INSTITUT DE
STATISTIQUE

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

DISCUSSION
PAPER

0903

A COMPUTATIONALLY EFFICIENT, CONSISTENT BOOTSTRAP FOR INFERENCE WITH NON-PARAMETRIC DEA ESTIMATORS

KNEIP, A., SIMAR, L. and P. W. WILSON

A Computationally Efficient, Consistent Bootstrap for Inference with Non-parametric DEA Estimators

Alois Kneip Léopold Simar Paul W. Wilson*

January 2009

Abstract

We develop a tractable, consistent bootstrap algorithm for inference about Farrell-Debreu efficiency scores estimated by non-parametric data envelopment analysis (DEA) methods. The algorithm allows for very general situations where the distribution of the inefficiencies in the input-output space may be heterogeneous. Computational efficiency and tractability are achieved by avoiding the complex double-smoothing procedure in the algorithm proposed by Kneip et al. (2008). In particular, we avoid technical difficulties in the earlier algorithm associated with smoothed estimates of a density with unknown, nonlinear, multivariate bounded support requiring complicated reflection methods. The new procedure described here is relatively simple and easy to implement: for particular values of a pair of smoothing parameters, the computational complexity is the same as the (inconsistent) naive bootstrap. The resulting computational speed allows the bootstrap to be iterated in order to optimize the smoothing parameters. From a practical viewpoint, only standard packages for computing DEA efficiency estimates, i.e., solving linear problems, are required for implementation. The performance of the method in finite samples is illustrated through some simulated examples.

*Kneip: Institut für Gessellschafts- und Wirtschaftswissenschaften, Statistische Abteilung, Universität Bonn, Adenauerallee 24-26, 53113 Bonn, Germany; email kneip@uni-bonn.de. Simar: Institut de Statistique, Université Catholique de Louvain, Voie du Roman Pays 20, Louvain-la-Neuve, Belgium; email simar@stat.ucl.ac.be. Wilson: The John E. Walker Department of Economics, 222 Sirrine Hall, Clemson University, Clemson, South Carolina 29634–1309, USA; email pww@clemson.edu. Financial support from the "Interuniversity Attraction Pole", Phase VI (No. P6/03) from the Belgian Government (Belgian Science Policy) is gratefully acknowledged. This work was made possible by the Palmetto cluster and a Condor pool deployed and maintained by Clemson Computing and Information Technology. We are grateful for technical support from Sebastien Goasguen, J. Barr von Ohesen, and the staff from the Cyberinfrastructure Technology Integration group at Clemson University. Of course, the usual caveats apply. *JEL* classification codes: C12, C14, C15

1 Introduction

Non-parametric data envelopment analysis (DEA) estimators have been widely used in studies of productive efficiency by firms, government agencies, national economies, and other decision-making units; Gattoufi et al. (2004) cite more than 1,800 published articles appearing in more than 400 journals. DEA estimators rely on linear programming methods along the lines of Charnes et al. (1978, 1979) and Färe et al. (1985) to estimate efficiency measures proposed by Debreu (1951), Farrell (1957), Shephard (1970), and others. DEA estimators measure efficiency relative to an ***estimate*** of an unobserved ***true*** frontier, conditional on observed data resulting from an underlying data-generating process (DGP). Under certain assumptions the DEA ***frontier*** estimator is a consistent, maximum likelihood estimator (Banker, 1993), with rates of convergence given by Korostelev et al. (1995). Consistency and convergence rates of DEA ***efficiency*** estimators have been established by Kneip et al. (1998); see Simar and Wilson (2000b) for a survey of the statistical properties of DEA estimators.

Although DEA estimators have been widely used, inference about the underlying efficiencies that are estimated remains problematic. Gijbels et al. (1999) derived the asymptotic distribution of a DEA efficiency estimators in the case of one input and one output, permitting classical inference in this special, limited case. However, one of the attractive features of DEA estimators is that they simultaneously allow both multiple inputs as well as multiple outputs. Simar and Wilson (1998, 2000a) proposed bootstrap methods for inference about efficiency based on DEA estimators in a multivariate framework, but consistency of these procedures has not been established.

Jeong (2004) derived the limiting distribution of DEA efficiency estimators under variable returns to scale for the special case $p = 1,\ q \geq 1$ in the input orientation (or $p \geq 1,\ q = 1$ in the output orientation), where p and q denote the numbers of inputs and outputs, respectively. Kneip et al. (2008) derived the limiting distribution of DEA efficiency estimators under variable returns to scale, with arbitrary numbers of inputs and outputs. This distribution contains several unknown quantities, and is not useful in a practical sense for inference. Kneip et al. (2008) also proposed two bootstrap procedures for inference about efficiency, and proved consistency of both methods. The first approach uses sub-sampling, where bootstrap samples of size $m < n$ are drawn (independently, with replacement) from the empirical

distribution of the original n sample observations. Unfortunately, there seems to be no reliable way of optimizing the size m of the sub-samples—bootstrap sample sizes in the inner loop of an iterated bootstrap along the lines of Hall (1992) are typically far too small to yield reliable information about coverages of the outer-loop bootstrap. Moreover, simulation results presented in Kneip et al. (2008) indicate that in finite-sample scenarios, coverages of confidence intervals for efficiency estimated by bootstrap sub-sampling are quite sensitive to the choice of the sub-sample size m.

The second, full-sample bootstrap procedure described by Kneip et al. (2008) requires for consistency not only smoothing the distribution of the observations as proposed in Simar and Wilson (1998, 2000a), but also smoothing of the initial DEA estimate of the frontier itself. This necessitates choosing values for two smoothing parameters. One of these can be optimized using existing methods from kernel density estimation, while a simple rule-of-thumb is provided for selecting the bandwidth used to smooth the frontier estimate. Simulation results presented in Kneip et al. indicate that the method works moderately well if smoothing parameters are chosen appropriately. In applications, one could in principle use cross-validation methods based on an iterated bootstrap to optimize the smoothing parameters Unfortunately, however, the method requires solving n auxiliary linear programs (each with $(p+q+1)$ constraints and $(n+1)$ weights, where $(p+q)$ is the sum of input and output dimensions and n represents sample size) for ***each*** of B bootstrap replications. As will be seen below in Section 4.4, the number of linear programs that must be solved to implement the method proposed in Kneip et al. (2008) greatly exceeds the number that must be solved in the method proposed in this paper. Consequently, cross-validation using an iterated bootstrap would be computationally infeasible with the method proposed by Kneip et al. (2008) in any situation with a reasonable number of observations.

The bootstrap procedure proposed in this paper is proved to be consistent and offers a dramatic reduction in computational burden relative to the full-sample bootstrap proposed by Kneip et al. (2008). Moreover, simulation results presented below in Section 6 indicate that the method proposed here performs as well as or better in terms of coverages of estimated confidence intervals than the full-sample method in Kneip et al. (2008). Although the new procedure also requires two smoothing parameters, it avoids the computational complexity resulting from the need to solve auxiliary linear programs on each bootstrap replication.

The resulting decrease in computational burden over the Kneip et al. (2008) method makes feasible a cross-validation technique that can be used to optimize the smoothing parameters. Even with the cross-validation, the computational burden remains small enough to make the method useful in practical, applied situations. Even in situations with many thousands of observations, recent advances in high-performance and high-throughput computing discussed below in Section 7 make the method useful to practitioners.

The naive bootstrap—based on re-sampling from the empirical distribution of the data—is attractive for its simplicity and low computational burden, but is inconsistent in situations where DEA efficiency estimators are used. As discussed by Simar and Wilson (1999b, 1999a), the inconsistency arises in part from the fact that when drawing from the empirical distribution, observations lying on the initial DEA frontier estimate are too-frequently selected.[1] The idea underlying the approach in this paper is to retain the simple features of the naive bootstrap to construct the part of bootstrap samples lying "far" from the estimated frontier, while drawing from a smooth, uniform distribution to construct the part of the bootstrap sample lying "near" the estimated frontier. The distinction between "near" and "far" is controlled by a smoothing parameter, while a second smoothing parameter controls the degree of smoothing applied to the estimated frontier. Since no distributions are estimated, and no auxiliary linear programs are needed, the speed of our procedure is similar to that of the naive bootstrap. Unlike the naive bootstrap for DEA estimators, however, our method is consistent.

The remainder of the paper evolves along the following lines: Section 2 establishes notation and describes DEA efficiency estimators as well as the underlying, true efficiencies that are estimated. Section 3 re-characterizes the model in a transformed space; this is necessary for implementing our bootstrap, as well as proving its consistency. The bootstrap algorithm is developed in Section 4, and proved to be consistent in Section 4.5. A data-driven method for determining sensible values of the required smoothing parameters is described in Section 5. Section 6 presents results of Monte Carlo experiments, giving estimates of realized coverages of confidence intervals estimated by the new bootstrap procedure. Additional results from Monte Carlo experiments showing how such methods might be expected to perform in

[1] In particular, while the empirical distribution is always a consistent estimator of the corresponding population distribution function, the empirical distribution function does not converge uniformly.

applied settings, where researchers must select specific values for smoothing parameters, are also given. Summary and conclusions are given in the final section.

2 Non-parametric DEA Estimators

In the production framework of Koopmans (1951), Debreu (1951), and Farrell (1957), firms transform a p-vector of input quantities into a q-vector of output quantities, with $p,\ q \geq 1$. Inputs are typically factors of production (e.g., labor, capital, materials, etc.), whereas outputs are typically goods, services, etc. In loose terms, a firm is said to become more efficient if it increases at least some of its output levels without increasing its input levels (output orientation), or alternatively if it reduces its use of at least some inputs without decreasing output levels (input orientation). Consider a fixed point of interest $(x_0, y_0) \in \mathbb{R}^p_+ \times \mathbb{R}^q_+$; this point may correspond to the input and output quantities used by a particular firm, or it might represent the input-output combination of a hypothetical firm. Let

$$\Psi = \{(x, y) \in \mathbb{R}^p_+ \times \mathbb{R}^q_+ \mid x \text{ can produce } y\} \tag{2.1}$$

denote the attainable set; Ψ is the set of feasible combinations of input vectors x and output vectors y. The Farrell-Debreu efficiency score for the point (x_0, y_0) is determined by the distance from (x_0, y_0) to the efficient frontier

$$\Psi^{\partial} = \left\{(x, y) \in \Psi \mid \left(\gamma x, \gamma^{-1} y\right) \notin \Psi \text{ for any } \gamma < 1\right\} \tag{2.2}$$

of the attainable set. The boundary Ψ^{∂} of Ψ constitutes the ***technology***. Microeconomic theory of the firm suggests that in perfectly competitive markets, firms operating in the interior of Ψ will be driven from the market, but makes no prediction of how long this might take.

We make standard assumptions on Ψ by adopting those of Shephard (1970) and Färe (1988).

Assumption 2.1. *Ψ is closed and strictly convex.*

Assumption 2.2. *$(x, y) \notin \Psi$ if $x = 0$, $y \geq 0, y \neq 0$, i.e., all production requires use of some inputs.*

Assumption 2.3. *for $\widetilde{x} \geq x$, $\widetilde{y} \leq y$, if $(x, y) \in \Psi$ then $(\widetilde{x}, y) \in \Psi$ and $(x, \widetilde{y}) \in \Psi$, i.e., both inputs and outputs are strongly disposable.*

Here and throughout, inequalities involving vectors are defined on an element-by-element basis; e.g., for $\widetilde{x},\ x \in \mathbb{R}^p_+$, $\widetilde{x} \geq x$ means that some number $\ell \in \{0,\ 1,\ \ldots,\ p\}$ of the corresponding elements of $\widetilde{x}$ and x are equal, while $(p - \ell)$ of the elements of $\widetilde{x}$ are greater than the corresponding elements of x. Assumption 2.3 is equivalent to an assumption of monotonicity of the technology.

The efficiency of a given point (x_0, y_0) is determined by the frontier Ψ^∂. The input-oriented Farrell-Debreu efficiency measure $\theta(x_0, y_0)$ is "radial" in the sense efficiency of a point (x_0, y_0) is defined in terms of how much all input quantities can be contracted, by the same proportion, without altering output levels to arrive at the boundary of Ψ; formally,

$$\theta(x_0, y_0) = \inf\{\theta \geq 0 \mid (\theta x_0, y_0) \in \Psi\}. \tag{2.3}$$

Similarly, the output-oriented Farrell-Debreu efficiency measure for the point (x_0, y_0) is defined by

$$\lambda(x_0, y_0) = \sup\{\lambda \geq 0 \mid (x_0, \lambda y_0) \in \Psi\}. \tag{2.4}$$

By construction, $\theta(x_0, y_0) \leq 1$ is the proportionate reduction of inputs this unit should perform to achieve (input) efficiency. If $\theta(x_0, y_0) = 1$, the unit is on the efficient frontier Ψ^∂ of Ψ. Similarly, $\lambda(x_0, y_0)$ gives the maximum, proportionate, feasible expansion of y_0, holding input quantities x_0 fixed, with $\lambda(x_0, y_0) = 1$ if $(x_0, y_0) \in \Psi^\partial$.

Since Ψ (and hence Ψ^∂) is unknown, it must be estimated from a sample $\mathcal{X}_n = \{(X_i, Y_i)\}_{i=1}^n$ of data on firms' input and output quantities. The next assumptions define a DGP; the framework here is similar to that in Simar (1996), Kneip et al. (1998), Simar and Wilson (1998, 2000a), and Kneip et al. (2008).

Assumption 2.4. *The n observations in $\mathcal{X}_n$ are identically, independently distributed (iid) random variables on the convex attainable set Ψ.*

Assumption 2.5. *(a) The (X, Y) possess a joint density f with support $\mathcal{D} \subseteq \Psi$; (b) f is continuous on $\mathcal{D}$; and (c) $f(\theta(x, y)x, y) > 0$ for all (x, y) in the interior of $\mathcal{D}$.*

Assumption 2.5(c) imposes a discontinuity in f at points in Ψ^∂ where $\theta(x,y)=1$, ensuring a significant, non-negligible probability of observing production units close to the production frontier. For points lying outside Ψ, $f \equiv 0$. In most practical situations, $\mathcal{D}=\Psi$; however, Assumption 2.5 does not exclude the possibility that $\mathcal{D}$ is a strict subset of Ψ.

