Centre for Efficiency and Productivity Analysis

Working Paper Series
No. WP09/2020

LASSO+DEA for small and big wide data

Ya Chen, Mike Tsionas and Valentin Zelenyuk

Date: October 2020

School of Economics
University of Queensland
St. Lucia, Qld. 4072
Australia

ISSN No. 1932 - 4398

LASSO+DEA for small and big wide data*

Ya Chen, Mike Tsionas and Valentin Zelenyuk

Abstract

In data envelopment analysis (DEA), the curse of dimensionality problem may jeopardize the accuracy or even the relevance of results when there is a relatively large dimension of inputs and outputs, even for relatively large samples. Recently, a machine learning approach based on the least absolute shrinkage and selection operator (LASSO) for variable selection was combined with SCNLS (a special case of DEA), and dubbed as LASSO-SCNLS, as a way to circumvent the curse of dimensionality problem. In this paper, we revisit this interesting approach, by considering various data generating processes. We also explore a more advanced version of LASSO, the so-called elastic net (EN) approach, adapt it to DEA and propose the EN-DEA. Our Monte Carlo simulations provide additional and to some extent, new evidence and conclusions. In particular, we find that none of the considered approaches clearly dominate the others. To circumvent the curse of dimensionality of DEA in the context of big wide data, we also propose a simplified two-step approach which we call LASSO+DEA. We find that the proposed simplified approach could be more useful than the existing more sophisticated approaches for reducing very large dimensions into sparser, more parsimonious DEA models that attain greater discriminatory power and suffer less from the curse of dimensionality.

Keywords: Data envelopment analysis; Data enabled analytics; Sign-constrained convex nonparametric least squares (SCNLS); Machine learning; LASSO; Elastic net; Big wide data

* This is a substantially revised version of CEPA WP02/2020. Feedback is very welcome.

1. Introduction

Data envelopment analysis (DEA) is a popular and effective tool to measure the relative efficiency of decision making units (DMUs) with multiple inputs and multiple outputs (Charnes et al., 1978).[2] DEA has been widely used to analyze many industries.[3] It has also been used as a data-driven tool for building a composite index and for balanced benchmarking (Sherman and Zhu, 2013). Recently, it was advocated that DEA can be viewed as a method (or tool) for data-oriented analytics in performance evaluation and benchmarking, as part of a broader toolbox for data enabled analytics (Zhu, 2020).

A well-known limitation of DEA (and virtually any non-parametric estimator) is the so-called curse of dimensionality—the dependency of the accuracy or explanatory power of the estimator on the dimension of the problem. This limitation becomes especially pronounced in the modern era of big wide data, which is often rich on dimensions describing various objects of interest, e.g., characteristics of customers, which can be viewed as numerous inputs of the customers being the DMUs.

One of the popular approaches in general statistics for dealing with such big wide data is LASSO—the Least Absolute Shrinkage and Selection Operator, proposed by Tibshirani (1996) (Meinshausen and Bühlmann, 2006; Tibshirani, 2011). In a nutshell, the idea of LASSO is to select a sparse (i.e., smaller in dimension) model by 'shrinking' the estimated effect of some of the variables to zero through l_1-type regularization or by penalty on the coefficients added to the standard (typically least squares) problem. Such shrinking or regularization usually leads to bias in the estimates of the coefficients, yet also helps in reducing the prediction error of the model (and the overall mean squared error), making it more parsimonious or sparse and easier to explain and use for modelling and, possibly, for predictive analytics. Such sparse models are then easier to handle and interpret in practice and have a higher out-of-

[2] The roots of DEA also go back to economic theory (activity analysis) modelling of Debreu (1951), Koopmans (1951a, b) and, most prominently, the seminal work of Farrell (1957).

[3] Also see reviews by Cook and Seiford (2009), Liu et al. (2013, 2016) and more recently by Emrouznejad and Yang (2018).

sample prediction accuracy and avoid the over-fitting challenges. This method is especially useful when the dimension of the problem is greater than the sample size.[4]

The first breakthrough in adapting LASSO to DEA appears to be due to Lee and Cai (2020), which is considered in the context of small datasets. In their interesting paper, Lee and Cai (2020) adapted LASSO by using the characterization of DEA known as the sign-constrained convex nonparametric least squares (SCNLS), proposed by Kuosmanen (2006) and Kuosmanen and Johnson (2010). They named it LASSO-SCNLS. This is the name we will also use here, interchangeably with the name LASSO-DEA, to attract the attention of a larger DEA audience, some of which might be less familiar with the interesting approach often referred to as SCNLS or by other names.[5]

In their simulation study, Lee and Cai (2020) used 9 inputs in their data generating process (DGP) and proposed a comparison of different methods by reducing the dimension of DEA one-by-one. They concluded that for a single-output case PCA-DEA is superior in most dimensions and LASSO-SCNLS dominates Group LASSO-SCNLS (adapting group LASSO to SCNLS) in most cases. For a multiple-output case they concluded that PCA-DEA and random methods perform poorly, and Group LASSO-SCNLS performs better than LASSO and generally dominates LASSO-SCNLS in most dimensions. They also concluded that the proposed LASSO-SCNLS method and its variants provide useful guidelines for the DEA with small datasets.

A key question arises: What about other versions of LASSO when it is adapted to DEA? In this paper we consider an extension of the basic LASSO (i.e., elastic net or EN) and adapt it to DEA and then perform more comprehensive Monte Carlo (MC) analysis starting with the scenarios considered by Lee and Cai (2020) and then some more general scenarios.

---

[4] E.g., see Tibshirani (2011) and Bühlmann and van de Geer (2011) for more details and references about LASSO.

[5] As pointed out by Kuosmanen (2006), SCNLS is an equivalent characterization of the output-oriented variable returns to scale (VRS) DEA model with a single-output case, which is also proven in Seijo and Sen (2011). Some attempts have been recently made to generalize this framework to multioutput by Kuosmanen and Johnson (2017) using direction distance function. Also, encouraging results from Wilson (2018) and Zelenyuk (2020) suggest that often one may still retain most of the relevant information by proxying all dimensions of outputs either via PCA or via price-based aggregation into total revenue.

From our extensive simulations, we find that among the different approaches we considered, the winners (typically by a small margin) vary across the scenarios, with no clear winner overall. More importantly, the difference among the approaches is usually not statistically significant—it is typically within two standard errors or even much less so.

Thus, while we believe that the theoretical contribution of Lee and Cai (2020), such as the pioneering work adapting Lasso for DEA context, is very important, it seems it would be fair to conclude from more extensive MC simulations that the other approaches are performing similarly well as LASSO-SCNLS.

While most of the DEA studies focus on relatively small data sets, and with relatively small dimensions, the modern age of development is increasingly demanding the analysis of big and often wide data[6], which has large dimensions. Current exceptions are the recent works by Misiunas et al. (2016), Khezrimotlagh et al. (2019), Charles et al. (2019) and Zelenyuk (2020) and we will try to contribute to this literature through this paper. In particular, for the big wide data cases where the true (yet unknown) model is sparse, we propose a simplified two-step approach to circumvent the curse of dimensionality of DEA. This approach involves two stages: standard LASSO methods are used to reduce the problem to a sparse problem at the first stage and then DEA is used at the second stage. Perhaps surprisingly, our MC simulations suggest this approach performs better than the more sophisticated LASSO-SCNLS.

The rest of the paper is organized as follows. Section 2 describes the LASSO-SCNLS approach that was proposed by Lee and Cai (2020). Section 3 introduces another

[6] Being a relatively new area of research, it appears that there is no one unanimously accepted definition of Big Data (Kitchin and McArdle, 2016). It is true that some scholars think of it as the case of a large number of observations—it is referred to as the 'Big long data' sometimes and was typically called 'large samples' in the past. Such data is actually a more preferred environment (in general and for DEA in particular) as it can enjoy the results of the large sample theories in statistics (e.g., see Zhu et al., 2018; Khezrimotlagh et al., 2019 on such contexts). Others think of Big Data as the environment where the number of variables is very large, potentially larger than the number of observations—it is a harder problem to deal with (especially for DEA), and is sometimes referred to as "Big wide data" (Lai et al., 2019; Zelenyuk, 2020). This area is what we will focus on, while acknowledging that there are also many other aspects of Big data, such as velocity, veracity (Erl et al., 2016), etc. Similar to Zelenyuk (2020), we focus on the case of very large input–output dimensions where the curse of dimensionality makes DEA applications practically infeasible.

(more advanced) version of LASSO, the so-called EN, adapting it to DEA and focuses on the case of variable correlation. Section 4 presents our simplified two-step approach. The results of MC simulations are presented in Section 5. Conclusions and directions for future research are discussed in Section 6.

2. The LASSO-SCNLS method

In this section, we first briefly describe LASSO--one of the main approaches in machine learning[7]. Then we describe SCNLS and then its combination with LASSO.

2.1. LASSO and group LASSO

For simplicity, consider a standard regression problem – a data set with n observations with standardized regressors x_{ij} and a dependent variable y_i for $i = 1, \dots, n$ and $j = 1, \dots, p$. And the researcher wants to fit or estimate the coefficients of the following linear regression model,

$$y_i = \alpha + \sum_{j=1}^{p} x_{ij}\beta_j\,, \quad i = 1, \dots, n \tag{1}$$

where α and β_j are the intercept and coefficients of regressors in the regression. Ordinary least squares, OLS, or its weighted versions (WLS) are the most common approaches to do so. However, when the number of regressors is very large, the OLS or WLS might be not very reliable. OLS is especially problematic when the dimension is larger than the number of observations, which is a more and more common case for modern data environments, sometimes referred to as big wide data. As a way to resolve the problem, and effectively anticipate the advance of the big wide data wave several decades before their arrival, Tibshirani (1996) suggested regularizing it by imposing an l_1-penalization or price on the total sum of coefficients, i.e., he suggested solving the following l_1-penalized regression problem

[7] E.g., see Mullainathan and Spiess (2017), Athey and Imbens (2019) and Hastie et al. (2009) for more details.

$$\min_{\alpha,\beta} \ \frac{1}{2}\sum_{i=1}^{n}(y_i - \alpha - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \qquad (2a)$$

where λ is a penalty price or tuning parameter chosen by the researcher (typically via some data-driven procedure like cross-validation).[8]

As pointed out by Tibshirani (1996) and exploited by many other works since then, the problem (2a) is also equivalent to setting $r = 1$ when solving the following more general constrained optimization problem (sometimes referred to as a bridge or general ridge regression)

$$\min_{\alpha,\beta} \ \frac{1}{2}\sum_{i=1}^{n}(y_i - \alpha - \sum_{j=1}^{p} x_{ij}\beta_j)^2 , \ \ s.t. \ \|\beta\|_r \leq s \qquad (2b)$$

where $\beta = (\beta_1, \ldots, \beta_p)'$ and $\|\beta\|_r$ denotes a suitable l_r-norm (e.g., $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ and $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$, which are the most common choices in practice) and s can be understood as the `budget' on the coefficients (which has one-to-one correspondence with λ).

It is also worth noting that choosing $r = 2$ leads to the so-called basic `ridge regression',[9] while choosing $r = 0$ leads to the so-called `best subset selection' approach, which is an interesting alternative to LASSO, although typically much more computer intensive than LASSO and we leave its exploration regarding DEA for future research. As pointed out by Tibshirani (1996), the LASSO estimate can be interpreted as a Bayesian posterior mode estimate when the regression parameters have independent Laplace (i.e., double-exponential) priors. The spirit of the LASSO is to use a prior which relies on the Laplace rather than the (more familiar) normal distribution.

[8] This implicitly assumes that the data for each input x_i is either in logs or standardized (by subtracting its sample mean and then dividing by its sample standard deviation), to ensure the same unit of measurement of the corresponding coefficients (β_i).

[9] It is in fact a special case of the so-called Tikhonov regularization for ill-posed problems, due to Tikhonov (1943) and later adapted to statistics by Foster (1961) who interpreted it as a Wiener–Kolmogorov filter and Hoerl (1962) who called it ridge regression.

A normal prior, essentially, imposes a quadratic penalty or l_2-norm, if centered at zero and, therefore, it allows for small coefficients but not coefficients being exactly zero which are corresponding to exclusion restrictions and are quite important in our framework as well as the 'large p, small n' framework. The Laplace prior is associated with a l_1(rather than l_2) penalty and can deal more effectively with the problem of proliferation of (irrelevant) regressors.

While being just a special case of (2b), an important distinctive feature of LASSO (e.g., relative to the ridge regression), is that besides shrinking the coefficients (to reduce the variance at the expense of inducing some bias), as is also done by the ridge regression, LASSO also (more importantly) does the variable selection, by shrinking some (and often many) coefficients of the regressors to zero, depending on the positive penalty parameter λ or the budget s. In particular, the larger is λ (or, equivalently, the smaller is s) the more regressors will be shrunk to zero. This, indeed, appears to be the most important aspect of LASSO in general when it is adapted to DEA.

The best subset selection approach also shrinks some coefficients to zero, yet does so algorithmically, with much more computational burden, because this is a non-convex optimization problem. Meanwhile, LASSO, uses the smallest order of the norm (l_1) in the penalty that guarantees the problem is convex and so is much easier to handle, especially with modern computing power and optimization methods.

