Supporting Information for "Simulation-Selection-Extrapolation: Estimation in High Dimensional Errors–in–Variables Models" by Linh Nghiem and Cornelis J. Potgieter

A Illustrating SIMEX performance in a high-dimensional setting

In both Sections 1 and 2 of the main paper, it was mentioned that SIMEX does not perform well in high-dimensional errors-in-variables models without suitably modifying the procedure. Specifically, standard SIMEX inflates the number of estimated nonzero components considerably, even when combined with a procedure such as the lasso. Here, a simulated example is presented to illustrate.

For the example, data pairs $(\boldsymbol{W}_i, Y_i)$ were generated according to the linear model $Y_i = \boldsymbol{X}_i^\top \boldsymbol{\theta} + \varepsilon_i$ with additive measurement error $\boldsymbol{W}_i = \boldsymbol{X}_i + \boldsymbol{U}_i$. Both the true covariates $\boldsymbol{X}_i$ and the measurement error components $\boldsymbol{U}_i$ were generated to be *i.i.d.* p-variate normal. Specifically, $\boldsymbol{X}_i \sim \mathrm{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma}$ having entries $\Sigma_{ij} = \rho^{|i-j|}$ with $\rho = 0.25$, and $\boldsymbol{U}_i \sim \mathrm{N}_p(\mathbf{0}, \boldsymbol{\Sigma}_u)$ with $\boldsymbol{\Sigma}_u = \sigma_u^2 I_{p\times p}$ with $\sigma_u^2 = 0.45$. The error components ε_i were simulated to be *i.i.d.* univariate normal, $\varepsilon_i \sim N(\mathbf{0}, \sigma_\varepsilon^2)$ with $\sigma_\varepsilon = 0.128$. The sample sizes was fixed at $n = 300$, and the number of covariates was $p = 500$. The parameter vector was taken to be $\boldsymbol{\theta} = \{1, 1, 1, 1, 1, 0, \dots, 0\}$ with $s = 5$ nonzero coefficients and $p - s = 495$ zero coefficients.

For the simulation step of SIMEX, a grid of $M = 13$ equally spaced λ-values ranging from 0.2 to 2 were used. For each value of λ, a total of $B = 100$ sets of pseudo-data were generated. In applying the lasso, the tuning parameter was chosen based on the one-standard-error rule based on 10-fold cross-validation. The lasso was implemented using the `glmnet` package in R. For the extrapolation step, a quadratic function was used.

The analysis of the simulated data shows that SIMEX applied to the lasso results in 174 nonzero parameter estimates. Of the 169 false positives, 156 are fairly small (less than 0.001 in absolute value), with 13 false positives being larger (greater than 0.001 in absolute value). Comparatively, a naive application of the lasso (not correcting for measurement error) gives only 5 non-zero parameter estimates. Implementing SIMEX, even when using a method such as the lasso that enforces sparsity, can result in an inflated number of variables in the model.

B A brief review of existing methodology

In Section 3 of the main paper, the SIMSELEX estimator is compared to several existing methods for fitting errors-in-variables models in high-dimensional settings. For the linear model, SIMSELEX is compared with the corrected lasso estimator of Sørensen et al. (2015) and the conic estimator of Belloni et al. (2017). For the logistic model, the SIMSELEX estimator is compared with the conditional scores lasso of Sørensen et al. (2015). These approaches are briefly reviewed in this section.

B.1 Linear Model

The corrected lasso estimator of Sørensen et al. (2015) is the solution to the optimization problem

$$\begin{aligned} \min_{\boldsymbol{\theta}} \ & L(\boldsymbol{\theta}) = \|Y - \boldsymbol{W}\boldsymbol{\theta}\|_2^2 - \boldsymbol{\theta}^\top \boldsymbol{\Sigma}_u \boldsymbol{\theta} \\ \text{s.t.} \ & \|\boldsymbol{\theta}\|_1 \le R \end{aligned}$$

where for p-dimensional vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_1 = \sum_{j=1}^p |x_j|$ and $\|\boldsymbol{x}\|_2^2 = \sum_{j=1}^p x_j^2$. Here, R is a tuning parameter that can be chosen based on cross-validation using an estimate of the unbiased loss function. Specifically, if the data are partitioned into random subset $\mathcal{P}_1, \ldots, \mathcal{P}_J$, each subset having size n/J, let $(\boldsymbol{W}_{(\mathcal{P}_j)}, Y_{(\mathcal{P}_j)})$ denote the data in the jth partition and let $(\boldsymbol{W}_{(-\mathcal{P}_j)}, Y_{(-\mathcal{P}_j)})$ denote the data excluding the jth partition. Also let $\hat{\boldsymbol{\theta}}_j$ denote the estimated parameter vector based on $(\boldsymbol{W}_{(-\mathcal{P}_j)}, Y_{(-\mathcal{P}_j)})$. Then the tuning parameter R can be chosen using cross-validation loss function

$$L_{CV}(R) = \sum_{j=1}^J \left\| Y_{\mathcal{P}_j} - \boldsymbol{W}_{\mathcal{P}_j} \hat{\boldsymbol{\theta}}_j \right\|_2^2 - \sum_{j=1}^J \hat{\boldsymbol{\theta}}_j^\top \boldsymbol{\Sigma}_u \hat{\boldsymbol{\theta}}_j.$$

The optimal tuning parameter R can be chosen either to minimize L_{CV}, or according to the one standard error rule (see Friedman et al. (2001)). Sørensen et al. (2015) prove that the corrected lasso performs sign-consistent covariate selection in large samples.

