Supplementary material to "Covariate Selection in High-Dimensional Generalized Linear Models With Measurement Error"

Øystein Sørensen, Kristoffer H. Hellton,
Arnoldo Frigessi, Magne Thoresen

January 5, 2018

A Proof of Proposition 1

It follows from the Taylor series expansion (9) that

$$\mu\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right) = \mu\left(\mathbf{x}_i^T\boldsymbol{\beta}^0\right) - \sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r.$$

This gives, for $j = 1, \ldots, p$,

$$\begin{aligned}
&\frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\left\{y_i - \mu\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)\right\}\right| = \\
&\frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\left\{\epsilon_i + \sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r\right\}\right| \leq \\
&\frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\epsilon_i\right| + \frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r\right| \leq \\
&\lambda + \frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r\right| \leq \\
&\lambda + \frac{1}{n}\sum_{i=1}^{n}\left|w_{ij}\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r\right|,
\end{aligned}$$

where we inserted $\epsilon_i = y_i - \mu\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)$ in the first step, we used the triangle inequality in the second step, we inserted the left bound in (7) in the third step, and finally used the generalized triangle inequality. Next, we have

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\left|w_{ij}\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r\right| \leq \\
&\frac{1}{n}\left(\sum_{i=1}^{n} w_{ij}^2\right)^{\frac{1}{2}}\left[\sum_{i=1}^{n}\left\{\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r\right\}^2\right]^{\frac{1}{2}} = \\
&\frac{1}{\sqrt{n}}\left[\sum_{i=1}^{n}\left\{\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r\right\}^2\right]^{\frac{1}{2}},
\end{aligned}$$

where we used Hölder's inequality in the first step and the assumption (5) that the covariates are standardized to have mean zero and unit variance in the second step. We now note that the last term above is the ℓ_2-norm $\|\mathbf{v}\|_2$ of a vector $\mathbf{v} \in \mathbb{R}^n$ with elements

$$v_i = \sum_{r=1}^{\infty} \mu^{(r)}\left(\mathbf{w}_i^T \boldsymbol{\beta}^0\right)(r!)^{-1}\left(-\mathbf{u}_i^T \boldsymbol{\beta}^0\right)^r, \quad i = 1, \ldots, n.$$

We thus have

$$\begin{aligned}
&\frac{1}{\sqrt{n}}\left[\sum_{i=1}^{n}\left\{\sum_{r=1}^{\infty}\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^r\right\}^2\right]^{\frac{1}{2}} \leq \\
&\frac{1}{\sqrt{n}}\sum_{r=1}^{\infty}\left[\sum_{i=1}^{n}\left\{\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\right\}^2\left(-\mathbf{u}_i^T\boldsymbol{\beta}^0\right)^{2r}\right]^{\frac{1}{2}} \leq \\
&\frac{1}{\sqrt{n}}\sum_{r=1}^{\infty}\left[\sum_{i=1}^{n}\left\{\frac{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)}{r!}\right\}^2\|\mathbf{u}_i\|_\infty^{2r}\|\boldsymbol{\beta}\|_1^{2r}\right]^{\frac{1}{2}} \leq \\
&\sum_{r=1}^{\infty}\frac{\delta^r\|\boldsymbol{\beta}\|_1^r}{r!\sqrt{n}}\left[\sum_{i=1}^{n}\left\{\mu^{(r)}\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)\right\}^2\right]^{\frac{1}{2}} = \\
&\sum_{r=1}^{\infty}\frac{\delta^r\|\boldsymbol{\beta}\|_1^r}{r!\sqrt{n}}\left\|\boldsymbol{\mu}^{(r)}\left(\mathbf{W}\boldsymbol{\beta}^0\right)\right\|_2,
\end{aligned}$$

where we used the triangle inequality in the first step, Hölder's inequality in the second step, and finally used the right bound in (7) in the second last step.

Putting the pieces together, it follows that

$$\frac{1}{n}\left|\sum_{i=1}^{n} w_{ij}\left\{y_i - \mu\left(\mathbf{w}_i^T\boldsymbol{\beta}^0\right)\right\}\right| \leq \lambda + \sum_{r=1}^{\infty}\frac{\delta^r}{r!\sqrt{n}}\left\|\boldsymbol{\beta}^0\right\|_1^r\left\|\boldsymbol{\mu}^{(r)}\left(\mathbf{W}\boldsymbol{\beta}^0\right)\right\|_2$$

for $j = 1, \ldots, p$, which proves that $\boldsymbol{\beta}^0 \in \Theta$.

B Any Fixed Point of Algorithm 1 Solves (10)

Consider a fixed point of the Algorithm 1. At a fixed point, $\beta^{(k+1)} = \beta^{(k)}$. It follows that

$$z_i = \mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)} + \left[y_i - \mu\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}\right]\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}^{-1}, \; i = 1, \ldots, n$$

and

$$\mathbf{V}^{(r)} = \left[\mu^{(r)}\left\{\mathbf{w}_1^T\boldsymbol{\beta}^{(k+1)}\right\}, \ldots, \mu^{(r)}\left\{\mathbf{w}_n^T\boldsymbol{\beta}^{(k+1)}\right\}\right]^T,$$

and accordingly,

$$\tilde{w}_{ij} = \sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}w_{ij}, \quad \tilde{z}_i = \sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}z_i.$$

Inserting this into the constraint set in (15), we get for the left-hand side

$$
\begin{aligned}
&\tilde{\mathbf{W}}^T\left(\tilde{\mathbf{z}}-\tilde{\mathbf{W}}\boldsymbol{\beta}^{(k+1)}\right)= \\
&\sum_{i=1}^{n}\tilde{w}_{ij}\left(\tilde{z}_i-\sum_{l=1}^{p}\tilde{w}_{il}\beta_l^{(k+1)}\right)= \\
&\sum_{i=1}^{n}\sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}w_{ij}\left(\sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}z_i-\sum_{l=1}^{p}\sqrt{\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}}w_{ij}\beta_l^{(k+1)}\right)= \\
&\sum_{i=1}^{n}\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}w_{ij}\left(z_i-\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right)= \\
&\sum_{i=1}^{n}\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}w_{ij}\left(\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}+\left[y_i-\mu\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}\right]\mu'\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}^{-1}-\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right)= \\
&\sum_{i=1}^{n}w_{ij}\left(y_i-\mu\left\{\mathbf{w}_i^T\boldsymbol{\beta}^{(k+1)}\right\}\right)= \\
&\mathbf{W}\left(\mathbf{y}-\boldsymbol{\mu}\left\{\mathbf{W}\boldsymbol{\beta}^{(k+1)}\right\}\right).
\end{aligned}
$$

