European Journal of Operational Research xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Interfaces with Other Disciplines

A more efficient algorithm for Convex Nonparametric Least Squares

Chia-Yen Lee [a], Andrew L. Johnson [b,*], Erick Moreno-Centeno [b,1], Timo Kuosmanen [c]

[a] Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan City 701, Taiwan
[b] Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77840, USA
[c] School of Business, Aalto University, 00101 Helsinki, Finland

ARTICLE INFO

Article history:
Received 7 March 2012
Accepted 26 November 2012
Available online xxxx

Keywords:
Convex Nonparametric Least Squares
Frontier estimation
Productive efficiency analysis
Model reduction
Computational complexity

ABSTRACT

Convex Nonparametric Least Squares (CNLSs) is a nonparametric regression method that does not require *a priori* specification of the functional form. The CNLS problem is solved by mathematical programming techniques; however, since the CNLS problem size grows quadratically as a function of the number of observations, standard quadratic programming (QP) and Nonlinear Programming (NLP) algorithms are inadequate for handling large samples, and the computational burdens become significant even for relatively small samples. This study proposes a generic algorithm that improves the computational performance in small samples and is able to solve problems that are currently unattainable. A Monte Carlo simulation is performed to evaluate the performance of six variants of the proposed algorithm. These experimental results indicate that the most effective variant can be identified given the sample size and the dimensionality. The computational benefits of the new algorithm are demonstrated by an empirical application that proved insurmountable for the standard QP and NLP algorithms.

1. Introduction

Convex Nonparametric Least Squares (CNLSs) is a nonparametric regression method used to estimate monotonic increasing (decreasing) and convex (concave) functions. Hildreth introduced the CNLS concept in his seminal work (1954), while Hanson and Pledger (1976) were the first to prove the statistical consistency of the CNLS estimator in the single regression case. The method has attracted attention primarily from statisticians, see Mammen (1991), Mammen and Thomas-Agnan (1999), and Groeneboom et al. (2001). Statistical properties such as consistency, Lim and Glynn (2012), Seijo and Sen (2012), and uniform convergence properties, Aguilera et al. (2012), have been shown. Functions of this type commonly arise in economics. For example, Varian (1982, 1984) describes monotonicity and convexity as standard regularity conditions in the microeconomic theory of utility and production functions.

Recently, CNLS has attracted considerable interest in the literature of productivity and efficiency analysis (Kuosmanen and Johnson, 2010), and we will focus our discussion on this domain. The two most common ways to estimate a frontier production function are Stochastic Frontier Analysis (SFA) and Data Envelopment Analysis (DEA) (e.g., Fried et al., 2008). The former one is a parametric regression method that requires a prior specification of the functional form of the frontier. The latter one is a nonparametric mathematical programming approach that avoids the functional form assumption, but also assumes away stochastic noise. Attractively, CNLS avoids the functional form assumption, building on the same axioms as DEA, but it also takes into account noise. CNLS estimates an average production function. However, CNLS can be used in a two-stage approach called Stochastic semi-Nonparametric Envelopment of Data (StoNED) to combine the main benefits of both DEA and SFA (Kuosmanen and Kortelainen, 2012). Since Kuosmanen (2008) introduced CNLS to the literature of productive efficiency analysis, several extensions to the methodology (Johnson and Kuosmanen, 2011 and Johnson and Kuosmanen, 2012; Mekaroonreung and Johnson, 2012) and empirical applications have been reported in such areas as agriculture (Kuosmanen and Kuosmanen, 2009), power generation (Mekaroonreung and Johnson, 2012), and electricity distribution (Kuosmanen, 2012). However, the computational complexity of CNLS presents a significant barrier for large-sample applications. This study focuses on this barrier and proposes a generic algorithm to reduce the computational time to solve the CNLS problem.[2]

A variety of work has been done on the computational aspects of CNLS. Hildreth (1954) developed an algorithm based on Karush–Kuhn–Tucker (KKT) conditions that can potentially take an infinite number of steps to identify the optimal solution.

* Corresponding author. Tel.: +1 979 845 9025.
E-mail addresses: cylee@mail.ncku.edu.tw (C.-Y. Lee), ajohnson@tamu.edu (A.L. Johnson), e.moreno@tamu.edu (E. Moreno-Centeno), Timo.Kuosmanen@aalto.fi (T. Kuosmanen).
[1] Tel.: +1 979 458 2335.

[2] The proposed method could potentially benefit a variety of nonparametric methods which impose shape constraints on the underlying function for example the estimator described in Du et al. (in press). Here we focus on the least squares estimation method CNLS.

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.11.054

Wilhelmsen (1976), and Pshenichny and Danilin (1978) provided algorithms projecting the data points of the dependent variable to the faces of a polyhedral cone; both algorithms converge in a finite number of iterations. Wu (1982) offered a simpler solution that also converges in a finite number of iterations. Dykstra (1983) proposed an iterative algorithm that is based on the projection onto the closed convex cones and minimizes a least squares objection function subject to concave restrictions. Goldman and Ruud (1993), and Ruud (1995) proposed an approach using a dual quadratic programming (QP) problem. They used a large number of parameters to cover all the permissible functions and obtained a smooth multivariate regression using a projection technique and structural restrictions of monotonicity and concavity. Fraser and Massam (1989) presented an algorithm to find the least square estimate of the mean in a finite number of steps by dividing the cone into subspaces. Meyer (1999) generalized Fraser and Massam's study and extended the algorithm to the case of more constraints than observations. Some related work exists applying bagging and smearing methods to convex optimization, Hannah and Dunson (2012). Recently, Kuosmanen (2008) transformed the infinite dimensional CNLS problem to a finite dimensional linearly-constrained quadratic programming problem (QP),[3] which enables one to solve the CNLS problem by using standard QP algorithms and solvers (such as CPLEX, MINOS, MOSEK). However, the number of constraints of the QP problem grows as a quadratic function of the number of observations. Standard QP algorithms are limited by the number of constraints, thus the computational burden when using quadratic programming to solve the CNLS problem is challenging even with relatively small sample sizes.

In light of the computational issues, the purpose of this study is to develop a more efficient approach to model the concavity constraints in the QP. For this purpose we use Dantzig et al.'s (1954, 1959) strategy to solve large scale problems by iteratively identifying and adding violated constraints (modern cutting plane methods for integer programming are based on this seminal work). Indeed, we show that Dantzig et al.'s strategy is not only useful when solving NP-hard problems[4] (he used the strategy to solve the travelling salesperson problem), but is also useful when solving large scale instances of problems that are solvable in polynomial time. Specifically, the underlying idea of the proposed generic algorithm is to solve a relaxed CNLS problem containing an initial set of constraints, those that are likely to be binding, and then iteratively add a subset of the violated concavity constraints until a solution that does not violate any constraint is found. In other words, the generic algorithm significantly reduces the computational cost to solve the CNLS problem by solving a sequence of QPs that contain a considerably smaller number of inequalities than the original QP formulation of the CNLS problem. Therefore, this algorithm has practical value especially in large sample applications and simulation-based methods such as bootstrapping or Monte Carlo studies.

The remainder or this paper is organized as follows, Section 2 introduce nonparametric regression, discusses the relationship between the Afriat inequalities and convex functions, and presents the QP formulation of the CNLS problem. Section 3 presents an algorithm to solve CNLS by identifying a set of initial constraints and iteratively adding constraints. Section 4 investigates the performance of the algorithm through Monte Carlo simulations. Section 5 presents an application that was previously too large to solve using the standard formulation of CNLS and describes the performance of the algorithm and Section 6 concludes.

2. CNLS and Afriat's theorem

In this section we will present the quadratic programming formulation of CNLS. An example will illustrate the general result that typically much fewer constraints are needed to solve CNLS than are included in the standard formulation. The function estimated by CNLS is typically not unique; however, the lower concave envelope of the function is. Thus we will present methods to estimate the lower concave envelope of the set of functions that are optimal to the CNLS formulation. We will also review the results of Afriat (1972) in which he proposed methods to impose convexity on the estimation of a production function. Afriat's results are important because they will provide insight into identifying binding constraints in the quadratic programming formulation of CNLS.