Assumption 2.6. *The function $\theta(x,y)$ is twice continuously differentiable for all $(x,y)\in\mathcal{D}$.*

Assumption 2.6 imposes some smoothness on the boundary Ψ^∂. This assumption is slightly stronger, but simpler, than a corresponding assumption needed by Kneip et al. (1998) to establish consistency of the DEA estimator of $\theta(x_0,y_0)$. We have adopted Assumption 2.6 from Kneip et al. (2008), where additional discussion is given.

The DEA estimator of Ψ is the convex hull of the free disposal hull of the sample $\mathcal{X}_n$, given by

$$\widehat{\Psi}_{\mathrm{DEA}} = \Big\{(x,y)\in\mathbb{R}_+^{p+q} \mid y \le \sum_{i=1}^n \gamma_i Y_i,\ x \ge \sum_{i=1}^n \gamma_i X_i, \sum_{i=1}^n \gamma_i = 1,\ \gamma_i \ge 0\ \forall\, i=1,\ldots,n\Big\}. \tag{2.5}$$

Substituting $\widehat{\Psi}_{\mathrm{DEA}}$ for Ψ in (2.3) yields the DEA estimator of $\theta(x_0,y_0)$ given by

$$\widehat{\theta}_{\mathrm{DEA}}(x_0,y_0) = \min_{\theta,\gamma_1,\,\ldots,\,\gamma_n} \Big\{\theta>0 \mid y_0 \le \sum_{i=1}^n \gamma_i Y_i,\ \theta x_0 \ge \sum_{i=1}^n \gamma_i X_i, \sum_{i=1}^n \gamma_i = 1,\ \gamma_i \ge 0\ \forall\, i=1,\ \ldots,\ n\Big\}. \tag{2.6}$$

Similarly, substituting $\widehat{\Psi}_{\mathrm{DEA}}$ for Ψ in (2.4) yields the DEA estimator of $\lambda(x_0,y_0)$ given by

$$\widehat{\lambda}_{\mathrm{DEA}}(x_0,y_0) = \max_{\lambda,\gamma_1,\,\ldots,\,\gamma_n} \Big\{\lambda>0 \mid \lambda y_0 \le \sum_{i=1}^n \gamma_i Y_i,\ x_0 \ge \sum_{i=1}^n \gamma_i X_i, \sum_{i=1}^n \gamma_i = 1,\ \gamma_i \ge 0\ \forall\, i=1,\ \ldots,\ n\Big\}. \tag{2.7}$$

Estimates are computed from (2.6) or (2.7) by solving a linear program using simplex, interior point, or other methods. We will say that *"we compute the DEA efficiency score of the unit (x_0,y_0) with respect to reference set given by $\mathcal{X}_n$"*. When needed to avoid ambiguities, we will write $\widehat{\theta}_{\mathrm{DEA}}(x_0,y_0 \mid \mathcal{X}_n)$ or $\widehat{\lambda}_{\mathrm{DEA}}(x_0,y_0 \mid \mathcal{X}_n)$, where the notation "$\mid \mathcal{X}_n$" is added to stress the fact that the reference set used by the DEA estimator in (2.6) is $\mathcal{X}_n$.

3 A Re-characterization of the Model

In order to develop our bootstrap algorithm and to prove its consistency, it is necessary to transform the model presented above in Section 2. To conserve space, we shall consider in the remainder of the paper only the input oriented case; all the results that we obtain extend to the output-oriented case after straightforward changes in notation. The transformation used here is adapted from Kneip et al. (2008) and is based on the coordinate system used in Jeong and Simar (2006). Transforming the model simplifies representation and computation of the quantities of interest.

Consider the point of interest, $(x_0, y_0) \in \mathbb{R}^p_+ \times \mathbb{R}^q_+$. Denote an orthonormal basis for x_0 by $\{v_j \mid j = 1, \ldots, p-1\}$.[2] Now consider a transformation r_{x_0} from $\mathbb{R}^p_+$ to $\mathbb{R}^{p-1} \times \mathbb{R}_+$:

$$r_{x_0} : x \mapsto \left(x'v_1,\ x'v_2,\ \ldots,\ x'v_{p-1},\ \frac{x'x_0}{||x_0||} \right), \tag{3.1}$$

where $||x_0|| = \sqrt{x_0'x_0}$ denotes the Euclidean norm of x_0. Then $(r^{(1)}_{x_0}(x),\ \ldots,\ r^{(p-1)}_{x_0}(x))$ is the coefficient vector of x in the space spanned by $\{v_j \mid j = 1, \ldots, p-1\}$ and $r^{(p)}_{x_0}(x)$ is the distance between x and x_0. Therefore, $r_{x_0}(x_0) = (0,\ \ldots,\ 0,\ ||x_0||)$. When $p = 1$, we have $r_{x_0}(x) = x$ for all $x \in \mathbb{R}_+$.

Next, define the transformation $h_{x_0,y_0} : \mathbb{R}^p_+ \times \mathbb{R}^q_+ \mapsto \mathbb{R}^{p-1+q} \times \mathbb{R}_+$ which maps (x, y) to (z, u), where

$$z = \left(r^{(1)}_{x_0}(x),\ \ldots,\ r^{(p-1)}_{x_0}(x),\ y^{(1)} - y^{(1)}_0,\ \ldots,\ y^{(q)} - y^{(q)}_0 \right)', \tag{3.2}$$

$$u = r^{(p)}_{x_0}(x) = \frac{x'x_0}{||x_0||}. \tag{3.3}$$

It will be useful to denote $z = (z^x, z^y)$, where z^x comprises the first $(p-1)$ components of z (corresponding to X) and z^y contains the last q components (of z) corresponding to Y. Note also that $h_{x_0,y_0}(x_0, y_0) = (0,\ 0,\ \ldots,\ 0,\ ||x_0||)$. Moreover, h_{x_0,y_0} is a one-to-one transformation, and the following inverse relation will be useful later: $(x, y) = h^{-1}_{x_0,y_0}(z, u)$, where

$$x = \sum_{j=1}^{p-1} z^{(j)} v_j + u\frac{x_0}{||x_0||}, \tag{3.4}$$

$$y = \left(z^{(p)},\ \ldots,\ z^{(p+q-1)} \right)' + y_0. \tag{3.5}$$

[2] Various methods exist for computing an orthonormal basis of a vector x_0; see, for example, Jeong and Simar (2006).

Applying the transformation in (3.2) to the points in $\mathcal{X}_n$ yields the set of transformed data in the new coordinate system (z, u):

$$\mathcal{Z}_n = \{(z_i, u_i) \mid (z_i, u_i) = h_{x_0,y_0}(X_i, Y_i),\ (X_i, Y_i) \in \mathcal{X}_n\}\,. \tag{3.6}$$

In the new coordinate system (z, u), the attainable set Ψ is represented by

$$G_{x_0,y_0} = \left\{(z, u) \in \mathbb{R}^{p-1+q} \times \mathbb{R}_+ \mid (z, u) = h_{x_0,y_0}(x, y),\ (x, y) \in \Psi\right\}, \tag{3.7}$$

where the subscript (x_0, y_0) has been attached to G to stress the fact that this particular description of Ψ is from the perspective of the point of interest, namely (x_0, y_0). Starting with some other point, we would obtain a different representation of Ψ. We can also represent the boundary (i.e., efficient frontier) of Ψ in the new coordinate system (z, u) in terms of the scalar-valued function

$$g(z \mid x_0, y_0) = \inf\{u > 0 \mid (z, u) \in G_{x_0,y_0}\}\,. \tag{3.8}$$

Hence the set G_{x_0,y_0} in (3.7) can be represented equivalently in terms of the function g as

$$G_{x_0,y_0} = \left\{(z, u) \in \mathbb{R}^{p-1+q} \times \mathbb{R}_+ \mid u \geq g(z|x_0, y_0)\right\}. \tag{3.9}$$

Furthermore, since the point of interest (x_0, y_0) is transformed by h_{x_0,y_0} into $(0, ||x_0||)$, we have

$$\theta(x_0, y_0) = ||x_0||^{-1} g(0|x_0, y_0). \tag{3.10}$$

The DEA estimator $\widehat{\Psi}_{\text{DEA}}$ of Ψ given in (2.5) can be characterized in terms of the (z, u)-space as

$$\widehat{G}_{\text{DEA},x_0,y_0} = \left\{(z, u) \in \mathbb{R}^{p-1+q} \times \mathbb{R}_+ \mid (z, u) = h_{x_0,y_0}(x, y),\ (x, y) \in \widehat{\Psi}_{\text{DEA}}\right\}. \tag{3.11}$$

Note that $\widehat{\Psi}_{\text{DEA}}$ can be rewritten as

$$\widehat{\Psi}_{\text{DEA}} = \Big\{(x, y) \in \mathbb{R}^{p+q}_+ \mid y = \sum_{i=1}^{n} \gamma_i Y_i - \sum_{\ell=1}^{q} \beta_\ell\, e_q^{(\ell)},\ x = \sum_{i=1}^{n} \gamma_i X_i + \sum_{j=1}^{p} \alpha_j\, e_p^{(j)},\ \sum_{i=1}^{n} \gamma_i = 1,$$
$$\gamma_i \geq 0\ \forall\ i = 1,\ \ldots, n,\ \beta_\ell \geq 0\ \forall\ \ell = 1,\ \ldots,\ q,\ \alpha_j \geq 0\ \forall\ j = 1,\ \ldots,\ p\Big\}, \tag{3.12}$$

with $e_m^{(k)}$ denoting the kth column of the identity matrix of order m. Here the inequalities on x and y appearing in (2.5) have been transformed into equalities by introducing the

positive parameters $\alpha_j,\ j = 1,\ \ldots,\ p$ and $\beta_\ell,\ \ell = 1,\ \ldots,\ q$. These constraints ensure free disposability of the set $\widehat{\Psi}_{\mathrm{DEA}}$. The αs and βs in (3.12) are equivalent to slacks in the constraints of the linear program appearing in (2.5).

Using (3.12), $\widehat{G}_{\mathrm{DEA},x_0,y_0}$ given in (3.11) can be written equivalently as

$$\begin{aligned}\widehat{G}_{\mathrm{DEA},x_0,y_0} = \Big\{\ & (z^x, z^y, u) \in \mathbb{R}^{p-1} \times \mathbb{R}^q \times \mathbb{R}_+ \mid z^y = \sum_{i=1}^{n} \gamma_i Z_i^y - \sum_{\ell=1}^{q} \beta_\ell\, e_q^{(\ell)}, \\ & z^x = \sum_{i=1}^{n} \gamma_i Z_i^x + \sum_{j=1}^{p} \alpha_j Z^x_{e_p^{(j)}},\ u = \sum_{i=1}^{n} \gamma_i U_i + \sum_{j=1}^{p} \alpha_j U_{e_p^{(j)}}, \\ & \sum_{i=1}^{n} \gamma_i = 1,\ \gamma_i \geq 0\ \forall\ i = 1,\ \ldots,\ n, \\ & \beta_\ell \geq 0\ \forall\ \ell = 1,\ \ldots,\ q,\ \alpha_j \geq 0\ \forall\ j = 1,\ \ldots,\ p\ \Big\}, \end{aligned} \tag{3.13}$$

where $(Z_i, U_i) = (Z_i^x, Z_i^y, U_i),\ i = 1,\ \ldots,\ n$ are the coordinates of the data points in $\mathcal{X}_n$ and $\left(Z^x_{e_p^{(j)}}, U_{e_p^{(j)}}\right)$ are the corresponding transformations of the $e_p^{(j)}$ to insure free disposability of inputs. Since the z^y-axes are parallel to the y-axis, we retain the same notation as before for free disposability of the outputs. The boundary of $\widehat{\Psi}_{\mathrm{DEA}}$, in the coordinate system (z, u), can now be described in terms of the scalar-valued function

$$\widehat{g}_{\mathrm{DEA}}(z|x_0, y_0) = \inf\{u > 0 \mid (z, u) \in \widehat{G}_{\mathrm{DEA},x_0,y_0}\}. \tag{3.14}$$

Hence the set $\widehat{G}_{\mathrm{DEA},x_0,y_0}$ can be represented equivalently as

$$\widehat{G}_{\mathrm{DEA},x_0,y_0} = \{(z, u) \in \mathbb{R}^{p-1+q} \times \mathbb{R}_+ \mid u \geq \widehat{g}_{\mathrm{DEA}}(z|x_0, y_0)\}. \tag{3.15}$$

As a practical matter, for any point z, $\widehat{g}_{\mathrm{DEA}}(z|x_0, y_0)$ can be computed as the solution of the linear program

$$\begin{aligned}\widehat{g}_{\mathrm{DEA}}(z|x_0, y_0) = \min \Big\{\ & u > 0 \mid z^y = \sum_{i=1}^{n} \gamma_i Z_i^y - \sum_{\ell=1}^{q} \beta_\ell\, e_q^{(\ell)}, \\ & z^x = \sum_{i=1}^{n} \gamma_i Z_i^x + \sum_{j=1}^{p} \alpha_j Z^x_{e_p^{(j)}},\ u = \sum_{i=1}^{n} \gamma_i U_i + \sum_{j=1}^{p} \alpha_j U_{e_p^{(j)}}, \\ & \sum_{i=1}^{n} \gamma_i = 1,\ \gamma_i \geq 0\ \forall\ i = 1,\ \ldots,\ n, \\ & \beta_\ell \geq 0\ \forall\ \ell = 1,\ \ldots,\ q,\ \alpha_j \geq 0\ \forall\ j = 1,\ \ldots,\ p\ \Big\}. \end{aligned} \tag{3.16}$$

It is easy to verify that the estimator of the Farrell-Debreu efficiency score of (x_0, y_0) can be recovered as

$$\widehat{\theta}_{\text{DEA}}(x_0, y_0) = ||x_0||^{-1}\widehat{g}_{\text{DEA}}(0|x_0, y_0). \tag{3.17}$$

Figure 1 illustrates $g(z|x_0, y_0)$ and $\widehat{g}_{\text{DEA}}(z|x_0, y_0)$ with $p = 2$ and $q = 1$, for the particular value $z^y = 0$. The Figure shows the transformed u-axis passing through the point (x_0, y_0); the single z^x-axis lies in the plane of the original x_1- and x_2-axes, while the z^y axis is orthogonal to this plane. The origin in (z, u)-space corresponds to the origin in the (x, y)-space. In the (z, u)-space, both the true frontier (represented by the smooth curve) and its DEA estimate (represented by the piecewise linear curve) can be described in terms of distance from the plane containing the z^x- and z^y-axes, in the direction parallel to the u-axis. The ability to describe the efficient frontier and its estimate in terms of a scalar-valued function will be very useful in developing a simplified bootstrap algorithm in the next section.