For the right-hand side of the inequality in (15), we get

$$
\lambda+\sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\left\|\boldsymbol{\beta}^{(k+1)}\right\|_1^r\left\|\mathbf{V}^{(r)}\right\|_2=\lambda+\sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\left\|\boldsymbol{\beta}^{(k+1)}\right\|_1^r\left\|\boldsymbol{\mu}^{(r)}\left\{\mathbf{W}\boldsymbol{\beta}^{(k+1)}\right\}\right\|_2.
$$

It follows that at any fixed point of Algorithm 1,

$$
\begin{aligned}
&\boldsymbol{\beta}^{(k+1)}= \\
&\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{\beta}\|_1:\frac{1}{n}\left\|\tilde{\mathbf{W}}^T\left(\tilde{\mathbf{z}}-\tilde{\mathbf{W}}\boldsymbol{\beta}\right)\right\|_\infty\leq\lambda+\sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\|\boldsymbol{\beta}\|_1^r\left\|\mathbf{V}^{(r)}\right\|_2\right\}= \\
&\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\|\boldsymbol{\beta}\|_1:\frac{1}{n}\left\|\mathbf{W}\left(\mathbf{y}-\boldsymbol{\mu}\left\{\mathbf{W}\boldsymbol{\beta}\right\}\right)\right\|_\infty\leq\lambda+\sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\|\boldsymbol{\beta}\|_1^r\left\|\boldsymbol{\mu}^{(r)}\left\{\mathbf{W}\boldsymbol{\beta}\right\}\right\|_2\right\}.
\end{aligned}
$$

Thus, any fixed point of Algorithm 1 is a solution to the original problem (10).

C Computing the Solution to (15)

We can simplify the computation of (15) by introducing the auxiliary variable $\mathbf{u}\in\mathbb{R}^p$ (see, e.g., Candes and Tao (2007, eq. (1.9)) or Boyd and Vandenberghe (2004, p. 617)). We then get the equivalent problem

$$
\begin{aligned}
&\text{mininimize } \mathbf{1}_p^T\mathbf{u} \text{ (with respect to } \mathbf{u},\boldsymbol{\beta}) \text{ subject to } -\mathbf{u}\leq\boldsymbol{\beta}\leq\mathbf{u}, \\
&-\sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\left(\mathbf{1}_p^T\mathbf{u}\right)^r\left\|\mathbf{V}^{(r)}\right\|_2\mathbf{1}_p+\frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}\boldsymbol{\beta}\leq\lambda\mathbf{1}_p+\frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{z}}, \text{ and} \\
&-\sum_{r=1}^{R}\frac{\delta^r}{r!\sqrt{n}}\left(\mathbf{1}_p^T\mathbf{u}\right)^r\left\|\mathbf{V}^{(r)}\right\|_2\mathbf{1}_p-\frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}\boldsymbol{\beta}\leq\lambda\mathbf{1}_p-\frac{1}{n}\tilde{\mathbf{W}}^T\tilde{\mathbf{z}}.
\end{aligned}
$$

When $R = 1$, (15) is thus equivalent to the linear program

$$\begin{aligned}
&\text{mininimize } \mathbf{1}_p^T \mathbf{u} \text{ (with respect to } \mathbf{u}, \boldsymbol{\beta}) \text{ subject to } -\mathbf{u} \leq \boldsymbol{\beta} \leq \mathbf{u}, \\
&-\frac{\delta}{\sqrt{n}} \mathbf{1}_p^T \mathbf{u} \left\| \mathbf{V}^{(1)} \right\|_2 \mathbf{1}_p + \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \boldsymbol{\beta} \leq \lambda \mathbf{1}_p + \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{z}}, \text{ and} \\
&-\frac{\delta}{\sqrt{n}} \mathbf{1}_p^T \mathbf{u} \left\| \mathbf{V}^{(1)} \right\|_2 \mathbf{1}_p - \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \boldsymbol{\beta} \leq \lambda \mathbf{1}_p - \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{z}},
\end{aligned}$$

which can be solved by standard software.

D Coordinate Descent Algorithm for GMUL

We describe here the coordinate descent algorithm used to solve (21). Our goal is to find a $\boldsymbol{\beta}$ minimizing the function

$$\begin{aligned}
&f(\boldsymbol{\beta}) = \\
&-n^{-1} \tilde{\mathbf{z}}^T \tilde{\mathbf{W}} \boldsymbol{\beta} + \boldsymbol{\beta}^T \left\{ (2n)^{-1} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} + \gamma_1 \mathbf{I}_p \right\} \boldsymbol{\beta} + \sum_{j=1}^{p} \omega_j^{(k)} |\beta_j| = \\
&-n^{-1} \sum_{i=1}^{n} \tilde{z}_i \sum_{j=1}^{p} \tilde{w}_{ij} \beta_j + (2n)^{-1} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \tilde{w}_{ij} \beta_j \right)^2 + \gamma_1 \sum_{j=1}^{p} \beta_j^2 + \sum_{j=1}^{p} \omega_j^{(k)} |\beta_j| .
\end{aligned}$$

The partial derivatives of $f(\boldsymbol{\beta})$ with respect to β_j, $j = 1 \ldots, p$, can be written as

$$\frac{\partial f}{\partial \beta_j} = -n^{-1} \sum_{i=1}^{n} \tilde{z}_i \tilde{w}_{ij} + n^{-1} \sum_{i=1}^{n} \tilde{w}_{ij} \sum_{l \neq j} \tilde{w}_{il} \beta_l + n^{-1} \beta_j \sum_{i=1}^{n} \tilde{w}_{ij}^2 + 2\gamma_1 \beta_j + \omega_j^{(k)} \tau_j,$$

where $\tau_j = 1$ if $\beta_j > 0$, $\tau_j = -1$ if $\beta_j < 0$, and $\tau_j \in [-1, 1]$ if $\beta_j = 0$. Setting $\partial f / \partial \beta_j = 0$, we find the analytical solution to (21),

$$\beta_j = \frac{n^{-1} \sum_{i=1}^{n} \tilde{w}_{ij} \left(\tilde{z}_i - \sum_{l \neq j} \tilde{w}_{ij} \beta_l \right) - \omega_j^{(k)} \tau_j}{n^{-1} \sum_{i=1}^{n} \tilde{w}_{ij}^2 + 2\gamma_1}$$

for $j = 1, \ldots, p$. Since $\boldsymbol{\tau}$ is implicitly defined, we compute $\boldsymbol{\beta}$ iteratively using the coordinate descent updates