Consider a production function with shape restrictions that is estimated via CNLS, and specify a regression model

$$y = f(\mathbf{x}) + \varepsilon$$

where y is the dependent variable, $\mathbf{x}$ is a vector of input variables, and ε is a random variable satisfying $E(\varepsilon|\mathbf{x}) = 0$. CNLS can be used to estimate any function that belongs to the class of functions, $\mathcal{F}$, satisfying monotonicity and concavity. As an example, in this paper we will primarily focus on the well know Cobb–Douglas functional form, $f(\mathbf{x}) = \prod_{m=1}^{M} x_m^{(\beta_m)}$. Note if $\sum_{m=1}^{M}\beta_m \leqslant 1$, then the Cobb–Douglas function belongs to the class $\mathcal{F}$. The specification of the Cobb–Douglas function we use in our Monte Carlo simulations for the data generation process is $y = \prod_{m=1}^{M} x_m^{\left(\frac{0.8}{M}\right)}$.[5] Note, the additive specification of the disturbance term in our regression model does not allow one to estimate the underlying Cobb–Douglas function by applying ordinary least squares (OLSs) to the log-transformed regression equation. In this case, the Cobb–Douglas function should be estimated by nonlinear regression. However, the additive version of the Cobb–Douglas regression model has been widely used, including such seminal work as Just and Pope (1978).

The production function, $f(\mathbf{x})$, could be estimated by assuming a parametric functional form and applying OLS or maximum likelihood methods; however in parametric functions such as translog, the regularity conditions (monotonicity and concavity) are typically difficult to impose (Henningsen and Henning, 2009). Recent developments in the nonparametric regression literature now allow the estimation of production functions consistent with monotonicity and concavity as described below.

2.1. CNLS estimation

Nonparametric regression is a method that does not specify the functional form *a priori*. The continuity, monotonicity and concavity constraints are enforced in the least squares estimation method CNLS (Hildreth, 1954; Kuosmanen, 2008),

$$\begin{aligned}
&\min_{\alpha,\beta,\varepsilon} \quad \sum_{i=1}^{n} \varepsilon_i^2 \\
&\text{s.t.} \\
&\qquad y_i = \alpha_i + \boldsymbol{\beta}_i'\mathbf{x}_i + \varepsilon_i \quad \text{for } i = 1,\ldots,n \qquad (a)\\
&\qquad \alpha_i + \boldsymbol{\beta}_i'\mathbf{x}_i \leqslant \alpha_h + \boldsymbol{\beta}_h'\mathbf{x}_i \quad \text{for } i,h = 1,\ldots,n \text{ and } i \neq h \qquad (b)\\
&\qquad \boldsymbol{\beta}_i \geqslant 0 \text{ for } i = 1,\ldots,n, \qquad (c)
\end{aligned} \tag{1}$$

where y_i denotes the output, $\mathbf{x}_i = (x_{i1}, \ldots, x_{iM})'$ is the input vector, and ε_i is the disturbance term that represents the deviation of firm

[3] Hereafter we will refer to a linearly-constrained quadratic programming problem simply as a quadratic program (QP).

[4] An algorithm is polynomial time if the algorithm's running time is bounded from above by a polynomial expression in the size of the input for the algorithm; for the purposes of this paper the inputs are the number of observations and the number of components of the input vector. NP-hard means non-deterministic polynomial-time hard; in practical terms, most computer scientists believe these problems cannot be solved in polynomial time.

[5] In the online appendix we consider a constant returns-to-scale version, $y = \prod_{m=1}^{M} x_m^{\left(\frac{1}{M}\right)}$.

Please cite this article in press as: Lee, C.-Y., et al. A more efficient algorithm for Convex Nonparametric Least Squares. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2012.11.054

i from the estimated function. Constraints (1a) represent a basic linear hyperplane and estimates intercept α_i and slope $\boldsymbol{\beta}_i = (\beta_{i1}, \ldots, \beta_{im})^{\mathrm{T}}$ parameters characterizing the marginal products of inputs for each observation. Constraints (1b) impose concavity using Afriat's inequalities. Finally, constraints (1c) impose monotonicity on the underlying unknown production function.

In general the parameters $(\alpha_i,\ \boldsymbol{\beta}'_i)$ estimated using CNLS in formulation (1) are non-unique; however the fitted values, $\hat{y}_i = \hat{\alpha}_i + \hat{\boldsymbol{\beta}}'_i\mathbf{x}_i$, are unique (Groeneboom et al., 2001). Thus one can calculate a lower concave envelope for the production function estimated using CNLS. Using the results from problem (1), Kuosmanen and Kortelainen (2012) propose to solve problem (2) to estimate the lower concave envelope.

$$\begin{aligned}\min_{\alpha_{le},\beta_{le}} \quad & \alpha_{le,i} + \boldsymbol{\beta}'_{le,i}\mathbf{x}_i \\ \text{s.t.} \quad & \alpha_{le,i} + \boldsymbol{\beta}'_{le,i}\mathbf{x}_i \geqslant \hat{y}_i \quad \text{for } i = 1, \ldots, n \end{aligned} \tag{2}$$

Here we us the notation $\alpha_{le,i}$ and $\boldsymbol{\beta}'_{le,i}$ to indicate these parameters are reestimated in (2) to find the lower concave envelope and may be distinct from the parameters estimated in (1). The optimal solution to problem (2) is unique and is also an optimal solution to problem (1). This uniqueness facilitates the analysis of the generic algorithm (described in the following section) and, thus, hereafter we refer to the optimal solution to problem (2) as the optimal solution to the CNLS problem.

We note that the models (1) and (2) can be combined in a single optimization problem by using a multi-criteria objective function and non-Archimedean weights to make the minimization of squared errors lexicopgraphically more important, but the use of a non-Archimedean has caused considerable debate in the closely related Data Envelopment Analysis (DEA) literature (see, for example, Boyd and Fare, 1984; Charnes and Cooper, 1984). Thus we prefer to maintain the two models in which model (1) estimates the fitted values $\hat{y}_i$ and model (2) calculates the lower concave envelope. The fitted values are unique (Groeneboom et al., 2001); however, the set of hyperplanes need not be unique. Thus the lower concave envelope is consistent with the minimum extrapolation principle, Banker et al. (1984), and provides a unique identification of the hyperplanes (Kuosmanen and Kortelainen, 2012).

As is clear from formulation (1) and (2), CNLS estimates the unknown production function using n hyperplane segments. However, typically the number of **unique** hyperplane segments is much lower than n (Kuosmanen and Johnson, 2010), which presents an opportunity to reduce the number of constraints and decrease the time required to solve problem (1). To illustrate this phenomenon and the CNLS estimator, we generated 100 observations of a single-input single-output equation, $y = x^{0.8} + v$. The observations, x, were randomly sampled from a Uniform [1,10] distribution and v was drawn from a normal distribution with standard deviation of 0.7. Fig. 1 shows the obtained CNLS estimator. Note that, in this case, the CNLS curve is characterized by seven unique hyperplanes (dashed lines) and the other 93 hyperplanes estimated are redundant (that is, even though we are estimating 100 hyperplanes, 93 of the estimated hyperplanes are identical to one of the 7 hyperplanes that form the lower concave envelope).[6]

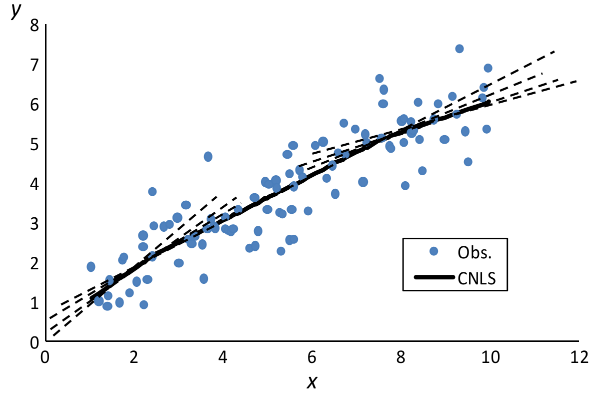

Fig. 1. The CNLS estimate, includes only seven unique hyperplanes (dashed lines).

