# Supplement To "Implementing Convex Optimization in R: Two Econometric Examples"

Zhan Gao* and Zhentao Shi†

## 1 Introduction

Due to space limitations, we append the main text with these supplementary materials. For the completeness of the paper, we briefly describe the benchmark simulation designs in the original papers, and report more details about the empirical application of Chinese GDP evaluation. Although in the main text the classifier-Lasso is introduced in the linear regression form of penalized least squares (PLS) to simplify presentation, Su et al. (2016) develop the theory with the general profile log-likelihood function as well as the penalized GMM (PGMM). This nonlinear profile log-likelihood can be reformulated with exponential cones, and PGMM can be handled under the same optimization framework as PLS.

Replication files are hosted at the `Github` repository https://github.com/zhao-gao/convex_prog_in_econometrics. We also provide instruction of installing `Rmosek` at the end of this document.

## 2 Classifier-Lasso

### 2.1 Data Generating Process

The simulation in the main text follows the linear static panel data DGP (DGP 1) in Su et al. (2016, p.2237). The observations are drawn from three groups with the proportion $n_1 : n_2 : n_3 = 0.3 : 0.3 : 0.4$. The observed data $(y_{it}, x_{it})$ are generated from

$$x_{it} = \left(0.2\mu_i^0 + e_{it1}, 0.2\mu_i^0 + e_{it2}\right)'$$
$$y_{it} = \beta_i^{0\prime} x_{it} + \mu_i^0 + \varepsilon_{it},$$

*Zhan Gao: zhangao@usc.edu. Address: Department of Economics, University of Southern California, 3620 South Vermont Ave. Kaprielian Hall, 300 Los Angeles, CA 90089-0253, USA.

†Zhentao Shi (corresponding author): zhentao.shi@cuhk.edu.hk or shizhentao@gmail.com. Address: Department of Economics, 9F Esther Lee Building, the Chinese University of Hong Kong, Sha Tin, New Territories, Hong Kong SAR, China. Tel: (852) 3943-1432. Fax (852) 2603-5805.

where $\mu_i^0$, $\varepsilon_{it}$, $e_{it1}$, $e_{it2} \sim$ i.i.d.N $(0,1)$. The true coefficients are $(0.4, 1.6)$, $(1,1)$, $(1.6, 0.4)$ for the three groups, respectively. In the implementation, the C-Lasso tuning parameter is specified as $\lambda = \frac{1}{2}\widehat{\sigma}_Y^2 T^{-\frac{1}{3}}$, where $\widehat{\sigma}_Y^2$ is the sample variance of demeaned dependent variable. Given the number of groups, we run the simulation for $R = 500$ replications and report the RMSE of the estimation of $\alpha_1$ and the probability of correct classification (correct ratio). Performance is measured by

$$\text{RMSE}(\hat{\beta}_1) = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{K}\frac{n_k}{n}\left(\hat{\alpha}_{k,1}^{(r)} - \alpha_{k,1}^{0}\right)^2}, \text{ and Correct Ratio} = \frac{1}{nR}\sum_{r=1}^{R}\sum_{i=1}^{n}\mathbf{1}\left\{\hat{g}_i^{(r)} = g_i^0\right\},$$

where $\widehat{g}_i^{(r)}$ and $g_i^{(0)}$ are the estimated and the true group identity of the $i$'s individual, respectively, and $\mathbf{1}\{\cdot\}$ is the indicator function.

## 2.2 More about Empirical Application

Su et al. (2016) provides an information criterion to automatic the choice of tuning parameters $K$ and $\lambda$. We go over $K = 1, 2, 3, 4$ and $\lambda = c\text{var}\,(y)\,T^{-\frac{1}{3}}$ where the constant $c$ takes one of the 10 equally spaced values between 0.001 and 0.01. In each of the three model specifications, the information criterion identifies two groups, as listed in Table 1.