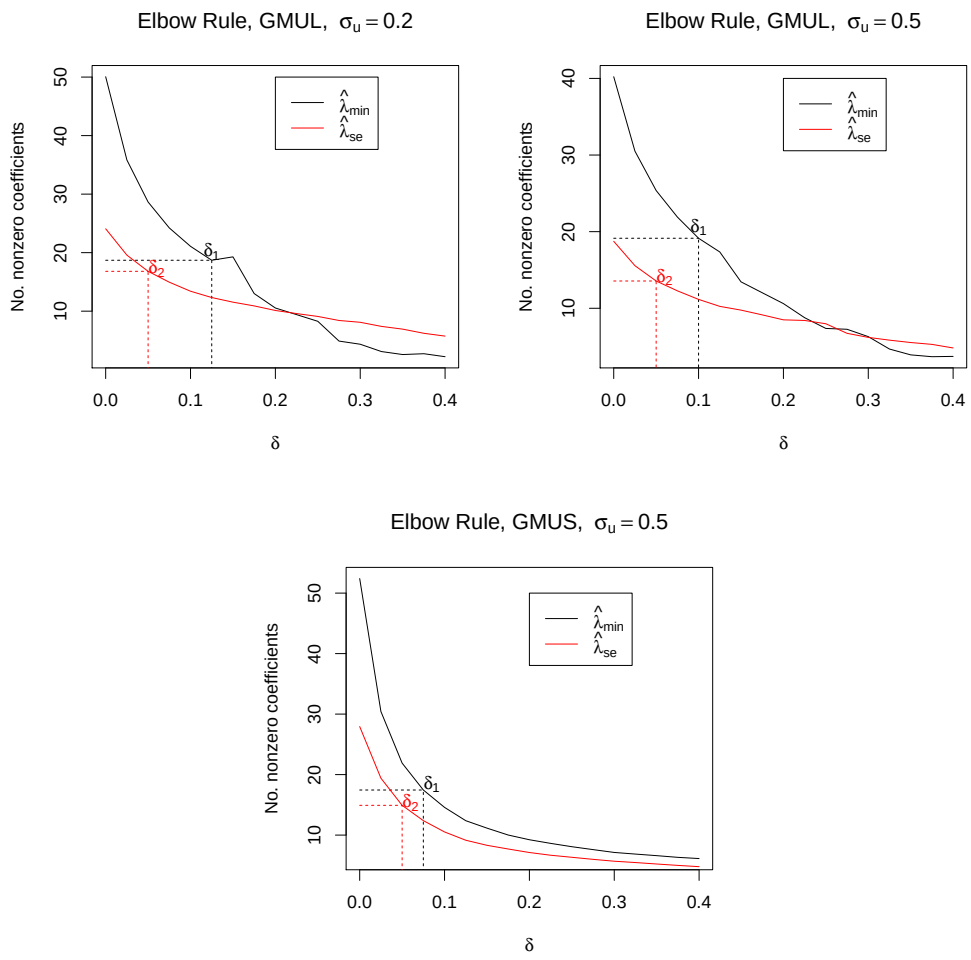
$$\hat{\beta}_j \leftarrow \frac{S \left\{ n^{-1} \sum_{i=1}^{n} \tilde{w}_{ij} \left(\tilde{z}_i - \sum_{l \neq j} \tilde{w}_{il} \hat{\beta}_l \right), \omega_j^{(k)} \right\}}{\frac{1}{n} \sum_{i=1}^{n} \tilde{w}_{ij}^2 + 2\gamma_1},$$

for $j = 1, \ldots, p, 1, \ldots$ until convergence (Friedman et al., 2007). $S(a, b)$ is the soft-thresholding operator (24). On convergence, we set $\boldsymbol{\beta}^{(k+1)} = \hat{\boldsymbol{\beta}}$.