Slightly different from the standard LASSO, the group LASSO considers the problem of selecting grouped variables (regressors) for accurate prediction in regression and shrinks the selected group of regressors simultaneously (Yuan and Lin, 2006). To be precise, let L be the set of input groups, and $g \in L$. The set I_g represents the set of input variables in g-th group. The group LASSO is then represented by[10]

[10] Here we use a similar notation as in Friedman et al. (2010) and Meier et al. (2008) also provided a similar notation.

$$\min_{\alpha,\beta} \ \frac{1}{2}\sum_{i=1}^{n}(y_i - \alpha - \sum_{g\in L}\sum_{j\in I_g} x_{igj}\beta_{gj})^2 + \lambda \sum_{g\in L}\sqrt{|I_g|}\sqrt{\sum_{j\in I_g}\beta_{gj}^2} \quad (3)$$

where α and β_{gj} are the intercept and coefficients of the regressors in the g-th group, and $\sqrt{|I_g|}$ accounts for the varying group sizes. That is, the group LASSO removes an entire group of input variables simultaneously when λ becomes larger (Hastie et al., 2009; Yuan and Lin, 2006). The group LASSO reflects many practical situations such as the multifactor analysis-of-variance problem and enjoys excellent performance (Yuan and Lin, 2006). It inherits the convex penalty and acts like the lasso at the group level (Meier et al., 2008). In fact, the group LASSO simplifies into the standard LASSO with a constant group size 1.

The last few decades witnessed many other variations of LASSO to address different aspects of modelling, some of which along with their comparisons can be found in Zou and Hastie (2005), Zou (2006), Meinshausen (2007), Tibshirani (2011) and Hastie et al. (2017), etc. Obviously, it is infeasible to adapt all of them to DEA in one modest paper such as ours and so we will consider just a few that we conjecture are among the most appealing for the context and also hope that future research will complete the rest of the picture.

2.2. SCNLS

DEA is usually known as a nonparametric linear programming approach for measuring productive efficiency. More recently, it was also interpreted as nonparametric least-squares regression with convexity constraints. In particular, Kuosmanen (2006) and Kuosmanen and Johnson (2010) showed that the inefficiency estimated by the output-oriented DEA with a variable returns to scale (VRS) approach (often referred to as the BCC model, due to Banker et al. (1984)) is equivalent to the efficiency estimated by SCNLS in a single-output case. In particular, the SCNLS representation is given by:

$$\min_{\alpha,\beta,\varepsilon} \ \sum_{i=1}^{n}\varepsilon_i^2$$

$$
\begin{aligned}
& s.t.\, y_i = \alpha_i + \textstyle\sum_{j=1}^{p} \beta_{ij} x_{ij} + \varepsilon_i,\ \forall i = 1, \ldots, n \\
& \alpha_i + \textstyle\sum_{j=1}^{p} \beta_{ij} x_{ij} \leq \alpha_h + \textstyle\sum_{j=1}^{p} \beta_{hj} x_{ij}\,,\ \forall i, h = 1, \ldots, n \\
& \beta_{ij} \geq 0,\ \forall i = 1, \ldots, n,\ j = 1, \ldots, p \\
& \varepsilon_i \leq 0,\ \forall i = 1, \ldots, n
\end{aligned} \tag{4}
$$

where ε_i is the (additive) inefficiency term and represents the deviation of a DMU_i from the estimated SCNLS frontier, and variables α_i, and β_{ij} are the intercept and slope parameters, respectively.

Note that the objective function in (4) minimizes the sum of squared residuals, just like in the standard least-squares regression, except that there are also additional constraints, turning the problem into a convex regression-type estimator of the boundary of the support of the data.[11] This similarity to the regression approach makes this form useful for adapting the LASSO, as was first noted by Lee and Cai (2020), which we discuss in the next section.[12] Moreover, in the second constraint of (4), we use the same notation $\forall i, h = 1, \ldots, n$ with Kuosmanen (2006) and Kuosmanen and Johnson (2010) while the notation $\forall i, h = 1, \ldots, n \; and \; i \neq h$ is used in Lee and Cai (2020).

2.3. LASSO-SCNLS

By combining SCNLS with LASSO, Lee and Cai (2020) proposed a LASSO-SCNLS method for variable selection (in a single-output case) as follows:

$$
\begin{aligned}
& \min_{\alpha, \beta, \varepsilon} \ \textstyle\sum_{i=1}^{n} \varepsilon_i^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{p} \beta_{ij} \\
& s.t.\, y_i = \alpha_i + \textstyle\sum_{j=1}^{p} \beta_{ij} x_{ij} + \varepsilon_i,\ \forall i = 1, \ldots, n
\end{aligned}
$$

[11] Model (4) is under an assumption of variable returns to scale (VRS). To get a constant returns to scale (CRS) version of model (4), one needs to add the following constraints: $\alpha_i = 0, \forall i = 1, \ldots, n$. If model (4) is adapted to sign-constrained isotonic nonparametric least squares (INLS), then we have a FDH model (Keshvari and Kuosmanen, 2013).

[12] Also see Tsionas and Izzeldin (2018) for related discussions.

$$\alpha_i + \sum_{j=1}^{p} \beta_{ij} x_{ij} \leq \alpha_h + \sum_{j=1}^{p} \beta_{hj} x_{ij}\,,\ \forall i, h = 1, \ldots, n \tag{5}$$
$$\beta_{ij} \geq 0,\ \forall i = 1, \ldots, n,\ j = 1, \ldots, p$$
$$\varepsilon_i \leq 0,\ \forall i = 1, \ldots, n$$

Note that the only difference between formulation (4) and (5) is its objective function: there is an additional penalty term in (5) for shrinking β_{ij}, just like LASSO has in (2a). For a given choice of λ, if $\beta_{ij} = 0$ for all DMUs for a certain input, then this input is removed from the analysis[13]. In particular, Lee and Cai (2020) use a binary search to identify the best λ to control the number of selected input variables and remove less correlated variables.[14]

2.4. Group LASSO-SCNLS[15]

Similar to LASSO-SCNLS, Group LASSO-SCNLS proposed in Lee and Cai (2020) adapts SCNLS by adding a penalty term with group variables in a similar fashion as was done for the standard group LASSO in Yuan and Lin (2006). In this way, it also shrinks the variables in the same group simultaneously. Specifically, Group LASSO-SCNLS shrinks β_{ij} for all observations in the same input i simultaneously as follows:[16]

$$\min_{\alpha,\beta,\varepsilon} \ \sum_{i=1}^{n} \varepsilon_i^2 + \lambda \sum_{j=1}^{p} \sqrt{\sum_{i=1}^{n} \beta_{ij}^2}$$
$$s.t.\ y_i = \alpha_i + \sum_{j=1}^{p} \beta_{ij} x_{ij} + \varepsilon_i,\ \forall i = 1, \ldots, n$$
$$\alpha_i + \sum_{j=1}^{p} \beta_{ij} x_{ij} \leq \alpha_h + \sum_{j=1}^{p} \beta_{hj} x_{ij}\,,\ \forall i, h = 1, \ldots, n \tag{6}$$

---

[13] It should be noted that we did not use model (9) in Lee and Cai (2020) for computing the optimal β_{ij} because the lower bounds for the alternate optima of Kuosmanen and Kortelainen (2012) is developed for CNLS rather than LASSO-SCNLS. In fact, we also run model (9) in Lee and Cai (2020). And the results show that each β_{ij} based on model (9) is larger than that based on model (8) in most cases.

[14] E.g., if the total number of dimensions is 10 (9 inputs and 1 output), in order to get a full sequence of reduced dimensions from 2 (one input and one output) to 9 (eight inputs and one output), Lee and Cai (2020) search for the tuning parameter λ by using a binary search to get a specific reduced dimension d =2, ..., 9.

[15] Lee and Cai (2020) actually used a notation 'Group LASSO-SCNLS'. In this paper we use 'Group LASSO-SCNLS' as an alternative to highlight the combination of group LASSO and SCNLS.

[16] Compared with the standard group LASSO in equation (3), Group LASSO-SCNLS proposed in Lee and Cai (2020) actually assumes a constant group size 1.

$\beta_{ij} \geq 0,\ \forall i = 1,\ldots,n,\ j = 1,\ldots,p$
$\varepsilon_i \leq 0,\ \forall i = 1,\ldots,n$

Except for the above methods, Lee and Cai (2020) also considered what they called the 'Random' method as well as what they called the 'PCA-DEA' method for reducing the dimensions and compared them to the LASSO-type approaches. To be more precise, their RM obtains a random sequence of removed variables by randomly selecting one input variable and removing it, one by one, until only one variable remains. Moreover, their PCA-DEA method replaces the input variables with fewer principal components (PCs) and thus the dimension (number of variables) is reduced by selecting fewer PCs.

3. Some generalizations

While having great advantages in optimally balancing the bias and variance and making the model sparse or parsimonious with potentially greater prediction accuracy, as with any method LASSO also has its limitations, which are expected to be inherited by the LASSO-SCNLS. In particular, as was noticed from the seminal work of Tibshirani (1996) and in many papers since then, one of the main weaknesses of the basic LASSO is that it may perform poorly in the case of high correlations among regressors. To address such data, Zou and Hastie (2005) proposed an improved version of LASSO that they called elastic net (EN) and we will adapt it to the DEA (via SCNLS representation) in this section. To do so, recall that the EN estimator is defined by the following problem:

$$\min_{\alpha,\beta} \ \frac{1}{2}\sum_{i=1}^{n}(y_i - \alpha - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda[\delta \sum_{j=1}^{p} |\beta_j| + (1-\delta)\sum_{j=1}^{p} \beta_j^2] \qquad (7)$$

i.e., it combines the benefit of the ridge regression and the LASSO, and remains a convex optimization problem.

Zou and Hastie (2005) pointed out that various real-world data and simulations showed that EN often outperforms LASSO, while enjoying a similar sparsity of

representation. In particular, EN encourages a grouping effect, such that strongly correlated regressors tend to be in or out of the model together.[17]

The EN-DEA (or EN-SCNLS) can be characterized by

$$\begin{aligned}
&\min_{\alpha,\beta,\varepsilon} \ \sum_{i=1}^{n} \varepsilon_i^2 + \lambda[\delta \sum_{j=1}^{p} \sum_{i=1}^{n} \beta_{ij} + (1-\delta) \sum_{j=1}^{p} \sum_{i=1}^{n} \beta_{ij}^2] \\
&s.t. \ y_i = \alpha_i + \sum_{j=1}^{p} \beta_{ij} x_{ij} + \varepsilon_i, \ \forall i = 1, \ldots, n \\
&\alpha_i + \sum_{j=1}^{p} \beta_{ij} x_{ij} \leq \alpha_h + \sum_{j=1}^{p} \beta_{hj} x_{ij}, \ \forall i, h = 1, \ldots, n \qquad (8) \\
&\beta_{ij} \geq 0, \ \forall i = 1, \ldots, n, \ j = 1, \ldots, p \\
&\varepsilon_i \leq 0, \ \forall i = 1, \ldots, n
\end{aligned}$$

Note that, as is similar to EN, we add the ridge-type (or l_2) penalty into SCNLS. Different from EN but similar to LASSO-SCNLS, we shrink β_{ij} in model (8) rather than β_j of EN. When $\delta = 1$, model (8) becomes LASSO-SCNLS. When $\delta = 0$, model (8) is an adaptation of the ridge regression, and could be called 'Ridge-DEA' (or Ridge-SCNLS).[18] The Ridge-DEA could be more useful than LASSO-DEA for usual $n > p$ where p is relatively small situations, if there are high correlations between inputs. Indeed, in this case the prediction performance of ridge regression is likely to dominate LASSO in standard regression setups (Tibshirani, 1996). In model (8), if $\beta_{ij} = 0$ for all DMUs for a certain input, then this input is removed from the analysis.

4. A simplified two-step approach

In the previous section, we described several approaches for circumventing the curse of dimensionality in DEA. We also noted that Lee and Cai (2020) were using a binary search to get a tuning parameter λ corresponding to a specific reduced dimension. However, from their study it is unclear whether the true relevant regressors are

[17] EN also removes the limitation on the number of selected variables and is therefore useful when the number of regressors is much bigger than the number of observations, where LASSO may often encounter more difficulties.

[18] In this paper, we set $\delta = 0.5$ for EN+DEA and EN-SCNLS. By following a similar rule for choosing the tuning parameters in Zou and Hastie (2005), we also tried other different values of δ (0.1, 0.3, 0.7 and 0.9) and re-ran the results. The main results were qualitatively the same as the findings in section 5.2.2.

identified in this way and if all the irrelevant regressors are set to zero. Indeed, the shrinkage of coefficients of the irrelevant regressors to zero (even if at the expense of potential bias for the coefficient of the relevant regressors) is one of the most important aspects of LASSO. For this reason, methods like cross-validation LASSO are usually used to obtain an `optimal' (under a certain criterion) value for the tuning parameter that attempts to identify which coefficients are set to zero and which are not. Lee and Cai (2020) have not discussed such an approach, perhaps because it is not working as well as it is in the standard LASSO cases. Indeed, our attempts to adapt a cross-validation to the LASSO-SCNLS also have not provided any optimistic evidence, and so we leave it for future research. Meanwhile, here we propose a simplified two-step approach to handle the curse of dimensionality of DEA via LASSO, which we will call 'LASSO+DEA'. Particularly, the algorithm could be summarized as follows:

Step 1*. As an exploration of the first order approximation of the complex multi-dimensional world, using standard cross-validation LASSO or its more advanced versions (e.g. EN) to select an optimal number of regressors due to an optimal choice of the tuning parameter* λ (e.g., selected *via 10-fold cross validation or other popular methods).*

Step 2*. Run DEA (or SCNLS) on only the variables selected as relevant in Step 1.*

It should be noted that the LASSO approach built in Lee and Cai's (2020) section 4.1.2 is a two-step approach, although there is still an important difference between our approach and theirs. In their work, they were "manually" choosing the tuning parameter λ. What we proposed and extensively investigated in simulations for the first stage is to use LASSO with data-driven selection of the tuning parameter λ (e.g., based on optimizing the classical cross validation criterion) to select the relevant variables *optimally*, before deploying the DEA on those selected variables. In spite of this difference, we benefit a lot from the interesting work of Lee and Cai's (2020).