2.2. Afriat's theorem

The method to impose concavity in CNLS is based on Afriat's theorem (Afriat, 1967; Afriat, 1972), which is a fundamental result in microeconomic theory (e.g., Varian, 1982, Varian, 1984). Afriat's theorem can be used for two purposes: (1) nonparametrically testing if a given set of data satisfies the regularity conditions (concavity) implied by the economic theory (Varian, 1984). If this is indeed the case, Afriat's numbers (defined in Theorem 1) can be used for constructing inner and outer bounds for the possible functions f that can describe the data. Or alternatively (2) Afriat's inequalities (defined in Theorem 1) have been used in the context of nonparametric regression to enforce global curvature conditions (Kuosmanen and Kortelainen, 2012). There are many potential applications in areas such as demand analysis, production analysis, and finance.

Kuosmanen (2008) transformed the infinite dimensional CNLS problem to the finite dimensional QP problem using Afriat's inequalities defined in Afriat's theorem:

Theorem 1. (Afriat's Theorem[7]):*If n is the number of observations and m is the number of inputs, the following conditions are equivalent:*

(i) There exists a continuous globally concave function f: $R^m \to R$ that satisfies $y_i = f(\mathbf{x}_i)$ in a finite number of points $i = 1,\ldots,n$.

(ii) There exist finite coefficients (from now Afriat's numbers) $\alpha_i, \beta_i = (\beta_{i1} \ldots \beta_{im})'$ such that $y_i = \alpha_i + \boldsymbol{\beta}'_i\mathbf{x}_i$ for $i = 1, \ldots, n$, that satisfy the following system of inequalities (henceforth Afriat's inequalities):

$$\alpha_i + \boldsymbol{\beta}'_i\mathbf{x}_i \leqslant \alpha_h + \boldsymbol{\beta}'_h\mathbf{x}_i \quad \text{for } i, h = 1, \ldots, n \quad \text{and } i \neq h.$$

The above statement of Afriat's theorem refers to f as a classic concave production function, but other applications (e.g., utility functions, cost/expenditure functions, distance functions, etc.) are equally possible. The following properties are derived from Afriat's theorem:

(1) Instead of concavity, convexity is easily implemented by reversing the sign of inequalities in condition ii above.

(2) Strict concavity (convexity) is obtained by using strict inequalities in condition ii above.

(3) Monotonicity can be imposed independently by inserting further constraints $\beta_i \geqslant 0, \forall i$ (increasing) or $\beta_i \leqslant 0$, $\forall i$ (decreasing).

[6] CNLS alone is used to estimate an average production function and is the focus of our paper. However, StoNED (an efficiency analysis method) uses CNLS in the first stage (Kuosmanen and Kortelainen, 2012) and the Jondrow decomposition (Jondrow et al., 1982) in the second stage. The Jondrow decomposition assumes homoskedasticity of both noise and inefficiency, thus the frontier production function is simply a parallel shift of the function estimated considering only noise. The shape of production function is unchanged, and thus does not affect the computational complexity.

[7] There are alternative equivalent statements of Afriat's theorem, see e.g., Afriat (1972) or Varian (1982) or Fostel et al. (2004); however we follow Kuosmanen (2008).

Please cite this article in press as: Lee, C.-Y., et al. A more efficient algorithm for Convex Nonparametric Least Squares. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2012.11.054

In the context of nonparametric regression, n should be generally large to accurately estimate a function in high dimensions; this is often referred to as the curse of dimensionality (Yatchew, 1998). The system of Afriat's inequalities, presented in condition ii above, involves $n(m+1)$ unknown variables and $n(n-1)$ inequality constraints, where the number of observations, n, is usually much larger than the number of inputs, m. When the data contain a large number of observations, imposing Afriat's inequalities can become computationally demanding. For example when $n = 100$, the number of inequalities is 9900.

3. A generic algorithm for CNLS model reduction

This section develops a generic algorithm based on the seminal work of Dantzig et al. (1954, 1959) to address the computational burden of solving CNLS. Specifically, Dantzig et al. proposed the following approach of solving large-scale problems: Solve a relaxed model containing only a subset of the constraints, and iteratively add violated constraints to the relaxed model until an optimal solution to the relaxed model is feasible for the original problem. Recall that, given n observations, CNLS requires $n\,(n-1)$ concavity constraints. If n is large, the number of concavity constraints is significant, because the performance of standard QP algorithms is limited by the number of constraints. To address this issue we use Dantzig et al.'s strategy: we start with a set of inequalities that are likely to be satisfied at equality in the optimal solution, and iteratively add violated constraints to the relaxed model until the optimal solution to the relaxed model is feasible for the CNLS problem. Hereafter we define *relevant constraints* as the set of inequalities from problem (1) that are satisfied at equality by the optimal solution to problem (2).[8]

The generic algorithm iterates between two operations: (A) solving model (1) but including only a subset V of the constraints in (1b), and (B) verifying whether the obtained solution satisfies all of the constraints in (1b); if it does then the algorithm terminates, otherwise V is appended with some of the violated constraints and the process is restarted. Section 3.1 gives two strategies to identify an initial subset of constraints that includes a large proportion of the relevant constraints with a relatively high level of accuracy. Section 3.2 describes three strategies to determine the subset of violated constraints to be added at each iteration. In order to give an algorithmic description of the generic algorithm, we use the following formulation, which we refer to as the relaxed CNLS problem (RCNLS).

$$
\begin{aligned}
&\min_{\alpha,\beta,\varepsilon} \quad \sum_{i=1}^{n} \varepsilon_i^2 \\
&\text{s.t.} \quad y_i = \alpha_i + \boldsymbol{\beta}_i'\mathbf{x}_i + \varepsilon_i \quad \text{for } i = 1,\dots,n \qquad (a)\\
&\qquad \alpha_i + \boldsymbol{\beta}_i'\mathbf{x}_i \leqslant \alpha_h + \boldsymbol{\beta}_h'\mathbf{x}_i \quad \forall (i,h) \in V \qquad (b)\\
&\qquad \boldsymbol{\beta}_i \geqslant 0 \quad \text{for } i = 1,\dots,n, \qquad (c)
\end{aligned}
\tag{3}
$$

Here V is a subset of all the observation pairs; thus the concavity constraints (3b) are a subset of all the concavity constraints (1b).

Generic Algorithm

```
1. Let $t = 0$ and let $V$ be a subset of the observation pairs.
2. Solve RCNLS to find an initial solution, $\left(\alpha_i^{(0)},\ \boldsymbol{\beta}_i^{(0)}\right)$.
3. Do until $\left(\alpha_i^{(t)},\ \boldsymbol{\beta}_i^{(t)}\right)$ satisfies all concavity constraints (Eqs. (1b)):
   3.1. Select a subset of the concavity constraints that $\left(\alpha_i^{(t)},\ \boldsymbol{\beta}_i^{(t)}\right)$ violates and let $V^{(t)}$ be the corresponding observation pairs.
   3.2. Set $V = V \cup V^{(t)}$.
   3.3. Solve RCNLS to obtain solution $\left(\alpha_i^{(t+1)},\ \boldsymbol{\beta}_i^{(t+1)}\right)$.
   3.4. Set $t = t + 1$.
```

Proposition 1. *The generic algorithm obtains an optimal solution to CNLS (problem (1)).*

Proof. The result follows from the following two observations: (1) For any t, (α_i^t, β_i^t) is an optimal solution to a relaxation of problem (1). (2) The termination condition for the generic algorithm (step 3) guarantees that, at termination, (α_i^t, β_i^t) is a feasible solution of problem (1).

3.1. Approaches to determine the set of initial constraints

Critical to the generic algorithm's performance is the identification of a set of initial concavity constraints that includes a large proportion of the relevant constraints. This section describes two methods for constructing such a set of initial constraints.

3.1.1. Elementary Afriat approach

For intuition, let us start from a univariate case $m = 1$. The number of unknowns is only $2n$, but we still have $n(n-1)$ inequality constraints. It is possible to reduce the number of inequalities by sorting the observed data in ascending order according to x. Without loss of generality, assume the data have been sorted as $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$. In this case, it is easy to show the following:

Elementary Afriat's theorem – univariate case (Hanson and Pledger, 1976): The following conditions are equivalent:

(i) There exists a continuous globally concave function f: $R \to R$ that satisfies $y_i = f(\mathbf{x}_i)$ in a finite number of points $i = 1,\dots,n$.