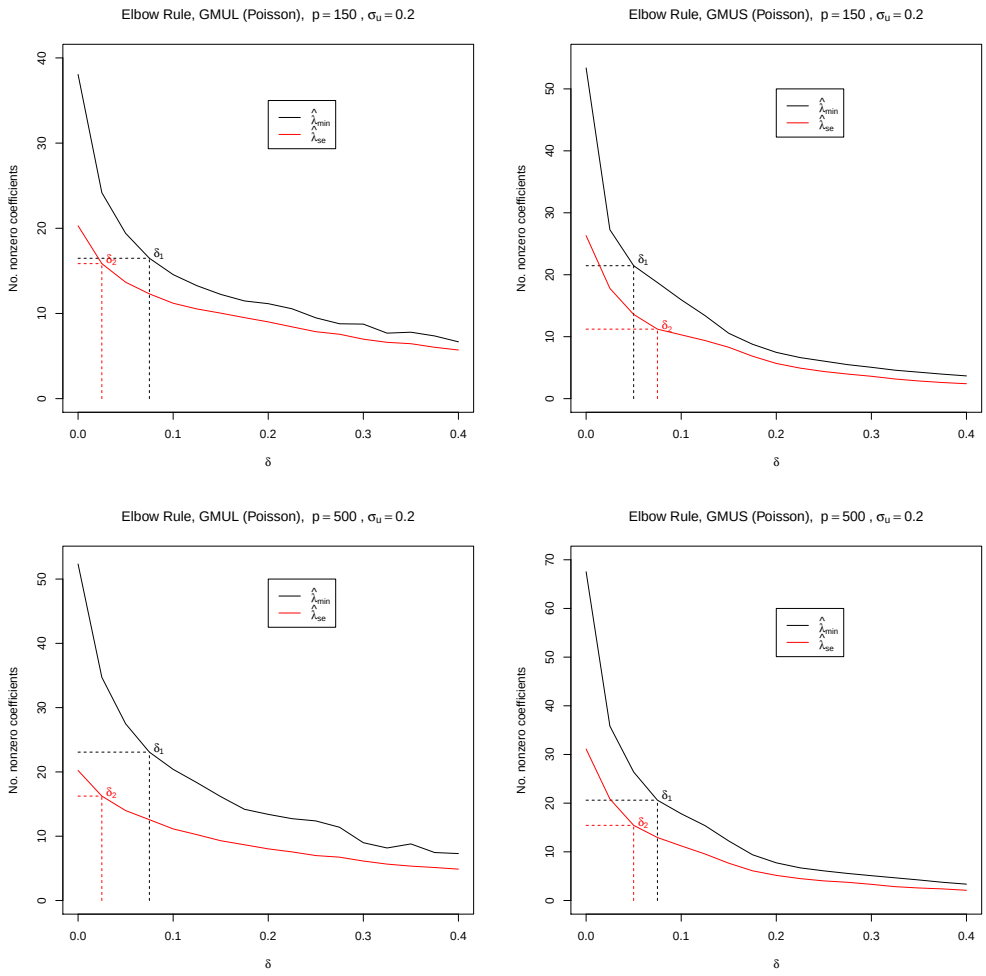
E Elbow Rule

Supplementary Figure S1 shows the elbow plot for the GMUL with $\sigma_u = 0.2$ as well as both GMUL and GMUS with $\sigma_u = 0.5$ in the logistic regression experiment. Comparing the top left plot to Figure 2 (left) in the main paper, we see that the elbow was less well defined for the GMUL compared to the GMUS. Studying the plots for $\sigma_u = 0.5$, we also see that the GMUS had the most well defined elbow.

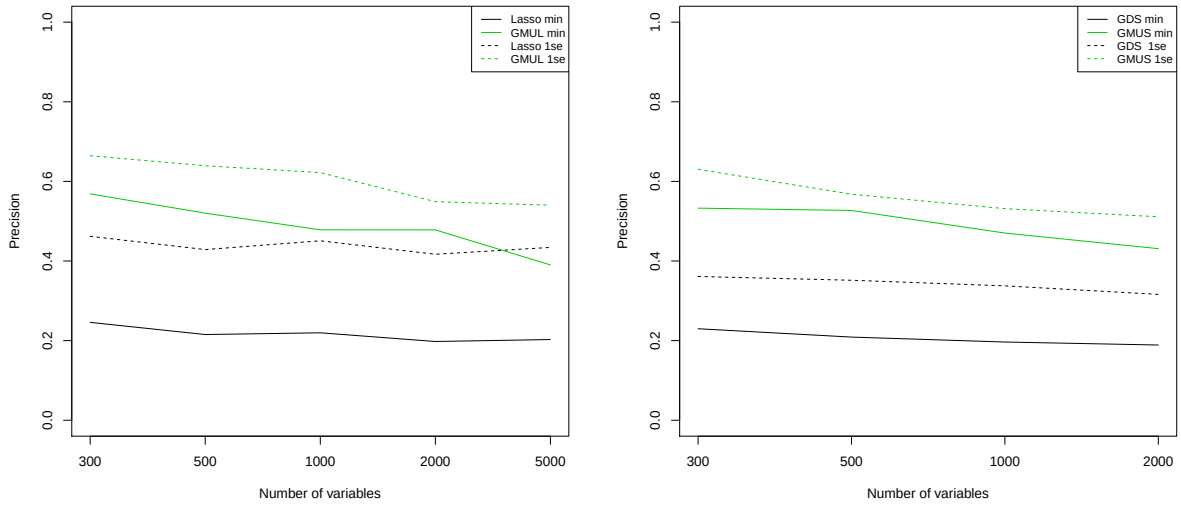
Supplementary Figure S2 shows the elbow plots for Poisson regression. Both the curves decreased sharply when δ become nonzero, and we chose the δ values where the decline began to level off.



Supplementary Figure S1: Elbow rule with logistic regression.



Supplementary Figure S2: Elbow rule with Poisson regression.



Supplementary Figure S3: a) Proportion of true non-zero coefficients among the estimated non-zero coefficients as function of the dimension p for lasso and GMUL, both with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$. b) Proportion of true non-zero coefficients among the estimated non-zero coefficients as function of the dimension p for GDS and GMUS, with both $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$.

F Simulations with High-Dimensional Data

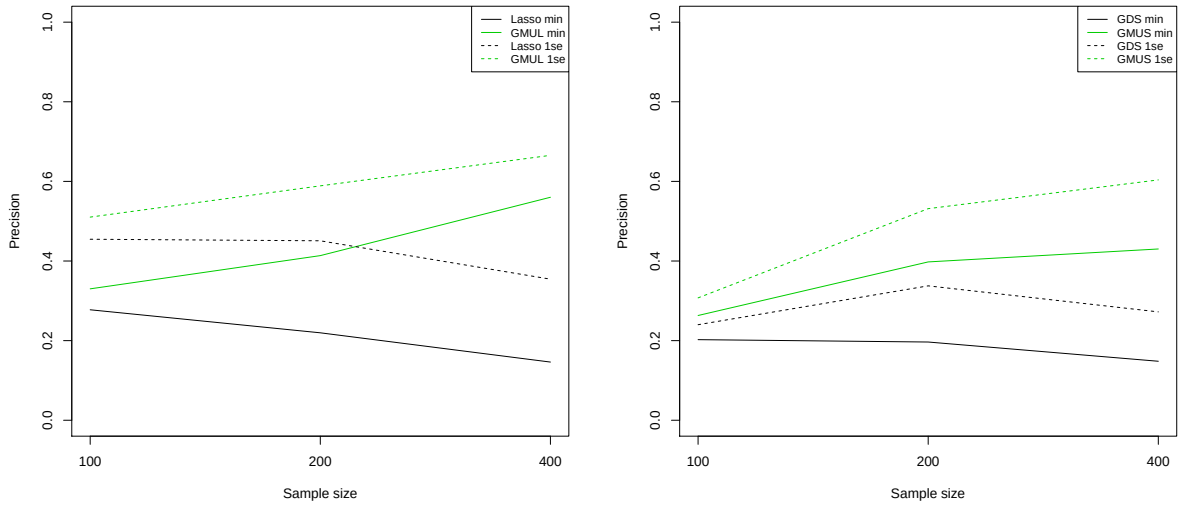
This section presents further simulation experiments comparing GMUL and GMUS to the naive lasso and GDS, respectively, in particular investigating the impact of the dimensionality p, the sample size n, and the size of the active set s. The simulations follow the setup of logistic regression given in Section 6: the matrix X had i.i.d. entries $x_{ij} \sim N(0,1)$ and the matrix W had i.i.d. entries $w_{ij} = x_{ij} + u_{ij}$ with $u_{ij} \sim N(0, \sigma_u)$, for $i = 1, \ldots, n$ and $j = 1, \ldots, p$. Outcomes y_i were binomially distributed with mean $\{1 + \exp(-\mathbf{x}_i^T \beta^0)\}^{-1}$ and non-zero regression coefficients set to $\beta_j^0 = 1$ for $j = 1, \ldots, s$.