An important advantage of this simplified approach is that many packages in popular software (Matlab, R, Python, etc.) are available to implement both stages, which have been tested on many synthetic and real data. We will discuss the

implementation of this approach in Matlab and compare how it performs relative to LASSO-SCNLS in the next section.

A disadvantage of this approach is, of course, that the first stage is only looking at the first order approximation (possibly in logs, so potentially modelling some curvature). On the other hand, when the dimension is very large such approximation of the more complex and highly multi-dimensional world might be the only feasible way to proceed.

5. Monte Carlo evidence and comparison

In this section, we first compare the above methods in the small data setting as was in Lee and Cai (2020). Then, we compare the LASSO+DEA and LASSO-SCNLS (EN-SCNLS) under the 'big wide data' context.

The performance and comparison of methods is analyzed with the help of approximate (or estimated) mean squared error (MSE), i.e.,

$$MSE_m = \frac{1}{n}\sum_{i=1}^{n}\left(y_{im}^{T} - (\theta_{im}^{*} \times y_{im})\right)^2 \quad \forall m = 1,\ldots,M \tag{9a}$$

where y_{im}^{T} is the true efficient output and $\theta_{im}^{*} \times y_{im}$ is the estimate and M is the number of Monte Carlo replications for the scenario of interest. Here θ_{im}^{*} is calculated by output-oriented BCC model (Lee and Cai, 2020), and $\theta_{im}^{*} = 1 - \varepsilon_{im}/y_{im}, \forall i = 1,\ldots,n$ as shown in Kuosmanen and Johnson (2010). When efficiency scores are compared (e.g., Wilson, 2018; Zelenyuk, 2020), equation (9a) is changed to:

$$MSE_m = \frac{1}{n}\sum_{i=1}^{n}(\theta_{im}^{T} - \theta_{im}^{*})^2 \quad \forall m = 1,\ldots,M \tag{9b}$$

where θ_{im}^{T} is the true efficiency for a DMU.

Then, MSE_m is averaged over the number of Monte Carlo replications (M), i.e.,

$$AMSE_M = \frac{1}{M}\sum_{m=1}^{M} MSE_m \tag{10}$$

To get a sense of significance in the differences of average MSE (AMSE)[19] across different methods, we also present the Monte Carlo standard errors for the averages of MSE, computed as

$$se(AMSE_M) = \frac{1}{M}\sqrt{\sum_{m=1}^{M}(MSE_m - AMSE_M)^2} \tag{11}$$

As pointed out by Lee and Cai (2020), their PCA-DEA uses PCs as inputs and therefore cannot generate a `meaningful' DEA-estimated frontier and that it is also difficult to interpret the relationship of new PCs with respect to the original input variables. Hence, in this paper, we exclude PCA-DEA from the performance comparison of dimension reduction methods. In particular, we compare the following six methods: 1) Random; 2) LASSO; 3) LASSO-SCNLS; 4) Group LASSO-SCNLS; 5) EN; 6) EN-SCNLS. Note that only parts of the six methods are compared in different simulations and scenarios.

The computer used in this paper is a MacBook Pro laptop with a 2.9 GHz dual-core Intel Core i5 CPU, 16 GB memory and a MacOS High Sierra 10.13.6 operating system. All the computations were done in MATLAB R2016a using codes programmed by the authors, involving the standard MATLAB library. Besides our own programming we adapted some earlier codes of Leopold Simar. It should be noted that the computing time significantly increases as the number of replications and/or the sample goes up. For example, the computing time for Tables 1, 3 and 4 was about 3 hours, 8 hours, and 25 days, respectively.

5.1. Comparisons for small data environments

[19] Besides the average MSE comparison, in this paper, we also examine the comparison for average mean absolute deviation (MAD), average bias, average Pearson coefficient, average Spearman coefficient and average Kendall coefficient. Because the definitions of other indicators such as MAD and bias are well-known and their formulations could be easily obtained by modifying (9a) or (9b), we did not show all of them for conciseness.

In this section, we first try to replicate the simulations of Lee and Cai (2020) using their DGP with 10 dimensions (9 inputs and 1 output).[20] Table 1 shows the results.

Table 1. AMSE and standard errors of four variable selection methods

	AMSE (s. e.)			
Dimensions	Random	LASSO	LASSO-SCNLS	Group LASSO-SCNLS
10	20.705 (0.890)	20.705 (0.890)	20.705 (0.890)	20.705 (0.890)
9	19.616 (0.834)	19.416 (1.020)	19.217 (0.919)	19.904 (0.882)
8	18.217 (0.786)	17.899 (1.008)	17.766 (0.929)	18.288 (0.906)
7	16.246 (0.869)	16.265 (0.996)	15.958 (0.872)	16.504 (0.889)
6	13.619 (0.836)	14.305 (1.049)	13.867 (0.879)	13.841 (0.876)
5	10.480 (0.850)	11.215 (0.881)	11.818 (0.862)	11.275 (0.898)
4	7.541 (0.720)	8.580 (0.680)	8.828 (0.651)	8.240 (0.644)
3	5.298 (0.541)	5.762 (0.640)	6.187 (0.451)	5.516 (0.526)
2	2.630 (0.343)	3.169 (0.399)	3.580 (0.309)	2.767 (0.312)

Note: The standard error of AMSE is shown in parenthesis. And equation (9a) is used for calculation.

As shown in Table 1, the AMSE decreases as the dimension goes down, which is the same as the results in Table 2 of Lee and Cai (2020). For each row or dimension, the AMSE in Table 1 of this paper is close to that in Table 2 of Lee and Cai (2020). And the AMSE under LASSO-SCNLS is larger than that under Group LASSO-SCNLS for a dimension no larger than 6. So LASSO-SCNLS does not always show better performance for higher dimensions than Group LASSO-SCNLS, which is a different conclusion from that in Lee and Cai (2020). Most importantly, when we take into account the standard errors of AMSE, our results suggest there is no clear dominance of any of the methods because the AMSE for the four methods is well within two standard errors from each other for most dimensions.

[20] See Appendix A for more details on their DGP setting.

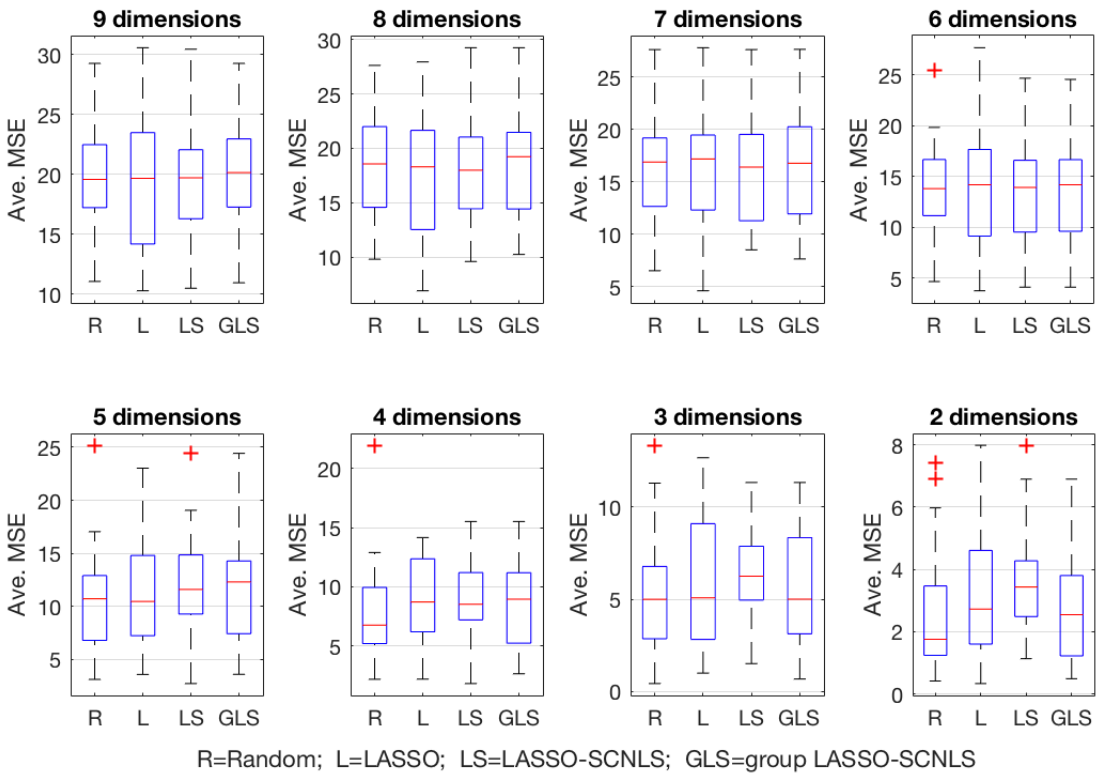

Figure 1. Box plot of MSEs

Figure 1 further shows the box plots for the MSEs and one can see that the difference in performance for the above four methods (Random, LASSO, LASSO-SCNLS and Group LASSO-SCNLS) is indeed fairly small. Taking into account the variation (e.g., via estimated iqr, standard errors, etc.), we can also conclude that there is no clear dominance for certain methods. We also explore more scenarios such as more general settings of the above DGP in Lee and Cai (2020) and the DGP setting considered in Wilson (2018) and other works and reach the same conclusion. Namely, none of these four approaches dominate the other, especially when considering measures of accuracy and variation. See appendix B for the details.

5.2. Comparisons for 'big wide data' environments

In this section, we first compare the shrinkage ability of cross-validation LASSO and LASSO-SCNLS to illustrate the rationale of our proposed LASSO+DEA. Then we compare the performance between the LASSO+DEA and LASSO-SCNLS (EN-SCNLS).

5.2.1. Shrinkage ability of cross-validation LASSO and LASSO-SCNLS

Consider a simple scenario where the total number of observations is $n = 100$, the total number of regressors is $p = 50$, while the total number of truly relevant regressors is $q = 5$. To be more precise and without loss of generality, suppose the first five regressors are regarded as the true regressors, corresponding to assumed beta coefficients $\beta_j^T = j,\ j = 1,\ldots,5$. Meanwhile, the other regressors are irrelevant regressors in the sense that their coefficients are zero. As is typical in practice, suppose the researcher does not know which regressors are relevant and what their relations to the dependent variable are and he/she needs to rely on LASSO to identify them.

To generate the synthetic data to illustrate the performance of different methods, we follow a DGP setting similar to Hastie et al. (2017), adapting it to the usual context considered in efficiency analysis. In particular, the dependent variable is generated as

$$y_{im} = \sum_{j=1}^{p} x_{ijm}\beta_{jm} + v_{im} - u_{im}, \quad i = 1,\ldots,n, m = 1,\ldots,M \tag{12}$$

where $u_{im} \sim N_+(0, {\delta_u}^2)$ and $v_{im} \sim N(0, {\delta_v}^2)$, i.i.d. for all $i = 1,\ldots,n$ while regressor x_i is generated from the multinormal. To be more precise, x_i are row entries of matrix $X \in \mathbb{R}^{n\times p}$ generated from $N_p(0, \Sigma_X)$, where $\Sigma_X = \zeta\delta_v\Sigma$ and ζ is the signal-to-noise parameter that we vary to explore its influence and $\Sigma \in \mathbb{R}^{p\times p}$ has an entry (w, t) equal to $\rho^{|w-t|}$ which allows for different degrees of correlation between the inputs.

For the results presented below, we have set $\delta_u = 0.5$ and $\delta_v = 0.05$, while the signal-to-noise ratio (SNR)[21] is set to $\zeta = 10$ and $\rho = 0.7$.

We will compare the performance of cross-validation LASSO and LASSO-SCNLS.[22] Besides the two methods, we also present ordinary least squares (OLS) as a

[21] This is a similar setting as in Hastie et al. (2017) and Bertsimas et al. (2016).

[22] Note that in Matlab, the LASSO procedure standardizes inputs X prior to fitting the model sequence by default, so we also do this to X prior to using it in LASSO-SCNLS. Moreover, for LASSO-SCNLS we use the same tuning parameter as the one identified by cross-validation LASSO.

benchmark. Note that because the true model is linear, the standard LASSO is indeed a suitable approach here, as is the OLS, except for modelling the inefficiency, which is effectively ignored by the LASSO and by the OLS. On the other hand, due to no correlation of the inefficiency with the regressors, ignoring the inefficiency term here should not affect the estimates of the slopes (which is the main focus here) and only affects the intercept, which is typically not important for LASSO and its variants (and can also be recovered if needed in the style done in SFA literature).