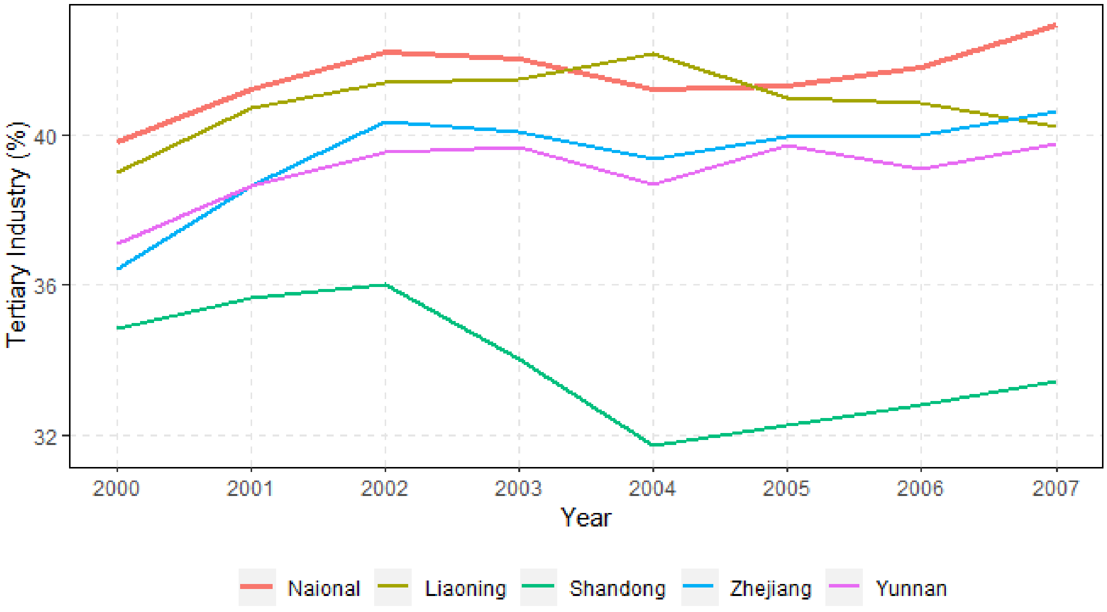
In the first specification where all five indicators are included, we observe small deviance across computing platforms. Liaoning, Zhejiang, Shandong and Yunnan are moved into group 2 according to `Rmosek` whereas they are left in the group 1 in `CVX` results. Since this is a short $T$ panel data with $T = 8$, the numerical stability is more fragile when we include more regressors.

A key observation in Chen et al. (2019) is that group 1 gathers Beijing, Shanghai and Hainan, the three provinces with the highest GDP shares of the tertiary sector associated with the provision of services. The `Rmosek` result retains this feature. As shown in the figure following Table 1, except for Liaoning in 2004, the tertiary industry GDP shares of the four provinces excluded from the Beijing-Shanghai-Hainan group doe not reach the national aggregate share. Given their relatively lower shares, it is sensible that `Rmosek` removes them out of the high-share group.

Table 1: Classification Results: Chen et al. (2019, Table A11)

| All 5 Indicators | | No Light | | No Light and Tax | |
|---|---|---|---|---|---|
| Group 1 | Group 2 | Group 1 | Group 2 | Group 1 | Group 2 |
| Beijing | Tianjin | Beijing | Tianjin | Beijing | Tianjin |
| Hebei | Jilin | Inner Mongolia | Hebei | Hebei | Liaoning |
| Shanxi | Heilongjiang | Liaoning | Shanxi | Shanxi | Shanghai |
| Inner Mongolia | **Liaoning** | Jilin | Heilongjiang | Inner Mongolia | Zhejiang |
| Shanghai | Jiangxi | Shanghai | Zhejiang | Jilin | Shandong |
| Jiangsu | Henan | Jiangsu | Jiangxi | Heilongjiang | Henan |
| Anhui | Hunan | Anhui | Shandong | Jiangsu | Hunan |
| Fujing | Guangdong | Fujian | Henan | Anhui | Guangdong |
| Hubei | **Zhejiang** | Hubei | Hunan | Fujian | Chongqing |
| Hainan | Guangxi | Hainan | Guangdong | Jiangxi | Guizhou |
| Qinghai | Chongqing | Qinghai | Guangxi | Hubei | Yunnan |
| Xinjiang | Sichuan | | Chongqing | Guangxi | Shaanxi |
| | **Shandong** | | Sichuan | Hainan | |
| | Guizhou | | Guizhou | Sichuan | |
| | Shaanxi | | Yunnan | Gansu | |
| | Gansu | | Shaanxi | Qinghai | |
| | **Yunnan** | | Gansu | Ningxia | |
| | Ningxia | | Ningxia | Xinjiang | |
| | | | Xinjiang | | |

Note: Provinces in bold highlight different results from Chen et al. (2019). The GDP Shares of Tertiary Sector of these provinces are reported in the following time series along with the national average.