(ii) There exist finite coefficients α_i, β_i: $y_i = \alpha_i + \beta_i x_i \forall i = 1,\dots,n$, that satisfy the following system of inequalities (original Afriat's inequalities):

$$\alpha_i + \beta_i x_i \leqslant \alpha_h + \beta_h x_i \quad \forall i,\ h = 1,\dots,n \quad \text{and } i \neq h.$$

(iii) There exist finite coefficients α_i, β_i: $y_i = \alpha_i + \beta_i x_i\ \forall i = 1,\dots,n$, that satisfy the following system of inequalities (elementary Afriat's inequalities):[9]

$$
\begin{aligned}
&\beta_i \leqslant \beta_{i-1} \quad \forall i = 2,\dots,n\\
&\alpha_i \geqslant \alpha_{i-1} \quad \forall i = 2,\dots,n
\end{aligned}
$$

Condition ii) involves $n\,(n-1)$ constraints, whereas condition iii) requires only $2(n-1)$ constraints. In the case of $n = 100$, the original conditions require 9900 inequalities, whereas our elementary condition requires only 198 inequalities. Thus, a substantial decrease in the number of inequalities is possible by using the prior ranking of the observed data and the transitivity of inequality relations. Moreover, note that imposing monotonicity in the single input case, condition iii) requires only a single constraint $\beta_n \geqslant 0$, whereas imposing monotonicity in the general case, condition ii) requires $n \cdot m$ constraints $\boldsymbol{\beta}_i \geqslant 0\ \forall\ i = 1,\dots,n$.

The elementary Afriat's theorem motivates the following method for generating an initial set of constraints when the production function being estimated has multiple inputs.[10] Arbitrarily pick one of the inputs (say, variable k) and index the observed data in ascending order according to the selected input (i.e., such that $\mathbf{x}_i \leqslant {}^k\mathbf{x}_{i+1}$, where the inequality compares only the kth entry of the input vector, but the entire input matrix is sorted). Then, let the initial set of

[8] As noted in Section 2.2, this solution is also an optimal solution to problem (1), and has the advantage of being unique (thus the relevant constraints are well-defined).

[9] In the case of ties (i.e., when $x_i = x_{i-1}$), the inequalities should be changed to equalities.

[10] Alternative methods such as kernel based methods or principle component analysis could also be used; however, we suggest the elementary Afriat approach because inputs are often highly correlated. If the inputs vectors are perfectly correlated, the elementary Afriat constraints would be sufficient to impose concavity.

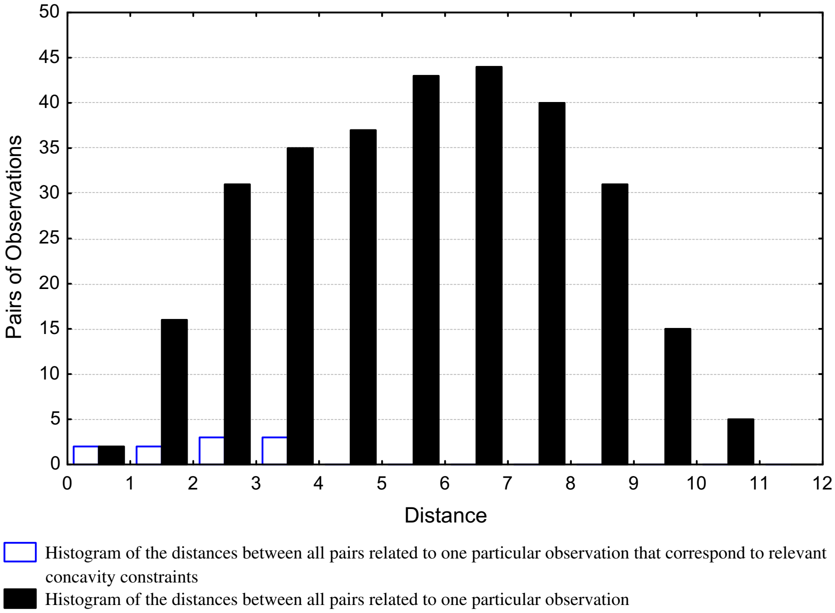

Fig. 2. The concavity constraints corresponding to nearby observations are significantly more likely to be relevant than those corresponding to distant observations.

observation pairs in V to be $\{(1,2),\ (2,3),\dots,(i-1,i),\ (i,+1),\dots,(n-1,n)\}$; thus, the explicit formulation for initial relaxed CNLS problem is as follows:

$$\min_{\alpha,\beta,\varepsilon}\left\{\sum_{i=1}^{n}\varepsilon_i^2 \;\middle|\; \begin{array}{ll} y_i = \alpha_i + \boldsymbol{\beta}_i'\mathbf{x}_i + \varepsilon_i & \forall i = 1,\dots,n; \\ \alpha_i + \boldsymbol{\beta}_i'\mathbf{x}_i \leqslant \alpha_{i+1} + \boldsymbol{\beta}_{i+1}'\mathbf{x}_i & \forall i = 1,\dots,n-1; \\ \boldsymbol{\beta}_i \geqslant 0 & \forall i = 1,\dots,n; \end{array}\right\} \qquad (4)$$

In the Monte Carlo simulations described in Section 4, we refer to the use of this strategy for determining the set of initial constraints as CNLS^r.

3.1.2. Sweet spot approach

The sweet spot approach aims to predict the relevant concavity constraints and uses these as the initial set of constraints. This approach is implemented as: for each observation i, include the concavity constraints corresponding to the observations whose distance[11] to observation i is less than a pre-specified threshold value δ_i (distance percentile parameter). The range between the zeroth percentile and the δ_i^{th} percentile is defined as the sweet spot. Empirically, we found that an effective value for δ_i is the 3rd percentile of the distances from all observations to observation i.[12] If both elementary Afriat approach and sweet spot approach are applied to identify initial set of constraints (sweet spot constraints), then we will refer to the approach as CNLS$^+$. The reminder of this section motivates CNLS$^+$.

As previously mentioned (and illustrated in Fig. 1), Kuosmanen and Johnson, 2010 showed that the number of unique hyperplanes to construct a CNLS production function is generally much lower than n. From Eq. (1b), observe that, in the optimal solution of CNLS (problem 1), the concavity constraints that are satisfied at equality correspond to pairs of observations that share a hyperplane in the CNLS function. Therefore only a small number of the concavity constraints are relevant. Moreover, it is reasonable to assume that observations that are close to each other are more likely to correspond to relevant concavity constraints than those that are far apart. The following simulation further motivates the idea that the relevant concavity constraints correspond to pairs of nearby observations.

We generated 300 observations of a two-input single-output equation, $y = x_1^{04}x_2^{04} + v$. The observations, x_1, x_2, were randomly sampled from a Uniform [1,10] distribution and v was drawn from a normal distribution with standard deviation of 0.7. Then we solved the CNLS problem (since it is not possible to directly solve problem 1 with more than 200 observations, we solved it using one of the algorithms herein proposed and based on Theorem 1 the results are equivalent) and identified the relevant constraints. Fig. 2 shows a histogram of the distances between all pairs related to one particular observation (black) and the histogram of the distances between all pairs related to one particular observation that correspond to relevant concavity constraints (white). One can observe that indeed, as previously argued, the concavity constraints corresponding to nearby observations are significantly more likely to be relevant than those corresponding to distant observations.

3.2. Strategies for selecting from the set of violated concavity constraints

We propose three strategies to select the violated (concavity) constraints (VCs) that are added in each iteration of the generic algorithm. The first strategy, referred to as one-VC-added CNLS (CNLS-O), is to select the most violated constraint from all concavity constraints (See table 1 for the definition of most violated). This strategy, in each iteration, adds at most one violated constraint to the set V. The second strategy, referred to as group-VC-added CNLS (CNLS-G), is to select, for each observation i, the most violated constraint among the $n-1$ concavity constraints related to observation i. This strategy, in each iteration, adds at most $n-1$ violated constraints to the set V. The last strategy, referred to as all-VC-added (CNLS-A), is to select all the violated constraints. This strategy adds at most $(n-1)^2$ violated constraints to the set V. Table 1 shows the summary of these three strategies and provides

[11] This study uses the Euclidean norm measured in the M+1 dimensional space of inputs and output to measure the distance between two observations. Alternatively the distance could be measure in an M dimensional space, however, experimental results indicated this did not have a significant effect in the computational time.

[12] The 3rd percentile worked well in our Monte Carlo simulations. In the empirical example given in Section 5, we test several different percentiles and show that indeed the 3rd percentile works well.

Table 1
Strategies for selecting from the set of violated concavity constraints.

