A Branch-and-Bound Algorithm for Instrumental Variable Quantile Regression

Guanglin Xu* Samuel Burer†

August 1, 2014

Abstract

This paper studies a statistical problem called *instrumental variable quantile regression* (IVQR). We model IVQR as a convex quadratic program with complementarity constraints and—although this type of program is generally NP-hard—we develop a branch-and-bound algorithm to solve it globally. We also derive bounds on key variables in the problem, which are valid asymptotically for increasing sample size. We compare our method with two well known global solvers, one of which requires the computed bounds. On random instances, our algorithm performs well in terms of both speed and robustness.

Keywords: Convex quadratic programming; complementarity constraints; branch-and-bound; quantile regression

1 Introduction

Least-squares linear regression [14] estimates the conditional expectation of an outcome scalar variable $\boldsymbol{b}$ by modeling

$$\boldsymbol{b} = \boldsymbol{a_1}^T x_1^* + \boldsymbol{\epsilon} \quad \text{with} \quad \mathrm{E}[\boldsymbol{\epsilon} \mid \boldsymbol{a_1} = a_1] = 0,$$

where $\boldsymbol{a_1}$ is an n_1-dimensional vector of covariates, x_1^* is an n_1-dimensional vector of coefficients, $\boldsymbol{\epsilon}$ is a random error component, and $\mathrm{E}[\cdot]$ denotes expectation. Clearly, $\mathrm{E}[\boldsymbol{b} \mid \boldsymbol{a_1} = a_1] = a_1^T x_1^*$, and computing an estimate $\hat{x}_1$ of x_1^* corresponds to minimizing the sum of squared residuals in a given sample $(b, A_1) \in \mathbb{R}^{m\times(1+n_1)}$, where the rows of (b, A_1) correspond to the individual observations. Specifically, $\hat{x}_1 := \arg\min_{x_1} \|b - A_1 x_1\|_2$.

*Department of Management Sciences, University of Iowa, Iowa City, IA, 52242-1994, USA. Email: guanglin-xu@uiowa.edu.

†Department of Management Sciences, University of Iowa, Iowa City, IA, 52242-1994, USA. Email: samuel-burer@uiowa.edu.

Quantile regression [4, 11] is related to least squares linear regression in spirit, but it models the conditional quantile of the outcome variable. For simplicity, we restrict our attention to the median case:

$$\boldsymbol{b} = \boldsymbol{a_1}^T x_1^* + \boldsymbol{\epsilon} \quad \text{with} \quad P(\boldsymbol{\epsilon} \le 0 \mid \boldsymbol{a_1} = a_1) = \tfrac{1}{2}.$$

Given a sample (b, A_1), the associated estimation problem is to minimize the absolute deviation:

$$\hat{x}_1 \in \operatorname{Arg}\min_{x_1} \|b - A_1 x_1\|_1 \qquad \longleftrightarrow \qquad \begin{array}{ll} \min_{x_1, x_3^+, x_3^-} & e^T x_3^+ + e^T x_3^- \\ \text{s. t.} & x_3^+ - x_3^- = b - A_1 x_1 \\ & x_3^+, x_3^- \ge 0. \end{array}$$

Here $x_1 \in \mathbb{R}^{n_1}$, $x_3^+, x_3^- \in \mathbb{R}^m$, and $e \in \mathbb{R}^m$ is the vector of all ones. (Note that we reserve the notation x_2 for the next paragraph.)

When there is sampling bias, i.e., sampling exhibits $P(\boldsymbol{\epsilon} \le 0 \mid \boldsymbol{a_1} = a_1) \ne \frac{1}{2}$, the estimate $\hat{x}_1$ provided by quantile regression may be inaccurate. In such cases, the presence of additional covariates $\boldsymbol{a_2}$, called *instruments*, can often be exploited to correct the bias [5, 9], i.e., sampling with both $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ properly exhibits $P(\boldsymbol{\epsilon} \le 0 \mid \boldsymbol{a_1} = a_1, \boldsymbol{a_2} = a_2) = \frac{1}{2}$. While this could serve as the basis for a model $\boldsymbol{b} = \boldsymbol{a_1}^T x_1^* + \boldsymbol{a_2}^T x_2^* + \boldsymbol{\epsilon}$, the hope is to minimize the effect of $\boldsymbol{a_2}$ so that the model depends clearly on the endogenous covariates $\boldsymbol{a_1}$, not the instruments $\boldsymbol{a_2}$. For example, the most desirable case would have $x_2^* = 0$.

Hence, in *instrumental variable quantile regression (IVQR)*, we define the estimator $\hat{x}_1$ of x_1^* such that the instruments $\boldsymbol{a_2}$ do not help in the conditional quantile. In other words, we (ideally) choose $\hat{x}_1$ such that

$$0 \in \operatorname{Arg}\min_{x_2} \|b - A_1 \hat{x}_1 - A_2 x_2\|_1$$

where $x_2 \in \mathbb{R}^{n_2}$ is a variable and (b, A_1, A_2) is the sample data. (Such a desirable $\hat{x}_1$ may not exist; see the next paragraph.) The corresponding LP is

$$\begin{array}{ll} \min_{x_2, x_3^+, x_3^-} & e^T x_3^+ + e^T x_3^- \\ \text{s. t.} & x_3^+ - x_3^- + A_2 x_2 = b - A_1 \hat{x}_1 \\ & x_3^+, x_3^- \ge 0 \end{array} \tag{1}$$

Note that $\hat{x}_1$ is not a variable in (1). Rather, given an estimate $\hat{x}_1$ of x_1^*, the purpose of (1) is to verify that $\hat{x}_2 = 0$ leads to minimal model error. So the overall IVQR problem is to find a value $\hat{x}_1$ having this desired property. This is a type of inverse optimization problem because we desire that part of the optimal solution have a pre-specified value (namely, $\hat{x}_2 = 0$).

In actuality, there may not exist an $\hat{x}_1$ providing an optimal $\hat{x}_2 = 0$ as just described. So instead we choose $\hat{x}_1$ such that that $\hat{x}_2$ optimizes (1) with minimum Euclidean norm. We will show in Section 2.1 that the resulting problem is a *convex quadratic program with complementarity constraints* (CQPCC), which is generally NP-hard to solve.

The IVQR problem was introduced in [5], where the authors carried out a statistical analysis of the estimation of x_1^* and provided asymptotic normality and standard-error calculations. For $n_1 = 1$, the authors presented a simple, effective enumeration procedure for calculating the estimate. However, they also pointed out that, for larger n_1, their enumeration procedure would suffer from the curse of dimensionality. This provides another perspective on the difficulty of solving the CQPCC mentioned in the previous paragraph.

Our paper is organized as follows. In Section 1.1, we briefly review the relevant literature, especially inverse optimization, partial inverse optimization, linear programs with complementarity constraints, and techniques for non-convex quadratic programs, and in Section 1.2, we establish the notation we use in the paper. Then in Section 2, we discuss the IVQR problem in detail. Section 2.1 formulates IVQR as a CQPCC and proposes a tractable relaxation by dropping the complementarity constraints. In Section 2.2, we derive valid bounds on key variables in IVQR, which hold asymptotically for increasing sample size in both light- and heavy-tailed models. The bounds will be used by one of the two global optimization solvers in Section 4 but are also of independent interest.

In Section 3, we propose a convex-QP-based B&B algorithm to solve IVQR globally. Our B&B algorithm works by enforcing the complementarity constraints via linear constraints in nodes of the tree. We detail the scheme and structure of the B&B algorithm in Section 3.1 and describe important implementation issues in Section 3.2. Section 4 empirically compares our algorithm with two optimization solvers—Couenne (version 0.4) and CPLEX (version 12.4)—on three types of randomly generated instances. In particular, CPLEX solves a mixed-integer model of the CQPCC using the bounds derived in Section 2.2. We conclude that our algorithm is quite efficient and robust. Section 5 gives some final thoughts.