In principle, the performance of LASSO-SCNLS should also be reasonably good here because it is nonparametric and should be able to estimate the truth whether it is linear or not (although it is less efficient than the parametric LASSO). The only problem is the noise, however, this is fairly small relative to the inefficiency.[23]

Table 2 presents the results from a typical draw from the DGP described above in this section. The second column shows the true coefficients β_j^T. The third column shows the estimated coefficients with the tuning parameter selected via 10-fold cross-validation LASSO. The fourth column presents LASSO-SCNLS with the same tuning parameter as was used for the standard LASSO. Here we report the average of all the observations' coefficient β_{ij} for the jth regressor, i.e. $\frac{1}{n}\sum_{i=1}^{n}\beta_{ij}$. The fifth to seventh columns present the OLS-estimated coefficients, the corresponding t-statistics and the standard errors respectively, to get a sense of the performance of the simplest approach among those we consider.

Note that despite the very large dimension of regressors or inputs, OLS is still doing reasonably well and is able to identify correctly all the relevant regressors, and just very few of the irrelevant regressors are identified at different levels of significance. This was persistent across many replications and for various values of parameters. Of course, when the number of regressors is increasing, the degrees of freedom decrease and thus the reliability of OLS also decreases but when it gets larger than the sample size, OLS is unavailable. In this case the LASSO becomes particularly

[23] An alternative is to use StoNED (Kuosmanen and Kortelainen, 2012) here to model the noise explicitly.

useful, and in general, although it tends to introduce some bias, the prediction accuracy usually improves relative to OLS (Tibshirani, 1996). Moreover, it is still possible to obtain standard errors and related statistics even if $p > n$. From Table 2 one can see that LASSO indeed performed well, often better than OLS, which persisted across most replications and for different values of parameters. Most importantly, it was able to correctly identify the relevant regressors and set all others to zero in the draw below as well as in most (yet of course not all) of the replications in our study.[24]

Table 2. Performance comparison between cross-validation LASSO and LASSO-SCNLS

Inputs	β_j^T	LASSO	LASSO_SCNLS	OLS	t_stat	se_b
1	1	0.781	0.527	1.000	72.623	0.014
2	2	1.934	0.893	2.015	105.403	0.019
3	3	3.049	2.810	2.989	156.760	0.019
4	4	3.922	2.446	3.992	211.507	0.019
5	5	4.802	3.555	4.993	284.720	0.018
6	0	0	0.061	-0.005	-0.241	0.020
7	0	0	0.012	0.003	0.145	0.019
8	0	0	0.025	-0.017	-1.009	0.017
9	0	0	0.165	-0.001	-0.089	0.016
10	0	0	0.055	0.021	1.114	0.019
11	0	0	0.021	-0.003	-0.159	0.018
12	0	0	0.028	0.001	0.051	0.018
13	0	0	0.087	0.033	1.695	0.020
14	0	0	0.067	-0.041	-2.655	0.015
15	0	0	0.003	0.024	1.497	0.016
16	0	0	0.008	-0.029	-1.626	0.018
17	0	0	0.099	0.024	1.531	0.016
18	0	0	0.000	-0.024	-1.218	0.019
19	0	0	0.059	-0.004	-0.229	0.017
20	0	0	0.001	0.011	0.567	0.019
21	0	0	0.033	-0.022	-1.247	0.018
22	0	0	0.037	0.031	1.831	0.017
23	0	0	0.090	-0.035	-1.851	0.019
24	0	0	0.096	0.011	0.642	0.018
25	0	0	0.081	0.050	2.468	0.020
26	0	0	0.212	-0.027	-1.393	0.019
27	0	0	0.068	0.020	1.161	0.017
28	0	0	0.013	-0.012	-0.879	0.014
29	0	0	0.019	0.001	0.058	0.020
30	0	0	0.156	0.008	0.423	0.019
31	0	0	0.124	0.016	0.924	0.017
32	0	0	0.016	0.000	-0.002	0.017
33	0	0	0.018	-0.038	-1.710	0.022
34	0	0	0.001	0.005	0.273	0.019

[24] For example, according to our results from 1000 replications, in all 1000 replications our approach selected all five true regressors and discarded all irrelevant regressors. See more discussions in the Appendix C. Also see Hastie et al. (2017) for more examples of the performance of LASSO and other methods in other contexts.

35	0	0	0.011	-0.015	-0.684	0.022
36	0	0	0.012	0.041	1.930	0.021
37	0	0	0.009	-0.030	-1.722	0.018
38	0	0	0.010	0.007	0.373	0.017
39	0	0	0.018	0.028	1.787	0.015
40	0	0	0.139	-0.023	-1.071	0.022
41	0	0	0.002	-0.007	-0.358	0.019
42	0	0	0.017	0.012	0.608	0.020
43	0	0	0.002	0.008	0.474	0.017
44	0	0	0.039	0.006	0.318	0.020
45	0	0	0.045	-0.006	-0.331	0.017
46	0	0	0.022	0.013	0.645	0.020
47	0	0	0.035	0.001	0.041	0.018
48	0	0	0.032	-0.012	-0.645	0.019
49	0	0	0.029	-0.006	-0.254	0.022
50	0	0	0.001	-0.004	-0.233	0.016

On the other hand, note that the estimated slope coefficients via LASSO-SCNLS are fairly far from the given (true) beta coefficients. Moreover, in virtually all the replications we analyzed, it seems that LASSO-SCNLS fails to discard the irrelevant regressors, i.e., it does not shrink the corresponding beta coefficients to zero even though we used the same tuning parameter that was optimal for the LASSO for the linear model (which is true here) and so, in a sense, is the best one can get here so far. In principle, it might be possible to develop a better method for an optimal selection of the tuning parameter λ specifically for LASSO-SCNLS that would help the latter perform well relative to the simplified approach, yet we were not able to do so (as apparently also the case for Lee and Cai (2020)) and hope that future research will help in this direction. Meanwhile, the simplified approach can already utilize many existing resources (available for Matlab, R, Python, etc.) and as a result, appears to be a more reliable approach to use in practice so far.

Indeed, since LASSO-SCNLS showed very poor performance in a very simple (linear) world, it would be hard to believe that somehow it gets better for a more complicated world and so we therefore, do not expect it. To explore this conjecture, we also considered other scenarios when the true output has a non-linear relationship to inputs, yet satisfies convexity and monotonicity (as required by SCNLS). Here it is worth noting that the latter two requirements (convexity and monotonicity) limit the degree of non-linearity substantially, making the linear approximation a fairly close

approximation and so the simplified approach is still expected to work reasonably well, especially in combination with the use of log-log (yet linear in parameters) or other popular in practice specifications.

In various simulations that look at different degrees of non-linearity and different noise, we indeed noted that as the magnitude of the error term goes up compared with the input values, or if the degree of nonlinearity of the true output increases, the accuracy of the cross-validation LASSO identifying the true regressors decreases, leading to more instances of irrelevant regressors identified as relevant (See the next sub-section and Appendix C for more discussions and details).

5.2.2. Comparisons of the simplified two-step approach and LASSO-SCNLS (EN-SCNLS)

Here, we will compare the performance between the LASSO+DEA (EN+DEA)[25] and LASSO-SCNLS (EN-SCNLS). We use the same DGP as Zelenyuk's (2020, Section 4.1), but only consider the single output case and keep only a few of many inputs as relevant (and set other coefficients to zero). In particular, note that the relationship between the output and the inputs is non-linear here (Cobb-Douglass) and so the standard LASSO that uses a linear model in original units would be misspecified here, yet may still provide a fair approximation. Taking logarithmic transformation would obviously make this relationship linear, and we indeed do so here. Also, note that there is no noise in this DGP.

In the following part, we consider 100 inputs and 1 output, three sample sizes (20, 50 and 100) with 200 replications in our Monte Carlo simulations. Note that only 5 out of 100 inputs are relevant (i.e. true regressors) under the so-called sparsity assumption. As shown above, the goal is first to see if cross validation LASSO is able to select a good value for λ that helps identify the true relevant regressors and discard the irrelevant regressors, and then to compare the performance by comparing the

[25] We consider the simplified two-step approach for both cross-validation LASSO and cross-validation EN because we consider both LASSO-SCNLS and EN-SCNLS in the following part. And we call it LASSO+DEA and EN+DEA, respectively.

true efficiency scores to their estimates between the LASSO+DEA (EN+DEA) and LASSO-SCNLS (EN-SCNLS). We use a similar procedure in section 5.2.1 for assigning the optimal tuning parameter λ of LASSO (EN) to LASSO-SCNLS (EN-SCNLS). We also consider two cases of γ, i.e. 1 and 0.5, which show different degrees of returns to scale in Zelenyuk (2020). Table 3 reports the results. Similar to Zelenyuk (2020), equation (9b) is used for calculation and result comparison.

Table 3. Monte Carlo Comparisons of LASSO+DEA (EN+DEA) and LASSO-SCNLS (EN-SCNLS)

	Sample size: 20				Sample size: 50				Sample size: 100			
Indicators	LASSO+ DEA	LASSO-SCNLS	EN+DEA	EN-SCNLS	LASSO+ DEA	LASSO-SCNLS	EN+DEA	EN-SCNLS	LASSO+ DEA	LASSO-SCNLS	EN+DEA	EN-SCNLS
$\gamma = 1$												
Ave. MSE	0.061	0.128	0.072	0.128	0.056	0.126	0.063	0.126	0.064	0.124	0.074	0.124
s. e.	0.003	0.002	0.003	0.002	0.003	0.001	0.003	0.001	0.003	0.001	0.003	0.001
Ave. MAD	0.193	0.307	0.210	0.307	0.178	0.302	0.191	0.302	0.194	0.299	0.211	0.299
s. e.	0.005	0.003	0.005	0.003	0.005	0.002	0.006	0.002	0.005	0.001	0.005	0.001
Ave. bias	-0.115	-0.307	-0.135	-0.307	-0.115	-0.302	-0.132	-0.302	-0.162	-0.299	-0.183	-0.299
s. e.	0.010	0.003	0.011	0.003	0.009	0.002	0.009	0.002	0.008	0.001	0.008	0.001
Ave. Pearson	0.524	0.024	0.452	0.025	0.542	0.077	0.491	0.079	0.544	0.076	0.477	0.076
s. e.	0.018	0.016	0.020	0.016	0.019	0.010	0.020	0.010	0.016	0.007	0.017	0.007
Ave. Spearman	0.505	0.046	0.451	0.047	0.541	0.121	0.505	0.122	0.552	0.140	0.490	0.140
s. e.	0.017	0.015	0.018	0.016	0.016	0.010	0.017	0.010	0.013	0.007	0.015	0.007
Ave. Kendall	0.377	0.032	0.334	0.033	0.400	0.084	0.371	0.084	0.404	0.095	0.356	0.095
s. e.	0.014	0.011	0.014	0.011	0.013	0.007	0.013	0.007	0.011	0.005	0.012	0.005
Ave. input	2.700	78.215	3.670	74.430	4.705	99.435	5.850	99.630	6.075	100.000	7.680	100.000
s. e.	0.219	1.644	0.284	1.264	0.413	0.185	0.461	0.117	0.350	0.000	0.421	0.000
Ave. selected relevant regressors	1.015	4.025	1.015	4.045	1.755	4.970	1.725	4.985	2.845	5.000	2.855	5.000
s. e.	0.036	0.089	0.038	0.077	0.068	0.014	0.065	0.009	0.078	0.000	0.078	0.000
Ave. selected irrelevant regressors	1.685	74.190	2.655	70.385	2.950	94.465	4.125	94.645	3.230	95.000	4.825	95.000
s. e.	0.202	1.572	0.265	1.218	0.366	0.175	0.414	0.114	0.304	0.000	0.374	0.000

$\gamma = 0.5$												
Ave. MSE	0.058	0.128	0.071	0.128	0.031	0.126	0.044	0.126	0.026	0.124	0.037	0.124
s. e.	0.003	0.002	0.003	0.002	0.002	0.001	0.003	0.001	0.002	0.001	0.003	0.001
Ave. MAD	0.188	0.307	0.211	0.307	0.124	0.302	0.151	0.302	0.111	0.299	0.134	0.299
s. e.	0.005	0.003	0.005	0.003	0.005	0.002	0.006	0.002	0.004	0.001	0.005	0.001
Ave. bias	-0.165	-0.307	-0.188	-0.307	-0.095	-0.302	-0.127	-0.302	-0.070	-0.299	-0.101	-0.299
s. e.	0.007	0.003	0.008	0.003	0.006	0.002	0.007	0.002	0.006	0.001	0.007	0.001
Ave. Pearson	0.599	0.033	0.508	0.030	0.754	0.108	0.669	0.104	0.783	0.090	0.709	0.090
s. e.	0.020	0.017	0.023	0.017	0.015	0.010	0.018	0.010	0.012	0.007	0.016	0.007
Ave. Spearman	0.577	0.063	0.497	0.052	0.734	0.139	0.658	0.140	0.764	0.143	0.695	0.143
s. e.	0.019	0.016	0.021	0.016	0.013	0.010	0.016	0.010	0.011	0.008	0.015	0.008
Ave. Kendall	0.444	0.045	0.378	0.037	0.573	0.096	0.506	0.096	0.601	0.097	0.540	0.097
s. e.	0.016	0.012	0.017	0.011	0.011	0.007	0.014	0.007	0.010	0.005	0.013	0.005
Ave. input	2.605	76.935	3.680	71.025	2.210	99.640	3.530	99.450	2.435	100.000	3.570	100.000
s. e.	0.200	1.669	0.240	1.210	0.209	0.108	0.318	0.144	0.231	0.000	0.288	0.000
Ave. selected relevant regressors	1.055	4.055	1.075	3.870	1.185	4.995	1.270	4.995	1.465	5.000	1.625	5.000
s. e.	0.027	0.085	0.028	0.081	0.034	0.005	0.040	0.005	0.056	0.000	0.060	0.000
Ave. selected irrelevant regressors	1.550	72.880	2.605	67.155	1.025	94.645	2.260	94.455	0.970	95.000	1.945	95.000
s. e.	0.187	1.604	0.229	1.162	0.189	0.107	0.295	0.144	0.195	0.000	0.246	0.000

As can be seen in Table 3, the results based on LASSO+DEA and EN+DEA show better performance than those based on LASSO-SCNLS and EN-SCNLS. For example, for 20 observations and $\gamma = 1$, the average MSE is 0.061 and 0.072 for LASSO+DEA and EN+DEA while it is 0.128 for LASSO-SCNLS and EN-SCNLS. The average MAD for LASSO+DEA and EN+DEA are smaller than those for LASSO-SCNLS and EN-SCNLS, while the average bias for LASSO+DEA and EN+DEA are larger (due to negative bias) than those for LASSO-SCNLS and EN-SCNLS. And all three correlation indicators (Pearson, Spearman and Kendall) for LASSO+DEA and EN+DEA are also significantly higher than those for LASSO-SCNLS and EN-SCNLS.