The conic estimator of Belloni et al. (2017) is also the solution to an optimization problem,

$$\begin{aligned} \min_{\boldsymbol{\theta}, t} \ & \|\boldsymbol{\theta}\|_1 + \lambda t \\ \text{s.t} \ & \left\| \frac{1}{n} \mathbf{W}^\top (Y - \mathbf{W}\boldsymbol{\theta} + \boldsymbol{\Sigma}_u \boldsymbol{\theta}) \right\|_\infty \le \mu t + \tau, \quad t \ge 0, \quad \|\boldsymbol{\theta}\|_2 \le t. \end{aligned}$$

where for p-dimensional vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_\infty = \max_{j=1,\ldots,p} |x_j|$. This method requires the selection of three tuning parameters, here denoted μ, τ and λ. The optimal choices of these tuning parameters depend on the underlying model structure, including the rate at which the number of nonzero model coefficients increases with sample size. Belloni et al. (2017) do suggest tuning parameter values for application. Furthermore, these authors also proved that under suitable sparsity conditions, their conic estimator has smaller minimax efficiency bound than the Matrix Uncertainty Selection estimator of Rosenbaum et al. (2010). We are not aware of any comparison, numerical or otherwise, of the corrected lasso estimator and the conic estimator. This comparison is presented as part of our simulation study in Section 3.1 of the main paper.

B.2 Logistic Regression

For the logistic regression model, the SIMSELEX estimator is compared with the conditional scores lasso estimator developed by Sørensen et al. (2015) and the Generalized Matrix Uncertainty Selector (GMUS) developed by Sørensen et al. (2018). The conditional scores lasso estimator is computed by solving the set of estimating equations

$$\sum_{i=1}^{n}\left(Y_i - F\left\{\eta_i - \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{\Sigma}_u \boldsymbol{\theta}\right\}\right)\begin{pmatrix}1\\ \boldsymbol{W}_i + Y_i \boldsymbol{\Sigma}_u \boldsymbol{\theta}\end{pmatrix} = \mathbf{0} \text{ subject to } \|\boldsymbol{\theta}\|_1 \leq R$$

where $\eta_i = \mu + \boldsymbol{\theta}^\top(\boldsymbol{W}_i + Y_i\boldsymbol{\Sigma}_u\boldsymbol{\theta})$ and $F(.)$ is the logit function. Note that this is a system of $p+1$ estimating equations. Sørensen et al. (2015) also illustrate how the conditional scores lasso can be applied to other GLMs. For the simulation studies in section 3.2 of the main paper, the tuning parameter R is chosen to be $1.5\left\|\hat{\beta}_{\text{naive}}\right\|_1$, where $\hat{\beta}_{\text{naive}}$ denotes the naive lasso.