1.1 Related literature

As mentioned above, the IVQR problem is a type of inverse optimization problem [2, 21] because ideally we would like the optimal solution $\hat{x}_2$ of (1) to be zero. Since the other variables x_3^+, x_3^- in (1) do not have desired values, IVQR is in fact a type of *partial inverse optimization problem*, which is similar to a regular inverse problem except that only certain parts of the desired optimal solution are specified. Research on partial inverse optimization problems has been active in recent decades; see [3, 7, 12, 17, 18, 19, 20]. In particular, Heuberger [8] was one of the first to investigate partial inverse optimization problems. In many cases, partial inverse optimization is NP-hard. For example, solving partial inverse linear programming typically involves explicitly handling the complementarity conditions of the primal-dual optimality conditions.

In Section 2.1, we will show that IVQR can be formulated as a convex QP with complementarity constraints (CQPCC). Even *linear programs with complementarity constraints* (LPCCs) are known to be NP-hard since, for example, they can be used to formulate NP-hard nonconvex quadratic optimization problems [16]; see also [13, 10]. It is well known that LPCCs can be formulated as mixed-integer programs when the nonnegative variables involved in the complementarity constraints are explicitly bounded. In Section 4, we will employ a similar technique to reformulate IVQR as a convex quadratic mixed-integer program, which can be solved by CPLEX (version 12.4). Complementarity constraints can also be handled using general techniques for bilinear problems although we do not do so in this paper; see [6, 15] for example.

1.2 Notation

$\mathbb{R}^n$ refers to n-dimensional Euclidean space represented as column vectors, and $\mathbb{R}^{m\times n}$ is the set of real $m \times n$ matrices. The special vector $e \in \mathbb{R}^n$ consists of all ones. For $v \in \Re^n$, both v_i and $[v]_i$ refer to the i-th component of v. For $v, w \in \mathbb{R}^n$, the Hadamard product of v and w is denoted by $v \circ w := (v_1 w_1, ..., v_n w_n)^T$. For a matrix A, we denote by the row vector A^i the i-th row of A. For a scalar $p \geq 1$, the p-norm of $v \in \Re^n$ is defined as $\|v\|_p := (\sum_{i=1}^n |v_i|^p)^{1/p}$. The ∞-norm is defined as $\|v\|_\infty := \max_{i=1}^n |v_i|$. For a given minimization problem, the notation $\operatorname{Arg\,min}$ denotes the set of optimal solutions. If the optimal solution set is known to be a singleton, i.e., there is a unique optimal solution, we write $\arg\min$ instead. Our probability and statistics notation is standard. $P(\cdot)$ and $\mathrm{E}[\cdot]$ denote probability and expectation, respectively, and $P(\cdot|\cdot)$ and $\mathrm{E}[\cdot|\cdot]$ are the conditional variants. For an event E, we also write $\mathbb{1}\{E\}$ for the indicator function for E.

2 The IVQR Problem and Its Details

In this section, we formulate the IVQR problem as a CQPCC (convex quadratic program with complementarity constraints) and state a natural CQP relaxation that will serve as the root node in our B&B algorithm discussed in Section 3. We also derive asymptotic bounds on critical variables in the CQPCC that hold with high probability as the sample size increases.

The bounds will in particular be required by one of the other global solvers in Section 4.

2.1 A CQPCC representation of the IVQR problem

Recall problem (1), which expresses our goal that $\hat{x}_2 = 0$ lead to minimal model error given the estimate $\hat{x}_1$ of x_1^*. Its dual is

$$\begin{array}{ll} \max_y & (b - A_1\hat{x}_1)^T y \\ \text{s.t.} & A_2^T y = 0 \\ & -e \le y \le e \end{array} \tag{2}$$

Note that $A_2^T y = 0$ reflects that (1) optimizes only with respect to x_2, while $\hat{x}_1$ is considered fixed. The full optimality conditions for (1) and (2), including complementary slackness, are

$$x_3^+ - x_3^- + A_2 x_2 = b - A_1\hat{x}_1$$

$$x_3^+, x_3^- \ge 0 \tag{3}$$

$$A_2^T y = 0 \tag{4}$$

$$-e \le y \le e \tag{5}$$

$$x_3^+ \circ (e - y) = x_3^- \circ (y + e) = 0 \tag{6}$$

Now consider the full IVQR problem in which x_1 is a variable. The optimality conditions just stated allow us to cast the IVQR problem as the task of finding a feasible value of x_1 satisfying the following system, where x_2, x_3^+, x_3^-, and y are also variables:

$$x_3^+ - x_3^- + A_1 x_1 + A_2 x_2 = b \tag{7}$$

$$\text{(3)–(6)}$$

$$x_2 = 0.$$

In comparison to the preceding optimality conditions, equation (7) highlights that x_1 is a variable, and the equation $x_2 = 0$ expresses our goal that zero is the optimal solution of (1). As mentioned in the Introduction, however, the constraint $x_2 = 0$ may be too stringent, and so we relax it to the weaker goal of finding a solution $(x_1, x_2, x_3^+, x_3^-, y)$ such that x_2 has minimum Euclidean norm:

$$\begin{array}{ll} \min_{x_1,x_2,x_3^+,x_3^-,y} & \|x_2\|_2^2 \\ \text{s.t.} & \text{(3)–(7)}. \end{array} \tag{8}$$

This is our CQPCC formulation of the IVQR problem. In particular, the objective taken together with (3)–(5) and (7) form a CQP (convex quadratic program), and (6) enforces the complementarity constraints. For completeness, we prove that (8) is feasible.

Proposition 1. *The IVQR problem (8) is feasible.*

Proof. Consider the primal problem (1) with $\hat{x}_1$ fixed. As x_3^+ and x_3^- are nonnegative, their difference $x_3^+ - x_3^-$ is equivalent to a vector of free variables. Hence, (1) is feasible. Furthermore, as x_3^+, and x_3^- are nonnegative, the objective function $e^T x_3^+ + e^T x_3^-$ of (1) is bounded below, and hence (1) has an optimal solution. Then so does the dual (2) by strong duality. Those primal and dual optimal solutions, in addition, satisfy complementary slackness, exhibiting a feasible solution of (8). □

We present an alternative formulation (9) of (8), which will be the basis of the rest of the paper since it will prove convenient for the development in Section 3. We first add slack variables to (8) to convert all inequalities (except nonnegativity) into equations:

$$
\begin{array}{ll}
\min_{x_1,x_2,x_3^+,x_3^-,y,s^+,s^-} & \|x_2\|_2^2 \\
\text{s. t.} & x_3^+ - x_3^- + A_1x_1 + A_2x_2 = b, \quad x_3^+, x_3^- \geq 0 \\
 & A_2^T y = 0, \quad y + s^+ = e, \quad -e + s^- = y, \quad s^+, s^- \geq 0 \\
 & x_3^+ \circ s^+ = x_3^- \circ s^- = 0
\end{array}
$$

where $s^+, s^- \in \mathbb{R}^m$. Then we eliminate y:

$$
\begin{array}{ll}
\min_{x_1,x_2,x_3^+,x_3^-,s^+,s^-} & \|x_2\|_2^2 \\
\text{s. t.} & x_3^+ - x_3^- + A_1x_1 + A_2x_2 = b, \quad x_3^+, x_3^- \geq 0 \\
 & A_2^T(e - s^+) = 0, \quad e - s^+ = -e + s^-, \quad s^+, s^- \geq 0 \\
 & x_3^+ \circ s^+ = x_3^- \circ s^- = 0.
\end{array}
\tag{9}
$$

Due to the presence of the complementarity constraints, (9) is very likely difficult to solve since even linear programs with complementarity constraints (LPCCs) are NP-hard [13]. We will propose in Section 3, however, that (9) can be solved practically by a B&B algorithm, which employs polynomial-time CQP relaxations. For example, if we simply eliminate the complementarities, the resulting relaxation is tractable:

$$
\begin{array}{ll}
\min_{x_1,x_2,x_3^+,x_3^-,s^+,s^-} & \|x_2\|_2^2 \\
\text{s. t.} & x_3^+ - x_3^- + A_1x_1 + A_2x_2 = b, \quad x_3^+, x_3^- \geq 0 \\
 & A_2^T(e - s^+) = 0, \quad e - s^+ = -e + s^-, \quad s^+, s^- \geq 0.
\end{array}
\tag{10}
$$

This relaxation will indeed serve as the root relaxation in Section 3, and all node relaxations will be derived from it. It can be easily solved by numerous solvers.