Strategy	Formula to identifying the violated constraints to add	Upper bound on the # of VC added per iteration
CNLS-O (one-VC-added)	$\max\limits_{i,h}\left\{\alpha_i^t + \boldsymbol{\beta}_i^{t'}\mathbf{x}_i - \left(\alpha_h^t + \boldsymbol{\beta}_h^{t'}\mathbf{x}_i\right)\right\} > 0$	1
CNLS-G (group-VC-added)	$\max_h\left\{\alpha_i^t + \boldsymbol{\beta}_i^{t'}\mathbf{x}_i - \left(\alpha_h^t + \boldsymbol{\beta}_h^{t'}\mathbf{x}_i\right)\right\} > 0 \quad \forall i$	n
CNLS-A (all-VC-added)	$\alpha_i^t + \boldsymbol{\beta}_i^{t'}\mathbf{x}_i - \left(\alpha_h^t + \boldsymbol{\beta}_h^{t'}\mathbf{x}_i\right) > 0 \quad \forall i,h$	$(n-1)^2$

Table 2
CNLS$^+$-G significantly outperforms the other methods for two dimensional problems with at least 100 observations.

Number of observations	Average run time measured in seconds (standard deviation)						
	CNLS	CNLS^r-O	CNLS^r-G	CNLS^r-A	CNLS$^+$-O	CNLS$^+$-G	CNLS$^+$-A
25	0.4 (0.02)	24.8 (1.81)	4.2 (0.26)	1.4 (0.09)	19.5 (2.22)	3.7 (0.65)	1.1 (0.17)
50	5.8 (0.61)	70.4 (2.69)	9.4 (1.38)	4.0 (0.80)	44.3 (5.72)	6.4 (0.90)	2.6 (0.73)
100	367.9 (40.93)	210.2 (10.04)	83.0 (12.32)	63.9 (13.32)	116.7 (9.40)	24.0 (5.77)	28.6 (11.81)
150	4168 (600.93)	474.6 (50.69)	393.9 (135.27)	292.6 (66.61)	158.4 (23.69)	30.6 (6.55)	98.0 (40.72)
200	N/A	953.7 (118.36)	1424 (180.32)	820.8 (371.21)	236.9 (37.67)	47.2 (12.12)	101.3 (22.31)
250	N/A	1779 (188.44)	2555 (861.06)	1344 (413.18)	392.9 (50.88)	62.8 (14.20)	131.2 (84.82)
300	N/A	3144 (445.20)	8084 (2519.76)	1833 (585.07)	672.2 (95.83)	93.4 (18.45)	172.6 (204.05)

N/A: system out of memory.

the formulas to identify and quantify the violated constraints (VC) to add.

4. Monte Carlo simulations

This section describes four simulation studies analyzing the performance of the variants of the generic algorithm and comparing them to directly solving CNLS (1). These experiments were performed on a personal computer (PC) with an Intel Core i7 CPU 1.60 GHz and 8 GB RAM. The optimization problems were solved in GAMS 23.3 using the CPLEX 12.0 QCP (Quadratically Constrained Program) solver. The six variants analyzed are all the possible combinations between determining the initial constraint set (CNLS^r and CNLS$^+$) and selecting the violated constraints to add (CNLS-O, CNLS-G and CNLS-A). The first simulation study investigates the performance of the algorithms as a function of the number of observations. The second simulation study investigates the performance of the algorithms as a function of the number of inputs. The third simulation compares the algorithms by simultaneously varying the number of inputs and the number of observations. Finally, the fourth simulation study aims to determine the largest problem sizes that can be solved with the variants that were found to be the most effective in the other three studies.

The first study assumed a two-input, one-output production function, $f(x) = x_1^{0.4}x_2^{0.4}$, and the corresponding regression equation, $y = x_1^{0.4}x_2^{0.4} + v$, where the observations x_1 and x_2 were randomly sampled from a Uniform [1,10] distribution and v was drawn from a normal distribution with standard deviation of 0.7. This study simulates seven scenarios, each with different number of observations. Each of the rows in Table 2 correspond to one scenario and give the average time, in seconds and standard deviation (shown

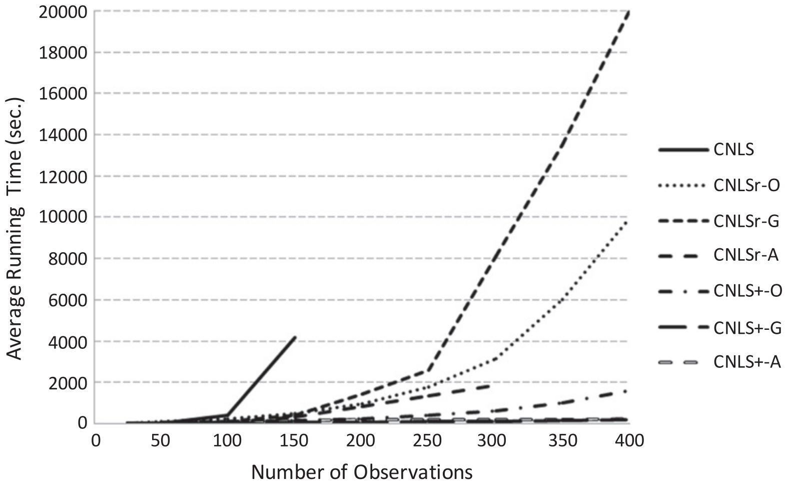

Fig. 3. CNLS$^+$-G has the least running time for problems with at least 100 observations.

Please cite this article in press as: Lee, C.-Y., et al. A more efficient algorithm for Convex Nonparametric Least Squares. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2012.11.054

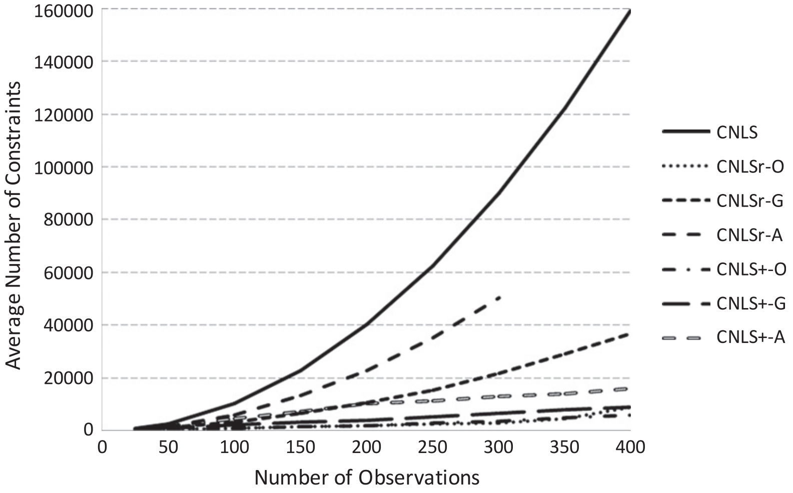

Fig. 4. CNLS$^+$-G adds almost the same the number of constraints as CNLS^r-O and CNLS$^+$-O (which add the fewest constraints).

in parenthesis) that each algorithm required to solve the problem. The averages and standard deviations were obtained by simulating each scenario 10 times.

Table 2 shows that CNLS$^+$-G significantly outperforms the other methods for two dimensional problems with at least 100 observations. An extended version of Table 2, Table 9, is shown in the online appendix. Table 9 shows that: (1) CNLS^r-O adds the least number of constraints to construct the CNLS production function but requires the most iterations which leads to longer computation times; (2) conversely, CNLS^r-A requires the least number of iterations, but adds the most constraints; (3) the original CNLS model generates 90,000 constraints for a problem with 300 observations, but actually, on average, in such model only 1238 constraints are relevant (including 938 concavity constraints). In the two input case Figs. 3 and 4 illustrate the average running time and average constraints needed, respectively, for each of the seven strategies CNLS, CNLS^r-O, CNLS^r-G, CNLS^r-A, CNLS$^+$-O, CNLS$^+$-G and CNLS$^+$-A while increasing the number of observations.

The second study assumed an M-input one-output production function, $y = \prod_{m=1}^{M} x_m^{\left(\frac{0.8}{M}\right)} + v$, where the observations x_m were randomly sampled form a Uniform[1,10] distribution and v was a drawn from a normal distribution with standard deviation of 0.7.