4 The Bootstrap

4.1 Basics of the idea

Any bootstrap for inference about the efficiency $\theta(x_0, y_0)$ of the point (x_0, y_0) requires an initial estimate $\widehat{\theta}_{\text{DEA}}(x_0, y_0 \mid \mathcal{X}_n)$ and a corresponding set of bootstrap values $\widehat{\theta}^*_b(x_0, y_0)$, $b = 1, \ldots, B$. Each of the B bootstrap values must be computed by applying the original estimator to a bootstrap sample $\mathcal{X}^*_n = \{(X^*_i, Y^*_i\}^n_{i=1}$; i.e., $\widehat{\theta}^*(x_0, y_0) = \widehat{\theta}_{\text{DEA}}(x_0, y_0 \mid \mathcal{X}^*_n)$. Then the empirical distribution of these B values can be used to approximate the sampling distribution of the desired quantities, e.g., the distribution of $\widehat{\theta}_{\text{DEA}}(x_0, y_0) - \theta(x_0, y_0)$ or $\widehat{\theta}_{\text{DEA}}(x_0, y_0)/\theta(x_0, y_0)$. The approximated sampling distributions can the be used, as is typical, for purposes of inference (i.e., estimation of confidence intervals or testing hypotheses), bias correction, etc.

The key to any bootstrap procedure is to generate the bootstrap samples $\mathcal{X}^*_{n,b}$ in a way that allows the procedure to provide a consistent approximation of the underlying sampling distribution. The naive bootstrap, where bootstrap samples are constructed by drawing (with replacement) from the original sample, is known to be inconsistent in the present context; see Simar and Wilson (1999b, 1999a) for discussion. As noted in Section 1, Kneip et al. (2008) developed a consistent bootstrap procedure. But, their approach involves substantial

computational costs due to the auxiliary linear programs that must be solved each time a bootstrap sample $\mathcal{X}_{n,b}^*$ is constructed. This problem is avoided in the framework that follows.

The inconsistency of the naive bootstrap in the context of DEA estimators arises from the fact that the support of (X, Y) is bounded and unknown; an additional problem is that the DEA estimator of the boundary of support does not possess the properties of the original boundary. To give an example, under the assumptions listed in Section 2, the function g defined in (3.8) must be differentiable and convex, whereas its DEA estimate given in (3.14) is not; instead, it is a piecewise linear function. The double-smoothing procedure developed by Kneip et al. (2008) addresses these problems by (i) drawing bootstrap data from a smooth estimate of the density of (X, Y) with unknown boundary estimated by (ii) a smooth, convex version of the DEA boundary.

Since most of the problems arise near the boundary, the approach taken here is to generate points "near" the boundary from a local uniform distribution along (below) a smoothed version of the DEA frontier. For points "far" from the boundary, naive re-sampling will be used. However, consistency of our bootstrap procedure will require smoothing the initial frontier estimate, and projecting X_i onto the smoothed estimate. The remainder of this section provides the necessary details.

4.2 Smoothing the initial frontier estimate

In order to smooth the initial DEA frontier estimate, we will work in the (z, u)-space defined in Section 3. Let $h \in (0, 1]$ and define

$$g^{(h)}(z \mid x_0, y_0) = g(0 \mid x_0, y_0) + h^2 \left[g(h^{-1}z \mid x_0, y_0) - g(0 \mid x_0, y_0) \right]. \tag{4.1}$$

This is a smoothed version of $g(z \mid x_0, y_0)$ defined in (3.8), with the degree of smoothing controlled by h. The corresponding DEA estimator of $g^{(h)}(z \mid x_0, y_0)$ is given by

$$\widehat{g}_{\mathrm{DEA}}^{(h)}(z \mid x_0, y_0) = \widehat{g}_{\mathrm{DEA}}(0 \mid x_0, y_0) + h^2 \left[\widehat{g}_{\mathrm{DEA}}(h^{-1}z \mid x_0, y_0) - \widehat{g}_{\mathrm{DEA}}(0 \mid x_0, y_0) \right]. \tag{4.2}$$

Note that if $h = 1$ there is no smoothing, and $g^{(h)}(z \mid x_0, y_0) = g(z \mid x_0, y_0)$ and $\widehat{g}_{\mathrm{DEA}}^{(h)}(z \mid x_0, y_0) = \widehat{g}_{\mathrm{DEA}}(z \mid x_0, y_0)$. The following properties are easily verified:

(i) For $h < 1$, $g^{(h)}$ as well as $\widehat{g}_{\mathrm{DEA}}^{(h)}$ are convex functions (this follows from Lemma 1 in Kneip et al., 2008);

(ii) We have the following identities:

$$g^{(h)}(0 \mid x_0, y_0) = g(0 \mid x_0, y_0) = ||x_0||\, \theta(x_0, y_0), \tag{4.3}$$

$$\widehat{g}^{(h)}_{\mathrm{DEA}}(0 \mid x_0, y_0) = \widehat{g}_{\mathrm{DEA}}(0 \mid x_0, y_0) = ||x_0||\, \widehat{\theta}_{\mathrm{DEA}}(x_0, y_0); \tag{4.4}$$

(iii) The second derivatives of $g^{(h)}$ and g at $z = 0$ are the same: $g^{''}(0|x_0, y_0) = g^{(h),''}(0|x_0, y_0)$.

Following the reasoning in Kneip et al. (2008), it can be proven that the difference between $\widehat{g}^{(h)}_{\mathrm{DEA}}(z \mid x_0, y_0)$ and $g^{(h)}(z \mid x_0, y_0)$ is of smaller order than $n^{-2/(p+q+1)}$, so that a bootstrap based on $\widehat{g}^{(h)}_{\mathrm{DEA}}$ will provide the same results as a bootstrap relying on $g^{(h)}$; hence from the above properties, asymptotic distributions based on $g^{(h)}$ will coincide with those based on g.

In the (z, u) coordinate system, (X_i, Y_i) has coordinates (Z_i, U_i) for all i. Using (4.2), we can estimate the smoothed DEA frontier in the (z, u)-space by

$$\widehat{g}^{(h)}_{\mathrm{DEA}}(Z_i \mid x_0, y_0) = \widehat{g}_{\mathrm{DEA}}(0 \mid x_0, y_0) + h^2 \left[\widehat{g}_{\mathrm{DEA}}\left(h^{-1}Z_i \mid x_0, y_0\right) - \widehat{g}_{\mathrm{DEA}}(0 \mid x_0, y_0)\right], \tag{4.5}$$

where $\widehat{g}_{\mathrm{DEA}}\left(h^{-1}Z_i \mid x_0, y_0\right)$ is obtained by solving the linear program

$$\begin{aligned}
\widehat{g}_{\mathrm{DEA}}\left(h^{-1}Z_i \mid x_0, y_0\right) \;=\; & \min \Big\{ u > 0 \mid h^{-1}Z_i^y = \sum_{k=1}^{n} \gamma_k Z_k^y - \sum_{\ell=1}^{q} \beta_\ell e_q^{(\ell)}, \\
& h^{-1}Z_i^x = \sum_{k=1}^{n} \gamma_k Z_k^x + \sum_{j=1}^{p} \alpha_j Z^x_{e_p^{(j)}},\; u = \sum_{k=1}^{n} \gamma_k U_k + \sum_{j=1}^{p} \alpha_j U_{e_p^{(j)}}, \\
& \sum_{k=1}^{n} \gamma_k = 1,\; \gamma_k \geq 0 \;\forall\; k = 1,\; \ldots,\; n, \\
& \beta_\ell \geq 0 \;\forall\; \ell = 1,\; \ldots,\; q,\; \alpha_j \geq 0 \;\forall\; j = 1,\; \ldots,\; p \Big\}
\end{aligned} \tag{4.6}$$

obtained by substituting $h^{-1}Z_i$, $h^{-1}Z_i^y$, and $h^{-1}Z_i^x$ for z, z^y, and z^x in (3.16).

Given the coordinate $(Z_i, \widehat{g}^{(h)}_{\mathrm{DEA}}(Z_i|x_0, y_0))$ of the projected frontier point in the (z, u)-space, we can recover X_i^∂ in the original coordinate system by computing

$$X_i^\partial = \sum_{j=1}^{p-1} Z_i{}^{(j)} v_j + \widehat{g}^{(h)}_{\mathrm{DEA}}(Z_i \mid x_0, y_0)\, \frac{x_0}{||x_0||}, \tag{4.7}$$

which can be written equivalently as

$$X_i^\partial = X_i - \frac{(x_0' X_i) x_0}{||x_0||^2} + \widehat{g}^{(h)}_{\mathrm{DEA}}(Z_i \mid x_0, y_0)\, \frac{x_0}{||x_0||}. \tag{4.8}$$

4.3 Extrapolation

The linear program in (4.6) may be infeasible in some cases. This will happen if the point to be projected onto the DEA boundary is such that in the u-direction, the line parallel to u passing through $(h^{-1}Z_i^x, h^{-1}Z_i^y)$ does not intersect the convex hull of the points (Z_k, U_k), $k = 1, \ldots, n$, which appear on the right-hand side of the constraints of the linear program in (4.6). In cases where this happens we will use an extrapolation method adapted from Kneip et al. (2008). For $h = 1$, the linear program has a feasible solution, so we first identify, for each point in the original sample indexed by $i = 1, \ldots, n$, the minimal value of h that avoids an infeasible solution to the linear program that appears in (4.6); denote these values h_i, $i = 1, \ldots, n$. Then, for each i, if h_i is greater than the chosen value of h for the smoothing in (4.5), we will instead compute $\widehat{g}_{\mathrm{DEA}}\left(h_i^{-1}Z_i \mid x_0, y_0\right)$ and then extrapolate this value in an appropriate way to define a sensible value for $\widehat{g}_{\mathrm{DEA}}\left(h^{-1}Z_i \mid x_0, y_0\right)$.

Necessarily, there will be a few points in the sample $\mathcal{X}_n$ with maximal values for output quantities. Such points will lie on the part of the DEA frontier that is parallel to one (or more) of the x-axes. Due to the assumption of free disposability, these portions of the DEA frontier estimate stretch toward infinity in the direction of the x-axes with which it is parallel. For these points, $h_i = 1$; there is nothing to smooth and we will only extrapolate. Asymptotically, these points will have no influence on the properties of the bootstrap for any point (x_0, y_0) in the interior of $\widehat{\Psi}_{\mathrm{DEA}}$; the strategy taken here avoids numerical difficulties.

4.3.1 Computation of h_i

To compute the maximal expansion (i.e., the minimal value of h) that avoids having $h^{-1}Z_i^y$ lie above the convex hull in the output direction, we return to the original coordinate system (x, y). The maximal expansion of an observation (X_i, Y_i) to a point (X_i, Y_i^λ), $Y_i^\lambda = \lambda_i Y_i$, is determined by the maximum value of λ such that $\widehat{\theta}_{\mathrm{DEA}}(X_i, Y_i^\lambda \mid \mathcal{X}_n)$ exists, i.e., has a feasible solution. This maximal expansion can be determined by a linear program similar to the one that defines the DEA estimator of the Farrell output-oriented efficiency measure. Let $\mathcal{K} = \left\{k \mid \widehat{\lambda}_{\mathrm{DEA}}\left(X_k, Y_k \mid \mathcal{X}_n\right) = 1\right\}$. Then the maximal expansion for the ith observation

(X_i, Y_i) in $\mathcal{X}_n$ is given as the solution to the linear program

$$\widehat{\lambda}_{DEA}\left(X_i, Y_i \mid \{(X_i, Y_k) \mid k \in \mathcal{K}\}\right) = \max\Big\{\ \lambda > 0 \mid \lambda Y_i \leq \sum_{k\in\mathcal{K}} \gamma_k Y_k, \\ \sum_{k\in\mathcal{K}} \gamma_k = 1,\ \gamma_k \geq 0\ \forall\ k \in \mathcal{K}\Big\}. \qquad (4.9)$$

This linear program can be seen to to be the same as the linear program appearing in the definition of the DEA estimator of the output-oriented Farrell efficiency measure after adding $X_i \geq \sum_{k\in\mathcal{K}} \gamma_k X_i$ to the constraints; however, such a constraint is redundant here, and consequently is omitted in the formulation of (4.9). Finally, the maximal expansion for the ith observation is given by

$$h_i = \left[\widehat{\lambda}_{DEA}\left(X_i, Y_i \mid \{(X_i, Y_k) \mid k \in \mathcal{K}\}\right)\right]^{-1}. \qquad (4.10)$$

4.3.2 Extrapolation

Suppose that h is the current chosen value for smoothing the DEA frontier. There are two possibilities:

(i) If $h_i \leq h$, no extrapolation is needed since $\widehat{g}_{\text{DEA}}\left(h^{-1}Z_i \mid x_0, y_0\right)$ is feasible. In this case, $\widehat{g}^{(h)}_{\text{DEA}}(h^{-1}Z_i \mid x_0, y_0)$ can be computed as written in (4.5).

(ii) If $h_i > h$, extrapolation is required. In this case, compute $\widehat{g}^{(h)}_{\text{DEA}}(h^{-1}Z_i \mid x_0, y_0)$ by replacing $\widehat{g}_{\text{DEA}}\left(h^{-1}Z_i \mid x_0, y_0\right)$ on the right-hand side of (4.5) with the extrapolated value $\widetilde{g}_{\text{DEA}}\left(h^{-1}Z_i \mid x_0, y_0\right)$ where the details for computing $\widetilde{g}_{\text{DEA}}(\cdot)$ are provided in the Appendix (see equation A.2).

It is important to note that computation of the smoothed frontier, the maximal expansions h_i, and any necessary extrapolation must be done only once, before computation of the bootstrap replications. Consequently, the computational burden of the smoothing is negligible. This will become more clear in the remainder of this section, where we list the steps required by our bootstrap algorithm.