2.2 Variable bounds

The B&B algorithm that we will present in Section 3 can solve (9) directly, even though the feasible set is unbounded. However, one of the other algorithms, with which we will compare, requires *a priori* bounds on the variables x_3^+ and x_3^- of (9). So in this subsection, we derive bounds for these variables. The derived bounds are also of interest from the statistical point of view.

Since the difference $x_3^+ - x_3^-$ is closely related to the error $\boldsymbol{\epsilon}$ of the model, it suffices to bound $\boldsymbol{\epsilon}$. However, one can expect that $\boldsymbol{\epsilon}$ is unbounded in general, and so some additional assumptions are required to bound $\boldsymbol{\epsilon}$ with high probability as the sample size m grows larger. We will focus on two specific, representative examples—one in which $\boldsymbol{\epsilon}$ has light tails and one in which the tails of $\boldsymbol{\epsilon}$ are heavy—and we prove explicit bounds on $\boldsymbol{\epsilon}$ that hold with high probability for large m. These bounds will subsequently be incorporated into (9) and used in Section 4 by one of the solvers.

Suppose that data (b, A) with $\epsilon := b - Ax^*$ is a random sample following the quantile-regression model

$$\boldsymbol{b} = \boldsymbol{a}^T x^* + \boldsymbol{\epsilon} \quad \text{with} \quad P(\boldsymbol{\epsilon} \leq 0 \mid \boldsymbol{a} = a) = \tfrac{1}{2}.$$

This is exactly the model considered in this paper except that the subscript 1 appearing on $\boldsymbol{a}$, a, and A has been dropped for notational convenience. We start by stating two lemmas that will facilitate the details of Example 1 (light tails) and Example 2 (heavy tails) below. The proofs are included in the Appendix.

Lemma 1. *For a random sample $(b, A) \in \mathbb{R}^{m\times(1+n)}$ with $\epsilon := b - Ax^*$ and a given constant $C > 1$, the probability $P\left(\|\epsilon\|_\infty > C\|b\|_\infty\right)$ is bounded above by both*

$$P\left(\frac{C}{C+1} < \frac{\|\epsilon\|_\infty}{\|Ax^*\|_\infty} < \frac{C}{C-1}\right)$$

and

$$m \max_{i=1}^{m} P\left(|\epsilon_i| > C \max_{k=1}^{m}\left\{|\epsilon_k| \cdot \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\}\right\}\right).$$

where $\mathbb{1}$ is the indicator function.

Lemma 2. *For any normal random variable $Z \sim \mathcal{N}(0, \sigma^2)$ and any $\theta \geq 1$, it holds that*

$$\frac{1}{2\theta} \cdot \epsilon(\theta) \leq P(Z > \theta\sigma) \leq \frac{1}{\theta} \cdot \epsilon(\theta), \qquad \text{where } \epsilon(\theta) := \frac{1}{\sqrt{2\pi}} \exp(-\theta^2/2).$$

In addition, consider q identically distributed copies $Z_1, \ldots, Z_q$ of Z, where q is large enough *so that* $\log(q) \geq 1$ *and* $q/(8\pi \log(q)) \geq \sqrt{q}$*. If* $\theta = \sqrt{\log(q)}$*, then*

$$P\left(\max_{1\leq p\leq q} Z_p \leq \theta\sigma\right) \leq \exp(-q^{1/4}).$$

We are now ready to give the light- and heavy-tailed examples that suggest reasonable asymptotic bounds on the error. In particular, both Examples 1 and 2 show that the bound $2\|b\|_\infty$ will be appropriate (with high probability for large sample size) for many situations of interest. So we can enforce $x_3^+ \leq 2\|b\|_\infty e$ and $x_3^- \leq 2\|b\|_\infty e$ in the formulation (9) of IVQR.

Example 1 (light tails)**.** *For the case* $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$*, let* $(b, A) \in \Re^m \times (1+n)$ *be a random sample with* $\epsilon := b - Ax^*$*. Then the inequality* $\|\epsilon\| \leq 2\|b\|_\infty$ *holds almost surely as* $m \to \infty$*.*

To explain Example 1, set $C = 2$. Lemma 1 implies $P(\|\epsilon\|_\infty > 2\|b\|_\infty) \leq m \max_{i=1}^m p_i$, where

$$p_i := P\left(|\epsilon_i| > 2\max_k \left\{|\epsilon_k| \cdot \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\}\right\}\right).$$

We claim that each product $m \cdot p_i \to 0$ as $m \to \infty$. In particular, we will show that, independently of i,

$$p_i \leq \exp(-\tfrac{m}{72}) + 2(\tfrac{3}{m})^2 + \exp\left(-(\tfrac{m}{3})^{1/4}\right) \tag{11}$$

so that

$$\begin{aligned} P(\|\epsilon\|_\infty > 2\|b\|_\infty) &\leq m\left(\exp(-\tfrac{m}{72}) + 2(\tfrac{3}{m})^2 + \exp\left(-(\tfrac{m}{3})^{1/4}\right)\right) \\ &= \exp\left(\log(m) - \tfrac{m}{72}\right) + \tfrac{18}{m} + \exp\left(\log(m) - (\tfrac{m}{3})^{1/4}\right) \\ &\to 0. \end{aligned}$$

This shows that one can asymptotically expect the error to be at most $2\|b\|_\infty$.

To prove the inequality (11), fix the index i; say $i = m$ without loss of generality. If more than $q := \lfloor \frac{m}{3} \rfloor$ of the terms $\epsilon_k a_k^T x^*$ are nonnegative (including the first q terms without loss of generality) and $|\epsilon_m| \leq 2\max_{1\leq k\leq q} |\epsilon_k|$, then

$$\begin{aligned} |\epsilon_m| &\leq 2\max_{k=1}^{q} |\epsilon_k| \\ &= 2\max_{k=1}^{q}\{|\epsilon_k| \cdot \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\}\} \\ &\leq 2\max_{k=1}^{m}\{|\epsilon_k| \cdot \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\}\}. \end{aligned}$$

Logically, this ensures the contrapositive implication

$$\begin{aligned}|\epsilon_m| > 2 \max_{k=1}^{m} \{|\epsilon_k| \cdot \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\}\} \quad &\implies \\ &\sum_{k=1}^{m} \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\} \leq q \quad \text{or} \quad |\epsilon_m| > 2 \max_{k=1}^{q} |\epsilon_k|.\end{aligned}$$

So $p_m \leq \alpha + \beta$, where

$$\begin{aligned}\alpha &:= P\left(\sum_{k=1}^{m} \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\} \leq q\right) \\ \beta &:= P\left(|\epsilon_m| > 2 \max_{k=1}^{q} |\epsilon_k|\right).\end{aligned}$$

We next bound α and β separately.