This study simulates seven scenarios, each with a different number of inputs. Each of the rows in Table 3 correspond to one scenario and give the average time (in seconds) that each variant required to solve the problem. The averages were obtained by simulating each scenario 10 times. Table 3 shows that CNLS$^+$-G outperforms all other variants. Only in the 8-input scenario a variant, CNLS^r-G, is slightly better than CNLS$^+$-G. Recall that, in this scenario, CNLS is estimating an eight-dimensional function therefore, in the context of nonparametric regression and its curse of dimensionality, 100 observations is, if at all, barely enough to obtain meaningful results. Indeed, from Table 3, one can observe that, for a fixed number of observations and as the number of inputs increase, the performance of CNLS^r-G improves with respect to the performance of CNLS$^+$-G. However, we consider that, for practical purposes, this improvement is not relevant because, to obtain a meaningful nonparametric regression curve, the number of observations should grow exponentially as the number of dimensions increase (see Yatchew (1998), curse of dimensionality).

Table 3
CNLS$^+$-G outperforms the other methods for 100 observations in problems with 2–7 inputs.

# Of inputs	Average running time (seconds)						
	CNLS	CNLS^r-O	CNLS^r-G	CNLS^r-A	CNLS$^+$-O	CNLS$^+$-G	CNLS$^+$-A
2	367.9	210.2	83.0	63.9	116.7	24.0	28.6
3	385.2	245.0	87.8	139.9	184.2	27.2	72.0
4	304.1	256.8	54.4	117.3	200.2	28.4	88.9
5	298.9	241.3	72.2	101.7	207.2	37.3	73.6
6	271.5	274.1	63.4	94.9	214.7	45.5	76.1
7	295.5	252.0	51.9	83.1	224.4	43.1	53.6
8	288.2	245.0	34.8	79.1	227.7	35.1	51.4

Table 10, found in the online appendix, is an extended version of Table 3. From Table 10 one can observe the following interesting phenomenon: For a fixed number of observations, as the number of inputs increase, the number of relevant concavity constraints decreases slightly. Nevertheless, this reduction is minuscule so, for practical purposes, we conclude that the dimensionality of the problem has little impact on the number of hyperplanes required.

The third study assumed the same production function as the second study. The observations and noise were also sampled from the same distributions used in the second study. In this study the number of inputs is varied from two to eight and the number of observations is one of {25, 50, 100, 200, 300, 400, 500, 600, 700}. Thus the third study consists of 63 scenarios and, as before, each scenario was simulated 10 times to obtain the average performance of each algorithm. Table 4 gives, for each scenario, the best strategy in terms of average solution time. Note that CNLS and CNLS$^+$-A are the best strategies when the number of observations is small (in fact, too small to be useful in practice) while CNLS$^+$-G and CNLS^r-G are the best strategies when the number of observations is large (practical-sized problems). For high dimensional models, when the number of observations ranges from 100 to 400, CNLS^r-G dominates CNLS$^+$-G. This is because CNLS$^+$-G takes more iterations to identify the violated concavity constraints then CNLS^r-G even though CNLS$^+$-G uses fewer constraints on average. Also note the CNLS$^+$-G method average time to solve a 500 observation formulation is less than a 400 observation formulation. A 500 observation formulation adds more concavity constraints initially based on the distance criteria and uses fewer iterations to reach optimal solution leading to a shorter run time. In general, the CNLS$^+$-G is suggested in large scale or high dimensionality scenarios.

Reviewing the results in Tables 3 and 4, the run time for the entire proposed algorithm variant is reported. Consider the example with 300 observations, 2 inputs and solved using variant

Please cite this article in press as: Lee, C.-Y., et al. A more efficient algorithm for Convex Nonparametric Least Squares. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2012.11.054

Table 4
In most scenarios CNLS+-G is the best strategy to solve problem (1).

Obs.	Performance	Number of inputs						
		2	3	4	5	6	7	8
25	Best strategy	CNLS	CNLS	CNLS	CNLS	CNLS	CNLS	CNLS
	Avg. constr.	625	625	625	625	625	625	625
	Avg. time (seconds)	0.4	0.5	0.5	0.5	0.5	0.4	0.5
50	Best strategy	CNLS$^+$-A	CNLS$^+$-A	CNLS$^+$-A	CNLS$^+$-A	CNLS$^+$-A	CNLS$^+$-A	CNLS$^+$-A
	Avg. constr.	1303	1131	1187	1176	1163	1098	990
	Avg. iter.	2.4	3.2	3.0	3.3	2.5	2.8	3.0
	Avg. time (seconds)	2.6	3.0	3.4	3.7	2.9	3.1	3.0
100	Best strategy	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS^r-G
	Avg. constr.	2111	1933	1813	1737	1731	1615	1464
	Avg. iter.	20.7	21.1	23.3	28.2	32.8	31.5	35.7
	Avg. time (seconds)	24.0	27.2	28.4	37.3	45.5	43.1	34.8
200	Best strategy	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS^r-G	CNLS^r-G	CNLS^r-G
	Avg. constr.	4477	4506	4414	4268	5677	4995	4751
	Avg. iter.	18.5	23.7	25.3	31.3	22.0	19.0	18.7
	Avg. time (seconds)	47.2	94.8	137.5	200.6	135.1	86.4	70.4
300	Best strategy	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS^r-G	CNLS^r-G	CNLS^r-G	CNLS^r-G
	Avg. constr.	6712	6743	7169	9793	10,725	10,276	8386
	Avg. iter.	14.6	22.0	27.7	23.0	26.3	26.0	18.0
	Avg. time (seconds)	93.4	276.1	476.0	536.6	611.2	505.9	171.9
400	Best strategy	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS^r-G	CNLS^r-G	CNLS^r-G
	Avg. constr.	10,015	10,106	10,117	10,491	18,095	14,336	11,262
	Avg. iter.	15.1	22.1	30.3	38.5	32.7	22.3	13.7
	Avg. time (seconds)	224.8	456.5	1278	2034	2215.6	861.6	198.0
500	Best strategy	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G
	Avg. constr.	11,232	11,692	12,153	12,689	12,078	11,186	11,639
	Avg. iter.	5.0	6.6	9.0	10.3	8.7	7.3	7.0
	Avg. time (seconds)	135.3	209.5	358.4	626.5	835.1	808.3	594.7
600	Best strategy	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G
	Avg. constr.	14,699	15,022	15,339	15,454	15,669	14,948	14,231
	Avg. iter.	3.7	4.7	6.0	6.0	6.4	5.1	3.6
	Avg. time (seconds)	236.3	276.2	572.2	721.7	876.8	905.2	741.2
700	Best strategy	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G	CNLS$^+$-G
	Avg. constr.	18,684	18,948	19,384	19,660	19,934	19,423	19,309
	Avg. iter.	3.0	4.4	5.3	5.6	6.0	6.3	4.9
	Avg. time (seconds)	580.5	715.8	832.3	1147	1700	2764	1952

Table 5
Within a 5 hours limit, CNLS$^+$-G can solve problems with a larger number of observations than CNLS^r-G.

Proposed models	Number of inputs		
	2	5	8
CNLS^r-G	400	800	700
CNLS$^+$-G	1300	800	1000

Table 6
Descriptive statistics.

All districts ($N = 604$)	2006–2007		2007–2008	
	Mean	Std. dev.	Mean	Std. dev.
Performance score (Y)	95.64	6.05	95.82	6.31
Admin. expenditure per pupil (X_1)	1196.10	264.23	1122.97	316.49
Building operation exp. per pupil (X_2)	1906.37	429.91	1786.22	441.63
Instructional exp. per pupil (X_3)	5400.14	868.57	5018.06	722.53
Pupil support exp. per pupil (X_4)	972.46	312.48	897.02	256.89

Table 7
Running time (seconds) in educational data set.

Year	CNLS	CNLS^r-O	CNLS^r-G	CNLS^r-A	CNLS$^+$-O	CNLS$^+$-G	CNLS$^+$-A
2006–2007	N/A	11,974	32	N/A	8521	150	N/A
2007–2008	N/A	7489	5333	N/A	8715	704	N/A

N/A: system out of memory.