4.4 Efficient computation

For given values of the two smoothing parameters τ and h, the computational burden of our bootstrap will be similar to that of the naive bootstrap. As noted above, the only

additional calculations required by our procedure are the computation of the h_i in (4.10), and computation of $\widehat{g}^{(h)}_{\rm DEA}(Z_i|x_0,y_0)$ in (4.5), and these can be performed before initiating the bootstrap loop. Given a sample $\mathcal{X}_n$ and a point of interest (x_0,y_0), the complete bootstrap algorithm for inference about the input efficiency $\theta(x_0,y_0)$ can now be described for given values of τ and h, and the number of bootstrap replications B.

Algorithm #1:

[1] Form the set $\mathcal{Z}_n$; i.e., for each $i = 1,\ ,\ldots,\ n$, transform each observation in $\mathcal{X}_n$ from coordinates (X_i,Y_i) to coordinates (Z_i,U_i). In the new coordinate system, (x_0,y_0) has coordinates $(0,U_0)$ where $U_0 = ||x_0||$.

[2] Compute $\widehat{\delta}_i = \frac{\widehat{g}_{DEA}(Z_i|(x_0,y_0),\mathcal{Z}_n)}{U_i} \leq 1\ \forall\ i = 1,\ ,\ldots,\ n$, and compute $\widehat{\theta}_0 = \widehat{\theta}_{DEA}(x_0,y_0 \mid \mathcal{X}_n) = \frac{\widehat{g}_{DEA}(0|(x_0,y_0),\mathcal{Z}_n)}{U_0}$.

[3] Using extrapolation as necessary, compute (smoothed) frontier points (Z_i,U_i^{∂}) where $U_i^{\partial} = \widehat{g}^{(h)}_{DEA}(Z_i \mid (x_0,y_0),\mathcal{Z}_n)$.

[4] Loop over steps [4.1]–[4.3] B times:

[4.1] Draw independently, uniformly, and with replacement from the set of integers $\{i\}_{i=1}^n$ n times to create a set of labels $\mathcal{J} = \{j_i\}_{i=1}^n$.

[4.2] For each $i = 1,\ \ldots,\ n$, set $Z_i^* = Z_{j_i}$ and

$$U_i^* = \begin{cases} U_{j_i}^{\partial}/\widehat{\delta}_{j_i} & \text{if } \widehat{\delta}_{j_i} < (1-\tau); \\ U_{j_i}^{\partial}/\xi_{j_i}^* & \text{otherwise,} \end{cases}$$

where $\xi_{j_i}^*$ is a random, independent draw from a uniform distribution on the interval $[(1-\tau),\ 1]$, to construct a bootstrap sample $\mathcal{Z}_n^* = \{(Z_i^*,U_i^*)\}_{i=1}^n$.

[4.3] Compute $\widehat{\theta}_0^* = \frac{\widehat{g}_{DEA}(0|(x_0,y_0),\mathcal{Z}_n^*)}{U_0}$.

[5] Use the set $\mathcal{B} = \{\widehat{\theta}_{0,\ell}^*\}_{\ell=1}^B$ of bootstrap values to estimate a $(1-\alpha)\times 100$-percent confidence interval.

Note that in step [4.3], we use the un-smoothed g-function. in addition, $\widehat{\theta}_0^*$ is equivalent to $\widehat{\theta}_{DEA}(x_0,y_0 \mid \mathcal{X}_n^*)$, where the set $\mathcal{X}_n^*$ is recovered from $\mathcal{Z}_n^*$ using the transformation in

(3.4)–(3.5). However, it is not necessary to recover $\mathcal{X}_n^*$ since $\widehat{\theta}_0^*$ can be computed directly using $\mathcal{Z}_n^*$.

The confidence interval estimate in step [5] can be computed by any of various methods. To remain in line with the theoretical results in Kneip et al. (2008), the bootstrap values in the set $\mathcal{B}$ can be used, together with the initial estimate $\widehat{\theta}_0$ obtained in step [1], to approximate the conditional distribution of $\left(\frac{\widehat{\theta}_0}{\theta_0} - 1\right)$ given the sample $\mathcal{X}_n$. This can then be used to approximate the unknown distribution of $\left(\frac{\widehat{\theta}_0}{\theta_0} - 1\right)$. For $\alpha \in (0, 1)$, let φ_α denote the α-quantile such that

$$\Pr\left[\left(\frac{\widehat{\theta}_0}{\theta_0} - 1\right) \leq \varphi_\alpha \mid \mathcal{X}_n\right] = \alpha. \tag{4.11}$$

Then an estimate $\widehat{\varphi}_\alpha$ of φ_α is obtained by taking the α-quantile of the empirical distribution of $\left(\frac{\widehat{\theta}_{0,\ell}^*}{\widehat{\theta}_0} - 1\right)$, constructed from $\widehat{\theta}_0$ and the B elements of the set $\mathcal{B}$. Finally, setting

$$\widehat{c}_{\mathrm{lo},\alpha} = \widehat{\theta}_0 \times \left(1 + \widehat{\varphi}_{1-\alpha/2}\right)^{-1} \tag{4.12}$$

and

$$\widehat{c}_{\mathrm{hi},\alpha} = \widehat{\theta}_0 \times \left(1 + \widehat{\varphi}_{\alpha/2}\right)^{-1} \tag{4.13}$$

gives an estimate $(\widehat{c}_{\mathrm{lo},\alpha},\ \widehat{c}_{\mathrm{hi},\alpha})$ of a $(1-\alpha) \times 100$-percent confidence interval for θ_0.

The fixed point of interest (x_0, y_0) might be one of the observed input-output pairs in $\mathcal{X}_n$, or it might represent a hypothetical firm (e.g., the mean or median firm). Let $n' = n$ if (x_0, y_0) is in $\mathcal{X}_n$, or $n' = n + 1$ otherwise. Depending on the need for extrapolation in step [3], the number η of linear programs that must be solved to implement Algorithm #1 is such that $(n' + 2n + B) < \eta \leq (n' + 3n + B)$. This contrasts with the full-sample bootstrap in Kneip et al. (2008) where $(n + n' + B(n + 1))$ linear programs must be solved. Hence the method proposed here requires solving $(B - 2)n$ fewer linear programs than the method proposed in Kneip et al. (2008), leading to a substantial reduction in computational burden.

Two issues remain—to prove consistency of the bootstrap in Algorithm #1, and to show how sensible, data-driven values of the smoothing parameters τ and h can be obtained. We address these issues next.

4.5 Consistency

The results in this section build on the theoretical results obtained by Kneip et al. (2008). In particular, Assumptions 2.1–2.6 listed in Section 2— and which also appear in Kneip et al.—require a twice continuously differentiable function g as well as the existence of a joint density $\bar{f}$ of $(\theta(X_i, Y_i), Z_i)$. Then it can be shown that there exists a continuous distribution function F_{x_0,y_0} such that for any $\gamma > 0$,

$$F_{x_0,y_0}(\delta) = \lim_{n\to\infty} \Pr\left[n^{\frac{2}{p+q+1}} \left(\frac{\widehat{\theta}_{DEA}(x_0, y_0)}{\theta(x_0, y_0)} - 1 \right) \leq \gamma \right]. \tag{4.14}$$

The structure of F_{x_0,y_0} depends only on the (matrix of) second derivatives $g''(0|x_0, y_0)$ of $g(z|x_0, y_0)$ at $z = 0$ and on the value $\bar{f}(1, 0)$ of the joint density at the point $(1, 0)$. For more details on the properties of F_{x_0,y_0}, see Kneip et al. (2008).

The following theorem establishes consistency of our bootstrap procedure.

Theorem 4.1. *Let $h \to 0$ and $n^{-\frac{1}{p+q+1}}/h \to 0$ as well as $\tau \to 0$ and $n^{-\frac{2}{p+q+1}}/\tau \to 0$ as $n \to \infty$. Under Assumptions 2.1–2.6,*

$$\sup_{\gamma>0} \left| F_{x_0,y_0}(\gamma) - \Pr\left(n^{\frac{2}{p+q+1}} \left(\frac{\widehat{\theta}_{DEA}(x_0, y_0|\mathcal{X}_n^*)}{\widehat{\theta}_{DEA}(x_0, y_0))} - 1 \right) \leq \gamma \mid \mathcal{X}_n \right) \right| \xrightarrow{p} 0 \qquad \text{as } n \to \infty. \tag{4.15}$$

Proof: For $j_i \in \mathcal{J}$ let $\widetilde{\delta}_i := \widehat{\delta}_{j_i}$, and define $\hat{\delta}_i^* = \widetilde{\delta}_i$ if $\widehat{\delta}_{j_i} < (1 - \tau)$, and $\hat{\delta}_i^* := \xi_{j_i}^*$ otherwise. Here, $\mathcal{J}$, $\widehat{\delta}_{j_i}$ and $\xi_{j_i}^*$ are defined in steps [2], [4.1] and [4.2] of Algorithm #1. Since $U_i^* = \widehat{g}_{DEA}^{(h)}(Z_i^* \mid (x_0, y_0), \mathcal{Z}_n)/\hat{\delta}_i^*$, the bootstrap sample $\mathcal{Z}_n^* = \{(U_i^*,\ Z_i^*)\}$ (or $\mathcal{X}_n^* = \{(X_i^*, Y_i^*)\}$) can be equivalently represented by the corresponding sample $\{(\hat{\delta}_i^*,\ Z_i^*)\}$. Furthermore, $\{(\widetilde{\delta}_i, Z_i^*)\}$ is obviously an *iid naive re-sample* of $(\hat{\delta}, Z)$-coordinates of the original sample $\mathcal{Z}_n$.

The asymptotic results of Kneip et al. (2008) are derived in a (θ, Z)-coordinate system. It is therefore important to analyze the relation between δ and efficiencies θ. Let $\theta_i := \theta(X_i, Y_i)$, $\delta_i := g(Z_i|x_0, y_0)/U_i$ denote true efficiencies and δ-coordinates of the original sample points $\mathcal{X}_n = \{(X_i, Y_i)\}$. With $Z_i = (Z_i^x, Z_i^y)$ we then obtain

$$U_i = \frac{g((\theta_i Z_i^x, Z_i^y)|x_0, y_0)}{\theta_i} \tag{4.16}$$

as well as

$$U_i = \frac{\hat{g}_{\mathrm{DEA}}((\widehat{\theta}_{DEA}(X_i, Y_i) Z_i^x, Z_i^y)|x_0, y_0, \mathcal{Z}_n)}{\widehat{\theta}_{DEA}(X_i, Y_i)}, \tag{4.17}$$

which imply

$$\delta_i = \theta_i \frac{g((Z_i^x, Z_i^y)|x_0, y_0)}{g((\theta_i Z_i^x, Z_i^y)|x_0, y_0)} \tag{4.18}$$

as well as

$$\widehat{\delta}_i = \widehat{\theta}_{DEA}(X_i, Y_i) \frac{\hat{g}_{\mathrm{DEA}}((Z_i^x, Z_i^y)|x_0, y_0, \mathcal{Z}_n)}{\hat{g}_{\mathrm{DEA}}((\widehat{\theta}_{DEA}(X_i, Y_i) Z_i^x, Z_i^y)|x_0, y_0, \mathcal{Z}_n)}. \tag{4.19}$$

It follows from (4.18)–(4.19) that there is a smooth, bijective mapping $(\theta_i, Z_i) \to (\delta_i, Z_i)$ and the joint density $\bar{f}$ of (θ_i, Z_i) induces a joint density $\bar{f}_\delta$ of (δ_i, Z_i). Moreover,

$$\frac{g((z^x, z^y)|x_0, y_0)}{g((\theta z^x, z^y)|x_0, y_0)} = 1 + O((\theta - 1)z^x) \quad \text{as } (\theta, z) \to (1, 0), \tag{4.20}$$

which implies that $\bar{f}(1, 0) = \bar{f}_\delta(1, 0)$.

Now for an arbitrary $b > 0$, define the set $C(b)$ by

$$C(b) = \{(\delta, Z) \mid 1 - \delta \leq b^2 \text{ and } |Z^{(j)}| \leq b \ , \forall\ j = 1, \ \ldots, \ p + q - 1\}. \tag{4.21}$$

For $\gamma > 0$ let $A[g, \gamma, n; b]$ denote the following event: for some $k \leq n$ and $i_1, \ \ldots, \ i_k \in \{1, \ \ldots, \ n\}$, there exist some $(X_{i_1}, Y_{i_1}), \ \ldots, \ (X_{i_k}, Y_{i_k})$ with $(\delta_{i_1}, Z_{i_1}), \ \ldots, \ (\delta_{i_k}, Z_{i_k}) \in \widetilde{C}(b \cdot n^{-\frac{1}{p+q+1}})$, as well as some $\alpha_1 \geq 0, \ \ldots, \ \alpha_k \geq 0$ with $\sum_{j=1}^k \alpha_j = 1$ such that $\sum_{j=1}^k \alpha_j \sum_{l=1}^{p-1} Z_{i_j}^{(l)} v_l = 0$, $\sum_{j=1}^k \alpha_j Z_{i_j}^{(r)} = 0$, $r = p, \ldots, p+q-1$, and $\sum_{j=1}^k \alpha_j \frac{g(Z_{i_j}|x_0, y_0)}{\delta_{i_j} g(0|x_0, y_0))} - 1 \leq \gamma n^{-\frac{2}{p+q+1}}$.

In addition, let $A[g^{(h)}, \gamma, n; b]$ be the corresponding event if g is replaced by $g^{(h)}$. Similarly, replace (δ_i, Z_i) by the bootstrap sample $(\hat{\delta}_i^*, Z_i^*)$ and g by $g^{(h)}$ or $\widehat{g}_{\mathrm{DEA}}^{(h)}$ to define the events $\tilde{A}[g^{(h)}, \gamma, n; b]^*$ and $\tilde{A}[\widehat{g}_{\mathrm{DEA}}^{(h)}, \gamma, n; b]^*$, respectively.