Because each $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$ conditional on a_k with $P(\epsilon_k \geq 0 \mid \boldsymbol{a} = a_k) = P(\epsilon_k \leq 0 \mid \boldsymbol{a} = a_k) = \frac{1}{2}$, we have $P(\epsilon_k a_k^T x^* \geq 0 \mid \boldsymbol{a} = a_k) = \frac{1}{2}$. Then, interpreting both ϵ_k and a_k as random, this means $P(\epsilon_k a_k^T x^* \geq 0) = \frac{1}{2}$. Therefore each $Y_k := \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\} - \frac{1}{2}$ is a bounded random variable with mean 0. By Azuma's inequality, we have

$$\alpha = P\left(\sum_{k=1}^{m} Y_k \leq q - \frac{m}{2}\right) \leq P\left(\sum_{k=1}^{m} Y_k \leq -\frac{m}{6}\right) \leq \exp(-\tfrac{m}{72}).$$

Finally, to bound β, set $t = 2\sqrt{\log(q)}$. Logically,

$$|\epsilon_m| > 2 \max_{k=1}^{q} |\epsilon_k| \quad \implies \quad |\epsilon_m| > t\sigma \quad \text{or} \quad 2 \max_{k=1}^{q} |\epsilon_k| \leq t\sigma.$$

So $\beta \leq \gamma + \delta$, where

$$\begin{aligned}\gamma &:= P\left(|\epsilon_1| > t\sigma\right) \\ \delta &:= P\left(2 \max_{k=1}^{q} |\epsilon_k| \leq t\sigma\right).\end{aligned}$$

To bound γ, we use Lemma 2 with $\theta = t$ to show that, for m large enough,

$$\begin{aligned}\gamma &= P(|\epsilon_1| > t\sigma) = 2P(\epsilon_1 > t\sigma) \\ &\leq \frac{2}{t} \cdot \frac{1}{\sqrt{2\pi}} \exp(-t^2/2) = \frac{1}{\sqrt{\log(q)}} \cdot \frac{1}{\sqrt{2\pi}} \exp(-2\log(q)) \\ &\leq q^{-2} \leq 2(\tfrac{3}{m})^2.\end{aligned}$$

To bound δ, we apply Lemma 2 with $\theta = \frac{1}{2}\sqrt{\log(q)}$ to conclude $\delta \le \exp(-q^{1/4}) \le \exp(-(\frac{m}{3})^{1/4})$.

In total, we have $p_i \le \alpha + \beta \le \alpha + \gamma + \delta \le \exp(-\frac{m}{72}) + 2(\frac{3}{m})^2 + \exp(-(\frac{m}{3})^{1/4})$, which is (11).

Example 2 (heavy tails)**.** *Consider the case when* $\mathrm{E}[|\boldsymbol{a}^T x^*|^q] \le K$ *for some integer* $q > 0$ *and scalar* $K > 0$ *and* $\boldsymbol{\epsilon}$ *satisfies* $P(|\boldsymbol{\epsilon}| < t) \le 1 - t^{-k}$, *where* $k + 1 < q$. *Let* $(b, A) \in \Re^m \times (1+n)$ *be a random sample with* $\epsilon := b - Ax^*$. *Then the inequality* $\|\epsilon\| \le 2\|b\|_\infty$ *holds almost surely as* $m \to \infty$.

From Jensen's inequality and the standard inequality $\|\cdot\|_\infty \le m^{1/q}\|\cdot\|_q$, we see

$$\mathrm{E}\left[\|Ax^*\|_\infty\right] \le \|E[Ax^*]\|_\infty \le m^{1/q}\,\|E[Ax^*]\|_q = m^{1/q}\left(\sum_{i=1}^m \mathrm{E}\left[|a_i^T x^*|^q\right]\right)^{1/q} \le m^{1/q}K^{1/q}.$$

Hence, by Markov's inequality,

$$P(\|Ax^*\|_\infty > m^{1/(k+1)}K^{1/q}) \le \frac{E[\|Ax^*\|_\infty]}{m^{1/(k+1)}K^{1/q}} \le \frac{m^{1/q}K^{1/q}}{m^{1/(k+1)}K^{1/q}} = \frac{m^{1/q}}{m^{1/(k+1)}},$$

which goes to 0 as $m \to \infty$ because $k + 1 < q$. Moreover,

$$P(\|\epsilon\|_\infty < t) = \Pi_{i=1}^m P(|\epsilon_i| < t) \le (1 - t^{-k})^m,$$

which, substituting $t = Cm^{1/(k+1)}K^{1/q}$, implies

$$\begin{aligned} P(\|\epsilon\|_\infty < Cm^{1/(k+1)}K^{1/q}) &\le (1 - (Cm^{1/(k+1)}K^{1/q})^{-k})^m \\ &= (1 - C^{-k}m^{-k/(k+1)}K^{-k/q})^m. \end{aligned}$$

Note that the last quantity goes to 0 as $m \to \infty$ because $k/(k+1) < 0$. By Lemma 1 and taking $C \ge 2$,

$$\begin{aligned} P(\|\epsilon\|_\infty > C\|b\|_\infty) &\le P\left(\tfrac{C}{C+1} \le \tfrac{\|\epsilon\|_\infty}{\|Ax\|_\infty} \le \tfrac{C}{C-1}\right) \\ &\le P\left(\tfrac{\|\epsilon\|_\infty}{\|Ax\|_\infty} \le C\right) \\ &= P\left(\|\epsilon\|_\infty \le C\|Ax\|_\infty\right) \\ &\le P(\|Ax^*\|_\infty > m^{1/(k+1)}K^{1/q}) + P(\|\epsilon\|_\infty \le Cm^{1/(k+1)}K^{1/q}) \\ &\to 0 + 0 = 0. \end{aligned}$$

3 A CQP-Based Branch and Bound Algorithm

In Section 2.1, we mentioned that the CQP relaxation (10)—and ones derived from it—would be used within a branch-and-bound (B&B) algorithm to solve IVQR via the reformulation (9). In this section we present the algorithm in detail and discuss important implementation issues. For the moment, we do not refer to the bounds derived in Section 2.2 since our own algorithm does not require them; we will need the bounds for CPLEX in Section 4.

3.1 The scheme and structure of the algorithm

Our B&B algorithm aims to enforce more and more of the complementarity constraints in (9) further and further down in a dynamically constructed tree. Complementarities are enforced using added linear equalities. For example, a single complementarity $[x_3^+]_i[s^+]_i = 0$ is enforced in one branch by $[x_3^+]_i = 0$ and in a second branch by $[s^+]_i = 0$. This is analogous to branching on a 0-1 binary variable z in integer programming, where one branch forces $z = 0$ and another $z = 1$.

To describe the B&B algorithm formally, for each node of the tree, let F_x^+ and F_s^+ be two disjoint subsets of the index set $\{1, \ldots, m\}$, and separately let F_x^- and F_s^- be two disjoint subsets of the same. Also define $G^+ := \{1, 2, ..., m\} \backslash (F_x^+ \cup F_s^+)$ and $G^- := \{1, 2, ..., m\} \backslash (F_x^- \cup F_s^-)$. The node will enforce complementarities associated with $F_x^+ \cup F_s^+$ and $F_x^- \cup F_s^-$ by solving the following CQP relaxation:

$$
\begin{array}{lll}
\min_{x_1,x_2,x_3^+,x_3^-,s^+,s^-} & \|x_2\|_2^2 & \\
\text{s. t.} & x_3^+ - x_3^- + A_1x_1 + A_2x_2 = b, \quad x_3^+, x_3^- \geq 0 & \\
& A_2^T(e - s^+) = 0, \quad e - s^+ = -e + s^-, \quad s^+, s^- \geq 0 & (12) \\
& [x_3^+]_i = 0 \ \ \forall\, i \in F_x^+, \qquad [s^+]_i = 0 \ \ \forall\, i \in F_s^+ & \\
& [x_3^-]_j = 0 \ \ \forall\, j \in F_x^-, \qquad [s^-]_j = 0 \ \ \forall\, j \in F_s^-. &
\end{array}
$$

This problem is the basic (or root) relaxation (10) with added linear inequalities that enforce the complementarities $[x_3^+]_i[s^+]_i = 0$ for all $i \in F_x^+ \cup F_s^+$ and $[x_3^-]_j[s^-]_j = 0$ for all $j \in F_x^- \cup F_s^-$. On the other hand, any complementarities corresponding to G^+ or G^- are relaxed compared to (9). Note that, while the feasible set of problem (12) is unbounded in the variables x_3^+ and x_3^-, the objective function is bounded below since it is nonnegative. Hence, (12) is always solvable or infeasible.