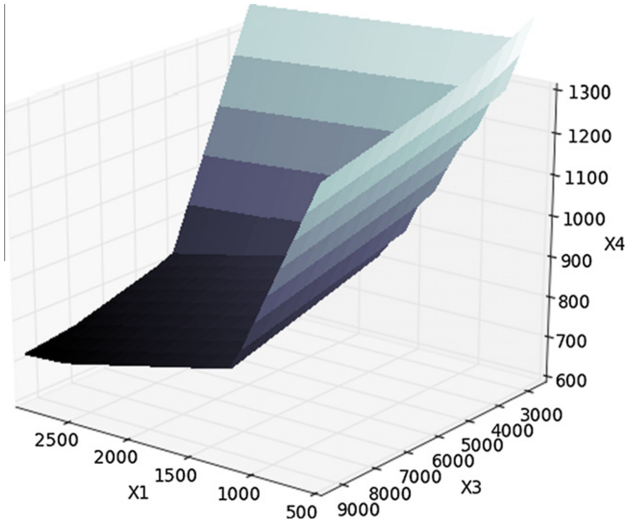

Fig. 5. The frontier of input space by administration expenditure per pupil (X_1), instructional expenditure per pupil (X_3), and pupil support expenditure per pupil (X_4) given fixed performance score and building operation expenditure per pupil on average in 2006–2007.

CNLS^r-G, to investigate the run time of each step of the algorithm, we find the initial solution, step 2, includes 600 constraints and takes less than 1 second; steps 3.1 and 3.2 are also very fast, taking less than 1 second because they are simple calculations; and step 3.3 presents a significant burden because the run time increases with the number of constraints. The 1st iteration including 900 constraints takes 1 second and the 80th iteration including around 22,000 constraints takes 14 minutes. These computational times are representative of the general results of our

Table 8
More than 26% of the relevant constraints are found in the sweet spot.

Year	2006–2007			2007–2008		
Distance percentile	3rd	6th	9th	3rd	6th	9th
Benchmarks						
Total number of CNLS constraints (A)	364,816	364,816	364,816	364,816	364,816	364,816
Number of relevant constraints (B)	2348	2348	2348	1731	1731	1731
Performance of the CNLS$^+$-G Algorithm						
CNLS$^+$-G running time (sec.)	150	1277	2833	704	1322	5125
Number of Constraints						
Total number of constraints in CNLS$^+$-G, (C)=(D)+(E)+(F)+(G)	13,285	23,554	33,823	21,785	23,553	34,422
Number of linear regression constraints[a] (D)	604	604	604	604	604	604
Number of ordering constraints[b] $\in V$ (E)	603	603	603	603	603	603
Number of sweet spot constraints[c] $\in V$ (F)	10,872	21,744	32,616	17,587	21,744	32,616
Number of VC[d] added, V (G)	1206	603	0	2991	602	599
Number of relevant constraints						
Total number of relevant constraints found in CNLS$^+$-G, (H)=(I)+(J)+(K)+(L)=(B)	2348	2348	2348	1731	1731	1731
Number of relevant constraints found in linear regression, (I)	604	604	604	604	604	604
Number of relevant constraints found in ordering constraints, (J)	10	10	10	7	7	7
Number of relevant constraints found in the sweet spot (K)	971	1407	1734	453	684	811
Number of relevant constraints found in the VC added (L)	763	327	0	667	436	309
Ratios assessing the effectiveness of the CNLS$^+$-G Algorithm						
Percentage of constraint reduction (1-C/A) (%)	96.4	93.5	90.7	94.0	93.5	90.6
Percentage of relevant constraints to CNLS constraints (B/A) (%)	0.64	0.64	0.64	0.47	0.47	0.47
Percentage of relevant constraints to CNLS$^+$-G constraints (B/C) (%)	17.7	10.0	6.9	8.0	7.4	5.0
Percentage of sweet spot constraints that are relevant constraints (K/F) (%)	8.9	6.5	5.3	2.6	3.2	2.5
Percentage of relevant constraints in the sweet spot to all relevant constraints (K/H) (%)	41.4	60.0	73.9	26.2	39.5	46.9
Percentage of VC added that are relevant constraints (L/G) (%)	63.3	54.2	N/A	22.3	72.4	51.6

[a] These are shown in (1a).
[b] These are the second type of constraints shown in Eq. (4).
[c] These are all the second type of constraints shown in Eq. (3), excluding the ordering constraints.
[d] These are the third type of constraints shown in Eq. (3).

experiments with alternative variants of the proposed algorithm and varying the size of the problem instance in terms of observations and inputs.

The final simulation study aims to determine the limit on the problem sizes (in terms of number of observations) that can be solved within 5 hours with the CNLS^r-G and CNLS$^+$-G variants. The results, presented in Table 5, indicate that CNLS$^+$-G can solve problems with a larger number of observations than CNLS^r-G.

5. Empirical study

This section demonstrates the computational benefits of the generic algorithm, and in particular the benefits of the CNLS$^+$-G variant. This variant was applied to an empirical study about State of Ohio kindergarten through twelfth grade schools for the 2006–2007 and the 2007–2008 school years. The dataset in this study is thoroughly described in Johnson and Ruggiero (in press). There are four classes of expenditures per pupil as inputs: administrative (X_1), building operation (X_2), instructional (X_3), and pupil support (X_4). The input price of each expenditure is deflated by an index of first-year teacher salaries and is measured on a per student basis. The output is an index of student performance (Y) developed by the State of Ohio. This index aggregates the measure of 30 statewide outcome goals including standardized tests in an overall measure of performance. Descriptive statistics of 604 observations are reported in Table 6.

Previously an analyst would have to use other production function estimation methods such as Data Envelopment Analysis (DEA)[13] or Stochastic Frontier Analysis (SFA) with their modeling limitations, (Kuosmanen and Kortelainen, 2012), because a CNLS model was computational infeasible. To illustrate the computational benefits of the proposed methods, the running time of 2006–2007 and 2007–2008 with respect to proposed models are shown in Table 7. The standard CNLS formulation (problem 1) cannot be directly solved due to out-of-memory errors. The CNLS$^+$-G algorithm performs well in both cases.

Fig. 5 shows the frontier of input space spanned by factors administration expenditure per pupil (X_1), instructional expenditure per pupil (X_3), and pupil support expenditure per pupil (X_4) given performance score and building operation expenditure per pupil are fixed at their averages for the data 2006–2007. The figure illustrates the substitutability among input factors.

Recall that in the sweet spot approach, used in CNLS$^+$-G, the initial set of constraints, V, in RCNLS is built as follows: For each observation i, include the concavity constraints corresponding to the observations whose distance to observation h is less than a pre-specified threshold value δ_i (distance percentile parameter). Also recall that the range between the zeroth percentile and the δ_i^{th} percentile is defined as the Sweet Spot. Throughout the paper, the threshold value in CNLS$^+$-G is set to the 3rd percentile, because this threshold was found to work well for our Monte Carlo simulations. Here we investigate the effects of the distance percentile parameter in the empirical study. For this purpose Table 8 shows a sensitivity analysis using 3rd, 6th, and 9th percentile respectively on the data for both periods. We make the following observations:

1. Typically a higher percentile will result in a longer running time because more constraints are added initially.
2. CNLS$^+$-G reduced the number of constraints by more than 90% using any percentile.
3. The percentage of relevant constraints included in the CNLS formulation is low, 0.47–0.64%, in contrast, the percentage of relevant constraints included in CNLS$^+$-G is relatively high, 5.0–17.7%. That is, the ratio of relevant constraints to all the constraints included throughout the execution of the algorithm is 10–30 times greater in CNLS$^+$-G than in CNLS.
4. The percentage of sweet spot constraints that are relevant constraints increases as the percentile used to define the sweet spot increases. When the 9th percentile is used, more than 46.9% of

[13] A variety of computational efficient algorithms exist for DEA; for a particular fast example see Dulá (2011).

all the relevant constraints needed are found within the sweet spot, thus the method for defining the sweet spot is effective at identifying relevant constraints. However, as the percentile increases, the ratio of sweet spot constraints to relevant constraints decreases; thus we get an effect of diminishing returns.

5. Our strategy for selecting violated constraints is reasonable efficient as 22.3–72.4% of added violated constraints (VC) are relevant constraints.
6. The results of our Monte Carlo simulation validate our proposal of using the 3rd percentile because of the significant benefits in terms of running time. This in part can be attributed to the higher percentage of CNLS$^+$-G constraints that are relevant constraints (17.7% and 8.0%).