For the simplified two-step approach, according to our results in Table 3, for $\gamma = 0.5$, as the sample size gets larger, the average MSE, MAD and bias reduce and continue to be much smaller while the average Pearson, Spearman and Kendall increase and continue to be much larger for LASSO+DEA and EN+DEA relative to LASSO-SCNLS or EN-SCNLS. For the LASSO-SCNLS or EN-SCNLS, for $\gamma = 0.5$, as the sample size gets larger, there is only a slight decrease for the average MSE, MAD and bias while the average Pearson even reduces for the 100-observation case. It should be noted that for $\gamma = 1$, as the sample size gets larger from 20 to 100, the shrinkage performance for the simplified two-step approach does not work as well as expected (a slight increase in MSE, MAD and bias for LASSO+DEA and EN+DEA, and a slight decrease in Pearson, Spearman and Kendall for EN+DEA). For both cases of returns to scale, the performance for LASSO+DEA is slightly better than that for EN+DEA. And the results for LASSO-SCNLS and EN-SCNLS are almost the same with small differences of correlation coefficients.

Most importantly, for different sizes of observations, the average number of selected inputs is small for the simplified two-step approach, which shows a stable shrinkage of coefficients of irrelevant variables to zero while it is fairly large and varies too much for LASSO-SCNLS and EN-SCNLS. Moreover, the simplified two-step approach shows significantly better performance for excluding irrelevant regressors

although LASSO-SCNLS and EN-SCNLS select more relevant regressors. In fact, the reasons for LASSO-SCNLS and EN-SCNLS's better performance in selecting more relevant regressors is due to its lower ability for input shrinkage (e.g. for 100 observations, LASSO-SCNLS and EN-SCNLS cannot shrink any regressors). Besides substantially lower shrinkage ability we also observe a substantially larger bias in the estimate of β coefficients for LASSO-SCNLS and EN-SCNLS relative to LASSO and EN, suggesting that SCNLS does not perform well when embedded with LASSO, at least in the current form[26].

This Monte Carlo evidence suggests that LASSO+DEA and EN+DEA performs better than LASSO-SCNLS and EN-SCNLS in MSE, MAD and bias, while LASSO-SCNLS and EN-SCNLS performed better than LASSO+DEA and EN+DEA in the average selected relevant regressors, yet poorer in the average selected irrelevant regressors. However, note that LASSO+DEA is a two-step approach and LASSO-SCNLS is all-in-one with more constraints, and so the two approaches have different model structures. This difference should be kept in mind when comparing these two approaches. As a potential caveat though, such comparisons are quite common in practice. In particular, while LASSO-SCNLS selected the irrelevant regressors more frequently, there might be a way to improve it with another step or with additional constraints or other refinements to shrink away the irrelevant regressors, which might be a fruitful direction for future research[27]. Nevertheless, the simplified two-step approach appears to show better performance.

5.2.3. MC simulations of LASSO+DEA and EN+DEA for big wide data

In this section, we will show the performance of the LASSO+DEA and EN+DEA for the contexts of big wide data using a same DGP setting in section 5.2.2. In the following part, we consider four cases of MC simulations: (Case 1) $n = 100, p = 1000$; (Case 2)

[26] We thank anonymous referee for pointing this out.

[27] We thank anonymous referee for pointing this out.

$n = 1000, p = 1000$; (Case 3) $n = 1000, p = 5000$; (Case 4) $n = 5000, p = 5000$. Note that we consider 200 replications in all the four cases. Table 4 reports the MC results.

Table 4. MC results of LASSO+DEA and EN+DEA

	Case 1: $n = 100$, $p = 1000$		Case 2: $n = 1000$, $p = 1000$		Case 3: $n = 1000$, $p = 5000$		Case 4: $n = 5000$, $p = 5000$	
Indicators	LASSO+ DEA	EN+DEA	LASSO+ DEA	EN+DEA	LASSO+ DEA	EN+DEA	LASSO+ DEA	EN+DEA
$\gamma = 1$								
Ave. MSE	0.052	0.057	0.033	0.051	0.036	0.058	0.010	0.014
s. e.	0.003	0.003	0.002	0.002	0.002	0.002	0.000	0.001
Ave. MAD	0.170	0.179	0.132	0.169	0.136	0.182	0.068	0.079
s. e.	0.005	0.005	0.003	0.004	0.004	0.004	0.001	0.002
Ave. bias	-0.088	-0.095	-0.131	-0.168	-0.135	-0.182	-0.068	-0.079
s. e.	0.010	0.010	0.003	0.004	0.004	0.004	0.001	0.002
Ave. Pearson	0.576	0.531	0.791	0.683	0.772	0.632	0.925	0.900
s. e.	0.018	0.019	0.009	0.014	0.012	0.016	0.002	0.005
Ave. Spearman	0.581	0.546	0.785	0.692	0.770	0.646	0.916	0.890
s. e.	0.016	0.016	0.008	0.011	0.010	0.013	0.002	0.005
Ave. Kendall	0.432	0.402	0.639	0.540	0.624	0.496	0.813	0.779
s. e.	0.013	0.013	0.009	0.011	0.010	0.012	0.003	0.006
Ave. input	6.130	7.170	6.000	8.410	6.430	9.640	4.835	5.525
s. e.	0.586	0.697	0.267	0.369	0.305	0.457	0.068	0.135
Ave. selected relevant regressors	1.835	1.685	4.450	4.515	4.395	4.445	4.665	4.695
s. e.	0.065	0.062	0.046	0.044	0.046	0.044	0.039	0.038
Ave. selected irrelevant regressors	4.295	5.485	1.550	3.895	2.035	5.195	0.170	0.830
s. e.	0.549	0.657	0.254	0.359	0.294	0.448	0.053	0.122
$\gamma = 0.5$								
Ave. MSE	0.023	0.035	0.017	0.023	0.016	0.024	0.011	0.012
s. e.	0.002	0.003	0.001	0.001	0.001	0.002	0.000	0.000
Ave. MAD	0.103	0.127	0.086	0.103	0.083	0.103	0.064	0.069
s. e.	0.004	0.005	0.003	0.003	0.003	0.004	0.001	0.001
Ave. bias	-0.055	-0.088	-0.070	-0.090	-0.065	-0.087	-0.026	-0.032
s. e.	0.006	0.007	0.003	0.004	0.003	0.005	0.001	0.001
Ave. Pearson	0.782	0.707	0.865	0.824	0.864	0.814	0.880	0.870
s. e.	0.016	0.018	0.006	0.009	0.008	0.012	0.002	0.003
Ave. Spearman	0.767	0.686	0.853	0.812	0.854	0.807	0.885	0.874
s. e.	0.015	0.017	0.006	0.009	0.007	0.010	0.002	0.003
Ave. Kendall	0.608	0.538	0.718	0.673	0.718	0.668	0.776	0.763
s. e.	0.013	0.015	0.007	0.009	0.008	0.010	0.002	0.004
Ave. input	3.025	4.255	4.165	5.080	4.040	5.090	4.215	4.545
s. e.	0.505	0.483	0.206	0.251	0.198	0.256	0.060	0.086

Ave. selected relevant regressors	1.155	1.195	3.535	3.685	3.340	3.470	4.175	4.315
s. e.	0.028	0.033	0.058	0.057	0.056	0.058	0.052	0.049
Ave. selected irrelevant regressors	1.870	3.060	0.630	1.395	0.700	1.620	0.040	0.230
s. e.	0.492	0.470	0.185	0.230	0.174	0.230	0.026	0.062

As can be seen in Table 4, for both cases of returns to scale, the performance indicators (average MSE, MAD, bias, Pearson, Spearman, Kendall, the number of selected irrelevant regressors) of LASSO+DEA are slightly better than those of EN+DEA for the above four cases from Case 1 to Case 4, although the average number of selected relevant regressors of LASSO+DEA is slightly smaller than that of EN+DEA in most cases. In Case 4, the performance difference between LASSO+DEA and EN+DEA is small.

In the above four cases, the performance of LASSO+DEA and EN+DEA is the best in Case 4. As the sample size gets larger, the performance of LASSO+DEA and EN+DEA generally gets better. Namely, the average MSE, MAD and bias reduce and the average Pearson, Spearman and Kendall increase. Most importantly, the average number of selected relevant regressors increases to nearly 5, and the average number of selected irrelevant regressors almost decreases to 0. It implies that the proposed approach identifies most of the true regressors and discards most of the irrelevant regressors.

Overall, the simplified two-step approach seems to show good performance for small data, and even better performance for big wide data.

6. Conclusions

This study revisits the LASSO variable selection or dimension reduction in DEA by Lee and Cai (2020). Using the same idea, we also adapted the EN approach (known to perform better for correlated data) to the SCNLS. Unlike previous results, our Monte Carlo simulations suggest that none of the considered approaches generally dominate the others in the scenarios considered by Lee and Cai (2020). The degree of

complication of these methods appears not to be justified by their capabilities, even for the simple scenarios that we considered. This encouraged us to consider a simplified two-step approach for addressing the curse of dimensionality in DEA. Namely, we suggest using standard LASSO or its extensions (like EN or other variants) at the first stage to identify the relevant inputs and then, at the second stage, use the desired DEA approach on the relevant inputs. Our Monte Carlo simulations suggest this approach can indeed be not just much simpler but also much more useful for addressing the big wide data context where dimensions are very large.

The idea of this paper is to propose promising innovations to DEA, by adapting some discoveries from statistical and Machine Learning literatures. Obviously, this is just the beginning and there is still a lot of work to do and in many directions. One of such directions is to develop statistical theory, including asymptotic theory for such an approach or its variations. While developing such theory may require years or even decades (and would require a separate paper(s) for a statistical journal), we believe the proposed approach can still be useful as an exploratory tool even before such technical papers are completed. A good example of such step-wise approach in research is, in fact, the DEA method itself—it was proposed and became very popular in both theory and practice decades before its statistical properties were established. Other fruitful directions for future research may include an extension of this approach to take into account the network structure of production processes, encompassing various interesting developments in Network DEA.

Moreover, this paper extends the LASSO-based DEA approach to encompass its generalized version such as EN. How to select the value of δ is an additional and separate issue, although Zou and Hastie (2005) provided a solution for the choice of the two tuning parameters in the EN. The two tuning parameters (λ and δ) in model (8) may potentially have some 'interaction effect' in the process of variable selection, which may depend on the design of the scenarios or real data or the sampling design in practice (i.e., depend on the data generating process). Considering different

approaches for such selections and investigating their performance is a challenging computational task in itself that requires a separate paper and so we leave it for future research.

Recently, DEA has also been applied to streaming data for identifying outliers (Dulá and López, 2013) and a future direction of research can be to synthesize such approaches with our simplified approach. It is also worth noting that for high-dimensional data streams in big data context, outlier detection may fail as data tends to become equally distant from each other and such approaches as Zhang et al. (2009) can also be explored for adaptation to the DEA context.

Finally, in the era of big data, it becomes challenging to deal not only with a large number of observations but a large number of inputs or environmental (contextual) variables as well. Efficient ways to deal with these problems remain open in the agenda and are likely to use novel techniques, for example, Bayesian compression (Guhaniyogi and Dunson, 2015) that accommodates uncertainty in the subspace reduction and exhibits a near parametric rate of convergence of the predictive distribution in the 'large p, small n' case. Moreover, an adaptation of the so-called two-stage-DEA (typically done via truncated regression and bootstrap as proposed by Simar and Wilson (2007)) can also be adapted to the context of big wide data featuring very large dimensions of environmental variables, where a few key regressors need to be selected via LASSO. In this context, a fruitful subject in itself left for future research is to consider the issues of potential endogeneity and the questions of performing causal inference, e.g. by adapting ideas from Belloni et al. (2012, 2013) and Athey (2015), to mention a few. All in all, we hope our paper will encourage more research on these interesting questions that are important for both theorists and practitioners.