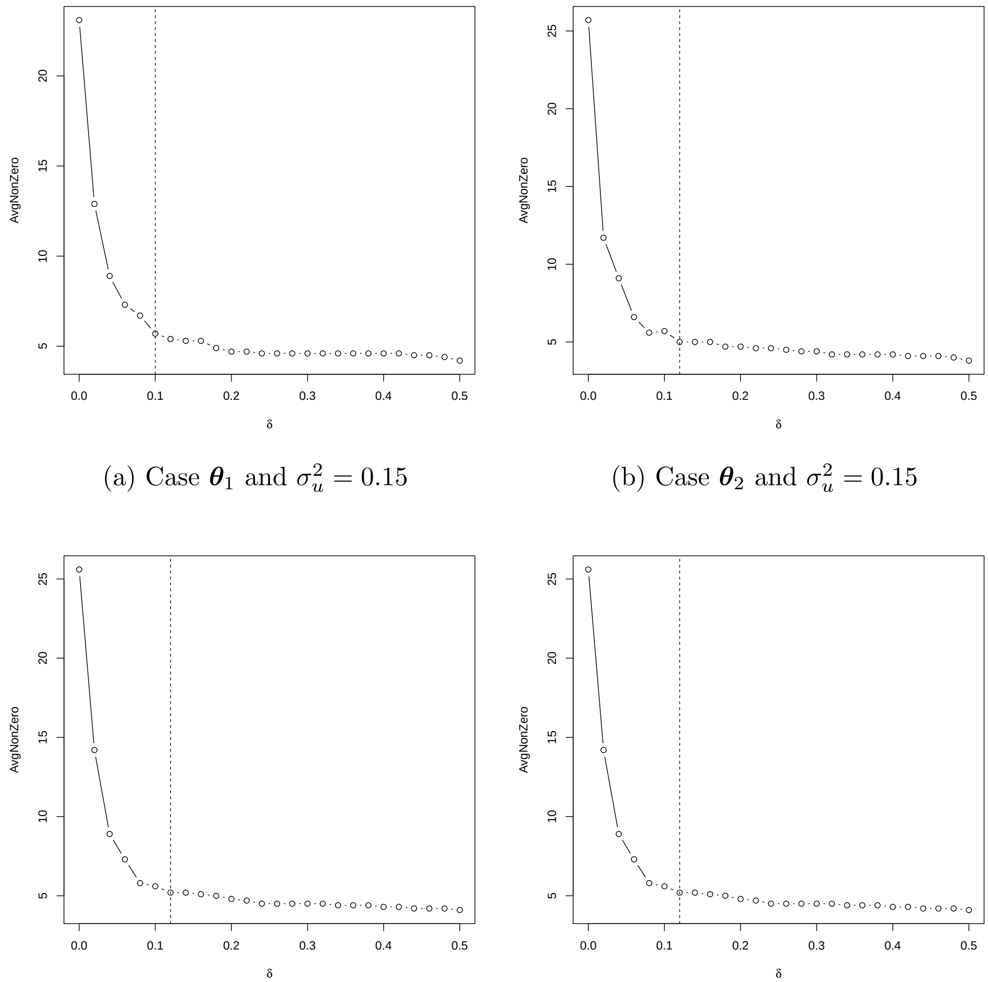
The GMUS estimator is defined as

$$\hat{\beta}_{MU} = \arg\min\{\|\beta\|_1 : \beta \in \Theta\}, \text{ where}$$

$$\Theta = \left[\beta \in \mathbb{R}^p : \left\|\frac{1}{n}\sum_{i=1}^{n} w_{ij}(y_i - F(\boldsymbol{w}_i^\top \beta))\right\|_\infty \leq \lambda + \frac{\delta}{\sqrt{n}}\|\beta\|_1 \left\|F'(\boldsymbol{W}\beta)\right\|_2\right]$$

where $F'(\boldsymbol{W}\beta) = \{F'(\boldsymbol{w}_1^\top\beta), \ldots, F'(\boldsymbol{w}_n^\top\beta)\}^\top$, with $F'(.)$ denotes the first derivative of $F(.)$. The tuning parameter λ is chosen to be equal to the tuning parameter when computing the naive lasso, while the tuning parameter δ was chosen following the elbow rule. More specifically, a grid of δ-values is chosen. For each value of δ in the grid, the GMUS is computed. Finally, the number of non-zero coefficients is plotted as a function of R, and the optimal R is chosen as the point at which the plot elbows i.e. starts to become flat. Note that finding this elbow for the GMUS is somewhat subjective and the authors do not provide an automated way of performing this selection.

For the simulation study in Section 3.2 of the main paper, the tuning parameter δ was chosen in a manner identical to the simulation study presented in Sørensen et al. (2015). First, $N_0 = 100$ samples were simulated using the data generation mechanism outlined. For the jth simulated dataset, let $R = \delta\left\|\hat{\boldsymbol{\theta}}_{\text{naive}}\right\|_1$, where $\left\|\hat{\boldsymbol{\theta}}_{\text{naive}}\right\|_1$ denotes the ℓ_1 norm of the naive lasso estimator. Let $(\delta, \text{NZ}_j(\delta))$ denote the curve of the number of non-zero coefficients as a function of λ. These curves were then averaged, resulting in curve $(\delta, \overline{\text{NZ}}(\delta))$ where $\overline{\text{NZ}}(\delta) = N_0^{-1}\sum_j \text{NZ}_j(\delta)$. The value of δ used subsequently to evaluate the conditional scores lasso estimators in the simulation study was the point at which the curve $\overline{\text{NZ}}(\delta)$ elbows. For each given simulation configuration, a different value of δ was calculated.



(a) Case $\boldsymbol{\theta}_1$ and $\sigma_u^2 = 0.15$

(b) Case $\boldsymbol{\theta}_2$ and $\sigma_u^2 = 0.15$

(c) Case $\boldsymbol{\theta}_1$ and $\sigma_u^2 = 0.30$

(d) Case $\boldsymbol{\theta}_2$ and $\sigma_u^2 = 0.30$

Figure B.1: Elbow plots choosing tuning parameters in implementation of conditional scores lasso estimator in the logistic regression simulation.

In the simulation study in Section 3.2 of the main paper, the GMUS estimator was computed only for the case of $p = 500$. The elbow plots for the settings associated with Normal measurement error were presented below. The tuning parameters in the simulation study with Laplace measurement error were chosen to be the same as the chosen value in the similar setting with Normal measurement error.

C Additional Simulation Results for Linear Regression, Logistic Regression, and Cox Survival Model

This section presents the simulation results corresponding to the case of $\boldsymbol{\theta}_2 = (1, 1, 1, 1, 1, 0, \ldots, 0)$. All the other simulation configurations are the same as outlined in the Section 3 of the main paper.

These results are discussed and interpreted in the main paper, with the tabulated summaries included here for completeness.

Table C.1: Comparison of estimators for linear regression with with the case of $\boldsymbol{\theta_2}$ based on ℓ_2 estimation error, average number of false positive (FP), and average number of false negative (FN) across 500 simulations.