From the arguments in the proof of Theorem 4 of Kneip et al. (2008), it follows that asymptotically the difference between g and $g^{(h)}$ is negligible, and that $|\Pr(A[g, \gamma, n; b]) - \Pr(A[g^{(h)}, \delta, n; b])| \to 0$ as $n \to \infty$. Hence by Theorem 1 of Kneip et al. (2008), for every $\epsilon > 0$ there exists a $b_\epsilon > 0$ such that for all $b \geq b_\epsilon$, all $\gamma > 0$, and all n sufficiently large,

$$\left| \Pr\left[n^{\frac{2}{p+q+1}} \left(\frac{\widehat{\theta}_{DEA}(x_0, y_0)}{\theta(x_0, y_0)} - 1 \right) \leq \gamma \right] - \Pr(A[g^{(h)}, \gamma, n; b]) \right| \leq \epsilon. \tag{4.22}$$

Note that the definition of the events $A[\cdot]$ used here is slightly different from the one used in Kneip et al. (2008) which is based on (θ, Z)-coordinates. However, due to (4.20) the difference is asymptotically negligible. Reasoning similar to that in the proof of Theorem 4

of Kneip et al. (2008), and by Theorems 1 and 2 of Kneip et al., it can be shown that $\Pr\Big\{\Big|\Big[\Pr\Big(n^{\frac{2}{p+q+1}}\Big(\frac{\widehat{\theta}_{DEA}(x_0,y_0|\mathcal{X}_n^*)}{\widehat{\theta}_{DEA}(x_0,y_0)}-1\Big)\le\gamma\mid\mathcal{X}_n\Big)-\Pr\Big(\tilde{A}[\widehat{g}_{\rm DEA}^{(h)},\gamma,n;b]^*\mid\mathcal{X}_n\Big)\Big]\Big|\le\epsilon\mid\mathcal{X}_n\Big\}\ \to\ 1.$ Furthermore, $|\tilde{A}[\widehat{g}_{\rm DEA}^{(h)},\gamma,n;b]^*-\tilde{A}[g^{(h)},\gamma,n;b]^*|=o_p(1)$, since by our assumptions on h and by Theorem 1 of Kneip et al., $\sup_{(\delta,z)\in C(bn^{-\frac{1}{p+q+1}})}\Big|\widehat{g}_{\rm DEA}^{(h)}(z\mid x_0,y_0)-g^{(h)}(z\mid x_0,y_0)\Big|=o_p(n^{-\frac{2}{p+q+1}})$.

Therefore, for every $\epsilon>0$, there exists a $b_\epsilon>0$ such that for all $b\ge b_\epsilon$ and all $\delta>0$,

$$\Pr\Big(\Big|\Pr\left[n^{\frac{2}{p+q+1}}\left(\frac{\widehat{\theta}_{\rm DEA}(x_0,y_0|\mathcal{X}_n^*)}{\widehat{\theta}_{\rm DEA}(x_0,y_0)}-1\right)\le\gamma\mid\mathcal{X}_n\right]-\Pr\left[\tilde{A}[g^{(h)},\gamma,n;b]^*\mid\mathcal{X}_n\right]\Big|\le\epsilon\mid\mathcal{X}_n\Big)\to1 \tag{4.23}$$

as $n\to\infty$.

By (4.22), (4.23), and by the continuity of the limit distribution established in (4.14), it only remains to show that for all $\gamma>0$ and all sufficiently large b,

$$\Big|\Pr(A[g^{(h)},\gamma,n;b])-\Pr(\tilde{A}[g^{(h)},\gamma,n;b]^*\mid\mathcal{X}_n)\Big|=o_p(1). \tag{4.24}$$

Any difference in the probabilities of the above events reflects a corresponding difference between the true distribution of (δ_i,Z_i) and the bootstrap distribution of $(\hat{\delta}_i^*,Z_i^*)$ given $\mathcal{X}_n$.

The probability that an observation (δ_i,Z_i) falls into $C(bn^{-\frac{1}{p+q+1}})$ is equal to $\pi_n=\bar{f}_\delta(1,0)2^{p+q-1}b^{p+q+1}\cdot n^{-1}$. For large n the distribution of the number k of points (δ_i,Z_i) falling into $C(bn^{-\frac{1}{p+q+1}})$ follows approximately a Poisson distribution with parameter $\pi_n\cdot n$, while the conditional distribution of (δ_i,Z_i) given $(\delta_i,Z_i)\in C(bn^{-\frac{1}{p+q+1}})$ is approximately equal to a uniform distribution on $C(bn^{-\frac{1}{p+q+1}})$. Also note that the matrix of partial second derivatives of $g^{(h)}$ at $z=0$ is equal to the matrix $g''(0|x_0,y_0)$ of partial second derivatives of g at $z=0$. Hence, it is easy to verify (see Proposition 1 of Kneip et al., 2008) that

$$\lim_{n\to\infty}\left|\Pr\big(A[g^{(h)},\gamma,n;b]\big)-\sum_{k=1}^{\infty}\Pr\big(U[g^{(h)},\gamma,k;b]\big)\frac{n\pi_n}{k!}e^{-n\pi_n}\right|=0 \tag{4.25}$$

for all $\delta>0,\ b>0$. Here, based on k iid random variables $(\vartheta_1,\zeta_1),\ \ldots,\ (\vartheta_k,\zeta_k)$ possessing a uniform distribution on $[0,b]\times[-b,b]^{p+q-1}$, we use $U[g^{(h)},\gamma,k;b]$ to denote the following event: there exist some $\alpha_1\ge0,\ \ldots,\ \alpha_k\ge0$ with $\sum_{j=1}^k\alpha_j=1$ such that $\sum_{j=1}^k\alpha_j\sum_{l=1}^{p-1}\zeta_j^{(l)}v_l=0$, $\sum_{j=1}^k\alpha_j\zeta_j^{(r)}=0$, $r=p,\ldots,p+q-1$, and $\sum_{j=1}^k\alpha_j\frac{1}{2g''(0|x_0,y_0)}\zeta_j'g''(0|x_0,y_0)\zeta_j+\sum_{j=1}^k\alpha_j\vartheta_j\le\gamma$.

Now, let $\hat{\pi}_n := \Pr[(\hat{\theta}_i^*, Z_i^*) \in C(bn^{-\frac{1}{p+q+1}})|\mathcal{X}_n]$, and let $\mathcal{S}(\mathcal{X}; b)$ denote the set of z-coordinates of all original observations satisfying $\widehat{\delta}_i > 1 - \tau$ as well as $|Z_i^{(j)}| \leq bn^{-\frac{1}{p+q+1}}$, $j = 1, \ldots, p+q-1$. Provided that $0 < \hat{\pi}_n$ and that $n\hat{\pi}_n = O_p(1)$, arguments similar to those above may be used to show that

$$\left| \Pr(\tilde{A}[g^{(h)}, \gamma, n; b]^*|\mathcal{X}_n)) - \sum_{k=1}^{\infty} \Pr(\tilde{U}[g^{(h)}, \gamma, k; b] \frac{n\hat{\pi}_n}{k!} e^{-n\hat{\pi}_n} \right| = o_p(1) \tag{4.26}$$

for all $\delta > 0, b > 0$ as $n \to \infty$. Here, $\tilde{U}[g^{(h)}, \delta, k; b]$ is defined similar to $U[g^{(h)}, \delta, k; b]$, but has to be based on k iid random variables $(\vartheta_1, \zeta_1^*), \ldots, (\vartheta_k, \zeta_k^*)$, where ϑ_j follows a uniform distribution on $[0, b]$, ζ_j^* follows (conditionally on $\mathcal{X}_n$) a discrete uniform distribution on $\mathcal{S}(\mathcal{X}; b)$, and ϑ_j is independent of ζ_j^*, $j = 1, \ldots, k$.

In order to analyze the difference between $n\pi_n$ and $n\hat{\pi}_n$, first note that the construction of our bootstrap sample implies $\hat{\pi}_n = \frac{b^2 n^{-\frac{2}{p+q+1}}}{\tau} \Pr[(\widetilde{\delta}_i, Z_i^*) \in D(\tau, bn^{-\frac{1}{p+q+1}})|\mathcal{X}_n]$, where $D(\tau, bn^{-\frac{1}{p+q+1}})$ is the set of all (δ, z) with $(1, z) \in C(bn^{-\frac{1}{p+q+1}})$ and $1 - \delta < \tau$. Furthermore, we can infer from (4.14) and (4.20) that $|\widehat{\delta}_i - \delta_i| = O_p(n^{-\frac{2}{p+q+1}}) = o_p(\tau)$. Recall that $\widetilde{\delta}_i = \widehat{\delta}_{j_i}$. With $\delta_i^* = \delta_{j_i}$ we therefore obtain that $\Pr[(\widetilde{\delta}_i, Z_i^*) \in D(\tau, bn^{-\frac{1}{p+q+1}})|\mathcal{X}_n] = \Pr[(\delta_i^*, Z_i^*) \in D(\tau, bn^{-\frac{1}{p+q+1}})|\mathcal{X}_n](1 + o_p(1))$. Obviously, $\{(\delta_i^*, Z_i^*)\}$ is an iid re-sample of the true (δ, z)-coordinates $\{(\delta_1, Z_1), \ldots, (\delta_n, Z_n)\}$ of the original observations. Well-known basic properties of empirical distributions, together with our assumptions on the asymptotic behavior of τ, lead to

$$\begin{aligned}
n\hat{\pi}_n &= \frac{nb^2 n^{-\frac{2}{p+q+1}}}{\tau} \Pr[(\delta_i^*, Z_i^*) \in D(\tau, bn^{-\frac{1}{p+q+1}})|\mathcal{X}_n](1 + o_p(1)) \\
&= \frac{nb^2 n^{-\frac{2}{p+q+1}}}{\tau} \Pr[(\delta_i, Z_i) \in D(\tau, bn^{-\frac{1}{p+q+1}})](1 + o_p(1)) \\
&\quad + O_p\left(\frac{nb^2 n^{-\frac{2}{p+q+1}}}{\tau} \sqrt{n^{-1} \Pr[(\delta_i, Z_i) \in D(\tau, bn^{-\frac{1}{p+q+1}})]} \right) \\
&= \frac{nb^2 n^{-\frac{2}{p+q+1}}}{\tau} \tau n^{-(p+q-1)/(p+q+1)} \bar{f}_\delta(1, 0) 2^{p+q-1} b^{p+q-1} (1 + o_p(1)) + O_p(n^{-\frac{1}{p+q+1}}/\tau^{1/2}) \\
&= n\pi_n(1 + o_p(1)).
\end{aligned} \tag{4.27}$$

Standard results on the convergence of the empirical distribution together with a straightforward generalization of the above argument can now be used to show also that the conditional distribution of ζ_j^*, given $\mathcal{X}_n$, asymptotically coincides with a uniform distribution on

$[-b,b]^{p+q-1}$. More precisely,

$$\sup_{\mathcal{C}} \left| \Pr[(\zeta_j^* \in \mathcal{C} \mid \mathcal{X}_n] - \frac{\lambda(\mathcal{C})}{(2b)^{p+q-1}} \right| = o_p(1),$$

where the supremum refers to all $(p+q)$-dimensional subintervals $\mathcal{C}$ of $[-b,b]^{p+q-1}$, and where $\lambda(\mathcal{C})$ is the Lebesgue measure of $\mathcal{C}$. Therefore, $|\Pr(\tilde{U}[g^{(h)},\delta,k;b] - \Pr(U[g^{(h)},\delta,k;b]| = o_p(1)$ for all k. Relation (4.24) then follows from (4.25)–(4.27), completing the proof. ∎

From the proof of Theorem 4.1 it is possible to gain some insight about the smoothing parameter τ. An important point in the proof is to approximate the value of $n\pi_n$ by $n\hat{\pi}_n = \frac{nb^2 n^{-\frac{2}{p+q+1}}}{\tau} \Pr\left[(\delta_i^*, Z_i^*) \in D(\tau, bn^{-\frac{1}{p+q+1}}) \mid \mathcal{X}_n\right]$. From (4.27), the standard deviation (due to re-sampling error) of this approximation is proportional to $O_p(n^{-\frac{1}{p+q+1}}/\tau^{1/2})$. Note that the term $(1 + o_p(1))$ in the last line of (4.27) hides a bias term. The leading bias term corresponds to the error in approximating the conditional distribution of δ_i (given Z_i) by a uniform distribution. Since this corresponds to a local constant approximation, this results in $E(n\hat{\pi}_n) = n\pi_n(1 + O_p(\tau))$. Hence, with $\tau \sim n^{-r/(p+q+1)}$ for some r, we must deal with a squared bias of order $n^{-2r/(p+q+1)}$, and a variance of order $n^{-(2-r)/(p+q+1)}$. An optimal value then is $r = 2/3$, leading to $\tau \sim n^{-2/3(p+q+1)}$.

5 Optimizing the Smoothing Parameters

Although the bootstrap in Algorithm #1 is consistent in the sense of Theorem 4.1, the problem of choosing appropriate values of the smoothing parameters τ and h in real-world applications remains. We propose a simple rule-of-thumb for choosing τ, and then employ an iterated bootstrap along the lines of Hall (1992) to optimize the choice of value for for h. The iterated bootstrap is made feasible by the computational simplicity of Algorithm #1—recall that the auxiliary computations required for determining the frontier points U_i^{∂} are done only once in step [3], before the bootstrap loop in begins in step [4].

Recall from the discussion in Section 4.5 that $\tau = O\left(n^{-2/3(p+q+1)}\right)$. Reflecting the estimates $\{\widehat{\delta}_i\}_{i=1}^n$ obtained in step [2] of Algorithm #1 around unity yields a set of $2n$ points that are symmetric around 1. Freedman and Diaconis (1981) proposed a robust normal-reference rule for selecting bin-widths for histogram estimators of probability density functions; their bin-width selection rule is

$$\widehat{w} = 2(\mathrm{IQ})(2n)^{-1/3}, \tag{5.28}$$

where IQ is the interquartile range. The term $2n$ appears in (5.28) due to the fact that we have $2n$ points after reflection.[3] The inter-quartile range of our reflected data is $2(1-\widehat{\delta}_{\text{med}})$, where $\widehat{\delta}_{\text{med}}$ is the sample median of the original n estimates $\{\widehat{\delta}_i\}_{i=1}^n$. To adjust for the fact that we have only n real observations, and not $2n$, we multiply (5.28) by $0.5^{-1/3}$. In addition, to obtain a value for τ that is of the correct order, we multiply the right-hand side of (5.28) by $n^{-2/3(p+q+1)}/n^{-1/3}$ to obtain our rule of thumb, i.e.,

$$\widehat{\tau} = 4\left(1-\widehat{\delta}_{\text{med}}\right) n^{-2/3(p+q+1)}. \tag{5.29}$$

Having fixed τ using the rule in (5.29), we can now focus our efforts on optimizing the choice of h. Here, we describe a data-driven, cross-validation technique that shares some features of cross-validation used to determine bandwidths in non-parametric density and regression estimation.