### 2.3 Code Snippets and Extra Examples of C-Lasso

In this section, we provide several code snippets to demonstrate the key formulation steps. Notice that the code blocks below are illustrations of the corresponding parts of the optimization problems, which are extracted from the full functions and hence not self-contained. For implementation and replication, please refer to the repository via the link provided in Introduction.

#### 2.3.1 Lasso and C-Lasso

We start with Lasso. In matrix notation, the Lasso problem is

$$\min_{\theta} \lambda \left( e'\beta^{+} + e'\beta^{-} \right) + \frac{t}{n}$$

$$\text{s.t.} \quad \left[ \begin{array}{ccc|ccc} X & -X & I_n & & \mathbf{0}_{n\times 3} & \\ & \mathbf{0}_{2\times(n+2p)} & & -\frac{1}{2} & 1 & 0 \\ & & & -\frac{1}{2} & 0 & 1 \end{array} \right] \theta = \begin{bmatrix} y \\ -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \; \|(v, s)\|_2 \leq r, \; \beta^{+}, \beta^{-} \geq 0$$

where the inequality for a vector is taken elementwisely. The following annotated R code snippet implements the matrix form.

```r
P = list(sense = "min")

# Linear coefficients in objective
P$c = c(rep(lambda, 2*p), rep(0, n), 1/n, 0, 0)

# The matrix in linear constraints
A = as.matrix.csr(X)
A = cbind(A, -A, as(n, "matrix.diag.csr"), as.matrix.csr(0, n, 3))
A = rbind(A, cbind(as.matrix.csr(0, 2, 2*p + n),
                                  as.matrix.csr(c(-.5, -.5, 1, 0, 0, 1), 2, 3)))
P$A = as(A,"CsparseMatrix")

# Right-hand side of linear constraints
P$bc = rbind(c(y, -0.5, 0.5), c(y, -0.5, 0.5))

# Constraints on variables
P$bx = rbind(c(rep(0, 2 * p), rep(-Inf, n), rep(0, 3)), c(rep(Inf, 2*p+n+3)))

# Conic constraints
P$cones = matrix(list("QUAD", c(n+2*p+3, (2*p+1):(2*p+n), n+2*p+2)), 2, 1)
rownames(P$cones) = c("type", "sub")
```

```
result = mosek(P, opts = list(verbose = verb))
xx = result$sol$itr$xx
coef = xx[1:p] - xx[(p+1):(2*p)]
```

We then take a step further to C-Lasso. The convexity is manifest when we write the problem in matrix form

$$\min_{\alpha_{\tilde{k}},\theta} \frac{1}{nT} e' t + \frac{\lambda}{n} \gamma' w$$

$$\text{s.t.} \quad t_i \geq 0,\ \|(\nu_i, s_i)\|_2 \leq r_i,\ \|\mu_i\|_2 \leq w_i, \text{ for all } i = 1, 2, \cdots, n$$

$$\left[\begin{array}{cccccc} \operatorname{diag}(X_1, \ldots, X_n) & \mathbf{I}_{Tn} & \mathbf{0} & \multicolumn{2}{c}{} & \mathbf{0} \\ \mathbf{I}_{np} & \mathbf{0} & -I_{np} & \multicolumn{2}{c}{\mathbf{0}} & -\mathbf{1}_n \otimes \mathbf{I}_p \\ \multicolumn{3}{c}{\mathbf{0}} & \mathbf{I}_2 \otimes \mathbf{I}_n & -\frac{1}{2}\mathbf{1}_2 \otimes \mathbf{I}_n & \mathbf{0} \end{array}\right] \begin{bmatrix} \theta \\ \alpha_{\tilde{k}} \end{bmatrix} = \begin{bmatrix} y \\ \mathbf{0}_{np} \\ -\frac{1}{2} e_n \\ \frac{1}{2} e_n \end{bmatrix}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix and $\mathbf{1}_n$ is the $n \times 1$ column of ones.