As earlier the GDS and lasso were computed via ten-fold cross-validation with both $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$, and the solutions of the GMUL and GMUS were computed over a grid of δs with λ fixed to $\hat{\lambda}_{min}$ or $\hat{\lambda}_{se}$. Each setup of parameters p, n and s was simulated in 100 independent Monte Carlo experiments, and the elbow rule was applied to the average number of non-zero coefficients over the 100 simulations to select the optimal δ. The `Rglpk` package (Theussl and Hornik, 2013) used for computing the linear programming part of the GDS and GMUS turned out to be inefficient for problems with p in the order of thousands. Hence, the linear programming part of the algorithms was instead implemented using the IBM CPLEX Optimizer, available through the IBM Academic Initiative [1]. The `Rcplex` package (Bravo et al., 2016) was used as an R interface to this solver.

In the first simulation setup, the sample size, the measurement error variance and the size of the active set were fixed to $n = 200$, $\sigma_u = 0.2$ and $s = 10$, while the dimension varied with p set to 300, 500, 1000, 2000 and 5000 for GMUL and 300, 500, 1000 and 2000 for GMUS. Performance of each method was measured by the precision averaged over the 100 simulations. Supplementary Figure S3a) shows, for increasing dimension, the average precision of the lasso and GMUL, both with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$. The precision of all methods decreased with increasing dimension, but the decrease was relatively small. The GMUL always had a higher precision than the standard lasso, and the estimators using $\hat{\lambda}_{se}$ had a higher precision than the estimators using $\hat{\lambda}_{min}$.

Supplementary Figure S3b) shows, for increasing dimension, the average precision of the GDS and GMUS,

[1] https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/



Supplementary Figure S4: a) Proportion of true non-zero coefficients among the estimated non-zero coefficients as function of the sample size n for lasso and GMUL, both with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$. b) Proportion of true non-zero coefficients among the estimated non-zero coefficients as function of the sample size n for GDS and GMUS, both with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$.

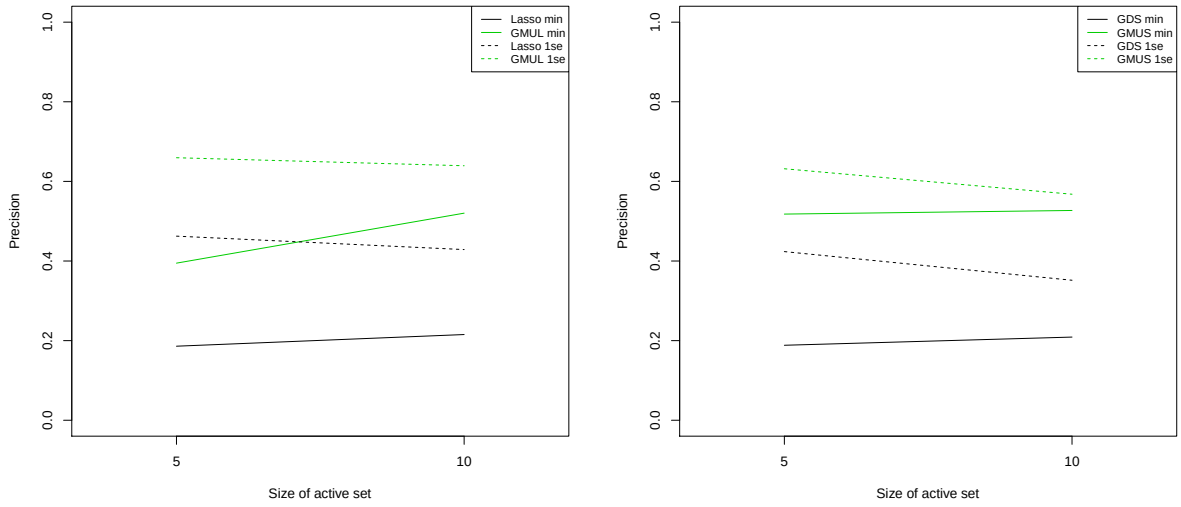
both with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$. The precision decreased with increasing dimension, but the decrease was relatively small. The GMUS always had a higher precision than the GDS, and we see the same behavior as with GMUL, where the estimators using $\hat{\lambda}_{se}$ had a higher precision than the estimators using $\hat{\lambda}_{min}$. Since $\hat{\lambda}_{se} \geq \hat{\lambda}_{min}$ by construction, the estimates obtained using $\hat{\lambda}_{se}$ have a smaller number of non-zero coefficients. As long as the true non-zero coefficients are retained, these make up a larger proportion of the estimated active as the penalization increases, thus increasing the precision. The tables S1 to S5 show detailed results for the simulations for $p = 300$, 500, 1000, 2000, and 5000.

Note that the parameter settings for the simulations described in Table S2 were identical to those of Table 1 (top) in the main text. The results are in good agreement between the two tables, but not identical. The reason for this is that the results were obtained in two independent simulation experiments; Table S2 is part of the simulations with high-dimensional data of Section F, whereas Table 1 is part of the simulations in Section 6. Thus, the choices of δ_1 and δ_2 based on the elbow rule were different, and accordingly the results in Table 1 (top) and Table S2 are not in exact agreement.

In the second simulation setup, the dimension, the measurement error variance, and the size of the active set were fixed to $p = 1000$, $\sigma_u = 0.2$, and $s = 10$, while the sample size varied with n set to 100, 200 and 400.