The idea is to begin as before with steps [1]–[3] of Algorithm #1, performing the auxiliary computation before entering the bootstrap loop in step [4]. On each bootstrap replication, a bootstrap sample $\mathcal{X}_n^*$ is drawn, and a bootstrap estimate $\widehat{\theta}_0^*$ is obtained. In the "real world," $\widehat{\theta}_0$ is an estimator of θ_0, conditional on the sample $\mathcal{X}_n$. The bootstrap uses the analogy principle: in the "bootstrap world," $\widehat{\theta}_0^*$ is an estimator of $\widehat{\theta}_0$, which is the "true value" analogous to the true value θ_0 in the "real world."

Bootstrap iteration involves creating another analogy, similar to the one described above. On each bootstrap replication, an inner bootstrap loop is performed, conditional on the sample $\mathcal{X}_n^*$, and drawing samples $\mathcal{X}_n^{**}$ on each pass through the inner loop. In this "inner bootstrap world," $\widehat{\theta}_0^*$ becomes the "true value" to be estimated by $\widehat{\theta}_0^{**}$ based on $\mathcal{X}_n^{**}$. At the end of the entire exercise, we have what amounts to a Monte Carlo experiment, whose results can be used to assess coverage of the estimated confidence intervals for θ_0 obtained with $\widehat{\tau}$ determined by (5.29) and a particular value of h. In order to optimize h, given $\widehat{\tau}$, the procedure can then be embedded in a univariate optimization algorithm, such as the golden-section search algorithm described by Kiefer (1953).

Given data $\mathcal{X}_n$, the point of interest (x_0, y_0), and a value for $h \in (0, 1]$, our iterated bootstrap consists of the following steps:

[3]Scott (1979) derived the histogram bin-width that minimizes asymptotic mean square error when using a histogram to estimate a normal density function. Scott proposed replacing the standard error appearing in his expression with the sample standard deviation. The rule of Freedman and Diaconis (1981) is more robust with respect to departures from normality. See Scott (1992) for additional discussion.

Algorithm #2:

[1] Perform steps [1]–[3] of Algorithm #1 to obtain $\widehat{\theta}_0$, the sets Z_n and $\{\widehat{\delta}_i\}_{i=1}^n$, and the set of frontier points $\{U_i^{\partial}\}_{i=1}^n$.

[2] Set $\widehat{\tau}$ using (5.29).

[3] Set $k = 0$, $\mathcal{B}_h = \varnothing$.

[4] Loop over steps [4.1]–[4.9] B_1 times:

[4.1] Draw independently, uniformly, and with replacement from the set of integers $\{i\}_{i=1}^n$ n times to create a set of labels $\mathcal{J} = \{j_i\}_{i=1}^n$.

[4.2] For each $i = 1, \ldots, n$, set $Z_i^* = Z_{j_i}$ and

$$U_i^* = \begin{cases} U_{j_i}^{\partial}/\widehat{\delta}_{j_i} & \text{if } \widehat{\delta}_{j_i} < (1-\widehat{\tau}); \\ U_{j_i}^{\partial}/\xi_{j_i}^* & \text{otherwise,} \end{cases}$$

where $\xi_{j_i}^*$ is a random, independent draw from a uniform distribution on the interval $[(1-\widehat{\tau}),\ 1]$, to construct a bootstrap sample $\mathcal{Z}_n^* = \{(Z_i^*, U_i^*)\}_{i=1}^n$.

[4.3] Compute $\widehat{\theta}_0^* = \frac{\widehat{g}_{DEA}(0|(x_0,y_0),\mathcal{Z}_n^*)}{U_0}$ and add $\widehat{\theta}_0^*$ to the set $\mathcal{B}_h$.

[4.4] Analogous to step [2] in Algorithm #1, compute $\delta_i^* = \frac{\widehat{g}_{DEA}(Z_i^*|(x_0,y_0),\mathcal{Z}_n^*)}{U_i^*} \leq 1 \; \forall \; i = 1, \ , \ldots, \ n$.

[4.5] Using extrapolation as necessary, compute (smoothed) frontier points $(Z_i^*, U_i^{\partial *})$ where $U_i^{\partial *} = \widehat{g}_{DEA}^{(h)}(Z_i^* \mid (x_0,y_0), \mathcal{X}_n^*) \; \forall \; i = 1, \ \ldots, \ n$.

[4.6] Set $\mathcal{B}_h^* = \varnothing$.

[4.7] Loop over steps [4.7.1]–[4.7.3] B_2 times:

[4.7.1] Draw independently, uniformly, and with replacement from the set of integers $\{i\}_{i=1}^n$ n times to create a set of labels $\mathcal{J}^* = \{j_i^*\}_{i=1}^n$.

[4.7.2] For each $i = 1, \ \ldots, \ n$, set $Z_i^{**} = Z_{j_i}^*$ and

$$U_i^{**} = \begin{cases} U_{j_i}^{\partial *}/\widehat{\delta}_{j_i^*}^* & \text{if } \widehat{\delta}_{j_i^*}^* < (1-\tau); \\ U_{j_i}^{\partial *}/\xi_{j_i^*}^{**} & \text{otherwise,} \end{cases}$$

where $\xi_{j_i^*}^{**}$ is a random, independent draw from a uniform distribution on the interval $[(1-\tau),\ 1]$, to construct a bootstrap sample $\mathcal{Z}_n^{**} = \{(Z_i^{**}, U_i^{**})\}_{i=1}^n$.

[4.7.3] Compute $\widehat{\theta}_0^{**} = \frac{\widehat{g}_{DEA}(0|(x_0,y_0),\mathcal{Z}_n^{**})}{U_0}$ using the un-smoothed g-function; add $\widehat{\theta}_0^{**}$ to the set $\mathcal{B}_h^*$.

[4.8] Use the estimate $\widehat{\theta}_0^*$ computed in step [5.3] and the set $\mathcal{B}_h^* = \{\widehat{\theta}_{0,\ell}^{**}\}_{\ell=1}^{B_2}$ of bootstrap values to estimate a $(1-\alpha)\times 100$-percent confidence interval $\left[\widehat{c}_{\mathrm{lo},\alpha}^*,\ \widehat{c}_{\mathrm{hi},\alpha}^*\right]$ for $\widehat{\theta}_0$.

[4.9] If $\widehat{\theta}_0 \in \left[\widehat{c}_{\mathrm{lo},\alpha}^*,\ \widehat{c}_{\mathrm{hi},\alpha}^*\right]$ then increment k by 1.

[5.] Use the estimate $\widehat{\theta}_0$ computed in step [2] and the set $\mathcal{B}_h = \{\widehat{\theta}_{0,\ell}^*\}_{\ell=1}^{B_1}$ of bootstrap values to estimate a $(1-\alpha)\times 100$-percent confidence interval $[\widehat{c}_{\mathrm{lo},\alpha},\ \widehat{c}_{\mathrm{hi},\alpha}]$ for θ_0.

[6.] Compute $\widehat{\alpha}(h) = 1 - kB_2^{-1}$, the estimated size of the interval computed in step [5].

Using any one of various programming languages (e.g., Fortran, C, etc.) or interpreters (R, S-Plus, Matlab, etc.), it is straightforward to implement Algorithm #2 as a procedure (subroutine, function, etc.). The list of required inputs includes the point of interest, (x_0, y_0); the sample data, $\mathcal{X}_n$; the numbers of replications for the outer and inner bootstrap loops, B_1 and B_2; the desired test size, α; and a value for h on the interval $(0, 1]$. The list of items returned from the procedure includes an estimate of efficiency for the point of interest, $\widehat{\theta}_0$; an estimated $(1-\alpha)\times 100$-percent confidence interval, $[\widehat{c}_{\mathrm{lo},\alpha},\ \widehat{c}_{\mathrm{lo},\alpha}]$; and $\widehat{\alpha}(h)$, the estimated size of the interval estimate. As noted above, this procedure can be used with a univariate optimization algorithm to determine $\widehat{h} = \mathrm{argmin}_h|\widehat{\alpha}(h)-\alpha|$ and the corresponding confidence interval estimate computed in step [5] of Algorithm #2.

Note that there is no need to make the convergence tolerance in the univariate algorithm used to optimize $\widehat{h}$ too small; this will avoid excessive computational burden. For $B_1 = B_2 = 1000$ bootstrap replications, the inner bootstrap loop in steps [4.7.1]–[4.7.3] amounts to a Monte Carlo experiment with 1,000 trials; hence, at 95-percent significance, $\widehat{\alpha}$ has an estimation error of about $\pm 2\sqrt{\frac{0.95\times 0.05}{1000}}$ or approximately ± 0.014. In this case, there is no reason to make the convergence tolerance smaller than 0.014. Care should be taken, however; in some situations (e.g., with small number of observations and large dimensionality $(p+q)$), it is conceivable that $|\widehat{\alpha}(h) - \alpha|$ might exceed the convergence tolerance for all $h \in (0, 1]$. In this case, the search should be stopped after some pre-defined number of iterations.

Algorithm #2 yields two confidence interval estimates; one in step [4.8], and the final interval in step [5]. In steps [4.8]–[4.9], one could compute interval estimates $\left[\widehat{c}_{\mathrm{lo},\alpha}^*,\ \widehat{c}_{\mathrm{hi},\alpha}^*\right]$ for

various values of α, and record whether each interval estimate covers $\widehat{\theta}_0$ in step [4.9]. After selecting the optimal bandwidth value $\widehat{h}$ using a univariate search as described above, one could then use the interval estimates $\left[\widehat{c}^*_{\mathrm{lo},\alpha},\ \widehat{c}^*_{\mathrm{hi},\alpha}\right]$ corresponding to this bandwidth to adjust the coverage of the final interval estimate computed in step [5]. In particular, one would select the interval estimate among the intervals $\left[\widehat{c}^*_{\mathrm{lo},\alpha},\ \widehat{c}^*_{\mathrm{hi},\alpha}\right]$ that covers $\widehat{\theta}_0$ with frequency closest to $(1-\alpha)$; denote the corresponding size as α^*. Then, in step [5], one would estimate a $(1-\alpha^*)\times 100$-percent interval instead of estimating a $(1-\alpha)\times 100$-percent interval. This is the idea of the iterated bootstrap discussed by Hall (1992).

6 Monte Carlo Experiments

We performed Monte Carlo experiments with DGPs similar to those used in the Monte Carlo experiments of Kneip et al. (2008), in order to facilitate comparison between the methods introduced in this paper and the earlier methods examined in Kneip et al..

In Model 1, we have one output and one input ($p = q = 1$). The DGP is simulated by drawing an "efficient" input observation x_e distributed uniformly on $[10, 20]$, and setting the output level $y = x_e^{0.8}$. We then compute the corresponding "observed" input value $x = x_e e^{0.2|\varepsilon|}$, where $\varepsilon \sim N(0,1)$ and is independent. The DGP for this case can therefore be written as

$$y = x^{0.8} e^{-0.16|\varepsilon|}. \tag{6.30}$$

We take the point $(x_0, y_0) = (7.5^{1.25}/0.6,\ 7.5)$ as the fixed point for which efficiency is estimated on each Monte Carlo trial; the true efficiency for this point is $\theta(x_0, y_0) = 0.6$.

In Model 2, we specify two inputs and two outputs ($p = q = 2$). Efficient input levels $x_{1e},\ x_{2e}$ are drawn from the uniform distribution on $[10, 20]$. Next, output levels are generated by drawing ω from the uniform distribution on the interval $\left[\frac{1}{9}\frac{\pi}{2},\ \frac{8}{9}\frac{\pi}{2}\right]$, and then setting $y_1 = x_{1e}^{0.4} x_{2e}^{0.4} \times \cos(\omega)$ and $y_2 = x_{1e}^{0.6} x_{2e}^{0.3} \times \sin(\omega)$. The corresponding "observed" input levels are obtained by setting $x_1 = x_{1e} e^{0.2|\varepsilon|}$ and $x_2 = x_{2e} e^{0.2|\varepsilon|}$, where $\varepsilon \sim N(0,1)$ as in Model 1. Efficiency is estimated for the fixed point $x_0 = (12.5/0.6,\ 12.5/0.6)$, $y_0 = \left(12.5^{0.8}/\sqrt{2},\ 12.5^{0.9}/\sqrt{2}\right)$ on each Monte Carlo trial. The true efficiency for this point is $\theta(x_0, y_0) = 0.6$, also as in Model 1. In both Models 1 and 2, the fixed points of interest lie roughly in the middle of the range of the output data.

Our first set of experiments were designed to analyze the sensitivity of Algorithm #1 with respect to choices for the pair (τ, h). We considered six cases comprised of Models 1 and 2 with either $n = 25$, $n = 100$ or $n = 1000$ observations. For each of the six cases, we first generated 1,280,000 samples $\mathcal{X}_n$ and applied the estimator defined in (2.6) for the corresponding fixed point (x_0, y_0) to obtain 1,280,000 realizations of the estimator $\widehat{\theta}_{\mathrm{DEA}}(x_0, y_0)$. We divided each realization of $\widehat{\theta}_{\mathrm{DEA}}(x_0, y_0)$ by $\theta(x_0, y_0) = 0.6$ to obtain an approximation of the sampling distribution of $\left(\frac{\widehat{\theta}_{\mathrm{DEA}}(x_0,y_0)}{\theta(x_0,y_0)} - 1\right)$. Given the fact that we use 1.28 million realizations, we expect that this approximation should be very accurate. Finally, we computed the 0.025 and 0.975 quantiles $(c_{\mathrm{lo},\alpha},\ c_{\mathrm{hi},\alpha})$ of our approximated sampling distribution. We treat these as "true" values of the upper and lower bounds estimated by Algorithm #1 with $\alpha = .05$.

Next, for each case, we performed 1,280 Monte Carlo trials. On each trial, we generated a sample $\mathcal{X}_n$ and applied Algorithm #1 with $B = 2,000$ bootstrap replications for each of 10,000 pairs (τ, h) where $\tau \in \{0.00,\ 0.01,\ \ldots,\ 0.99\}$ and $h \in \{0.01,\ 0.02,\ \ldots,\ 1.00\}$. Thus for each pair (τ, h), we obtain 1,280 estimates $(\widehat{c}_{\mathrm{lo},\alpha},\ \widehat{c}_{\mathrm{hi},\alpha})$; using the "true" values $(c_{\mathrm{lo},\alpha},\ c_{\mathrm{hi},\alpha})$ obtained as described above, we can estimate the mean-square error (MSE) of the estimated bounds.