Though more tedious than Lasso, the construction of the large matrix in the linear constraints is straightforward. The formulation of the conic constraints is illustrated in the following chunk of code.

```
CC = list()

# locate the variables related
bench = N*(2*p + TT) + p

for(i in 1:N){
        # find index of each variable
        s.i = bench + i
        r.i = bench + N + i
        nu.i = (N*p + (i-1)*TT + 1):(N*p + i*TT)
        w.i = bench + 3*N + i
        mu.i = (N*(TT+p) + (i-1)*p + 1):(N*(TT+p) + i*p)
        CC = cbind(CC, list("QUAD", c(r.i, nu.i, s.i)),
                                   list("QUAD", c(w.i, mu.i)) )
}
P$cones = CC
rownames(prob$cones) = c("type", "sub")
```

The penalty $\gamma_i$ can be coded as follows.

```
pen.generate = function(b, a, N, p, K, kk){

        # Output arg: gamma
        # Input args:
        #  b, a (estimate of last iteration)
        #  kk (current focused group)

        # compute all ||\beta_i - alpha_k||_2
        a.out.exp = aperm(array(a, c(K, p, N)), c(3, 2, 1))
        p.norm = sqrt(apply((b - a.out.exp)^2, c(1,3), sum))

        # leave kk out and take product
        ind = setdiff(1:K,kk)
        gamma = apply(p.norm[, ind], 1, prod)
        return(gamma)
}
```

### 2.3.2 Additional Example: Exponential/Logarithm Formulation

In limited dependent variable models it is common to see exponential, logarithm or power terms in objective functions. These nonlinear functions can be easily formulated with power cones and exponential cones.

**Example 1** (Poisson maximum likelihood estimator)**.** The Poisson maximum likelihood estimator is defined as

$$\min_{\beta} -\frac{1}{n}\sum_{i=1}^{n}\left(y_i x_i'\beta - \exp\left(x_i'\beta\right)\right)$$

where $y_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^p$ are observed data, and $\beta$ is the parameter of interests. This optimization problem involves the component $\exp\left(\sum_{j=1}^{p} x_{ij}\beta_j\right)$, which is non-separable. Define $v_i = x_i'\beta$, and the objective becomes

$$\min_{v,\beta} \frac{1}{n}\sum_{i=1}^{n}\left(-y_i v_i + \exp\left(v_i\right)\right).$$

As in REL, we introduce the auxiliary variable $t_i$ to replace $\exp\left(v_i\right)$ in the objective and the constraint $t_i \geq \exp\left(v_i\right)$ is equivalent to $\left(t_i, 1, v_i\right) \in \mathcal{K}_{\text{exp}}$ . The estimator can be then formulated as

$$\min_{t,v,\beta} \frac{1}{n}\sum_{i=1}^{n}\left(-y_i v_i + t_i\right) \quad \text{s.t. } v_i = x_i'\beta,\ \left(t_i, 1, v_i\right) \in \mathcal{K}_{\text{exp}}, \text{ for each } i$$

**Example 2** (Logistic regression)**.** Consider the simplest logistic regression

$$\max_{\beta} \sum_{i=1}^{n} y_i \left(x_i'\beta\right) - \log\left(1 + \exp\left(x_i'\beta\right)\right). \tag{1}$$

Introducing $t_i$ to replace the softplus function in the objective and $\phi_i = x_i'\beta$, we turn (1) into

$$\max_{t_i,\phi_i,\beta} \sum_{i=1}^{n} \left(y_i\phi_i + t_i\right) \quad \text{s.t.} - \log\left(1 + \exp\left(\phi_i\right)\right) \geq t_i, \;\; \phi_i = x_i'\beta \;\; \text{for each}\, i.$$