Acknowledgements

Ya Chen thanks and acknowledges the financial support from National Natural Science Foundation of China (Nos. 71601064, 72071067 and 71871081), and the Major Project of the National Social Science Foundation of China (No. 18ZDA064). Ya Chen

also thanks Jianhui Xie, and seminar and conference participants at The University of Queensland, the 2019 International Conference on Data Envelopment Analysis and Data Analytics, and the First Lecture Series of Wenlan Economic Measurement and Applied Econometrics in Wenlan School of Business at Zhongnan University of Economics and Law for valuable discussions and comments. Valentin Zelenyuk thanks and acknowledges the financial support from the Australian Research Council (from the ARC Future Fellowship grant FT170100401) and from The University of Queensland. We also thank Bao Hoang Nguyen and Evelyn Smart for proofreading and their feedback and Zhichao Wang for the checks of the coding and replications of the computations. These individuals and organizations are not responsible for the views expressed in this paper.

References

Athey, S. (2015). Machine learning and causal inference for policy evaluation. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 5–6.

Athey, S., Imbens, G. W. (2019). Machine learning methods economists should know about. Graduate School of Business, Stanford University, Working Paper No. 3776.

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. Management Science, 30(9), 1078–1092.

Belloni, A., Chen, D., Chernozhukov, V., & Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. Econometrica, 80(6), 2369–2429.

Belloni, A.; Chernozhukov, V., & Hansen, C. (2013). Inference on Treatment Effects after Selection Among High-Dimensional Controls. The Review of Economic Studies, 81(2), 608–650.

Bertsimas, D., King, A., & Mazumder, R. (2016). Best subset selection via a modern optimization lens. The Annals of Statistics, 44(2), 813–852.

Bühlmann, P., & van de Geer, S. (2011). Statistics for High-dimensional Data: Methods, Theory and Applications. New York: Springer.

Charles, V., Aparicio, J., & Zhu, J. (2019). The curse of dimensionality of decision-making units: A simple approach to increase the discriminatory power of data envelopment analysis, European Journal of Operational Research, 279(3), 929–940.

Charnes, A., Cooper, W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. European Journal of Operational Research, 2(6), 429–444.

Cook, W., & Seiford, L. (2009). Data envelopment analysis (DEA) - thirty years on. European Journal of Operational Research, 192(1), 1–17.

Debreu, G. (1951). The coefficient of resource utilization. Econometrica, 19(3), 273–292.

Dulá, J.H., & López, F. J. (2013). DEA with streaming data. Omega, 41(1), 41–47.

Emrouznejad, A. & Yang, G. L. (2018). A survey and analysis of the first 40 years of scholarly literature in DEA: 1978–2016, Socio-Economic Planning Sciences, 61, 4–8.

Farrell, M. J. (1957). The measurement of productive efficiency. Journal of the Royal Statistical Society. Series A (General), 120(3), 253–290.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso, arXiv:1001.0736.

Foster, M. (1961). An Application of the Wiener-Kolmogorov Smoothing Theory to Matrix Inversion. Journal of the Society for Industrial and Applied Mathematics. 9(3), 387–392.

Guhaniyogi, R., & Dunson, D. B. (2015). Bayesian compressed regression. Journal of the American Statistical Association, 110(512), 1500–1514.

Hastie, T., Tibshirani R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Second Edition. Springer. 2nd edition.

Hastie, T., Tibshirani R., & Tibshirani, R. J. (2017). Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso. arXiv:1707.08692.
Hoerl, A. E. (1962). Application of Ridge Analysis to Regression Problems. Chemical Engineering Progress, 58(3), 54–59.
Keshvari, A., & Kuosmanen, T. (2013). Stochastic non-convex envelopment of data: Applying isotonic regression to frontier estimation. European Journal of Operational Research, 231, 481–491.
Khezrimotlagh, D., Zhu, J., Cook, W., & Toloo, M. (2019). Data Envelopment Analysis and Big Data, European Journal of Operational Research, 274(3), 1047–1054.
Kitchin, R. and McArdle, G. (2016). What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets. Big Data & Society, 1–10.
Koopmans, T. C. (1951a). Activity Analysis of Production and Allocation. New York, NY: Wiley.
Koopmans, T. C. (1951b). Analysis of production as an efficient combination of activities. Activity Analysis of Production and Allocation, 13:33–97.
Kuosmanen, T. (2006). Stochastic nonparametric envelopment of data: Combining virtues of SFA and DEA in a unified framework. MTT Discussion papers No. 3/2006, Helsinki, Finland.
Kuosmanen, T., & Johnson, A. L. (2010). Data envelopment analysis as nonparametric least-squares regression. Operations Research, 58(1), 149–60.
Kuosmanen, T., & Kortelainen, M. (2012). Stochastic non-smooth envelopment of data: semi-parametric frontier estimation subject to shape constraints. Journal of Productivity Analysis, 38, 11–28.
Kuosmanen, T., & Johnson, A. L. (2017). Modeling joint production of multiple outputs in StoNED: Directional distance function approach. European Journal of Operational Research, 262, 792–801.
Lai, K., Twine, N., O'Brien, A., Guo, Y., Bauer, D. (2019). *Artificial intelligence and machine learning in bioinformatics*. In S. Raganathan, M. Gribskov, K. Nakai. C. Schonbach (Eds.), Encyclopedia of Bioinformatics and Computational Biology, (pp. 272-286). United States of America: Elsevier.
Lee, C.-Y., & Cai, J.-Y. (2020). LASSO variable selection in data envelopment analysis with small datasets, Omega, 91, 102019.
Liu, J. S., Lu, L. Y. Y., & Lu, W-M (2016). Research fronts in data envelopment analysis. Omega, 58, 33–45.
Liu, J. S., Lu, L. Y. Y., Lu, W-M, & Lin, B. J. Y. (2013). Data envelopment analysis: 1978–2010: a citation-based literature survey. Omega, 41, 3–15.
Marsaglia, G. (1972). Choosing a point from the surface of a sphere. Annals of Mathematical Statistics, 43, 645–646.
Meier, L., van de Geer, S., & Bühlmann, P. (2008). The group lasso for logistic regression. Journal of the Royal Statistical Society: Series B (Methodological), 70(1), 53–71.

Meinshausen, N. (2007). 'Relaxed lasso', Computational Statistics & Data Analysis, 52, 374–393.
Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. The Annals of Statistics, 34, 1436–1462.
Misiunas, N., Oztekin, A., Chen, Y., & Chandra, K. (2016). DEANN: A healthcare analytic methodology of data envelopment analysis and artificial neural networks for the prediction of organ recipient functional status. Omega, 58, 46–54.
Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. Journal of Economic Perspectives, 31(2), 87–106.
Muller, M. E. (1959). A note on a method for generating points uniformly on n-dimensional spheres. Communications of the Association for Computing Machinery, 2, 19–20.
Seijo, E., & Sen, B. (2011). Nonparametric least squares estimation of a multivariate convex regression function. The Annals of Statistics, 39(3), 1633–1657.
Sherman, H.D., & Zhu, J. (2013). Analyzing performance in service organizations, Sloan Management Review, 54, 4, 36-42.
Simar, L. & Wilson, P. W. (2007). Estimation and inference in two-stage, semi-parametric models of production processes. Journal of Econometrics, 136(1), 31–64.

Simar, L., & Zelenyuk, V. (2011). Stochastic FDH/DEA estimators for frontier analysis. Journal of Productivity Analysis, 36, 1–20.
Thomas Erl, Wajid Khattak, Paul Buhler (2016). *Big Data Fundamentals: Concepts, Drivers & Techniques*. Prentice Hall Press, NJ, United States.
Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288.
Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. Journal of the Royal Statistical Society, Series B (Methodological), 73(3), 273–282.
Tikhonov, A. N. (1943). "Об устойчивости обратных задач" [On the stability of inverse problems]. Doklady Akademii Nauk SSSR. 39(5), 195–198.
Tsionas, M. G., & Izzeldin, M. (2018). Smooth approximations to monotone concave functions in production analysis: An alternative to nonparametric concave least squares. European Journal of Operational Research, 271(3), 797–807.
Wilson, P. W. (2018). Dimension reduction in nonparametric models of production. European Journal of Operational Research, 267(1), 349–367.
Yuan M., & Lin Y. (2006). Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B (Methodological), 68(Part 1), 49–67.
Zelenyuk, V. (2020). Aggregation of inputs and outputs prior to data envelopment analysis under big data, European Journal of Operational Research, 282 (11), 172–187.
Zhang, J., Gao, Q. G., Wang, H., Liu, Q., & Xu, K. (2009). Detecting Projected Outliers in High-dimensional Data Streams. International Conference on Database and Expert Systems Applications, 629–644.

Zhu, J. (2020). DEA under big data: Data enabled analytics and network data envelopment analysis. Annals of Operations Research, https://doi.org/10.1007/s10479-020-03668-8.
Zhu, Q., Wu, J., Song, M. (2018). Efficiency evaluation based on data envelopment analysis in the big data context. Computers & Operations Research, 98, 291–300.
Zou, H. (2006). The adaptive lasso and its oracle properties. Journal of the American Statistical Association, 101, 1418–1429.
Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. Journal of the Royal Statistical Society, Series B (Methodological), 301–320.

Appendixes

Appendix A. The DGP setting in Lee and Cai (2020)

Lee and Cai (2020) considered three cases of Monte Carlo simulations for numerical comparison:

(1) Case 1: 25 observations with 10 dimensions (9 inputs and 1 output);

(2) Case 2: 100 observations with 10 dimensions (9 inputs and 1 output);

(3) Case 3: 25 observations with 11 dimensions (9 inputs and 2 outputs).

Cases 1 and 2 are for a single-output scenario while case 3 is for a two-output scenario. In each case, Lee and Cai (2020) replicated the DGP 30 times (i.e. 30 Monte Carlo replications) for calculating AMSE. Below we briefly describe the DGP for Monte Carlo simulations with a single output in Lee and Cai (2020).

Assume that the values of inputs are independently and identically distributed (i.i.d.) and generated from a uniform distribution of the interval (10, 20), i.e. $x_{ijm} \sim U(10,\ 20)$. The inefficiency term is $\epsilon_{im} \sim N_+(0, \sigma^2)$, where $\sigma = 0.7$. Consider a Cobb-Douglas (CD) production function with multiple inputs and a single output y_{im}^T as a true "smooth" frontier. In particular, the true output y_{im}^T and the observed output y_{im} are calculated by the following equations, respectively:

$$y_{im}^T = \prod_j x_{ijm}^{\left(\frac{1}{p+1}\right)},\ \forall i = 1, \ldots n, j = 1, \ldots, p, m = 1, \ldots, M$$

$$y_{im} = \prod_j x_{ijm}^{\left(\frac{1}{p+1}\right)} \times e^{-\epsilon_{im}},\ \forall i = 1, \ldots n, j = 1, \ldots, p, m = 1, \ldots, M \quad \text{(A.1)}$$

The methods described in Section 2 are used for dimension reduction. Then the output-oriented BCC model is used to calculate an observation or DMU's efficiency

θ_{im}^{*}. Run the DGP M times, and calculate the AMSE from M replications using equations (9a) and (10).

Appendix B. Performance comparison with more scenarios

B.1. Results based on more general DGP

Here we further analyze the performance of four different methods using more general DGPs. We consider two generalizations of DGP with a single output used in Lee and Cai (2020):

(1) DGP 1: more general production function. Following Zelenyuk (2020), we assume the power κ ($\frac{1}{p+1}$ in the Appendix A) of inputs is a random variable, i.e. $\kappa \sim U(0,\ 1)$;

(2) DGP 2: more general production function and distribution of inefficiency. Except for the assumption in DGP 1, following Simar and Zelenyuk (2011), we assume that the standard deviation of the inefficiency term depends on the inputs, i.e. $\epsilon_{im} \sim N_{+}\left(0, \sigma_{\mu}^{2}\right)$, where we set $\sigma_{\mu} = (x_1 + x_2 + \cdots + x_p)/180$. Namely, the standard deviation varies in an interval [0.5, 1].

Due to space limitations, here we only report the results for DGP 2 with 200 replications, which are shown in Figure B1.

As shown in Figure B1, there is still no clear dominance for the four methods. The random method shows a worse performance for small dimensions while it shows a better performance for the results in Figure 1. Meanwhile, LASSO here appeared to be better than LASSO-SCNLS for small dimensions.