Supplementary Figure S4a) shows, for increasing sample size, the average precision of the lasso and GMUL with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$. For GMUL, the precision increased with the sample size, whereas the lasso instead had a relatively small decrease in the precision with increasing sample size. Also in this setting, the estimators using $\hat{\lambda}_{se}$ always had a higher precision than the estimators using $\hat{\lambda}_{min}$.

Supplementary Figure S4b) shows, for increasing sample size, the average precision of the GDS and GMUS with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$. The overall picture is as in Figure S4a), but the decrease in precision with sample size was smaller for the GDS than for the lasso. The GMUS always had a higher precision than the GDS, and the difference increased with increasing sample size. The estimators using $\hat{\lambda}_{se}$ had a higher precision than the estimators using $\hat{\lambda}_{min}$. The Supplementary Tables S6 to S8 show detailed results for the simulations, for $n = 100$, 200, and 400.



Supplementary Figure S5: a) Proportion of true non-zero coefficients among the estimated non-zero coefficients as function of the active set s for lasso and GMUL, both with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$. b) Proportion of true non-zero coefficients among the estimated non-zero coefficients as function of the active set s for GDS and GMUS, both with $\hat{\lambda}_{min}$ and $\hat{\lambda}_{se}$.

In the third simulation setup, the dimension, sample size and measurement error variance were fixed to $p = 500$, $n = 200$, and $\sigma_u = 0.2$, while the active set was given as $s = 5$ and $s = 10$.

Supplementary Figures S5a) and S5b) show the average precision of the GMUL and the lasso, and the GMUS and the GDS, respectively, with the size of the active set equal to 5 or 10. The GMUL always performed better than the lasso, and the GMUS always performed better than the GDS. For all methods using λ_{min}, the precision was slightly higher with $s = 10$ than with $s = 5$, while the opposite was true with λ_{1se}. With λ_{min}, too many covariates were selected in general, so with increasing s, the proportion of correctly selected variables increased, thus increasing the precision. Using λ_{1se}, on the other hand, yielded a sparser solution. With increasing s, a larger proportion of the relevant covariates were not selected, thus decreasing the precision. The Supplementary Tables S9 and S10 show detailed results for the simulations for $s = 5$ and $s = 10$.

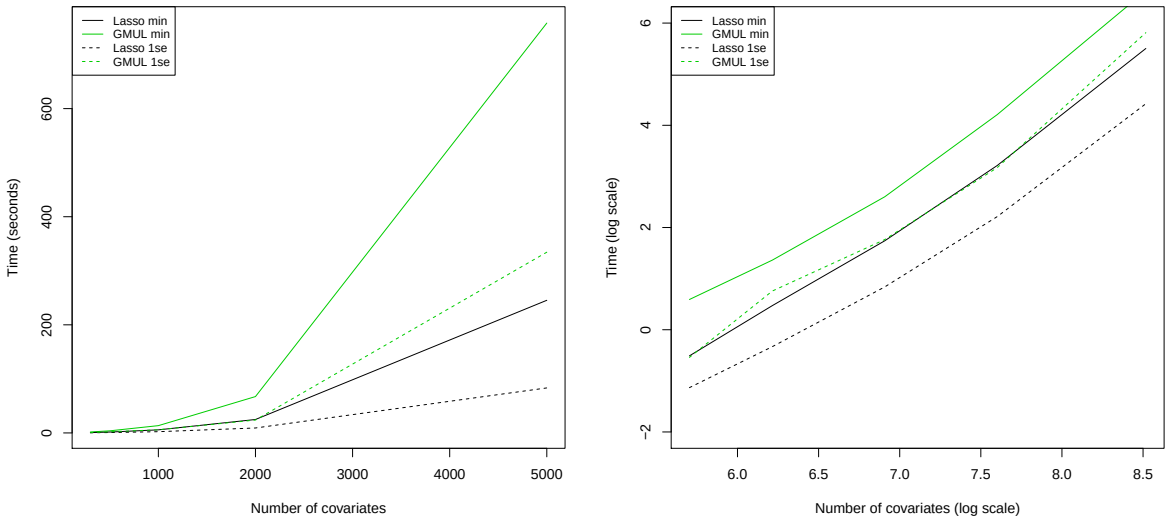
Supplementary Figures S6 and S7 show the average time until the stopping criterion was met, as a function of the number of covariates. The sample size, number of relevent covariates, and measurement error standard deviation were set to $n = 200$, $s = 10$, and $\sigma_u = 0.2$, respectively. The linearity of the logarithmic plots to the right in both figures show that the time until meeting the stopping criterion is a polynomial function of the number of covariates. We also note that the variants using $\hat{\lambda}_{se}$ were faster than those using $\hat{\lambda}_{min}$. This can be explained by the fact that the feasible set of the GMUS/GMUL is more constrained when using $\hat{\lambda}_{se}$, and thus, the iterative reweighing procedure converges faster.

References

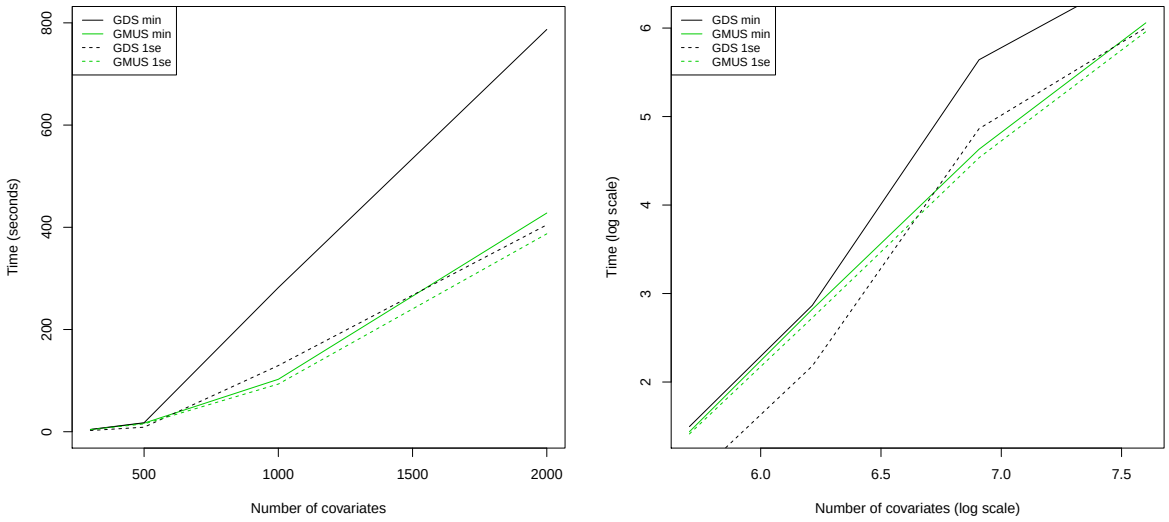
Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge, UK: Cambridge University Press.