Notice that $-\log\left(1 + \exp\left(\phi_i\right)\right) \geq t_i$ is equivalent to $\exp\left(\phi_i + t_i\right) + \exp\left(t_i\right) \leq 1$. Introducing $u_i$ and $v_i$, we transform it into

$$u_i + v_i \leq 1, \quad \left(u_i, 1, \phi_i + t_i\right) \in \mathcal{K}_{\text{exp}}, \quad \left(v_i, 1, t_i\right) \in \mathcal{K}_{\text{exp}},$$

and then we reach the standard form

$$\begin{aligned} &\max_{\theta} \begin{bmatrix} \mathbf{1}_n & y & \mathbf{0}_{1\times(2n+p)} \end{bmatrix} \theta \\ \text{s.t.} \quad &\begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{I}_n & \mathbf{0}_{n\times 2n} & -X \\ \mathbf{0}_{n\times 2n} & \mathbf{I}_n & \mathbf{I}_n & \mathbf{0}_{n\times p} \end{bmatrix} \theta \begin{matrix} = \\ \leq \end{matrix} \begin{bmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{1}_n \end{bmatrix} \\ &\left(u_i, 1, \phi_i + t_i\right) \in \mathcal{K}_{\text{exp}}, \quad \left(v_i, 1, t_i\right) \in \mathcal{K}_{\text{exp}}, \;\; \text{for each}\, i \end{aligned}$$

where $\theta = (t, \phi, u, v, \beta)$.

### 2.3.3 Additional Example: Penalized GMM

We consider the linear panel data model with latent group structures and endogeneity. After first-differencing, we have

$$\Delta y_{it} = \beta_i'\Delta x_{it} + \Delta\varepsilon_{it}$$

Let $z_{it} \in \mathbb{R}^m$ for some $m \geq p$ be the instrumental variables for $\Delta x_{it}$. The penalized GMM estimator is defined as the solution $(\boldsymbol{\beta}, \alpha)$ to

$$\min_{\boldsymbol{\beta},\alpha} \frac{1}{nT^2} \sum_{i=1}^{n} \left\| W_i^{\frac{1}{2}} z_i \left(\Delta y_i - \Delta x_i \beta_i\right) \right\|_2^2 + \frac{\lambda}{n} \sum_{i=1}^{n} \prod_{k=1}^{K} \left\|\beta_i - \alpha_k\right\|_2$$

where $W_i$ is an $m \times m$ positive-definite symmetric weighting matrix. As in Shi (2016b), PGMM problem can be formulated as

$$\min_{\boldsymbol{\beta},\alpha} \frac{1}{nT^2} \sum_{i=1}^{n} \|\tilde{y}_i - \tilde{x}_i\beta_i\|_2^2 + \frac{\lambda}{n} \sum_{i=1}^{n} \prod_{k=1}^{K} \left\|\beta_i - \alpha_k\right\|_2$$

by the transformations $\tilde{y}_i = W_i^{\frac{1}{2}} z_i \Delta y_i$ and $\tilde{x}_i = W_i^{\frac{1}{2}} z_i \Delta x_i$. The following iterative algorithm is essentially the same as PLS.

## 3 Relaxed Empirical Likelihood

The formulation of REL is inspired by Dantzig selector (Candes and Tao, 2007). Dantzig selector slacks the sup-norm of the first-order condition for optimality. REL carries over the idea of relaxation into estimating a finite-dimensional parameter in a structural economic model defined by many moment equalities.

**Example 3** (Dantzig selector)**.** Similar to Lasso, Dantzig selector produces a sparse solution to the linear regression model. Dantzig selector can be written as

$$\min_{\beta} \|\beta\|_1 \quad \text{s.t.} \quad \|X'(y - X\beta)\|_\infty \leq \lambda,$$

where $\lambda$ is a tuning parameter. It can be immediately reformulated as a linear programming problem

$$\begin{aligned} &\min_{\beta^+,\beta^-} \; e'\beta^+ + e'\beta^- \\ \text{s.t.} \;\; &X'y - \lambda e \leq \left(X'X\right)\left(\beta^+ - \beta^-\right) \leq X'y + \lambda e \\ &\beta^+, \beta^- \geq 0. \end{aligned}$$

It is readily solvable using the `R` package `quantreg` (Koenker, 2017).