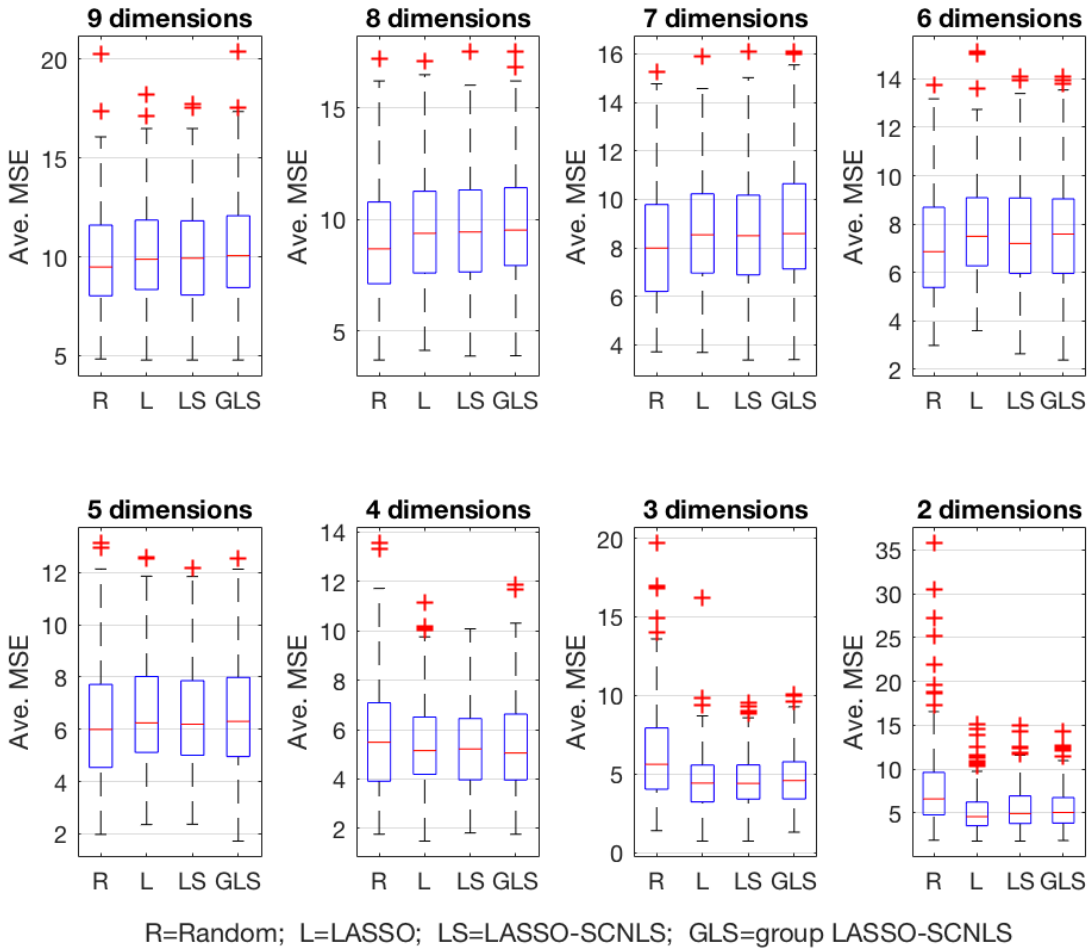

Figure B1. Results with 200 replications for DGP 2

B.2. MSE results of efficiency scores

Considering the average MSE are small and almost the same[28] in Table 6 of Lee and Cai (2020), the scale transformation used in Lee and Cai (2020) may play an important role. As a result, instead of using the average MSE of outputs in Lee and Cai (2020), it may be better to compare the average MSE of efficiency scores (equation (9b) in Section 5), as is often done in other studies (e.g., Wilson, 2018; Zelenyuk, 2020). In this section, we further compare the results of different methods based on the MSE of efficiency scores for DGP 2. We consider 200 replications and the results are shown

[28] We also re-run the simulations by using the same DGP in their paper as well as more general DGPs (our DGP 1 and DGP 2), and get similar results.

in Figure B2. We also find that there is no clear dominance of a certain method and no statistically significant difference among the methods.

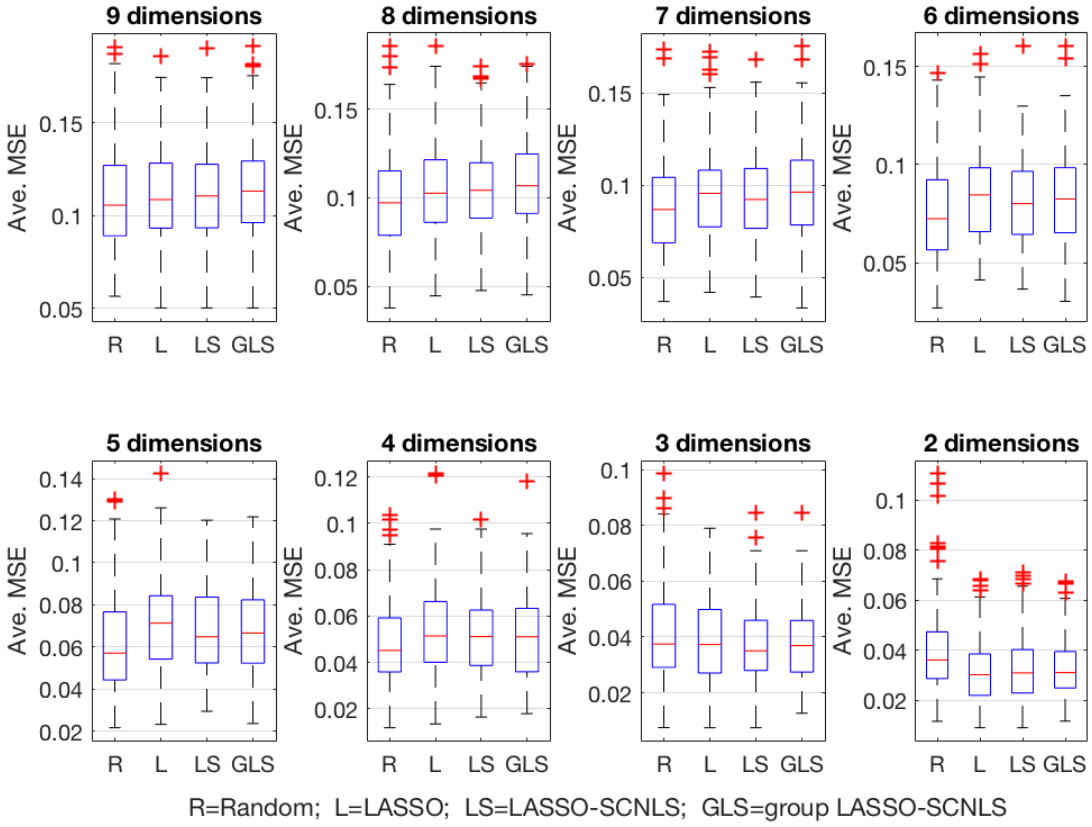

Figure B2. Results with 200 replications for MSE of efficiency scores

B.3. Results based on the DGP *in Wilson (2018)*

In order to show the performance of dimension reduction by EN-SCNLS and further compare the performance based on the introduced methods above, some Monte Carlo simulations are provided here. We consider four simulations based on the experiments used in Wilson (2018).

Particularly, p inputs and q outputs are generated as follows: Using the method in Muller (1959) and Marsaglia (1972) to generate $(p+q)$-tuples $u = [u_p, u_q]$ uniformly distributed on the surface of a closed $(p+q)$-ball with radius one centered at the origin. Here u_p and u_q are column vectors of lengths p and q, respectively. Set $x_{ijm} = \theta_{im}^{-1}(1 - |u_p|)$ and $y_{im} = |u_q|$ where $\theta_{im} = (1+\epsilon_{im})^{-1}$,

ϵ_{im} is drawn from $N_+(0, \sigma_\epsilon^2)$, and $\sigma_\epsilon = 0.25\sqrt{\pi/2}$ so that $E(\theta_{im}^{-1}) = 1.25$ and $E(\theta_{im}) \approx 0.8169$. Similarly, we can consider the case with $\sigma_\epsilon = 0.75\sqrt{\pi/2}$ as shown in Wilson (2018). Note that the inefficiency term θ_{im} is used to obtain observed inputs (the true input is divided by the inefficiency term). Similarly, we can use the inefficiency term to get observed outputs, and set $x_{ijm} = |u_p|$ and $y_{im} = \theta_{im}(1 - |u_q|)$. Then, we have four cases of simulations as follows:

- Case A: Output-oriented with $\sigma_\epsilon = 0.25\sqrt{\pi/2}$
- Case B: Output-oriented with $\sigma_\epsilon = 0.75\sqrt{\pi/2}$
- Case C: Input-oriented with $\sigma_\epsilon = 0.25\sqrt{\pi/2}$
- Case D: Input-oriented with $\sigma_\epsilon = 0.75\sqrt{\pi/2}$

We consider nine inputs and two outputs with 200 replications in our Monte Carlo simulations. And the outputs are aggregated into one output using a weighted Euclidian function used in (Zelenyuk, 2020) before dimension reduction. Similar to Section B.2, we focus on the comparison of efficiency scores. Due to space limitations, we only show the graphical results for the average MSE comparison, which are shown in Figures B3-B6. As can be seen in the figures below, the results based on different methods are similar. There is only a small average MSE difference for them.

