdot), \widehat{\mathbf{e}}(\cdot; \widehat{\boldsymbol{\beta}}) \right\}'$$

with $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_{\text{IPW}}$ or $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_{\text{EIF}}$ depending on the estimator considered, and the unknown densities entering the matrix $\boldsymbol{\Gamma}$ are replaced by some consistent estimators $\widehat{f}_{y(j)}(y)$, $j = 0, 1, \dots, J$, which are evaluated at $\widehat{q}_j(\tau)$, where $\widehat{q}_j(\tau)$ denotes a choice either in $\{\widehat{q}_{\text{IPW},j}(\tau_1), \dots, \widehat{q}_{\text{IPW},j}(\tau_{k_\tau})\}$ or in $\{\widehat{q}_{\text{EIF},j}(\tau_1), \dots, \widehat{q}_{\text{EIF},j}(\tau_{k_\tau})\}$ for, respectively, the IPW and EIF estimators.

We implement the estimators $\widehat{f}_{y(j)}(y)$ by using the IPW kernel density estimator

$$\widehat{f}_{y(j)}(y) = \left\{ h_n \sum_{i=1}^n \frac{d_i(j)}{\widehat{p}_j(\mathbf{x}_i)} \right\}^{-1} \sum_{i=1}^n \frac{d_i(j)}{\widehat{p}_j(\mathbf{x}_i)} K \left(\frac{y_i - y}{h_n} \right)$$

where h_n denotes a positive vanishing bandwidth sequence for each treatment level $j = 0, 1, \dots, J$. To construct a feasible version of this estimator, we use the ROT bandwidth choice $\hat{h}_n = 2.3449 \times \sigma_{y(j)} \times n^{-1/5}$, where $\sigma_{y(j)}^2 = \mathbb{V}\{y(j)\}$ is replaced by a consistent estimator. This choice may be justified using an integrated mean-squared error expansion for the infeasible kernel density estimator $\tilde{f}_{y(j)}(y) = \sum_{i=1}^n K\{(y_i(j) - y)/h_n\}/nh_n$. However, the ROT bandwidth choice \hat{h}_n is ad hoc for the estimator $\hat{f}_{y(j)}(y)$ and may not perform well. Deriving an optimal (ROT) bandwidth choice for the estimator $\hat{f}_{y(j)}(y)$ is beyond the scope of this article. An alternative way of choosing the bandwidths h_n could be based on some cross-validation procedure tailored to the particular structure of the IPW kernel density estimator $\hat{f}_{y(j)}(y)$.

7.3 Bootstrap estimator for the VCE

Our (nonparametric) bootstrap estimator for the VCE uses the following standard algorithm. For each of S bootstrap samples obtained using `bsample` (see [R] `bsample`), estimate the parameters using `poparms`. Then \hat{V}_{bs} is the variance matrix computed from these S random vectors. Under appropriate regularity conditions, it is possible to formally establish that this standard nonparametric bootstrap estimator will be consistent for the VCE.

8 Acknowledgments

We thank Daniel Millimet, Jeff Smith, and an anonymous referee for comments that improved this article. The first author gratefully acknowledges financial support from the National Science Foundation (SES 1122994).

9 References

- Busso, M., J. DiNardo, and J. McCrary. 2013. New evidence on the finite sample properties of propensity score reweighting and matching estimators. <http://emlab.berkeley.edu/~jmccrary/BDM2013.pdf>.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154.
- Cattaneo, M. D., and M. H. Farrell. 2011. Efficient estimation of the dose–response function under ignorability using subclassification on the covariates. In *Advances in Econometrics: Vol. 27—Missing Data Methods: Cross-Sectional Methods and Applications*, ed. D. M. Drukker, 93–127. Bingley, UK: Emerald.
- Chen, X., H. Hong, and A. Tarozzi. 2004. Semiparametric efficiency in GMM models of nonclassical measurement errors, missing data and treatment effects. Discussion Paper No. 1644, Cowles Foundation. <http://cowles.econ.yale.edu/P/cd/d16a/d1644.pdf>.

- . 2008. Semiparametric efficiency in GMM models with auxiliary data. *Annals of Statistics* 36: 808–843.
- Drukker, D. M., and V. Wiggins. 2004. Verifying the solution from a nonlinear solver: a case study: comment. *American Economic Review* 94: 397–399.
- Firpo, S. 2007. Efficient semiparametric estimation of quantile treatment effects. *Econometrica* 75: 259–276.
- Hahn, J. 1998. On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica* 66: 315–331.
- Heckman, J. J., H. Ichimura, and P. Todd. 1998. Matching as an econometric evaluation estimator. *Review of Economic Studies* 65: 261–294.
- Heckman, J. J., and E. J. Vytlačil. 2007. Econometric evaluation of social programs, part I: Causal models, structural models and econometric policy evaluation. In *Handbook of Econometrics*, ed. J. J. Heckman and E. Leamer, vol. 6B, 4779–4874. Amsterdam: Elsevier.
- Hirano, K., and G. W. Imbens. 2004. The propensity score with continuous treatments. In *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, ed. A. Gelman and X.-L. Meng, 73–84. Chichester, UK: Wiley.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189.
- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960.
- Horvitz, D. G., and D. J. Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association* 47: 663–685.
- Imbens, G. W. 2000. The role of propensity score in estimating dose–response functions. *Biometrika* 87: 706–710.
- . 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29.
- Imbens, G. W., W. Newey, and G. Ridder. 2007. Mean-squared-error calculations for average treatment effects. Working Paper 05.34, Institute of Economic Policy Research.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86.
- Kang, J. D. Y., and J. L. Schafer. 2007. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science* 22: 523–539.

- Khan, S., and E. Tamer. 2010. Irregular identification, support conditions, and inverse weight estimation. *Econometrica* 78: 2021–2042.
- Millimet, D. L., and R. Tchernis. 2009. On the specification of propensity scores, with applications to the analysis of trade policies. *Journal of Business and Economic Statistics* 27: 397–415.
- Morgan, S. L., and C. Winship. 2007. *Counterfactuals and Causal Inference: Methods and Principles for Social Research*. New York: Cambridge University Press.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- van der Laan, M. J., and J. M. Robins. 2003. *Unified Methods for Censored Longitudinal Data and Causality*. New York: Springer.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

About the authors

Matias D. Cattaneo is an associate professor of economics at the University of Michigan in the Department of Economics.

David Drukker is the director of econometrics at StataCorp.

Ashley Holland is an assistant professor of mathematics at Cedarville University in the Department of Science and Mathematics.