6. Conclusion

This study proposes a generic algorithm to reduce the time to solve the CNLS problem. This algorithm is necessary because current methods are very slow in the case of small sample sizes (100–300 observations) and, in our experience intractable for large sample sizes (>300). The underlying principles of this strategy are: (1) using a distance analysis to determine a set of initial constraints that are likely to be satisfied at equality in the optimal solution and (2) effectively identifying violated constraints which are iteratively added to the model. A particular variant of the generic algorithm, the CNLS$^+$-G variant, was determined to be the best algorithm by an extensive simulation study. CNLS$^+$-G was successfully applied to a real-life empirical study for which estimating CNLS was previously impossible using CPLEX and reasonable computational power. The distance analysis allows 25–75% of the relevant constraints to be identified initially. Although CNLS$^+$-G requires solving multiple quadratic programming problems, the largest instance that needs to be solved is at least 90% smaller in terms of the number of constraints required compared to the original CNLS formulation.

The generic algorithm to solve the CNLS problem is based on the strategy of Dantzig et al. (1954, 1959) to solve large scale problems by iteratively identifying and adding violated constraints. Most studies that apply Dantzig et al.'s strategy consider NP-hard problems. In contrast, we demonstrate that Dantzig et al.'s strategy is also valuable to solve problems that are theoretically tractable (i.e., in *P*), but that in practice were previously not solvable due to their large scale.

Appendix A. Supplementary material

Supplementary material associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ejor.2012.11.054.

References

Afriat, S.N., 1967. The construction of a utility function from expenditure data. International Economic Review 8, 67–77.
Afriat, S., 1972. Efficiency estimation of production functions. International Economic Review 13, 568–598.
Aguilera, N., Forzani L., Morin, P., 2012. On uniform consistent estimators for convex regression. Journal of Nonparametric Statistics 23 (4), 897–908.
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.
Boyd, G., Fare, R., 1984. Measuring the efficiency of decision making units: a comment. European Journal of Operational Research 15, 331–332.
Charnes, A., Cooper, W.W., 1984. The non-Archimedean CCR ratio for efficiency analysis: a rejoinder to boyd and fare. European Journal of Operational Research 15, 333–334.
Dantzig, G.B., Fulkerson, D.R., Johnson, S.M., 1954. Solution of a large-scale traveling salesman problem. Operations Research 2, 393–410.
Dantzig, G.B., Fulkerson, D.R., Johnson, S.M., 1959. On a linear-programming combinatorial approach to the traveling-salesman problem. Operations Research 7, 58–66.
Du, P., Parmeter, C.F., Racine, J.S., in press. Nonparametric Kernel Regression with Multiple Predictors and Multiple Shape Constraints, Statistica Sinica.
Dulá, J.H., 2011. An algorithm for data envelopment analysis. INFORMS Journal on Computing 23 (2), 284–296.
Dykstra, R.L., 1983. An algorithm for restricted least squares regression. Journal of the American Statistical Association 78 (384), 837–842.
Fostel, A., Scarf, H.E., Todd, M.J., 2004. Two new proofs of Afriat's theorem. Economic Theory 24 (1), 211–219.
Fraser, D.A.S., Massam, H., 1989. A mixed primal-dual bases algorithm for regression under inequality constraints: application to concave regression. Scandinavian Journal of Statistics 16, 65–74.
Fried, H.O., Lovell, C.A.K., Schmidt, S.S., 2008. The Measurement of Productive Efficiency and Productivity Growth. Oxford University Press, New York.
Goldman, S.M., Ruud, P.A., 1993. Nonparametric Multivariate Regression Subject to Constraint. Technical Report 93-213, Department of Economics, University of California at Berkeley.
Groeneboom, P., Jongbloed, G., Wellner, J.A., 2001. Estimation of convex functions: characterizations and asymptotic theory. Annals of Statistics 26, 1653–1698.
Hannah, L.A., Dunson, D.B., 2012. Ensemble methods for convex regression with applications to geometric programming based circuit design. In: Proceedings of the 29th International Conference on Machine Learning, ICML, vol. 1, pp. 369-376.
Hanson, D.L., Pledger, G., 1976. Consistency in concave regression. Annals of Statistics 4 (6), 1038–1050.
Henningsen, A., Henning, C., 2009. Imposing regional monotonicity on translog stochastic production frontiers with a simple three-step procedure. Journal of Productivity Analysis 32 (3), 217–229.
Hildreth, C., 1954. Point estimates of ordinates of concave functions. Journal of the American Statistical Association 49 (267), 598–619.
Johnson, A.L., Kuosmanen, T., 2011. One-stage estimation of the effects of operational conditions and practices on productive performance: asymptotically normal and efficient, root-n consistent StoNEZD method. Journal of Productivity Analysis 36 (2), 219–230.
Johnson, A.L., T. Kuosmanen, 2012. One-Stage and two-stage DEA estimation of the effects of contextual variables. European Journal of Operational Research 220 (2), 559–570.
Johnson, A.L., Ruggiero, J., in press. Nonparametric measurement of productivity and efficiency in education. Annals of Operations Research.
Jondrow, J., Lovell, C.A.K., Materov, I.S., Schmidt, P., 1982. On the estimation of technical inefficiency in the stochastic frontier production function model. Journal of Econometrics 19 (2-3), 233–238.
Just, R.E., Pope, R.D., 1978. Stochastic specification of production function and economic implications. Journal of Econometrics 7 (1), 67–86.
Kuosmanen, T., 2008. Representation theorem for convex nonparametric least squares. Econometrics Journal 11, 308–325.
Kuosmanen, T., Kuosmanen, N., 2009. Role of Benchmark Technology in Sustainable Value Analysis: an Application to Finnish Dairy Farms. Agricultural and Food Science 18 (3-4), 302–316.
Kuosmanen, T., 2012. Stochastic semi-nonparametric frontier estimation of electricity distribution networks: application of the StoNED method in the finnish regulatory model. Energy Economics 34, 2189–2199.
Kuosmanen, T., Johnson, A.L., 2010. Data envelopment analysis as nonparametric least-squares regression. Operations Research 58 (1), 149–160.
Kuosmanen, T., Kortelainen, M., 2012. Stochastic non-smooth envelopment of data: semi-parametric frontier estimation subject to shape constraints. Journal of Productivity Analysis 38 (1), 11–28.
Lim, E., Glynn, P.W., 2012. Consistency of multidimensional convex regression. Operations Research 60 (1), 196–208.
Mammen, E., 1991. Nonparametric regression under qualitative smoothness assumptions. Annuals of Statistics 19, 741–759.
Mammen, E., Thomas-Agnan, C., 1999. Smoothing splines and shape restrictions. Scandinavian Journal of Statistics 26, 239–252.
Mekaroonreung, M., Johnson, A.L., 2012. Estimating the Shadow Prices of SO2 and NOx for U. S. Coal Power Plants: a convex nonparametric least squares approach. Energy Economics 34 (3), 723–732.
Meyer, M.C., 1999. An extension of the mixed primal-dual bases algorithm to the case of more constraints than dimensions. Journal of Statistical Planning and Inference 81, 13–31.
Pshenichny, B.N., Danilin, M.Y., 1978. Numerical Methods in Extremal Problems. Mir Publishers, Moscow.
Ruud, P.A., 1995. Restricted Least Squares Subject to Monotonicity and Concavity Constraints. Working Paper, University of California at Berkeley.
Seijo, E., Sen, B., 2012. Nonparametric least squares estimation of a multivariate convex regression. Annals Statistics 39, 1633–1657.
Varian, H., 1982. The nonparametric approach to demand analysis. Econometrica 50, 945–973.
Varian, H., 1984. The nonparametric approach to production analysis. Econometrica 52, 579–597.
Wilhelmsen, D.R., 1976. A nearest point algorithm for convex polyhedral cones and applications to positive linear approximation. Mathematics of Computation 30 (133), 48–57.
Wu, C.F., 1982. Some algorithms for concave and isotonic regression. TIMS Studies in Management Science 19, 105–116.
Yatchew, A., 1998. Nonparametric regression techniques in economics. Journal of Economic Literature 36 (2), 669–721.

Please cite this article in press as: Lee, C.-Y., et al. A more efficient algorithm for Convex Nonparametric Least Squares. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2012.11.054