Bravo, H. C., K. Hornik, and S. Theussl (2016). Rcplex: R Interface to CPLEX. R package version 0.3-3.

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.



Supplementary Figure S6: Average time for the iterative reweighing algorithm for the GMUL to reach the stopping criterion, as a function of the number of covariates p. Each point is an average over the 100 Monte Carlo simulations at the chosen values of λ and δ, for the case with $n = 200$, $s = 10$, and $\sigma_u = 0.2$.



Supplementary Figure S7: Average time for the iterative reweighing procedure for the GMUS to reach the stopping criterion, as a function of the number of covariates p. Each point is an average over the 100 Monte Carlo simulations at the chosen values of λ and δ, for the case with $n = 200$, $s = 10$, and $\sigma_u = 0.2$.

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	9.76 (0.05)	33.88 (1.34)	0.25 (0.01)
GMUL($\hat{\lambda}_{min}, \delta_1$)	8.78 (0.12)	7.45 (0.36)	0.57 (0.01)
Lasso($\hat{\lambda}_{se}$)	9.29 (0.11)	13.55 (0.84)	0.46 (0.02)
GMUL($\hat{\lambda}_{se}, \delta_2$)	8.29 (0.13)	4.84 (0.31)	0.66 (0.01)
GDS($\hat{\lambda}_{min}$)	9.87 (0.03)	35.93 (1.05)	0.23 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	9.27 (0.08)	8.84 (0.35)	0.53 (0.01)
GDS($\hat{\lambda}_{se}$)	9.62 (0.06)	19.73 (0.92)	0.36 (0.01)
GMUS($\hat{\lambda}_{se}, \delta_2$)	8.82 (0.11)	5.89 (0.34)	0.63 (0.02)

Supplementary Table S1: $p = 300$ $n = 200$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.15$ $\delta_2 = 0.1$ GMUS: $\delta_1 = 0.06$ $\delta_2 = 0.06$

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	9.67 (0.07)	39.83 (1.43)	0.22 (0.01)
GMUL($\hat{\lambda}_{min}, \delta_1$)	8.64 (0.16)	9.01 (0.39)	0.52 (0.01)
Lasso($\hat{\lambda}_{se}$)	8.96 (0.13)	15.19 (0.92)	0.43 (0.02)
GMUL($\hat{\lambda}_{se}, \delta_2$)	8.18 (0.15)	5.34 (0.32)	0.64 (0.01)
GDS($\hat{\lambda}_{min}$)	9.58 (0.06)	40.19 (1.29)	0.21 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	8.33 (0.12)	8.14 (0.35)	0.53 (0.01)
GDS($\hat{\lambda}_{se}$)	9.08 (0.11)	20.03 (1)	0.35 (0.01)
GMUS($\hat{\lambda}_{se}, \delta_2$)	7.87 (0.14)	6.74 (0.36)	0.57 (0.01)

Supplementary Table S2: $p = 500$ $n = 200$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.15$ $\delta_2 = 0.12$ GMUS: $\delta_1 = 0.08$ $\delta_2 = 0.06$

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	9.1 (0.12)	39.55 (1.88)	0.22 (0.01)
GMUL($\hat{\lambda}_{min}$	7.8 (0.17)	10.32 (0.52)	0.48 (0.02)
Lasso($\hat{\lambda}_{se}$)	8 (0.19)	13.67 (1.06)	0.45 (0.02)
GMUL($\hat{\lambda}_{se}, \delta_2$)	7 (0.18)	5.23 (0.39)	0.62 (0.02)
GDS($\hat{\lambda}_{min}$)	9.2 (0.1)	43.97 (1.72)	0.2 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	8.02 (0.12)	10.21 (0.47)	0.47 (0.01)
GDS($\hat{\lambda}_{se}$)	8.54 (0.13)	22.44 (1.45)	0.34 (0.01)
GMUS($\hat{\lambda}_{se}, \delta_2$)	7.51 (0.15)	8.04 (0.52)	0.53 (0.02)

Supplementary Table S3: $p = 1000$ $n = 200$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.15$ $\delta_2 = 0.12$ GMUS: $\delta_1 = 0.09$ $\delta_2 = 0.06$

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	8.32 (0.15)	43.59 (2.09)	0.2 (0.01)
GMUL($\hat{\lambda}_{min}$	6.28 (0.21)	8.99 (0.55)	0.47 (0.02)
Lasso($\hat{\lambda}_{se}$)	6.91 (0.22)	15.46 (1.32)	0.41 (0.02)
GMUL($\hat{\lambda}_{se}, \delta_2$)	6.05 (0.21)	7.09 (0.59)	0.54 (0.02)
GDS($\hat{\lambda}_{min}$)	8.49 (0.15)	48.04 (2.3)	0.19 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	6.78 (0.16)	10.68 (0.57)	0.43 (0.02)
GDS($\hat{\lambda}_{se}$)	7.57 (0.2)	25.09 (1.81)	0.32 (0.02)
GMUS($\hat{\lambda}_{se}, \delta_2$)	6.33 (0.18)	8.09 (0.59)	0.51 (0.02)

Supplementary Table S4: $p = 2000$ $n = 200$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.15$ $\delta_2 = 0.12$ GMUS: $\delta_1 = 0.1$ $\delta_2 = 0.09$

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	7.13 (0.17)	40.74 (2.46)	0.2 (0.01)
GMUL($\hat{\lambda}_{min}$	5.53 (0.18)	10.78 (0.63)	0.39 (0.02)
Lasso($\hat{\lambda}_{se}$)	5.2 (0.24)	11.97 (1.15)	0.42 (0.03)
GMUL($\hat{\lambda}_{se}, \delta_2$)	4.5 (0.22)	6.17 (0.59)	0.52 (0.03)