p	Estimator	$\sigma_u^2 = 0.15$						$\sigma_u^2 = 0.30$					
		Normal			Laplace			Normal			Laplace		
		ℓ_2	FP	FN	ℓ_2	FP	FN	ℓ_2	FP	FN	ℓ_2	FP	FN
500	True	0.09	1.11	0.00	0.09	1.1	0.00	0.09	1.24	0.00	0.09	1.19	0.00
		(0.02)	(2.36)	(0.00)	(0.02)	(2.55)	(0.00)	(0.02)	(2.75)	(0.00)	(0.02)	(2.62)	(0.00)
	Naive	0.48	1.38	0.00	0.73	1.35	0.00	0.48	1.1	0.00	0.73	1.36	0.00
		(0.05)	(2.92)	(0.00)	(0.07)	(2.9)	(0.00)	(0.05)	(2.3)	(0.00)	(0.07)	(3.3)	(0.00)
	SIMSELEX	0.21	0.00	0.00	0.23	0.00	0.00	0.21	0.00	0.00	0.34	0.00	0.00
		(0.07)	(0.00)	(0.00)	(0.07)	(0.00)	(0.00)	(0.07)	(0.00)	(0.00)	(0.11)	(0.00)	(0.00)
	Conic	0.27	-	-	0.34	-	-	0.27	-	-	0.34	-	-
		(0.04)	-	-	(0.07)	-	-	(0.04)	-	-	(0.07)	-	-
	Corrected	0.29	2.48	0.00	0.4	2.19	0.00	0.29	2.55	0.00	0.4	2.32	0.00
		(0.05)	(4.5)	(0.00)	(0.08)	(3.85)	(0.00)	(0.05)	(4.18)	(0.00)	(0.08)	(4.09)	(0.00)
1000	True	0.09	1.04	0.00	0.09	1.29	0.00	0.09	1.79	0.00	0.09	1.33	0.00
		(0.02)	(2.36)	(0.00)	(0.02)	(2.74)	(0.00)	(0.02)	(4.35)	(0.00)	(0.02)	(3.33)	(0.00)
	Naive	0.5	1.78	0.00	0.75	1.24	0.00	0.5	1.79	0.00	0.75	1.63	0.00
		(0.06)	(5.09)	(0.00)	(0.07)	(2.75)	(0.00)	(0.06)	(4.47)	(0.00)	(0.07)	(3.56)	(0.00)
	SIMSELEX	0.23	0.00	0.00	0.24	0.00	0.00	0.23	0.00	0.00	0.35	0.00	0.00
		(0.07)	(0.00)	(0.00)	(0.07)	(0.00)	(0.00)	(0.07)	(0.00)	(0.00)	(0.1)	(0.00)	(0.00)
	Conic	0.27	-	-	0.37	-	-	0.27	-	-	0.37	-	-
		(0.04)	-	-	(0.07)	-	-	(0.04)	-	-	(0.07)	-	-
	Corrected	0.3	3.78	0.00	0.42	2.94	0.00	0.3	4.2	0.00	0.42	3.54	0.00
		(0.06)	(6.6)	(0.00)	(0.08)	(5.53)	(0.00)	(0.06)	(6.29)	(0.00)	(0.08)	(5.93)	(0.00)
2000	True	0.1	2.12	0.00	0.1	6.32	0.00	0.1	2.19	0.00	0.1	1.57	0.00
		(0.02)	(5.68)	(0.00)	(0.02)	(10.95)	(0.00)	(0.02)	(5.57)	(0.00)	(0.02)	(3.61)	(0.00)
	Naive	0.51	1.87	0.00	0.77	6.12	0.00	0.51	2.01	0.00	0.77	1.64	0.00
		(0.05)	(4.7)	(0.00)	(0.07)	(10.94)	(0.00)	(0.05)	(4.52)	(0.00)	(0.07)	(3.39)	(0.00)
	SIMSELEX	0.23	0.00	0.00	0.23	0.00	0.00	0.23	0.00	0.00	0.36	0.00	0.00
		(0.07)	(0.00)	(0.00)	(0.07)	(0.00)	(0.00)	(0.07)	(0.00)	(0.00)	(0.11)	(0.04)	(0.00)
	Conic	0.28	-	-	0.38	-	-	0.28	-	-	0.38	-	-
		(0.04)	-	-	(0.07)	-	-	(0.04)	-	-	(0.07)	-	-
	Corrected	0.3	5.66	0.00	0.43	4.76	0.00	0.3	5.64	0.00	0.43	4.36	0.00
		(0.05)	(9.41)	(0.00)	(0.08)	(9.62)	(0.00)	(0.05)	(8.18)	(0.00)	(0.08)	(6.5)	(0.00)

Table C.2: Comparison of estimators for logistic regression with with the case of $\boldsymbol{\theta_2}$ based on ℓ_2 estimation error, average number of false positive (FP), and average number of false negative (FN) across 500 simulations.