Now we discuss how to create new nodes in the tree, i.e., how to branch on a given node that is associated with sets $F_x^+, F_s^+, F_x^-, F_s^-, G^+, G^-$. First, select some $i \in G^+$ or $j \in G^-$ corresponding to a complementarity that has yet to be enforced. Then two child nodes are created as follows: one with $F_x^+ \leftarrow F_x^+ \cup \{i\}$ and one with $F_s^+ \leftarrow F_s^+ \cup \{i\}$ if an i was selected, or one with $F_x^- \leftarrow F_x^- \cup \{j\}$ and one with $F_s^- \leftarrow F_s^- \cup \{j\}$ if a j was selected. G^+ and G^- are also updated for the children as: $G^+ \leftarrow G^+ \backslash \{i\}$ if i was selected, or $G^- \leftarrow G^- \backslash \{j\}$ if j was selected. This form of branching leads to two special cases worth pointing out: (i) the root node corresponds to $F_x^+ = F_s^+ = F_x^- = F_s^- = \emptyset$ with no complementarities enforced; (ii) leaf nodes correspond to $G^+ = G^- = \emptyset$ with all complementarities enforced.

The next important concept of the B&B algorithm is fathoming, that is, removing a node from further consideration and, in particular, not branching on it. We may fathom a node if we are certain that the relaxation associated with that node does not contain any solutions for (9) that are better than what we have already encountered during the course of the algorithm. Let GUB ("global upper bound") denote the best feasible value of (9) encountered so far, including possibly gotten at the current node, e.g., when the optimal solution to the relaxation (12) is actually feasible for (9). Then we can fathom if the optimal value of (12) is greater than or equal to GUB. This fathoming rule includes a special case when (12) is infeasible, in which case the relaxation optimal value can be considered $+\infty$.

Another equally important concept for our B&B algorithm is how we evaluate the nodes in the tree. Evaluating a node involves solving the CQP relaxation (12) corresponding to the node and then fathoming the node (if possible) or branching on the node (if fathoming is not possible and if it is not a leaf node).

With these basic concepts introduced, we now have a whole picture of our B&B algorithm. Starting with the root node, the algorithm evaluates, adds, and fathoms nodes from the tree at each iteration. The algorithm finishes when all nodes generated by the algorithm have been evaluated and fathomed. The final GUB value is the optimal value, and the solution associated with GUB is an optimal solution.

3.2 Implementation issues

In this subsection, we will describe some important implementation issues for our B&B algorithm.

One of the most crucial issues is the method chosen to solve the CQP relaxation (12) at each node of the tree. As mentioned in Section 2.1, we use CPLEX 12.4 in our codes. In particular, we solve the relaxations using the dual simplex CQP pivoting method based on the following two considerations. First, preliminary results indicated that the dual pivoting method was more numerically stable and accurate compared to other methods, including the primal simplex CQP pivoting method and the interior-point (barrier) method. Secondly, the dual pivoting method is particularly useful for re-optimizing a problem when primal constraints are added, which is the case between parent and child nodes in our algorithm.

Another important issue is how to choose which complementarity on which to branch when child nodes are created. (Please refer to the conceptual discussion in the prior subsection.) We apply a maximum-violation approach, which is similar to most-fractional branching in integer programming. Suppose that we have just solved the CQP relaxation (12) at a node and have an associated optimal solution $(\hat{x}_1, \hat{x}_2, \hat{x}_3^+, \hat{x}_3^-, \hat{s}^+, \hat{s}^-)$. We then compute

$$i \in \operatorname{Arg}\max_k \left\{[\hat{x}_3^+]_k[\hat{s}^+]_k\right\} \quad \text{and} \quad j \in \operatorname{Arg}\max_l \left\{[\hat{x}_3^-]_l[\hat{s}^-]_l\right\}$$

and choose i if $[\hat{x}_3^+]_i[\hat{s}^+]_i$ is larger than $[\hat{x}_3^-]_j[\hat{s}^-]_j$ and otherwise choose j.

We also give a bit more detail about handling the optimal solution $(\hat{x}_1, \hat{x}_2, \hat{x}_3^+, \hat{x}_3^-, \hat{s}^+, \hat{s}^-)$ of the relaxation (12) at each node. (In our experience, CPLEX is quite stable and can always deliver an optimal solution or determine that (12) is infeasible.) First, the solution is is checked for feasibility in the original problem (9) up to a relative tolerance of 10^{-8}. If so, we update GUB, and the node is fathomed. On the other hand, if the solution is infeasible at this tolerance, then it means that some complementarity is violated, and we must branch.

In addition, we use a relative optimality tolerance for fathoming a node by its relaxed objective value. Given $\varepsilon > 0$, the optimal value LB ("lower bound") of the relaxation, and the current GUB, the node is fathomed if $(\text{GUB} - \text{LB})/\max\{1, \text{GUB}\} < \varepsilon$. Note that GUB is nonnegative in our setting, and we take $\epsilon = 10^{-6}$ in our implementation.

A best-bound strategy for selecting the next node to evaluate is employed in our B&B algorithm. After branching, the algorithm sorts the remaining nodes in the tree by their estimated relaxed optimal values, which are just the relaxed optimal values of their parents. Then the algorithm chooses the next node to evaluate as the one with the lowest estimated relaxed optimal value.

We adopt a heuristic method to generate an initial GUB for our B&B algorithm. From Proposition 1, we know that (1) is feasible and bounded below by 0 and thus has optimal solutions for each fixed x_1. Hence, we first solve the unconstrained linear least squares problem:

$$\hat{x}_1 = \arg\min_{x_1} \|A_1 x_1 - b\|_2.$$

Then, we solve (1) with $\hat{x}_1$ to get an optimal $\hat{x}_2$. We note that $\hat{x}_1$ and $\hat{x}_2$ are part of a feasible solution of the IVQR problem. Then our initial GUB is set to $\|\hat{x}_2\|_2^2$.

We can also add implied complementarity constraints to the nodes in the B&B tree. Consider a node in which $[s^+]_i = 0$ is enforced. Since the linear constraints of (12) imply $[s^+]_i + [s^-]_i = 2$, we have $[s^-]_i = 2$, which in turn implies $[x_3^-]_i = 0$. Similarly, when a node enforces $[s^-]_j = 0$, we see $[x_3^+]_j = 0$. Thus, the CQP in (12) can be strengthened as follows:

$$
\begin{array}{ll}
\min_{x_1,x_2,x_3^+,x_3^-,s^+,s^-} & \|x_2\|_2^2 \\
\text{s. t.} & x_3^+ - x_3^- + A_1 x_1 + A_2 x_2 = b, \quad x_3^+, x_3^- \geq 0 \\
& A_2^T(e - s^+) = 0, \quad e - s^+ = -e + s^-, \quad s^+, s^- \geq 0 \\
& [x_3^+]_i = 0 \ \ \forall\, i \in F_x^+, \qquad [s^+]_i = [x_3^-]_i = 0 \ \ \forall\, i \in F_s^+ \\
& [x_3^-]_j = 0 \ \ \forall\, j \in F_x^-, \qquad [s^-]_j = [x_3^+]_j = 0 \ \ \forall\, j \in F_s^-.
\end{array}
$$

4 Computational Experiments

In this section, we first discuss the procedure to generate test instances and describe three types of instances. Then, for the computational study, we test our B&B algorithm, referred to as *QPBB*, against two well known solvers on the three types of instances. All computational experiments were performed on an Intel Core 2 Quad CPU Q6700 running at 2.46 GHz×4 under the Linux operating system. We note that—when the compared algorithms report optimal solutions on an instance—we have found experimentally that the optimal values agree up to a relative accuracy of 10^{-6}. In other words, when an algorithm claims to have solved an instance optimally, it is independently verified by the other algorithms up to six significant figures.