Table 1 reports the estimated MSE obtained for each of the six cases we considered. The reported values of MSE are the sum of MSE for the lower and upper bounds of estimated 95-percent confidence intervals. Table 1 consists of two panels. In the first panel, minimum MSEs and corresponding values of τ and h among the 10,000 pairs (τ, h) are shown for each of the six cases described above. These results indicate that, holding dimensionality $(p + q)$ constant, MSE decreases as sample size n increases. For $p = q = 1$, the optimal value of τ decreases as n increases, but the optimal values of h remain constant at a very small value (0.01). For $p = q = 2$, the optimal value of τ decreases from 0.50 to 0.14 as n increases from 25 to 100, but the optimal value of h increases from 0.08 to 0.95. With $n = 1000$, the optimal value of τ for $p = q = 2$ is 0.99; with such a large value for τ, one can expect that almost all of the bootstrap values U_i^* generated in step [4.2] of Algorithm #1 will be drawn from a uniform distribution.

Figure 2 shows coverages of estimated 95-percent confidence intervals as a function of τ and h for the case where $p = q = 2$ and $n = 1000$. Note that in this figure, there is a ridge

along all values of τ and very small values of h (similar plots for the other cases we considered reveal similar phenomena; we do not include the plots here to save space). However, the high coverages found along this ridge are spurious; with two smoothing parameters, it is apparently possible to find combinations of τ and h that, by chance, produce coverages close to nominal values, but it is not reasonable to think that drawing the entire bootstrap sample from a uniform distribution would be meaningful. Doing so amounts to throwing away most information in the sample in order to make inference.

Consequently, in the second panel of results in Table 1, we consider MSEs of estimated confidence intervals among the pairs (τ, h) where $\tau \geq 0.2$. Here, the results seem more reasonable; i.e., for either $p = q = 1$ or $p = q = 2$, as sample size increases, the optimal value for h decreases, as does the MSE of the estimated confidence bounds. Moreover, the MSEs shown in the second panel are only slightly larger than the corresponding values in the top panel, particularly for the two cases where $n = 1000$. Overall, the results in Table 1 confirm the theoretical results in Theorem 4.1; i.e., the bootstrap in Algorithm #1 provides consistent inference.

In order to assess the usefulness of Algorithm #2 in applied situations, we conducted a second set of experiments. In these experiments, we considered the same six cases with the same DGPs as in the first set of experiments. On each of 1,024 Monte Carlo trials, we generated a sample of size n and used the golden-section search algorithm (Kiefer, 1953) with Algorithm #2 to find an "optimal" value $\widehat{h}$ for h and the corresponding confidence interval estimate (again, at 95-percent significance, or size $\alpha = 0.05$) as described above in Section 5. We considered two search strategies. The first strategy involves an unrestricted search over the closed interval $[0.01, 1.0]$. For the second strategy, we set $h = 1$ and apply Algorithm #2 to obtain $\widehat{\alpha}(h = 1)$; then we set $h = 0.5$ and apply Algorithm #2 again to obtain $\widehat{\alpha}(h = 0.5)$. If $|\alpha - \widehat{\alpha}(h = 0.5)| \leq |\alpha - \widehat{\alpha}(h = 1.0)|$, the search for an "optimal" value of h is made over the interval $[0.01, 1.0]$; otherwise, the search interval is $[0.5, 1.0]$.

Tables 2 and 3 report Monte Carlo estimates of coverages of confidence intervals obtained using Algorithm #2 with the unrestricted and restricted univariate searches for h, for each of the six cases described earlier (i.e., $n \in \{25,\ 100,\ 1000\}$ and $p = q \in \{1,\ 2\}$). Although we optimize the choice of h for confidence intervals with size $\alpha = 0.05$, in Tables 2 and 3 we report estimated coverages not only for 95-percent confidence interval estimates, but also

for 90- and 99-percent confidence interval estimates. The results are almost identical for the unrestricted and restricted searches (with $1,280$ Monte Carlo trials and $\alpha = 0.05$, the estimation error is $\pm 2\sqrt{\frac{0.95 \times 0.05}{1280}} \approx 0.012$; hence none of the differences between Tables 2 and 3 are significant). The results also indicate that embedding Algorithm #2 within the golden-section search algorithm to optimize h works quite well in the sense that one obtains, on average, confidence intervals with good coverage properties. At each of the three significance levels shown in Table 2, coverages improve with increasing sample size, both for $p = q = 1$ and $p = q = 2$. As expected, for a given sample size coverages are not quite as good when $p = q = 2$ as opposed to $p = q = 1$, but nonetheless coverages become reasonable by the time n reaches 1,000 observations; even with only 100 observations, coverages of 95- and 99-percent confidence interval estimates are close to the nominal values when $p = q = 2$. If one were interested in estimating 90-percent confidence intervals, one could perhaps improve the coverages shown in Table 2 by optimizing h for $\alpha = 0.1$ instead of $\alpha = 0.05$.[4]

Table 4 gives, again for each of the six cases described above, information about the distributions of $\widehat{\tau}$ (chosen by the rule of thumb given in (5.29)) and $\widehat{h}$ (chosen by the unrestricted and restricted golden-section searches with Algorithm #2). Each panel in Table 4 corresponds to p and q equal to either 1 or 2, and sample size equal to 25, 100, or 1,000. In each panel, the first set of results for $\widehat{h}$ correspond to unrestricted searches, while the second set of results for $\widehat{h}$ correspond to the restricted searches.

The results in Table 4 reveal that, for given dimensionality, the values $\widehat{\tau}$ decrease on average as sample size increases; in addition, the range of values of $\widehat{\tau}$ also decreases. Of course, this is expected since we use the rule of thumb given in (5.29). Perhaps more interesting, the results also show that while the coverage results shown in Tables 2 and 3 are similar for both the unrestricted and restricted searches for $\widehat{h}$, the values of $\widehat{h}$ that are chosen are in many cases quite different, particularly in the cases where $p = q = 1$. In the case

[4]One can compare the estimated coverages in Tables 2 and 3 with those given in Table 2 of Kneip et al. (2008) that were obtained using the double-smooth bootstrap described therein. The double-smooth method requires a smoothing parameter b for smoothing the initial frontier estimate (analogous to h in the method used in the current paper). Kneip et al. (2008) suggested a rule-of-thumb for choosing b; with $p = q = 1$, the rule gives values of 0.70, 0.60, and 0.48 for sample sizes 25, 100, and 800 (the largest sample size considered in Kneip et al., 2008). With $p = q = 2$, the rule for setting b gives values of 0.81, 0.74, and 0.64 for the same sample sizes. At 95-percent nominal significance, the coverages obtained with the earlier double-smooth bootstrap are about the same as those obtained here with $p = q = 1$, and about the same or slightly worse with $p = q = 2$.

where $p = q = 1$ and $n = 100$ (see the second panel in Table 4), the unrestricted search leads to a choice for h that is near 0 in each Monte Carlo trial, while the restricted search leads to a value for h close to 1 in more than 75-percent of Monte Carlo trials. Yet, results for estimated coverages in Tables 2 and 3 are nearly identical for the unrestricted and restricted searches in this case.[5]

As discussed above in Sections 1 and 4.4, the bootstrap methods described in this paper offer substantial reductions in computational burden over the full-sample method described in Kneip et al. (2008). Table 5 gives summary statistics on the distribution of computing times over the 1,024 Monte Carlo trials in each of our experiments using Algorithm #2 to optimize the choice of $\widehat{h}$. Code was written in Fortran and compiled (and optimized) using the Intel v10.1 Fortran compiler. All computations for a given Monte Carlo trial were performed on an individual core of an Intel Xeon E5410 quad-core processor running at 2.33GHz and with 6MB of L-2 cache and 12 GB of memory; computations for four trials were performed simultaneously on the four cores of a given processor. As expected, Table 5 reveals that computation time typically increases with sample size and with dimensionality, although in the case of unrestricted searches for $\widehat{h}$, computation times decrease on average when sample size is increased from 25 to 100 observations.[6] In addition, Table 5 shows that computation times required for unrestricted searches for $\widehat{h}$ are typically less than those required for restricted searches, holding sample size and dimensionality fixed.

With $n = 1,000$, $p = q = 2$, and an unrestricted search for $\widehat{h}$, Table 5 suggests that the ***expected*** required computing time to estimate a confidence interval is about 1,140 minutes, or about 19 hours. While this may seem like a substantial amount of time, it is far less time than would be required using the method proposed in Kneip et al. (2008). Recalling the discussion near the end of Section 4.4, for fixed values of the smoothing parameters, with $n = 1,000$ and $B = 2,000$ bootstrap replications, the Kneip et al. (2008) method would require solving about 334 times the number of linear programs required to implement Algorithm #1 in Section 4.4. If the full-sample bootstrap in Kneip et al. were iterated as in Algorithm #2, then the difference in number of linear programs that must be solved—

[5]The convergence tolerance for the golden-section method used to optimize the choice for h was set to 0.02 in our experiments. For $|\alpha - \alpha(h)|$ monotonically decreasing for h near 0 or 1, the algorithm can yield the same values—either 0.0102 or 0.9870—as h is driven toward either 0 or 1.

[6]This is due to the behavior described earlier; with the unrestricted search, $p = q = 1$, and $n = 100$, the "optimized" value of h that is chosen is always near 0, as shown in Table 4.

compared to the requirement for Algorithm #2—becomes even greater. Since most of the computation effort is consumed by solving linear programs, one might reasonably expect that a single Monte Carlo trial using the Kneip et al. full-sample, double-bootstrap in an iterated procedure analogous to Algorithm #2 might require ***at least*** $19 \times 334 = 6,346$ hours—264.4 days—running on a single core. Hence, while 19 hours might seem a long time, the alternative is far worse (i.e., by a factor of about 334).

7 Summary and Conclusions

As discussed in Section 1, DEA estimators have been widely applied, but typically without regard to inference about the underlying, true efficiency that is estimated. The work in this paper builds upon the earlier work in Kneip et al. (2008) to provide a consistent, tractable, computationally efficient method for inference. For the applied researcher, who typically has a single, finite sample of observations, we have shown that an iterative, cross-validation technique can be used to optimize one of the required smoothing parameters after fixing the other with a simple rule-of-thumb. Our simulation results show that in terms of coverages of estimated confidence intervals, the methods we propose work quite well even with modest sample sizes.

Despite the computational efficiency of the bootstrap methods developed above, with very large sample sizes, non-trivial computation times may still be required, although these are perhaps several orders of magnitude less than required by previously-available methods. Even with large samples, we expect that most researchers will be able to implement our methods on modern computing equipment using (perhaps several or many) multi-core processors capable of running simultaneously two or more threads; multi-core processors are now common in inexpensive desktop and even laptop computers. Algorithm #2 is trivially parallel in the sense that each of the B_1 loops in step [4] are independent of each other. By dividing the computation over perhaps J processors, one can reduce the required wall-time by a factor of almost $1/J$ with efficient coding using parallel programming techniques (see, for example, Gropp et al., 1999 or Chandra et al., 2001).[7] With the increasing availability of

[7]Near the end of December, 2008, rack-mounted units with dual Intel 2.3GHz Xeon E5410 quad-core processors could be purchased for about US$1600. Moreover, academic researchers in the U.S. can access massively parallel computers that are part of the TeraGrid project funded by the U.S. National Science Foundation (see http://www.teragrid.org/ for details). Alternatively, due to its trivially parallel nature,

both high-performance and high-throughput computing environments, the computational burden incurred while implementing the methods we have developed in this paper should be easily manageable..

Appendix A: Details of Extrapolation Procedure

The idea is to extrapolate the DEA frontier in such a way that the resulting frontier will satisfy the convexity assumption. This can be achieved by linear extrapolation but requires that we define two support points to provide the desired extrapolation. These points will be defined as points on the DEA frontier near the boundary in the output direction. Clearly, the set of points where $h_i = 1$ in (4.10) provides the most extreme set of points (where no extrapolation at all is allowed). We will iterate this idea of the algorithm of Section 4.3.1 to define a second layer of such (less) extreme points.[8]

Define the set of data points $\mathcal{X}_n^{(1)} = \mathcal{X}_n \setminus \{(X_i, Y_i) \mid h_i = 1\}$, *i.e.*, the data set after having eliminated the most extreme data points in the output direction. Next, define $\mathcal{K}^{(1)} = \left\{k \mid \widehat{\lambda}_{DEA}(X_k, Y_k \,|\, \mathcal{X}_n^{(1)}) = 1\right\}$. Then the maximal expansion for the ith observation $(X_i, Y_i) \in \mathcal{X}_n$ with respect to $\mathcal{X}_n^{(1)}$ will be given by

$$h_i^{(1)} = \left[\widehat{\lambda}_{DEA}^{(1)}(X_i, Y_i \,|\, (X_i, Y_k),\, k \in \mathcal{K}^{(1)})\right]^{-1}, \tag{A.1}$$

which is an analog of the linear program in (4.9), but with a smaller reference set since $\mathcal{K}$ has been replaced by $\mathcal{K}^{(1)}$. Note that for those points where $h_i = 1$, we will have $h_i^{(1)} > 1$. More generally, for all i, we have $h_i^{(1)} > h_i$. The points such that $h_i^{(1)} = 1$ will provide the second layer of extreme points.

Now for an arbitrary given point Z_i, $i = 1, \ldots, n$, in the case where the current value of h is smaller than h_i, the extrapolated value $\widetilde{g}_{\mathrm{DEA}}\,(h^{-1}Z_i |\, x_0, y_0)$ is easy to derive. For linear extrapolation we must determine the value at $h^{-1}Z_i$ of the line passing through the

it would be easy to divide the computations required to implement Algorithm #2 over a computing grid. Grid computing is an example of high-throughput computing, as opposed to high-performance computing on parallel machines. Many universities in the U.S., Europe, and elsewhere harness the computing power of separate machines spread across their campuses using scheduling software developed by the Condor Project (see http://www.cs.wisc.edu/condor/ for details).