p	Estimator	$\sigma_u^2 = 0.15$						$\sigma_u^2 = 0.30$					
		Normal			Laplace			Normal			Laplace		
		ℓ_2	FP	FN	ℓ_2	FP	FN	ℓ_2	FP	FN	ℓ_2	FP	FN
500	True	1.75	0.32	0.56	1.75	0.32	0.56	1.75	0.28	0.53	1.75	0.28	0.53
		(0.21)	(2.17)	(1.48)	(0.21)	(2.17)	(1.48)	(0.21)	(1.49)	(1.42)	(0.21)	(1.49)	(1.42)
	Naive	1.87	0.48	1.13	1.98	0.48	1.13	1.87	0.36	1.64	1.98	0.36	1.64
		(0.22)	(2.57)	(1.98)	(0.21)	(2.57)	(1.98)	(0.22)	(1.63)	(2.21)	(0.21)	(1.63)	(2.21)
	SIMSELEX	1.77	0.01	0.93	1.81	0.01	0.92	1.77	0.00	2.25	1.90	0.00	2.39
		(0.42)	(0.11)	(1.27)	(0.43)	(0.08)	(1.23)	(0.42)	(0.00)	(1.73)	(0.34)	(0.04)	(1.76)
	Cond	2.32	3.5	1.57	2.4	3.5	1.57	2.32	3.63	2.05	2.4	3.63	2.05
		(0.67)	(6.52)	(1.19)	(0.67)	(6.52)	(1.19)	(0.67)	(6.65)	(1.24)	(0.67)	(6.65)	(1.24)
	GMUS	1.61	0.91	0.02	1.77	0.91	0.02	1.61	0.41	0.1	1.77	0.41	0.1
		(0.08)	(1.17)	(0.13)	(0.07)	(1.17)	(0.13)	(0.08)	(0.73)	(0.3)	(0.07)	(0.73)	(0.3)
1000	True	1.75	0.35	0.47	1.77	0.35	0.47	1.75	0.32	0.62	1.77	0.32	0.62
		(0.18)	(1.71)	(1.32)	(0.21)	(1.71)	(1.32)	(0.18)	(1.47)	(1.56)	(0.21)	(1.47)	(1.56)
	Naive	1.89	0.52	1.23	1.99	0.52	1.23	1.89	0.46	2.05	1.99	0.46	2.05
		(0.21)	(2.29)	(2.03)	(0.2)	(2.29)	(2.03)	(0.21)	(3.18)	(2.34)	(0.2)	(3.18)	(2.34)
	SIMSELEX	1.8	0.01	1.06	1.81	0.01	1.08	1.8	0.00	2.79	1.92	0.00	2.80
		(0.4)	(0.12)	(1.35)	(0.41)	(0.13)	(1.41)	(0.4)	(0.04)	(1.76)	(0.34)	(0.00)	(1.80)
	Cond	2.46	4.83	1.7	2.43	4.83	1.7	2.46	3.99	2.19	2.43	3.99	2.19
		(0.66)	(8.76)	(1.19)	(0.68)	(8.76)	(1.19)	(0.66)	(7.22)	(1.19)	(0.68)	(7.22)	(1.19)
2000	True	1.78	0.56	0.57	1.76	0.56	0.57	1.78	0.52	0.66	1.76	0.52	0.66
		(0.19)	(3.02)	(1.46)	(0.21)	(3.02)	(1.46)	(0.19)	(3.25)	(1.56)	(0.21)	(3.25)	(1.56)
	Naive	1.91	0.84	1.36	2.02	0.84	1.36	1.91	0.48	2.08	2.02	0.48	2.08
		(0.21)	(4.69)	(2.09)	(0.19)	(4.69)	(2.09)	(0.21)	(2.08)	(2.33)	(0.19)	(2.08)	(2.33)
	SIMSELEX	1.83	0.00	1.19	1.83	0.00	1.35	1.83	0.00	3.03	1.96	0.00	3.07
		(0.41)	(0.00)	(1.34)	(0.37)	(0.04)	(1.56)	(0.41)	(0.00)	(1.72)	(0.30)	(0.04)	(1.75)
	Cond	2.46	5.76	1.78	2.43	5.76	1.78	2.46	5.82	2.36	2.43	5.82	2.36
		(0.65)	(10.06)	(1.22)	(0.63)	(10.06)	(1.22)	(0.65)	(10.22)	(1.22)	(0.63)	(10.22)	(1.22)

Table C.3: Comparison of estimators for Cox survival models for the case $\boldsymbol{\theta}_2$ based on ℓ_2 estimation error, average number of false positive (FP), average number of false negative (FN) across 500 simulations.

σ_u^2	p	ℓ_2			FP			FN		
		True	Naive	SIM-SELEX	True	Naive	SIM-SELEX	True	Naive	SIM-SELEX
0.15	500	0.88	1.32	1.03	3.92	2.65	0.00	0.00	0.00	0.00
		(0.11)	(0.09)	(0.17)	(3.93)	(3.44)	(0.00)	(0.00)	(0.00)	(0.00)
	1000	0.92	1.34	1.04	4.95	3.23	0.00	0.00	0.00	0.00
		(0.11)	(0.09)	(0.17)	(4.95)	(3.76)	(0.00)	(0.00)	(0.00)	(0.00)
	2000	0.95	1.37	1.08	5.23	3.63	0.00	0.00	0.00	0.00
		(0.1)	(0.09)	(0.17)	(5.15)	(4.47)	(0.00)	(0.00)	(0.00)	(0.00)
0.30	500	0.89	1.54	1.22	3.64	2.03	0.00	0.00	0.00	0.08
		(0.11)	(0.08)	(0.18)	(3.89)	(2.81)	(0.00)	(0.00)	(0.00)	(0.27)
	1000	0.92	1.56	1.25	4.78	2.47	0.00	0.00	0.00	0.11
		(0.11)	(0.09)	(0.19)	(5.31)	(3.61)	(0.00)	(0.00)	(0.00)	(0.31)
	2000	0.96	1.58	1.27	5.29	3.13	0.00	0.00	0.00	0.17
		(0.11)	(0.08)	(0.18)	(5.65)	(4.16)	(0.00)	(0.00)	(0.00)	(0.4)

D Comparison of extrapolation functions for SIMSELEX

Several extrapolation functions for the SIMEX procedure have been proposed in the literature. The quadratic function and nonlinear means function are used most frequently. In this section, the performance of SIMSELEX when using either the quadratic or nonlinear means function in the extrapolation step are compared. Web Table D.1 presents the mean and median ℓ_2 error across 500 simulations for both linear and logistic regression — the simulation configurations are as described in Section 3.1 (linear regression) and Section 3.2 (logistic regression) of the main paper.

In the case of linear regression, the nonlinear extrapolation function results in a SIMSELEX estimator with a smaller median ℓ_2 error, but a higher mean ℓ_2 error when compared to the quadratic extrapolation function. Specifically, for small measurement error variance ($\sigma_u^2 = 0.15$), the extrapolation methods give very consistent results as measured by mean and median ℓ_2 error. However, for large measurement error variance ($\sigma_u^2 = 0.3$), there are some instances where the mean ℓ_2 error for nonlinear extrapolation is much larger than for quadratic extrapolation.

In the case of logistic regression, the quadratic extrapolation function consistently outperforms the nonlinear means function regardless of whether mean or median ℓ_2 error is used as criterion. A closer inspection of the simulation results suggest one possible explanation for the superiority of quadratic extrapolation: in many of the simulated datasets, the nonlinear means function results in extrapolants very far from the true values. This results in the large mean and median ℓ_2 error values. We attempted increasing the value of B, the number of pseudo-datasets used for the simulation step, but this did not alleviate the problem. It might be possible that an increase in both the number of λ values and the value of B can improve performance of the nonlinear extrapolation function, but this becomes computationally demanding and seems unnecessary given the good performance of quadratic extrapolation.