4.1 Generation and description of instances

Test instances of the IVQR problem are randomly generated by specifying the following parameters: sample size (m), number of endogenous covariates (n_1), and number of instruments (n_2). We enforce that $m \geq n_1 = n_2$. Let $\mathcal{U}(0,1)$ and $\mathcal{N}(0,1)$ denote the standard uniform and normal distributions, respectively. Then our steps to generate random data $(b, A_1, A_2) \in \mathbb{R}^m \times \mathbb{R}^{m\times n_1} \times \mathbb{R}^{m\times n_2}$ for a single instance are shown in Algorithm 1. This procedure guarantees $P(\boldsymbol{\epsilon} \leq 0 \mid \boldsymbol{a_1} = a_1) \neq \frac{1}{2}$ but $P(\boldsymbol{\epsilon} \leq 0 \mid \boldsymbol{a_1} = a_1, \boldsymbol{a_2} = a_2) = \frac{1}{2}$. We refer the reader to [5] for details of the procedure and relevant theory.

Algorithm 1 Random Instance Generator

Inputs: m, n_1, n_2, $\alpha := (1, 2, \ldots, n_1)^T$, and $\beta := 2\alpha$
Outputs: (b, A_1, A_2)
for $i = 1, \ldots, m$ **do**
  Generate u from $\mathcal{U}(0,1)$
  For $j = 1, \ldots, n_2$, sample $[A_2]_{ij}$ independently from $\mathcal{N}(0,1)^2$ ("squared normal")
  For $j = 1, \ldots, n_1$, calculate $[A_1]_{ij} = [A_2]_{ij} + u\beta_j$
  Calculate $b_i = \sum_{j=1}^{n_1} (1 - u + u\alpha_j)[A_1]_{ij}$
end for

Instance Type	# Instances	(m, n_1, n_2)
Small-IVQR	100	(50, 5, 5)
Medium-IVQR	100	(100, 5, 5)
Large-IVQR	100	(200, 5, 5)

Table 1: Details of the three types of test instances.

We generate three types of instances using the above procedure. The first type, called *Small-IVQR*, contains 100 random instances with $(m, n_1, n_2) = (50, 5, 5)$. The second type, called *Medium-IVQR*, contains 100 random instances with $(m, n_1, n_2) = (100, 5, 5)$. The *Large-IVQR* type contains 100 random instances with $(m, n_1, n_2) = (200, 5, 5)$. We expect the Large-IVQR instances to be the most challenging since $2m$ equals the number of complementarities in (9), which directly effects the (potential) number of nodes in the B&B tree. Table 1 describes the three types of instances. We also generated instances with $n_2 > n_1$, but the computational results were the same as those presented with $n_1 = n_2$. So here we focus on the case $n_1 = n_2$.

4.2 Tests against Couenne

We first test our algorithm against the open-source global solver Couenne (version 0.4) [1] on the Small-IVQR instances. For our algorithm, we set the feasibility tolerance to 10^{-8} and the fathoming tolerance to 10^{-6}. Couenne incorporates a generic tolerance, which affects a number of aspects of its performance; we choose 10^{-6}. The per-instance time limit is set to 1800 seconds (30 minutes) for both algorithms. If the time limit is reached for either method on a particular instance, the method will return its current best objective value (GUB), but of course it is possible that the GUB returned is not the global optimal value.

We report our main observations as follows; see Figure 1. Our algorithm outperforms Couenne on all 100 instances in terms of computation time. Note that we present log-log plots of the CPU times, and the straight line defines $y = x$ in Figure 1. In addition, all instances are solved by our algorithm within the time limit, but Couenne exceeds time limit (30 minutes) on 1 instance denoted by a square in Figure 1. Finally, notice that our algorithm is faster than Couenne by up to three orders of magnitude.

We also ran Couenne on the Medium-IVQR instances. However, Couenne could not complete computation within the time limit of 1,800 seconds (30 minutes) on any of the 100 instances, whereas QPBB could solve each instance within the time limit (see Section 4.3.2 below for more details). Couenne also faced similar difficulties with the Large-IVQR instances.

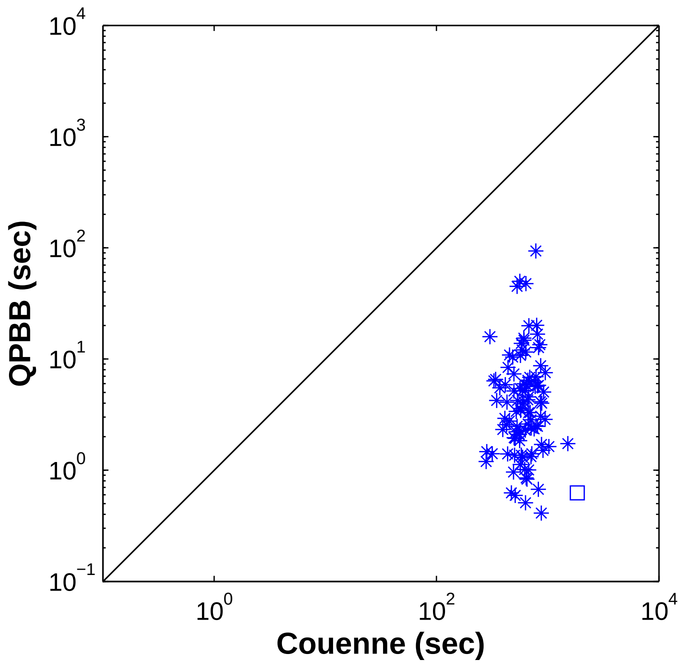

Figure 1: Comparison of the computation times (in seconds) between our B&B algorithm and Couenne on the 100 Small-IVQR instances. The x- and y-axes are log scales, and the diagonal line defines $y = x$. An asterisk represents an instance for which both algorithms report optimality; a square represents that Couenne exceeded the time limit of 1800 seconds.

4.3 Tests against CPLEX

In this subsection, we compare our algorithm against CPLEX's mixed-integer QP solver (version 12.4) applied to a mixed-integer programming (MIP) formulation of (9). For the MIP formulation, we use a standard technique for handling complementarity constraints by introducing binary variables together with a "big-M" approach. The big-M constants require *a priori* bounds on x_3^+ and x_3^-, and so we employ the bounds derived in Section 2.2. The MIP formulation is

$$\begin{array}{ll}
\min_{x_1,x_2,x_3^+,x_3^-,s^+,s^-,z^+,z^-} & \|x_2\|_2^2 \\
\text{s. t.} & x_3^+ - x_3^- + A_1 x_1 + A_2 x_2 = b, \quad x_3^+, x_3^- \geq 0 \\
& A_2^T(e - s^+) = 0, \quad e - s^+ = -e + s^-, \quad s^+, s^- \geq 0 \\
& x_3^+ \leq 2\,\|b\|_\infty\, z^+, \quad s^+ \leq 2(e - z^+) \\
& x_3^- \leq 2\,\|b\|_\infty\, z^-, \quad s^- \leq 2(e - z^-) \\
& z^+, z^- \in \{0,1\}^m,
\end{array} \qquad (13)$$

where $2\|b\|_\infty$ is the upper bound on both x_3^+ and x_3^- from Section 2.2.

We compare our algorithm with CPLEX on the Small-IVQR, Medium-IVQR, and Large-

IVQR instances. For the comparison, the per-instance time limit is set to 900 seconds (15 minutes) for both algorithms on the Small-IVQR instances, 1800 seconds (30 minutes) on the Medium-IVQR instances, and 10800 seconds (3 hours) on the Large-IVQR instances. If the time limit is reached for either algorithm on any instance, the algorithm will return its current best GUB. We use the same tolerances as those in Section 4.2 for our algorithm, and we use the default tolerances for CPLEX. For a fair comparison, we also introduce the upper bounds $x_3^+ \leq 2\,\|b\|_\infty\, e$ and $x_3^- \leq 2\,\|b\|_\infty\, e$ into formulation (9) and its relaxations within our algorithm.