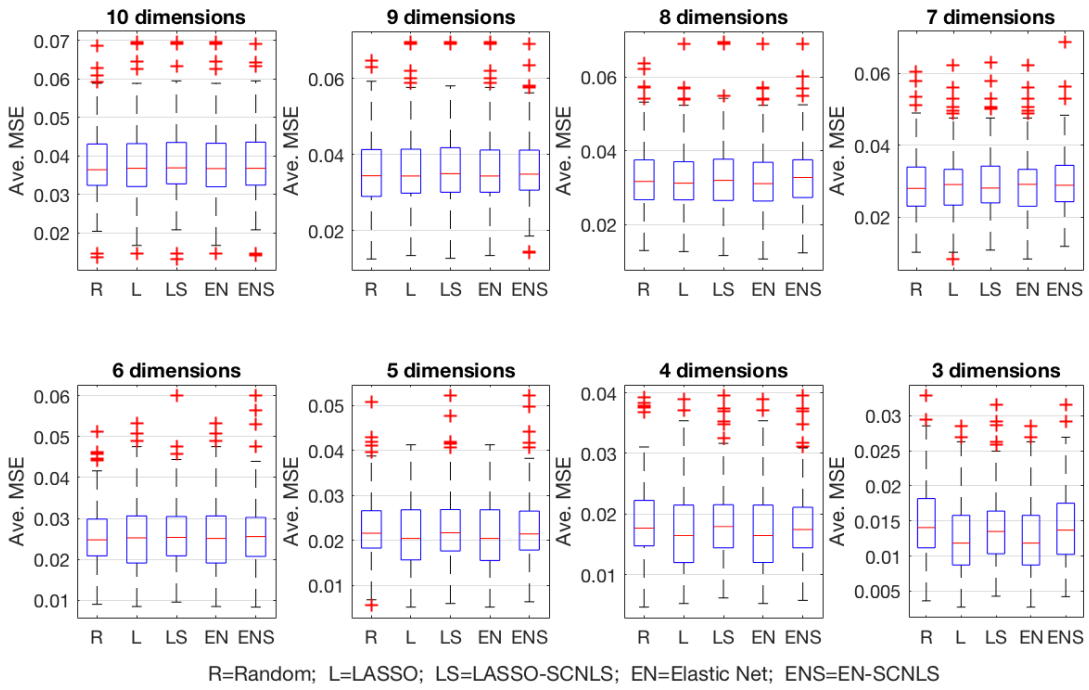

Figure B3. AMSE with 200 replications for Case A

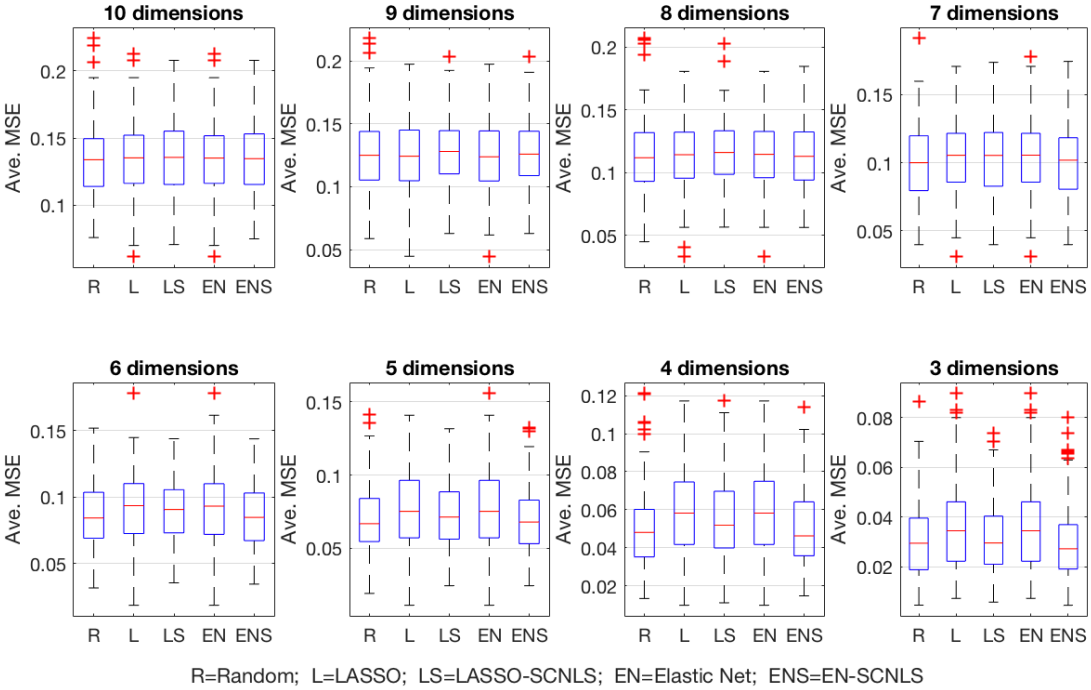

Figure B4. AMSE with 200 replications for Case B

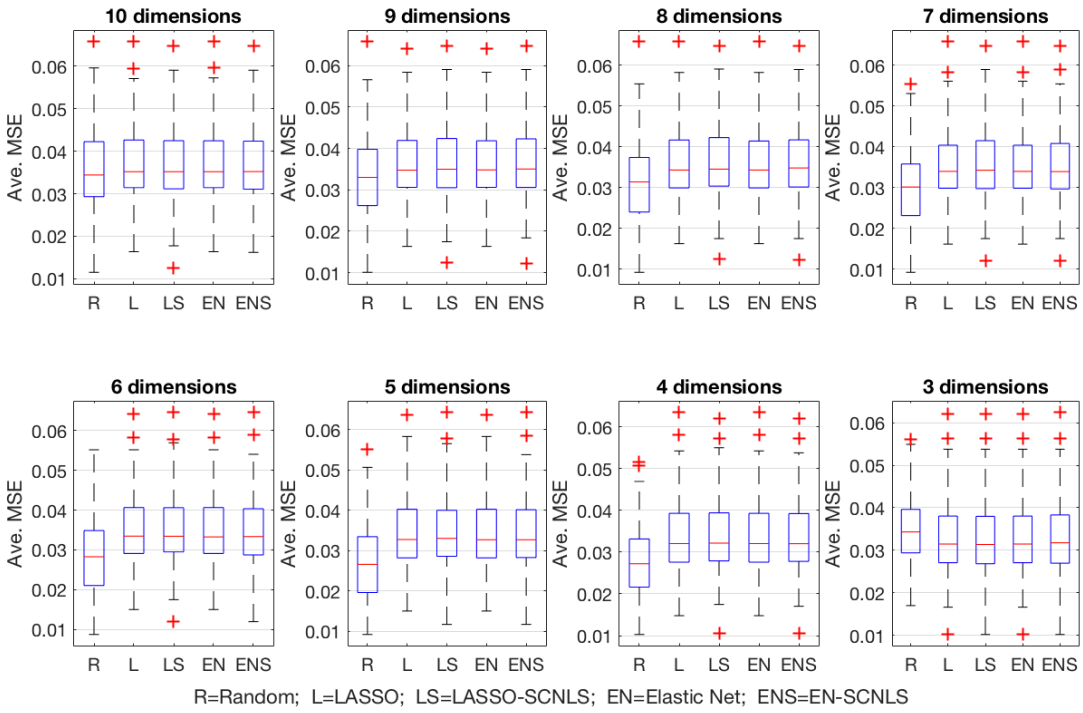

Figure B5. AMSE with 200 replications for Case C

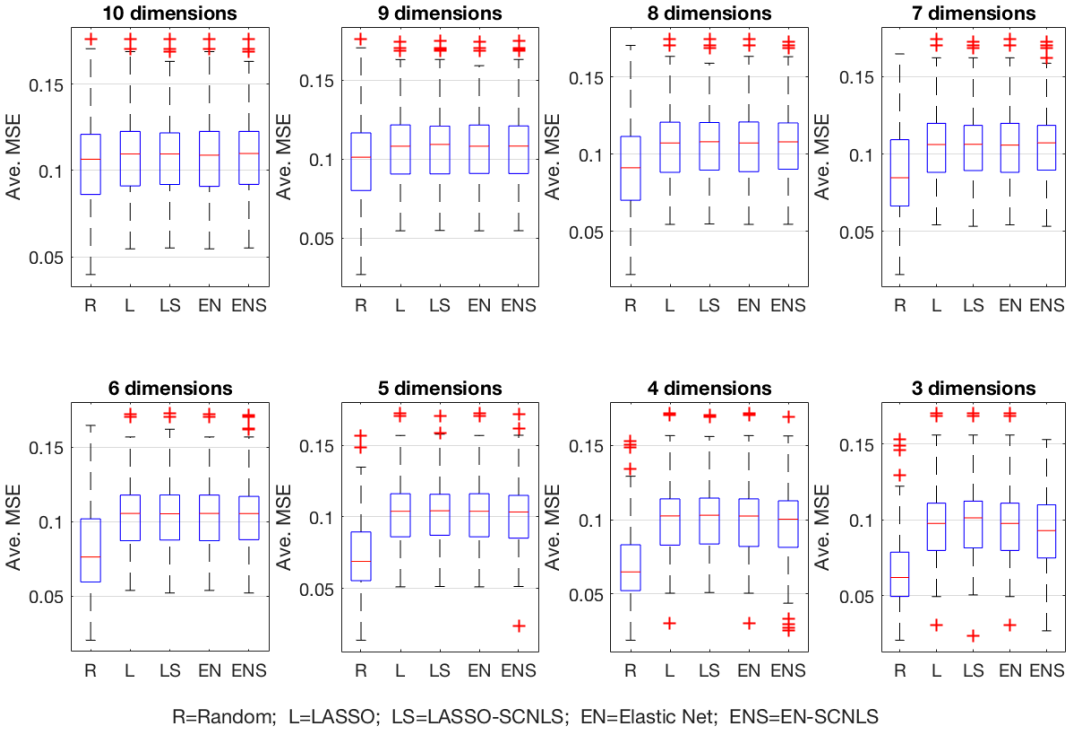

Figure B6. AMSE with 200 replications for Case D

Appendix C. More scenarios for shrinkage performance of cross-validation LASSO

C.1. More replications for cross-validation LASSO

We have run results for 1000 replications using the same DGP setting in Section 5.2.1. Here we only show the results of the first five replications (Rep. 1 to Rep. 5) in Table C1. As shown in Table C1, the cross-validation LASSO identifies all the true regressors and discards all the irrelevant regressors. In the last column of Table C1, we report the average coefficient for all regressors over 1000 replications. And the average estimated coefficients for the relevant regressors are very close to the corresponding true coefficients.

Table C1. Performance comparison for four cases

Inputs	β_j^T	Cross-validation LASSO					
		Rep. 1	Rep. 2	Rep. 3	Rep. 4	Rep. 5	Average
1	1	0.781	0.824	0.815	0.884	0.848	0.830
2	2	1.934	2.038	1.942	1.938	1.992	1.950
3	3	3.049	2.899	2.952	2.912	2.920	2.951
4	4	3.922	3.977	3.950	3.984	3.968	3.950
5	5	4.802	4.818	4.842	4.833	4.825	4.831
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0

34	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0

C.2. More cases of noise

In this section we consider six different cases of noise relative to inefficiency. All the other parameters are the same with the setting in Section 5.2.1, except for the δ_v. And the results are shown in Table C2. In all six cases, the cross-validation LASSO identifies all the true regressors but also discards all irrelevant regressors.

Table C2. Estimated coefficients for different noises

Inputs	β_j^T	Cross-validation LASSO					
		δ_v=0.001	δ_v=0.005	δ_v=0.01	δ_v=0.05	δ_v=0.1	δ_v=0.5
1	1	0.787	0.814	0.838	0.853	0.839	0.877
2	2	1.926	1.966	1.968	1.920	1.915	1.927
3	3	3.050	2.951	2.946	2.967	2.945	3.002
4	4	3.929	3.923	3.907	3.964	4.023	3.967
5	5	4.802	4.849	4.884	4.809	4.787	4.819
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0

23	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0

C.3. More cases of SNR

Moreover, we also consider different cases of SNR. Similarly, all the other parameters are the same with the setting in Section 5.2.1, except for the SNR. And the results are shown in Table C3. As shown in Table C3, the estimated coefficients of irrelevant regressors are close to their corresponding true coefficients. Note that irrelevant regressors may be selected although the number of the selected irrelevant regressors is very small.

Table C3. Estimated coefficients for different SNRs

Inputs	β_j^T	Cross-validation LASSO					
		SNR=0.05	SNR=0.14	SNR=0.42	SNR= 1.22	SNR=3.52	SNR=6.00
1	1	0.730	0.781	0.970	0.888	0.840	0.864
2	2	2.019	2.148	1.896	1.951	1.907	1.931
3	3	3.018	2.889	2.932	2.955	2.946	2.987
4	4	3.833	4.138	3.970	3.974	4.028	3.945
5	5	4.804	4.803	4.889	4.816	4.787	4.834
6	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0

C.4. Linearity and non-linearity

Except for the DGP setting in Section 5.2.1, we also consider the following cases:

I) $x_{ijm} \sim U(1,\ 20)$: $\delta_u = 2;\ \delta_v = 0.5;$

II) $x_{ijm} \sim U(1, 20)$: $\delta_u = 2$; $\delta_v = 1$;

III) $x_{ijm} \sim U(1, 20)$: $\delta_u = 2$; $\delta_v = 2$;

IV) $y_{im} = \sum_{j=1}^{5} x_{ijm} + 0.2 \times x_{i1m}^2$;

V) $y_{im} = \sum_{j=1}^{5} x_{ijm} + x_{i1m}^2$;

VI) $y_{im} = \sum_{j=1}^{5} x_{ijm} + 2 \times x_{i1m}^2$.

Note that in each case all the other settings are the same with Section 5.2.1 if not explicitly declared. In case I, II and III, except for the distribution of x_{ijm}, the value of δ_u and δ_v, all the other parameter settings are the same with Section 5.2.1. In case IV, V and VI[29], the distribution of x_{ijm} and other parameters are the same with Section 5.2.1, except for the non-linear assumption.

Comparing case I with case II and III, case III and II show more noise than case I. Case IV is slightly non-linear while case V and VI are more non-linear. And the results of cross-validation LASSO are shown in Tables C4, respectively.

Table C4. Estimated coefficients for four cases

Inputs	β_j^T	Cross-validation LASSO					
		I	II	III	IV	V	VI
1	1	0.911	0.931	0.931	0.782	0.801	0.630
2	2	1.907	1.930	1.927	1.915	1.847	1.695
3	3	2.892	2.914	2.901	3.092	3.263	3.522
4	4	3.912	3.935	3.937	3.835	3.491	2.996
5	5	4.905	4.935	4.940	4.848	5.051	5.125
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	-0.020	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0

[29] We take the first five inputs as relevant regressors for convenience as the order of the inputs is not important (due to the randomness of the simulated data).

17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	-0.006	-0.052	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0.004	0	0	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0
32	0	0	0	0.000	0	0	0
33	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0
43	0	0	0	0.031	0	0	0
44	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0
49	0	0	-0.003	-0.035	0	0	0
50	0	0	0	0	0	0	0

For case I, the cross-validation LASSO identifies the first five true inputs or regressors and discards all the other unnecessary regressors. The estimated beta coefficients are close to the given (true) beta coefficients. For case II, the cross-validation LASSO also identifies five true regressors. However, two unnecessary regressors are also identified. For case III, the cross-validation LASSO also identifies five true regressors but more (six) unnecessary regressors are identified. Hence, if the magnitude of the noise increases, the accuracy of the cross-validation LASSO discarding the unnecessary regressors decreases.

For case IV, the cross-validation LASSO still identifies the first five true regressors, and discards all the other unnecessary regressors. The estimated beta coefficients are also close to the given (true) beta coefficients. Hence, if the true output is non-linear but is not too far from linear, the cross-validation LASSO could still work reasonably well. For case V and VI, the cross-validation LASSO still identifies all five true regressors. However, it is possible that the cross-validation LASSO fails to identify all true regressors in case V and VI (as shown in the table below). Moreover, the estimated coefficients of the true regressors deviate from the corresponding true coefficients in cases V and VI. Hence, the further from the linearity, the less accurate the cross-validation LASSO should be expected to perform, yet there are apparently no better alternatives so far and a call for further research on this is in order.

Table C5. Results for performance statistics

	I	II	III	IV	V	VI
Ave. selected relevant regressors	5	5	5	5	4.995	4.836
Ave. selected irrelevant regressors	0.003	0.302	2.605	0.001	0.153	0.148
The number of correctness of regressor selection/discard	997	790	211	999	879	728
Ave. estimated β_{1m}	0.90	0.93	0.92	0.84	0.76	0.56
Ave. estimated β_{2m}	1.90	1.93	1.92	1.95	1.92	1.81
Ave. estimated β_{3m}	2.90	2.92	2.91	2.95	2.93	2.86
Ave. estimated β_{4m}	3.90	3.93	3.92	3.95	3.93	3.85
Ave. estimated β_{5m}	4.90	4.93	4.92	4.84	4.75	4.49

In fact, we try each case in Table C4 for 1000 replications. And the results for the performance statistics are shown in Table C5. In case I to IV, as shown in the first row of Table C5, in all the 1000 replications the cross-validation LASSO selected all five true regressors. However, in case I to IV the cross-validation LASSO may select irrelevant regressors. In case I, as shown in the third row of Table C5, in 997 replications the cross-validation LASSO selected all five true regressors and discarded all irrelevant regressors while the number decreases to 790 in case II. And as shown in the second row of Table C5, on average (over the 1000 replications) about 0.003 and 0.302 of 45

irrelevant regressors were selected in case I and II, respectively. In case III, only in 211 replications did the cross-validation LASSO select all five true regressors and discard all irrelevant regressors. While the method is not perfect and in some peculiar replications the number of selected irrelevant variables was relatively large, on average about 2.605 of 45 irrelevant regressors were selected. In case I to III, as shown in the last five rows of Table C5, the average estimated coefficients of the first five regressors were almost the same and close to their corresponding true coefficients.

In case IV, the cross-validation LASSO performed quite well. It selected all five true regressors and discarded all irrelevant regressors in 999 out of 1000 replications. In case V and VI, if the true output goes further away from the linearity, the performance decreases. Over the 1000 replications, on average about 4.836 of 5 relevant regressors were selected, and about 0.148 of 45 irrelevant regressors were selected in case VI. Moreover, from case IV to VI, the average estimated coefficients of the first five regressors decrease and gradually deviate from their corresponding true coefficients.