In constrast to Dantzig selector, REL uses a nonlinear objective function. It is still convex (in minus likelihood) but the `quantreg` which deals with linear programming problems only is no longer applicable. In the main text we formulate REL restriction with an exponential cone, as in the following snippet.

```r
# Exponential Cones
NUMCONES <- n
Prob$cones <- matrix(list(), nrow = 2, ncol = NUMCONES)
rownames(Prob$cones) <- c("type", "sub")
for (i in 1:n){
        Prob$cones[, i] <- list("PEXP", c(i, 2 * n + i, n + i))
}
```

### 3.1 Data Generating Process

The data generating process in Shi (2016a, Section 4.1) features the linear IV model with many IVs. The observed data $\{y_i\}_{i=1}^n$ are generated by the structural equation

$$y_i = x_i'\beta + e_i^{(0)}$$

where $\beta = (1,1)'$, $x_i = (x_{i1}, x_{i2})'$ are endogenous variables that are generated by $x_{i1} = 0.5z_{i1} + 0.5z_{i2} + e_i^{(1)}$ and $x_{i2} = 0.5z_{i3} + 0.5z_{i4} + e_i^{(2)}$, respectively, $e_i^{(0)}$ is the structural error, and $\left(e_i^{(1)}, e_i^{(2)}\right)$ are reduced-form errors. The observed data contains $m$ IVs $\{z_{ij}\}_{j=1}^m$ orthogonal to $e_i^{(0)}$ but the researcher does not know which one are relevant. We generate $\{z_{ij}\}_{j=1}^m \sim$ i.i.d.N $(0,1)$ and $\begin{pmatrix} e_i^{(0)} \\ e_i^{(1)} \\ e_i^{(2)} \end{pmatrix} \sim$ $\mathrm{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.25 & 0.15 & 0.15 \\ 0.15 & 0.25 & 0 \\ 0.15 & 0 & 0.25 \end{pmatrix}\right)$. The endogeneity comes from the correlation among all error terms. The orthogonality yields the moment restrictions $\mathbb{E}\left[z_i\left(y_i - x_i\beta\right)\right] = \mathbf{0}_m$ to identify $\beta$. We run $R = 500$ replications and report bias and RMSE for $\beta_1$ as Bias $= R^{-1}\sum_{r=1}^R \hat{\beta}_1 - \beta_1$ and RMSE $= \sqrt{R^{-1}\sum_{r=1}^R \left(\hat{\beta}_1 - \beta_1\right)^2}$.

## 4 Software Installation

Before we conclude this supplementary document, we briefly cover software installation if readers are interested in replicating our results and experimenting with their own convex optimization problems. CVXR is now available on CRAN as a standard R package. The latest version MOSEK 9.0 is shipped with Rmosek. To invoke install Rmosek in R, Windows users need Rtools. Once the prerequisites are satisfied, Rmosek can be installed by a command line similar to the following one:

```
source("<RMOSEKDIR>/builder.R")
attachbuilder()
install.rmosek()
```

More details are referred to the official installation manual at https://docs.mosek.com/9.0/rmosek/install-interface.html.

## References

Candes, E. and T. Tao (2007). The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics 35*(6), 2313–2351.

Chen, W., X. Chen, C.-T. Hsieh, and Z. Song (2019). A forensic examination of china's national accounts. *Brookings Papers on Economic Activities Spring*, 77–127.

Koenker, R. (2017). `quantreg`: Quantile regression r package version 5.33. https://cran.r-project.org/web/packages/quantreg/index.html.

Shi, Z. (2016a). Econometric estimation with high-dimensional moment equalities. *Journal of Econometrics 195*(1), 104–119.

Shi, Z. (2016b). Estimation of sparse structural parameters with many endogenous variables. *Econometric Reviews*, 1582–1608.

Su, L., Z. Shi, and P. C. Phillips (2016). Identifying latent structures in panel data. *Econometrica 84*(6), 2215–2264.