Table D.1: Monte Carlo mean and median ℓ_2 error of SIMSELEX estimator using nonlinear means (NL) and quadratic (Quad) extrapolation function for linear and logistic regression.

Model	p	ME type	$\sigma_u^2 = 0.15$				$\sigma_u^2 = 0.30$			
			Mean ℓ_2		Median ℓ_2		Mean ℓ_2		Median ℓ_2	
			NL	Quad	NL	Quad	NL	Quad	NL	Quad
Linear	500	Normal	0.34	0.32	0.31	0.32	0.5	0.5	0.44	0.5
		Laplace	0.37	0.33	0.31	0.32	0.55	0.51	0.47	0.52
	1000	Normal	0.34	0.34	0.32	0.34	1.14	0.53	0.48	0.53
		Laplace	0.33	0.34	0.32	0.34	0.52	0.51	0.46	0.51
	2000	Normal	0.35	0.35	0.32	0.34	0.79	0.54	0.5	0.55
		Laplace	0.92	0.36	0.34	0.36	0.58	0.55	0.5	0.54
Logistic	500	Normal	3.82	2.65	2.81	2.65	21.66	2.73	3.3	2.69
		Laplace	8.28	2.67	2.82	2.64	6.02	2.76	3.31	2.69
	1000	Normal	7.99	2.7	2.84	2.67	7.46	2.77	3.37	2.72
		Laplace	18.46	2.63	2.81	2.64	5.63	2.72	3.33	2.68
	2000	Normal	5.92	2.67	2.84	2.65	5.84	2.75	3.34	2.69
		Laplace	4.28	2.69	2.84	2.65	5.97	2.79	3.38	2.74

E SIMSELEX for Spline-Based Regression

This section provides implementation of SIMSELEX in the high-dimensional nonparametric regression setting and further demonstrates the flexibility of the procedure.

E.1 Spline Model Estimation

The proposed SIMSELEX algorithm can also be adapted to use for more flexible models such as regression using splines. Assume that the data $(\boldsymbol{W}_i, Y_i)$ are generated by an additive model $Y_i = \sum_{j=1}^{p} f_j(X_{ij}) + \epsilon_i$ with $\boldsymbol{W}_i = \boldsymbol{X}_i + \boldsymbol{U}_i$ and $\boldsymbol{U}_i$ having known covariance matrix $\boldsymbol{\Sigma}_u$. Also assume that $E[Y_i] = 0$, $i = 1, \ldots, n$. In practice, this can be achieved by centering the observed outcome variable. Furthermore, each of the functions $f_j(x)$ is assumed sufficiently smooth so that it can be well-approximated by an appropriately chosen set of basis functions. In this paper, the focus will be on an approximation using cubic B-splines with K knots. This model will have $p(K+3)$ regression coefficients that need to be estimated.

Now, assume that the true covariates $\boldsymbol{X}_i$ have been observed without measurement error. Let $\phi_{jk}(x)$, $j = 1, \ldots, p$, $k = 1, \ldots, K+3$ denote the resulting set of cubic B-spline basis functions where the knots for the jth covariate have been chosen as the $(100k)/(K+1)$th percentiles, $k = 1, \ldots, K$, of said covariate. The model to be estimated is then of the form $Y_i = \sum_{j=1}^{p} \sum_{k=1}^{K+3} \beta_{jk}\phi_{jk}(X_{ij}) + \epsilon_i$. In this setting, the jth covariate is selected if at least one of the coefficients $\beta_{jk}, k = 1, \ldots, K$ is nonzero. Therefore, it is natural to delineate all the coefficients β_{jk} into p groups, each corresponding to a covariate and containing $K + 3$ parameters. The model parameters are estimated by minimizing the penalized loss function

$$R(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[Y_i - \sum_{j=1}^{p} \sum_{k=1}^{K+3} \beta_{jk}\phi_{jk}(X_{ij}) \right]^2 + (1-\alpha)\kappa \sum_{j=1}^{p} \sqrt{\sum_{k=1}^{K+3} \beta_{jk}^2} + \alpha\kappa \sum_{j=1}^{p} \sum_{k=1}^{K+3} \|\beta_{jk}\| . \quad (1)$$

This loss function has been considered in Simon et al. (2013) for the sparse group lasso estimator. Let $\widehat{\boldsymbol{\beta}}^{\text{true}}$ denote the estimated coefficients from this model. The loss function (1) combines the lasso and group lasso penalties. The tuning parameter $\alpha \in [0, 1]$ balances overall parameter sparsity and within-group sparsity. While it is expected that only a few covariates will be selected, the nonlinear effect of each selected covariate may require a large number of basis functions to be accurately modeled. Therefore, strong overall sparsity but only mild within-group sparsity is expected. As per Simon et al. (2013), $\alpha = 0.05$ is used. The estimator of each function f_j is $\widehat{f}_j^{\text{true}}(x) = \sum_{k=1}^{K+3} \widehat{\beta}_{jk}^{\text{true}}\phi_{jk}(x)$ for all $j = 1, \ldots, p$.