[8] For the particular case of $p = q = 1$, the definition is obvious, since the univariate Z_i can be ordered and the most extreme data points will be given by the two (or three) highest order statistics of the Z_i, $i = 1, \ldots, n$.

points $\big((h_i)^{-1}Z_i, \widehat{g}_{\mathrm{DEA}}\big((h_i)^{-1}Z_i|\, x_0, y_0\big)\big)$ and $\Big((h_i^{(1)})^{-1}Z_i, \widehat{g}_{\mathrm{DEA}}\big((h_i^{(1)})^{-1}Z_i|\, x_0, y_0\big)\Big)$. Since Z_i is fixed, h^{-1} is the only parameter characterizing these curves in $\mathbb{R}^{p+q}$, the problem is thus to find a linear function of h^{-1} passing through the points above (i.e., those characterized by the values $h_i^{-1} and (h_i^{(1)})^{-1}$.

The formula for linear extrapolation is easy to derive. In particular,

$$\widetilde{g}_{\mathrm{DEA}}\left(h^{-1}Z_i|\, x_0, y_0\right) = a' \begin{pmatrix} 1 \\ h^{-1} \end{pmatrix}, \tag{A.2}$$

where the two-dimensional vector a is given by

$$a = \begin{pmatrix} 1 & (h_i)^{-1} \\ 1 & (h_i^{(1)})^{-1} \end{pmatrix}^{-1} \times \begin{pmatrix} \widehat{g}_{\mathrm{DEA}}\big((h_i)^{-1}Z_i|\, x_0, y_0\big) \\ \widehat{g}_{\mathrm{DEA}}\big((h_i^{(1)})^{-1}Z_i|\, x_0, y_0\big). \end{pmatrix} \tag{A.3}$$

From an operational point of view, note that for all $i = 1, \ldots, n$, the values of $(h_i, h_i^{(1)})$ and of $\widehat{g}_{\mathrm{DEA}}$ at the points $h_i^{-1}Z_i$, $(h_i^{(1)})^{-1}Z_i$ can be computed before the bootstrap loop and that they are independent of the current value of h and τ. Hence, they have to be computed only once, reducing the overall computational burden.

References

Banker, R. D. (1993), "Maximum likelihood, consistency and data envelopment analysis: a statistical foundation," *Management Science,* 39, 1265–1273.

Chandra, R., Dagum, L., Kohr, K., Maydan, D., McDonald, J., and Menon, R. (2001), *Parallel Programming in OpenMP,* San Francisco: Morgan Kaufmann Publishers.

Charnes, A., Cooper, W. W., and Rhodes, E. (1978), "Measuring the efficiency of decision making units," *European Journal of Operational Research,* 2, 429–444.

____ (1979), "Measuring the efficiency of decision making units," *European Journal of Operational Research,* 3, 339.

Debreu, G. (1951), "The coefficient of resource utilization," *Econometrica,* 19, 273–292.

Färe, R. (1988), *Fundamentals of Production Theory,* Berlin: Springer-Verlag.

Färe, R., Grosskopf, S., and Lovell, C. A. K. (1985), *The Measurement of Efficiency of Production,* Boston: Kluwer-Nijhoff Publishing.

Farrell, M. J. (1957), "The measurement of productive efficiency," *Journal of the Royal Statistical Society A,* 120, 253–281.

Freedman, D., and Diaconis, P. (1981), "On the histogram as a density estimator: L_2 theory," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gegiete,* 57, 453–476.

Gattoufi, S., Oral, M., and Reisman, A. (2004), "Data envelopment analysis literature: A bibliography update (1951–2001)," *Socio-Economic Planning Sciences,* 38, 159–229.

Gijbels, I., Mammen, E., Park, B. U., and Simar, L. (1999), "On estimation of monotone and concave frontier functions," *Journal of the American Statistical Association,* 94, 220–228.

Gropp, W., Lusk, E., and Skjellum, A. (1999), *Using MPI: Portable Parallel Programming with the Message-Passing Interface,* Cambridge, Massachusetts: The MIT Press.

Hall, P. (1992), *The Bootstrap and Edgeworth Expansion,* New York: Springer-Verlag.

Jeong, S. O. (2004), "Asymptotic distribution of DEA efficiency scores," *Journal of the Korean Statistical Society,* 33, 449–458.

Jeong, S. O., and Simar, L. (2006), "Linearly interpolated FDH efficiency score for nonconvex frontiers," *Journal of Multivariate Analysis,* 97, 2141–2161.

Kiefer, J. (1953), "Sequential minimax search for a maximum," *Proceedings of the American Mathematical Society,* 4, 502–506.

Kneip, A., Park, B., and Simar, L. (1998), "A note on the convergence of nonparametric DEA efficiency measures," *Econometric Theory,* 14, 783–793.

Kneip, A., Simar, L., and Wilson, P. W. (2008), "Asymptotics and consistent bootstraps for DEA estimators in non-parametric frontier models," *Econometric Theory,* 24, 1663–1697.

Koopmans, T. C. (1951), "An analysis of production as an efficient combination of activities," In *Activity Analysis of Production and Allocation,* ed. T. C. Koopmans, New York: John-Wiley and Sons, Inc. Cowles Commission for Research in Economics, Monograph 13.

Korostelev, A., Simar, L., and Tsybakov, A. B. (1995), "On estimation of monotone and convex boundaries," *Publications de l'Institut de Statistique de l'Université de Paris XXXIX,* 1, 3–18.

Scott, David (1979), "On optimal and data-based histograms," *Biometrika,* 66, 605–610.

____ (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization,* New York: John Wiley & Sons, Inc.

Shephard, R. W. (1970), *Theory of Cost and Production Functions,* Princeton: Princeton University Press.

Simar, L. (1996), "Aspects of statistical analysis in DEA-type frontier models," *Journal of Productivity Analysis,* 7, 177–185.

Simar, L., and Wilson, P. W. (1998), "Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models," *Management Science,* 44, 49–61.

____ (1999a), "Of course we can bootstrap DEA scores! but does it mean anything? logic trumps wishful thinking," *Journal of Productivity Analysis,* 11, 93–97.

____ (1999b), "Some problems with the Ferrier/Hirschberg bootstrap idea," *Journal of Productivity Analysis,* 11, 67–80.

____ (2000a), "A general methodology for bootstrapping in non-parametric frontier models," *Journal of Applied Statistics,* 27, 779–802.

____ (2000b), "Statistical inference in nonparametric frontier models: The state of the art," *Journal of Productivity Analysis,* 13, 49–78.

Table 1: Estimated Mean Square Error of Confidence Bounds ($\alpha = 0.05$)

n	$p = q$	τ	h	MSE
		$\tau \in [0, 1]$		
25	1	0.55	0.01	3.111E-05
100	1	0.52	0.01	1.217E-06
1000	1	0.46	0.01	2.714E-08
25	2	0.50	0.08	5.264E-04
100	2	0.14	0.95	2.299E-04
1000	2	0.99	0.06	9.440E-07
		$\tau \in [0, 0.2]$		
25	1	0.20	0.26	2.130E-04
100	1	0.20	0.07	4.211E-06
1000	1	0.18	0.04	6.085E-08
25	2	0.20	0.96	1.409E-03
100	2	0.14	0.95	2.299E-04
1000	2	0.20	0.20	3.661E-06

Table 2: Estimated Coverage, h Optimized for $\alpha = 0.05$ using Algorithm #2 with Unrestricted Golden-Section Search for $\widehat{h}$

n	$p = q$	$(1-\alpha) = 0.90$	$(1-\alpha) = 0.95$	$(1-\alpha) = 0.99$
25	1	0.824	0.896	0.957
100	1	0.876	0.929	0.978
1000	1	0.896	0.941	0.985
25	2	0.694	0.760	0.856
100	2	0.776	0.854	0.928
1000	2	0.840	0.920	0.979

Table 3: Estimated Coverage, h Optimized for $\alpha = 0.05$ using Algorithm #2 with Restricted Golden-Section Search for $\widehat{h}$

n	$p = q$	$(1-\alpha) = 0.90$	$(1-\alpha) = 0.95$	$(1-\alpha) = 0.99$
25	1	0.823	0.893	0.963
100	1	0.880	0.927	0.979
1000	1	0.899	0.942	0.985
25	2	0.691	0.768	0.856
100	2	0.788	0.861	0.931
1000	2	0.834	0.915	0.978

Table 4: Distibutions of Smoothing Parameters

	Min	Q1	Q2	Mean	Q3	Max
			$p = q = 1,\ n = 25$			
$\widehat{\tau}$	0.0882	0.1902	0.2275	0.2304	0.2670	0.4191
$\widehat{h}$	0.0122	0.5233	0.9870	0.7515	0.9870	0.9870
$\widehat{h}$	0.0148	0.5516	0.9870	0.7550	0.9870	0.9870
			$p = q = 1,\ n = 100$			
$\widehat{\tau}$	0.1189	0.1627	0.1747	0.1754	0.1882	0.2444
$\widehat{h}$	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102
$\widehat{h}$	0.3249	0.9870	0.9870	0.9693	0.9870	0.9870
			$p = q = 1,\ n = 1000$			
$\widehat{\tau}$	0.0959	0.1060	0.1084	0.1084	0.1109	0.1263
$\widehat{h}$	0.0102	0.0102	0.3832	0.3336	0.5760	0.9870
$\widehat{h}$	0.0644	0.3832	0.5276	0.5317	0.6138	0.9870
			$p = q = 2,\ n = 25$			
$\widehat{\tau}$	0.0334	0.1664	0.2227	0.2331	0.2889	0.6331
$\widehat{h}$	0.0126	0.1190	0.9870	0.6689	0.9870	0.9870
$\widehat{h}$	0.0111	0.1395	0.9870	0.6732	0.9870	0.9870
			$p = q = 2,\ n = 100$			
$\widehat{\tau}$	0.1135	0.1898	0.2105	0.2123	0.2322	0.3193
$\widehat{h}$	0.1824	0.6138	0.8686	0.7743	0.9870	0.9870
$\widehat{h}$	0.1402	0.6138	0.8686	0.7741	0.9870	0.9870
			$p = q = 2,\ n = 1000$			
$\widehat{\tau}$	0.1523	0.1731	0.1777	0.1776	0.1820	0.1991
$\widehat{h}$	0.4694	0.9870	0.9870	0.9435	0.9870	0.9870
$\widehat{h}$	0.4383	0.9870	0.9870	0.9410	0.9870	0.9870

NOTE: In each panel, the two rows labelled $\widehat{h}$ correspond to unrestricted and restricted searches for $\widehat{h}$, respectively.

Table 5: Distibutions of Computing Times in Minutes (Restricted and Unrestricted Searches for $\widehat{h}$

$p = q$	n	Min	Q1	Q2	Mean	Q3	Max
			Unrestricted Search				
1	25	6.68	35.89	67.05	67.04	98.57	132.40
1	100	42.74	45.57	45.86	45.78	46.16	47.20
1	1000	465.20	596.60	646.90	678.50	784.50	806.10
2	25	13.20	75.32	142.60	143.20	211.30	285.30
2	100	47.89	51.62	53.76	55.29	57.99	73.86
2	1000	1068.00	1098.00	1110.00	1140.00	1160.00	1478.00
			Restricted Search				
1	25	8.14	43.09	80.13	80.29	118.30	158.00
1	100	26.91	84.26	142.20	133.00	201.00	259.40
1	1000	621.60	747.70	1481.00	1700.00	2188.00	2893.00
2	25	16.12	87.26	169.10	169.60	249.40	334.80
2	100	51.25	160.30	271.70	258.70	381.60	506.50
2	1000	1109.00	1144.00	1326.00	1994.00	2468.00	3655.00

Figure 1: Re-characterization of the Model: Transformation from (x, y)-space to (z, u)-space

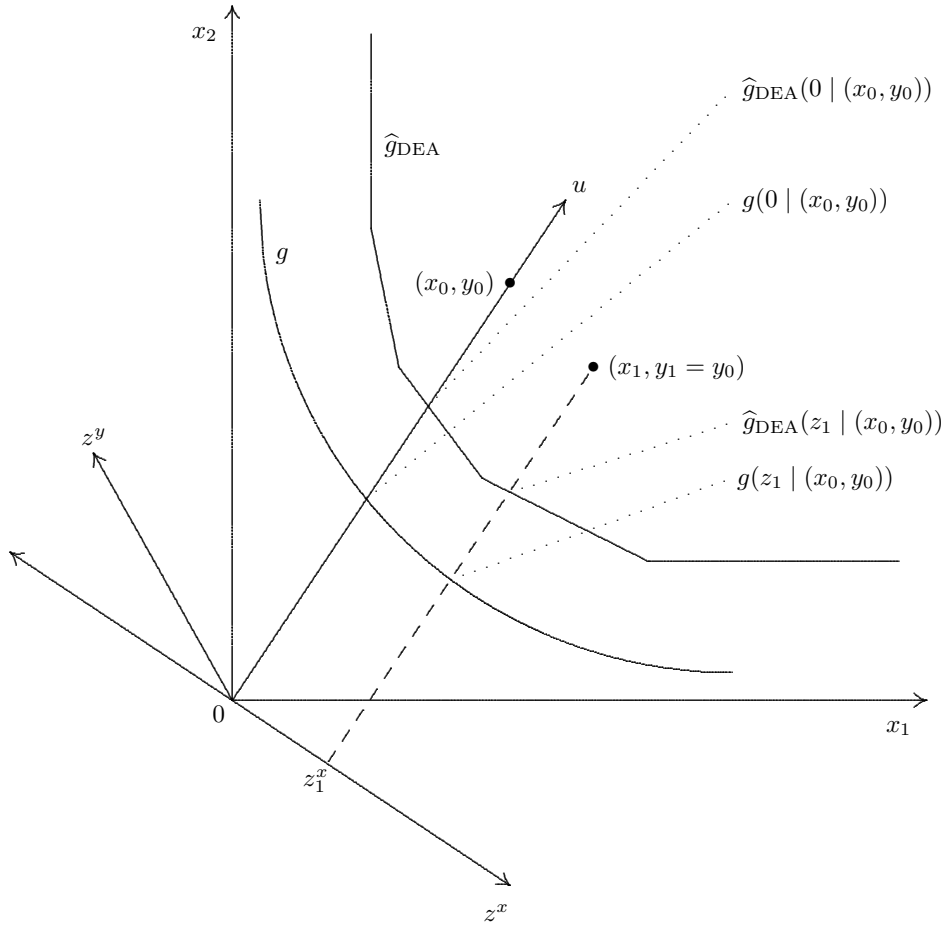

Figure 2: Coverage of Estimated 95-percent Confidence Intervals as a Function of τ and h ($p = q = 2$, $n = 1000$)