Supplementary Table S5: $p = 5000$ $n = 200$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.15$ $\delta_2 = 0.12$ GMUS: $\delta_1 = 0.09$ $\delta_2 = 0.06$, GMUS was not performed due to long calculation time

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	3.76 (0.23)	15.67 (1.32)	0.24 (0.02)
GMUL($\hat{\lambda}_{min}, \delta_1$)	3.39 (0.21)	10.04 (0.85)	0.29 (0.02)
Lasso($\hat{\lambda}_{se}$)	2.12 (0.22)	5.06 (0.77)	0.3 (0.03)
GMUL($\hat{\lambda}_{se}, \delta_2$)	1.95 (0.21)	3.52 (0.51)	0.34 (0.03)
GDS($\hat{\lambda}_{min}$)	4.68 (0.14)	21.25 (1.07)	0.2 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	3.65 (0.14)	11.49 (0.6)	0.26 (0.01)
GDS($\hat{\lambda}_{se}$)	4.15 (0.12)	14.41 (0.63)	0.24 (0.01)
GMUS($\hat{\lambda}_{se}, \delta_2$)	3.23 (0.12)	7.97 (0.37)	0.31 (0.01)

Supplementary Table S6: $p = 1000$ $n = 100$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.12$ $\delta_2 = 0.1$ GMUS: $\delta_1 = 0.08$ $\delta_2 = 0.08$

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	9.1 (0.12)	39.55 (1.88)	0.22 (0.01)
GMUL($\hat{\lambda}_{min}, \delta_1$)	8.24 (0.15)	13.81 (0.68)	0.41 (0.01)
Lasso($\hat{\lambda}_{se}$)	8 (0.19)	13.67 (1.06)	0.45 (0.02)
GMUL($\hat{\lambda}_{se}, \delta_2$)	7.25 (0.18)	6.22 (0.45)	0.59 (0.02)
GDS($\hat{\lambda}_{min}$)	9.2 (0.1)	43.97 (1.72)	0.2 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	8.31 (0.12)	14.32 (0.62)	0.4 (0.01)
GDS($\hat{\lambda}_{se}$)	8.54 (0.13)	22.44 (1.45)	0.34 (0.01)
GMUS($\hat{\lambda}_{se}, \delta_2$)	7.51 (0.15)	8.04 (0.52)	0.53 (0.02)

Supplementary Table S7: $p = 1000$ $n = 200$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.1$ $\delta_2 = 0.08$ GMUS: $\delta_1 = 0.06$ $\delta_2 = 0.06$

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	10 (0)	66.23 (2.43)	0.15 (0.01)
GMUL($\hat{\lambda}_{min}, \delta_1$)	9.99 (0.01)	9.15 (0.53)	0.56 (0.01)
Lasso($\hat{\lambda}_{se}$)	10 (0)	23.66 (1.46)	0.35 (0.01)
GMUL($\hat{\lambda}_{se}, \delta_2$)	9.97 (0.02)	6.04 (0.44)	0.67 (0.02)
GDS($\hat{\lambda}_{min}$)	10 (0)	66.23 (2.43)	0.15 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	10 (0)	29.93 (1.25)	0.28 (0.01)
GDS($\hat{\lambda}_{se}$)	10 (0)	23.66 (1.46)	0.35 (0.01)
GMUS($\hat{\lambda}_{se}, \delta_2$)	9.99 (0.01)	10.89 (0.76)	0.54 (0.02)

Supplementary Table S8: $p = 1000$ $n = 400$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.08$ $\delta_2 = 0.06$ GMUS: $\delta_2 = 0.04$ $\delta_2 = 0.04$

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	4.96 (0.02)	28.76 (1.65)	0.19 (0.01)
GMUL($\hat{\lambda}_{min}, \delta_1$)	4.9 (0.03)	8.29 (0.5)	0.43 (0.02)
Lasso($\hat{\lambda}_{se}$)	4.84 (0.05)	9.05 (0.82)	0.46 (0.02)
GMUL($\hat{\lambda}_{se}, \delta_2$)	4.78 (0.05)	3.56 (0.35)	0.66 (0.02)
GDS($\hat{\lambda}_{min}$)	4.95 (0.02)	28.58 (1.49)	0.19 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	4.82 (0.04)	5.73 (0.36)	0.52 (0.02)
GDS($\hat{\lambda}_{se}$)	4.88 (0.04)	10.78 (0.94)	0.42 (0.02)
GMUS($\hat{\lambda}_{se}, \delta_2$)	4.69 (0.05)	3.8 (0.36)	0.63 (0.02)

Supplementary Table S9: $p = 500$ $n = 200$ $s = 5$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.15$ $\delta_2 = 0.12$ GMUS: $\delta_1 = 0.08$ $\delta_2 = 0.06$

	#TP	#FP	#TP/(#TP + #FP)
Lasso ($\hat{\lambda}_{min}$)	9.67 (0.07)	39.83 (1.43)	0.22 (0.01)
GMUL($\hat{\lambda}_{min}, \delta_1$)	8.64 (0.16)	9.01 (0.39)	0.52 (0.01)
Lasso($\hat{\lambda}_{se}$)	8.96 (0.13)	15.19 (0.92)	0.43 (0.02)
GMUL($\hat{\lambda}_{se}, \delta_2$)	8.18 (0.15)	5.34 (0.32)	0.64 (0.01)
GDS($\hat{\lambda}_{min}$)	9.58 (0.06)	40.19 (1.29)	0.21 (0.01)
GMUS($\hat{\lambda}_{min}, \delta_1$)	8.33 (0.12)	8.14 (0.35)	0.53 (0.01)
GDS($\hat{\lambda}_{se}$)	9.08 (0.11)	20.03 (1)	0.35 (0.01)
GMUS($\hat{\lambda}_{se}, \delta_2$)	7.87 (0.14)	6.74 (0.36)	0.57 (0.01)

Supplementary Table S10: $p = 500$ $n = 200$ $s = 10$ $\sigma_u = 0.2$, GMUL: $\delta_1 = 0.15$ $\delta_2 = 0.12$ GMUS: $\delta_1 = 0.08$ $\delta_2 = 0.06$

Friedman, J., T. Hastie, H. Höfling, and R. Tibshirani (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics 1*(2), 302–332.

Theussl, S. and K. Hornik (2013). Rglpk: R/GNU linear programming kit interface. R package version 0.5-2.