Now, using the contaminated data $\boldsymbol{W}_i$, a similar procedure can be followed to obtain the naive estimator. Again, evaluate the knots of the model as equally spaced percentiles, this time of the covariates contaminated by measurement error. The corresponding cubic B-spline basis functions are denoted $\phi_{jk}^{W}(x)$. The naive estimator $\widehat{\boldsymbol{\beta}}^{\text{naive}}$ can be obtained by minimizing a function analogous to (1), but with true data X_{ij} replaced by contaminated data W_{ij} in the loss function. The naive estimator for function f_j is $\widehat{f}_j^{\text{naive}}(x) = \sum_{k=1}^{K+3} \widehat{\beta}_{jk}^{\text{naive}} \phi_{jk}^W(x)$ for all $j = 1, \ldots, p$.

To compute the SIMSELEX estimator, for each of the added noise level λ_m, generate B pseudodata $\tilde{\boldsymbol{W}}^{(b)}(\lambda_m)$, $b = 1, \ldots, B$ as before. Note that the same set of basis functions obtained for the naive estimate is used. Then, the estimate $\widehat{\beta}_{jk}^{(b)}(\lambda_m)$ for each set of pseudodata is obtained by minimizing a function analogous to (1), but with true data X_{ij} replaced by pseudodata $\tilde{W}_{ij}^{(b)}(\lambda_m)$ in the loss function. The estimates $\widehat{\beta}_{jk}^{(b)}(\lambda_m)$ are averaged across B samples to obtain $\widehat{\beta}_{jk}(\lambda_m)$ for each λ_m in the grid.

After the simulation step of SIMSELEX, the jth covariate is associated with $K+3$ "paths" $\left\{(\lambda_i, \hat{\beta}_{j1}(\lambda_i)), \ldots, ...(\lambda_i, \hat{\beta}_{j,K+3}(\lambda_i))\right\}$, each of which needs to be extrapolated to $\lambda = -1$. This is different from the parametric model settings considered in Section 3 of the main paper, where each covariate j is associated with only one parameter path $\theta_j(\lambda_i)$ that needs to be extrapolated to $\lambda = -1$. Therefore, the selection step for spline-based regression needs to be approached with some care. Here, two different approaches for selection step are considered.

The first approach for selection applies a variation of the group lasso to all $p(K+3)$ coefficients β_{jk}. This is done using a quadratic extrapolation function. Specifically, it is assumed that

$$\hat{\beta}_{jk}(\lambda_i) = \Gamma_{0jk} + \Gamma_{1jk}\lambda_i + \Gamma_{2jk}\lambda_i^2 + \varepsilon_{ijk}, \quad i = 1, \ldots, M, \quad j = 1, \ldots, p, \quad k = 1, \ldots, K+3$$

with ε_{ijk} zero-mean error terms. With this approach, the jth covariate is zeroed out if all the parameter estimates $\{\hat{\Gamma}_{ijk}\}_{i=0,1,2,\ k=1,\ldots,K}$ equal zero. Applying the group lasso, the loss function to be minimized is

$$R = \sum_{j=1}^{p} \left(\|\boldsymbol{\Theta}_j - \boldsymbol{\Lambda}\boldsymbol{\Gamma}_j\|_2^2 + \xi_3 \|\boldsymbol{\Gamma}_j\|_2 \right) \tag{2}$$

where

$$\boldsymbol{\Gamma}_j = \begin{bmatrix} \Gamma_{0j1} & \ldots & \Gamma_{0jK} \\ \Gamma_{1j1} & \ldots & \Gamma_{1jK} \\ \Gamma_{2j1} & \ldots & \Gamma_{2jK} \end{bmatrix}, \ \boldsymbol{\Theta}_j = \begin{bmatrix} \hat{\beta}_{j1}(\lambda_1) & \ldots & \hat{\beta}_{jK}(\lambda_1) \\ \vdots & & \vdots \\ \hat{\beta}_{j1}(\lambda_M) & \ldots & \hat{\beta}_{jK}(\lambda_M) \end{bmatrix}, \ \Lambda = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 \\ \vdots & \vdots & \vdots \\ 1 & \lambda_M & \lambda_M^2 \end{bmatrix},$$

and $\|.\|_2$ denotes the Frobenius norm (matrix version of the ℓ_2 norm). This is a very natural extension of the approach considered in Section 2.2 of the main paper. The tuning parameter ξ_3 can be chosen through cross-validation. Even though (2) is convex and block-separable, the minimization is computationally very expensive due to the number of model parameters. As such, an alternative approach intended to speed up computation was also considered.

The alternative approach considered for selection applies the group lasso not to each individual coefficient, but to the *norm* of each group of coefficients $\beta_{jk},\ k = 1, \ldots, K+3$ corresponding to the jth covariate. This is motivated by noting that the norm of a group of coefficients will only equal 0 if all the coefficients in the said group are equal to 0. More specifically, let $\hat{\beta}_j(\lambda_i) = [\hat{\beta}_{j1}(\lambda_i), \ldots, \hat{\beta}_{jK}(\lambda_i)]^\top$, $i = 1, \ldots, M$, $j = 1, \ldots, p$, and let $\hat{\eta}_{ij} = \left\|\hat{\beta}_j(\lambda_i)\right\|_q$ denote the corresponding

ℓ_q norm. The two scenarios considered are $q = 1$ and 2. The norm is modeled quadratically as

$$\hat{\eta}_{ij} = \Gamma_{0j} + \Gamma_{1j}\lambda_i + \Gamma_{2j}\lambda_i^2 + \varepsilon_{ij}, \quad i = 1, \ldots, M,$$

with ε_{ij} zero-mean error terms. The jth covariate is not selected if all the elements of the estimated vector $(\hat{\Gamma}_{0j}, \hat{\Gamma}_{1j}, \hat{\Gamma}_{2j})$ are equal to zero. The group lasso loss function to be minimized is

$$\tilde{R} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{p}\left(\hat{\eta}_{ij} - \Gamma_{0j} - \Gamma_{1j}\lambda_i - \Gamma_{2j}\lambda_i^2\right)^2 + \xi_4 \sum_{j=1}^{p}\sqrt{\Gamma_{0j}^2 + \Gamma_{1j}^2 + \Gamma_{2j}^2}. \tag{3}$$

Equation (3) is convex and block-separable, and can be minimized efficiently through proximal gradient descent methods. The tuning parameter ξ_4 can be chosen through cross-validation.