4.3.1 Tests on the Small-IVQR instances

We have the following three main observations; see Figure 2. First, both our algorithm and CPLEX report optimal solutions for all 100 instances within the time limit of 900 seconds. In addition, our algorithm outperforms CPLEX on 82 instances out of the 100 while it performs slightly worse on the remaining 18. Finally, notice that as the computation time required for an instance increases for either algorithm, our algorithm is more likely to perform better than CPLEX. This trend can be observed even more clearly on the Medium-IVQR instances discussed next.

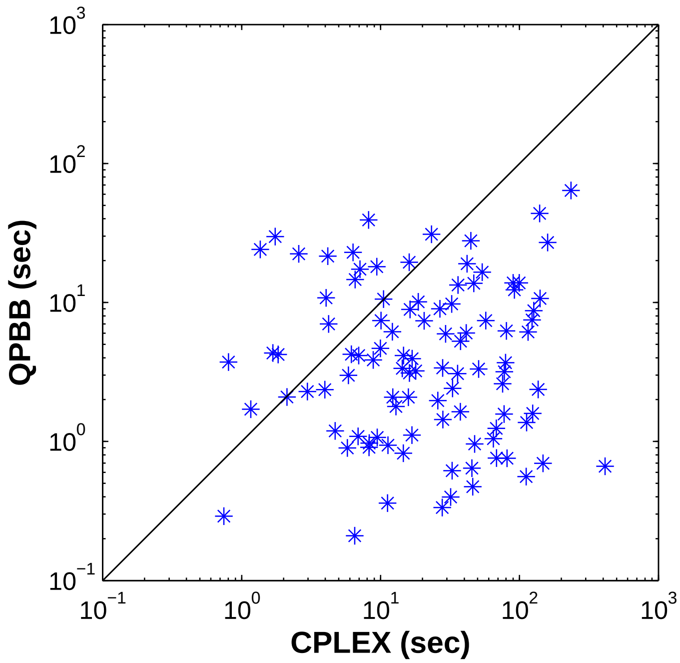

Figure 2: Comparison of the computation time (in seconds) between our B&B algorithm and CPLEX on 100 Small-IVQR instances. The x- and y-axes are log scales, and the diagonal line defines $y = x$. An asterisk represents an instances for which both algorithms report optimality.

4.3.2 Tests on Medium-IVQR instances

As just shown in Section 4.3.1, our algorithm is more likely to perform better than CPLEX on Small-IVQR instances requiring less computation time generally. Thus, in order to make a more solid conclusion on this point, we compare the performance of the two algorithms on the Medium-IVQR instances. Figure 3 shows the computation details. We see that our algorithm can solve all 100 instances to optimality while CPLEX only reports optimal solutions for 57 instances. For CPLEX's remaining 43 instances, it reaches the time limit on 42 instances and encounters numerical issues on 1. In addition, our algorithm performs better than CPLEX on 53 of the 57 for which both algorithms report optimal solutions.

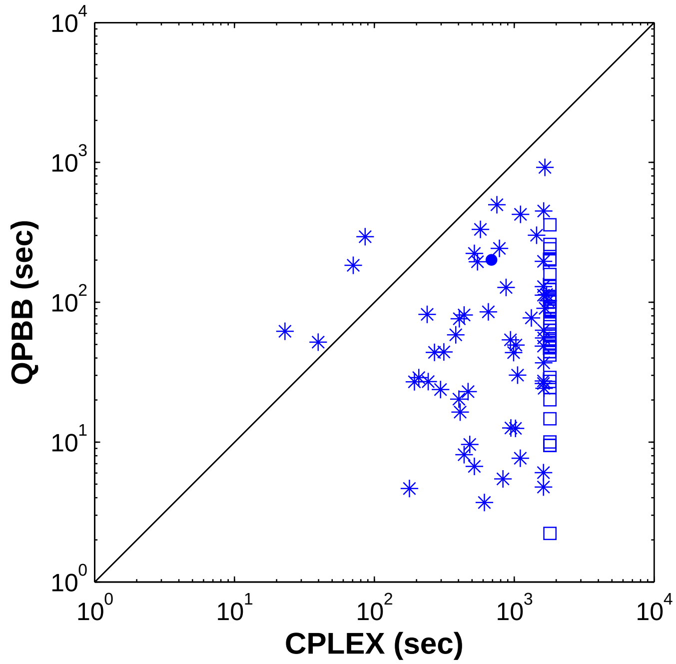

Figure 3: Comparison of the computation time (in seconds) between our B&B algorithm and CPLEX on the 100 Medium-IVQR instances. The x- and y-axes are log scales, and the diagonal line defines $y = x$. An asterisk represents an instance where both algorithms report optimality; a square represents an instances where CPLEX exceeded the time limit; and a dot represents an instance where CPLEX encountered numerical issues.

4.4 Additional experiments

To explore how well our algorithm scales, we tested both our algorithm and CPLEX on the Large-IVQR instances. We set the time limit for each algorithm to solve each instance as 10800 seconds (3 hours). However, CPLEX could not solve any instance within the time limit. Thus, we do not report the detailed results of CPLEX here. We kept the other settings

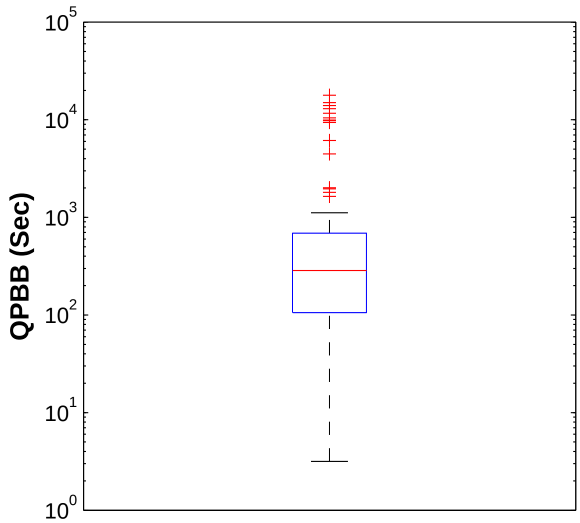

Figure 4: The box plot of the computation time (in seconds) of our B&B algorithm over the 88 Large-IVQR instances for which optimal solutions are obtained within the time limit. The y-axis is a log scale.

of our B&B algorithm the same as described in Section 4.2. We find that our algorithm report optimal solutions for 88 of 100 instances within the time limit. Figure 4 shows the box plot for the computation time of our B&B algorithm on those 88.

To improve the performance of our algorithm, we developed a variety of variants. In terms of branching rules, we also tried strong-branching and pseudo-cost branching rules in addition to the maximum-violation rule. It turned out the maximum-violation branching rule performs the best among these rules. For node selection rules, we tried depth-first search and breadth-first search rules besides the best-lower-bound rule. Results indicated that the best-lower-bound rule performed best.

5 Conclusions

In this paper, we have studied a problem arising in statistics called instrumental variable quantile regression (IVQR) and proposed a CQPCC formulation. We have introduced a relaxation scheme, which was then incorporated into a CQP-based B&B algorithm to solve the IVQR problem. We have tested our algorithm on three types of randomly generated instances against two well-known global solvers including Couenne and CPLEX. The computational results show that our B&B algorithm solves IVQR efficiently and robustly.

Acknowledgments

The authors would like to acknowledge the participation and contribution of Alexandre Belloni, who supplied the IVQR problem as an interesting global, quadratic problem. He also provided the code to generate instances of the IVQR problem consistent with the statistics literature.

References

[1] Couenne, a solver for non-convex MINLP problems. Technical report, http://www.coin-or.org/Couenne/.

[2] R. K. Ahuja and J. B. Orlin. Inverse optimization. *Operations Research*, 49(5):771–783, October 2001.

[3] M. C. Cai, C. W. Duin, X. G. Yang, and J. Z. Zhang. The partial inverse minimum spanning tree problem when weight increase is forbidden. *European Journal of Operational Research*, 188(2):23–28, January 2008.

[4] C. Chen. An introduction to quantile regression and the quantreg procedure. In *Proceedings of the Thirtieth Annual SAS Users Group International Conference*, 2005.

[5] V. Chernozhukov and C. Hansen. Instrumental variable quantile regression: A robust inference approach. *Journal of Econometrics*, 142(1):379 –398, January 2008.

[6] S. A. Gabriel, R. Gareia-Bertrand, P. Sahakij, and A. J. Concjo. A practical approach to approximate bilinear functions in mathematical programming problems by using schur's decomposition and sos type 2 variables. *Journal of Operational Research Society*, 57(8):995–1004, August 2006.

[7] S. Gentry. Partial inverse linear programming. Technical Report LIDS-P-2532, Massachusetts Inst. Tech., Lab for Information and Decision Systems, December 2001.

[8] C. Heuberger. Inverse combinatorial optimization: a survey on problems, methods, and results. *Journal of Combinatorial Optimization*, 8(3):329–361, Septermber 2004.

[9] J. Horowitz and S. Lee. Nonparametric instrumental variables estimation of a quantile regression model. *Econometrica*, 75(4):1191–1208, 2007.

[10] J. Hu. On the global solution of linear programs with linear complementarity constraints. *SIAM Journal on Optimization*, 19(1):445–471, February 2008.

[11] R. Koenker and K. F. Hallock. Quantile regression. *Journal of Economic Perspectives*, 15(4):143–156, Fall 2001.

[12] T. C. Lai and J. B. Orlin. The complexity of preprocessing. Technical report, Sloan School of Management, MIT, 2003.

[13] O. L. Mangasarian, 1997. Regularized linear programs with equilibrium constraints.

[14] D. Montgomery, E. Peck, and G. Vining. *Introduction to Linear Regression Analysis*. Wiley, 2012.

[15] H. D. Sherali and A. Alameddine. A new reformulation - linearization technique for bilinear programming problems. *Journal of Global Optimization*, 2(4):379–410, 1992.

[16] D. Vandenbussche and G. Nemhauser. A polyhedral study of nonconvex quadratic programs with box constraints. *Mathematical Programming*, 102(3):531–557, 2005.

[17] Q. Wang. Partial inverse most unbalanced spanning tree problem. *Przeglad Elektrotechniczny*, 88(1b):111–114, 2012.

[18] X. G. Yang. Complexity of partial inverse assignment problem and partial inverse cut problem. *RAIRO - Operations Research*, 35(1):117–126, January 2001.

[19] X. G. Yang and J. Z. Zhang. Inverse sorting problem by minimizing the total weighted number of changes and partial inverse sorting problems. *Computational Optimization and Applications*, 36(1):55–66, January 2007.

[20] X. G. Yang and J. Z. Zhang. Partial inverse assignment problems under l_1 norm. *Operations Research Letters*, 35(1):23–28, January 2007.

[21] J. Zhang and Z. Liu. Calculating some inverse linear programming problems. *Journal of Computational and Applied Mathematics*, 72(2):215–434, August 1996.

A Proof of Lemma 1

Proof. Since $Ax^* + \epsilon = b$, we have from the triangle inequality that $\|\epsilon\|_\infty \leq \|b\|_\infty + \|Ax^*\|_\infty$ and $\|Ax^*\|_\infty \leq \|b\|_\infty + \|\epsilon\|_\infty$. So the event $\|\epsilon\|_\infty > C\|b\|_\infty$ implies

$$\|\epsilon\|_\infty > C\|b\|_\infty \geq C(\|\epsilon\|_\infty - \|Ax^*\|_\infty) = C\|\epsilon\|_\infty - C\|Ax^*\|_\infty$$

and

$$\|\epsilon\|_\infty > C\|b\|_\infty \geq C(\|Ax^*\|_\infty - \|\epsilon\|_\infty) = C\|Ax^*\|_\infty - C\|\epsilon\|_\infty,$$

which together imply

$$\frac{C}{C+1} < \frac{\|\epsilon\|_\infty}{\|Ax^*\|_\infty} < \frac{C}{C-1}.$$

This proves the first bound. To prove the second, we note that, by definition,

$$\|\epsilon\|_\infty > C\|b\|_\infty \qquad \Longleftrightarrow \qquad \max_{i=1}^{m} |\epsilon_i| > C \max_{k=1}^{m} |a_k^T x^* + \epsilon_k|,$$

and so $\|\epsilon\|_\infty > C\|b\|_\infty$ implies that $|\epsilon_i| > C \max_{k=1}^m |a_k^T x^* + \epsilon_k|$ holds for at least one specific i. Hence,

$$\begin{aligned} P\left(\|\epsilon\|_\infty > C\|b\|_\infty\right) &\leq \sum_{i=1}^m P\left(|\epsilon_i| > C \max_{k=1}^m |a_k^T x^* + \epsilon_k|\right) \\ &\leq m \max_{i=1}^m P\left(|\epsilon_i| > C \max_k |a_k^T x^* + \epsilon_k|\right). \end{aligned} \tag{14}$$

Next, we claim $|a_k^T x^* + \epsilon_k| \geq |\epsilon_k| \cdot \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\}$ for each k. If $\epsilon_k a_k^T x^* < 0$, this is certainly true. Otherwise, if $\epsilon_k a_k^T x^* \geq 0$, then ϵ_k and $a_k^T x^*$ have the same (possibly zero) signs and hence $|a_k^T x^* + \epsilon_k| \geq |\epsilon_k|$. Thus,

$$|\epsilon_i| > C \max_{k=1}^m |a_k^T x^* + \epsilon_k| \qquad \Longrightarrow \qquad |\epsilon_i| > C \max_{k=1}^m \left\{|\epsilon_k| \cdot \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\}\right\}$$

and so

$$P\left(|\epsilon_i| > C \max_{k=1}^m |a_k^T x^* + \epsilon_k|\right) \leq P\left(|\epsilon_i| > C \max_{k=1}^m \left\{|\epsilon_k| \cdot \mathbb{1}\{\epsilon_k a_k^T x^* \geq 0\}\right\}\right). \tag{15}$$

Combining inequalities (14) and (15), we achieve the second bound. □

B Proof of Lemma 2

Proof. The first two-sided inequality is a standard fact about the normal distribution. For the last inequality, we have

$$P\left(\max_{1\leq p\leq q} Z_p \leq \theta\sigma\right) = \Pi_{p=1}^q P(Z_p \leq \theta\sigma) = (1 - P(Z > \theta\sigma))^q.$$

By the first part of the lemma and the standard fact that $0 < x < 1$ and $a > 0$ imply $(1-x)^a \leq \exp(-ax)$,

$$\begin{aligned} (1 - P(Z > \theta\sigma))^q &\leq \left(1 - \frac{1}{2\theta} \cdot \epsilon(\theta)\right)^q \\ &\leq \exp\left(-\frac{q}{2\theta} \cdot \epsilon(\theta)\right). \end{aligned}$$

Now substituting the definition of $\epsilon(\theta)$ and $\theta = \sqrt{\log(q)}$, we see

$$\begin{aligned} \exp\left(-\frac{q}{2\theta} \cdot \epsilon(\theta)\right) &= \exp\left(-\frac{q}{2\theta} \cdot \frac{1}{\sqrt{2\pi}} \exp(-\log(q)/2)\right) \\ &= \exp\left(-\frac{q}{2\theta} \cdot \frac{1}{\sqrt{2\pi}} q^{-1/2}\right) \\ &= \exp\left(-\frac{\sqrt{q}}{2\sqrt{\log(q)}} \cdot \frac{1}{\sqrt{2\pi}}\right) \\ &\leq \exp\left(-q^{1/4}\right), \end{aligned}$$

where the last inequality follows from the assumption that $q/(8\pi \log(q)) \geq \sqrt{q}$. This proves the result. □