Finally, if the jth covariate is chosen in the selection step, extrapolation is performed separately on each β_{jk} to get the SIMSELEX estimate for each coefficient, denoted by $\widehat{\beta}_{jk}^{\text{ssx}}$. Then, the SIMSELEX estimate for each function f_j is computed as $\widehat{f}_j^s(x) = \sum_{k=1}^{K+3} \widehat{\beta}_{jk}^{\text{ssx}} \phi_{jk}^W(x)$.

E.2 Simulation

Data pairs $(\boldsymbol{W}_i, Y_i)$ were generated according to the additive model $Y_i = \sum_{j=1}^{p} f_j(X_{ij}) + \epsilon_i$, and $\boldsymbol{W}_i = \boldsymbol{X}_i + \boldsymbol{U}_i$ with $f_1(t) = 3\sin(2t) + \sin(t)$, $f_2(t) = 3\cos(2\pi/3t) + t$, $f_3(t) = (1-t)^2 - 4$, $f_4(t) = 3t$, and $f_j(t) = 0$, $j = 5, \ldots, p$. The $s = 4$ non-zero functions have all been centered at 0. The true covariates X_{ij} were generated from a Gaussian copula model with correlation structure $\Sigma_{ij} = 0.25^{|i-j|}$, see Xue-Kun Song (2000) for more details. The covariates marginal were then rescaled to have a uniform distribution on $[-3, 3]$. The measurement errors $\boldsymbol{U}_i$ were generated to be *i.i.d.* p-variate normal, $\boldsymbol{U}_i \sim N_p(\boldsymbol{0}, \sigma_u^2 \boldsymbol{I}_p)$, with $\boldsymbol{I}_p$ the $p \times p$ identity matrix. Two values of σ_u^2 were considered, $\sigma_u^2 = 0.15$ and $\sigma_u^2 = 0.3$, corresponding to 5% and 10% noise-to-signal ratios for each individual covariate. Simulations were also done for number of covariates $p \in \{100, 500\}$. Although the NSR look small in each covariate, recall from Section 3.1 of the main paper that the change in total proportion of variability ΔV increases rapidly in multivariate space. For each configuration, $N = 500$ samples were generated.

For each simulated dataset, the true, naive, and SIMSELEX estimators were computed. We are unaware of any other method in the literature dealing with spline-based regression in the high-dimensional setting when covariates are subject to measurement error. For each covariate, the number of knots was chosen to be $K = 6$. As such, each function f_j is modeled by $K + 3 = 9$ basis functions. In the simulation step of SIMSELEX, $B = 40$ sets of pseudodata are generated for each level of added measurement error. The function estimators are evaluated using integrated squared error, ISE $= \sum_{j=1}^{p} \int \left(\widehat{f}_{ij}(x) - f_{ij}(x)\right)^2 dx$, as well as the number of false positive (FP) and false negative (FN) covariates selected.

Web Table E.1 compares the performance of the SIMSELEX estimator with alternative methods of doing variable selection in the case of $p = 100$ and with $\sigma_u^2 = 0.15$. Firstly, selection approach (2) using individual models for all the coefficients β_{jk} was implemented. Secondly, approach (3) was applied both for the ℓ_1 norm and for the ℓ_2 norm, calculated based on the groups of parameters corresponding to specific variables. The table reports the MISE, the number of false positives (FP) and false negatives, and also the average time (in seconds), all calculated for 500 simulated samples. The average time was recorded based on running the simulations on one node (memory 7GB) of ManeFrame II (M2), the high-performance computing cluster of Southern Methodist University in Dallas, TX.

Table E.1: Comparison of SIMSELEX variable selection methods for spline regression with $p = 100$.

Selection	MISE	FP	FN	Time (second)
All coefficients	17.32	21.50	0.00	819.00
ℓ_1 norm	17.17	10.06	0.00	59.70
ℓ_2 norm	16.76	4.62	0.00	56.68

Considering the results in Web Table E.1, selection based on the ℓ_2 norm gives the best result, while selection based on individually considering all the coefficients gives the worst results. The latter also takes more than 14 times longer to compute (on average) than the ℓ_2 approach. The ℓ_1 approach is comparable to ℓ_2 in terms of MISE and average computation time, but has a much higher average number of false positive selections. Therefore, the SIMSELEX estimator with selection using ℓ_2 norm for parameter groups is compared with the naive estimator. The results are summarized in Web Table E.2.

Table E.2: Comparison of estimators for high-dimensional spline regression model based on estimation error (MISE), average number of false positives (FP) and false negatives (FN). Standard errors in parentheses.

σ_u^2	Estimator	$p = 100$			$p = 500$		
		MISE	FP	FN	MISE	FP	FN
0.15	True	15.96	3.68	0.00	18.05	12.11	0.00
		(2.99)	(2.75)	(0.00)	(3.28)	(6.47)	(0.00)
	Naive	37.19	9.67	0.00	47.62	16	0.00
		(7.17)	(5.51)	(0.00)	(8.41)	(10.16)	(0.00)
	SIMSELEX	16.95	5.48	0.00	21.94	6.5	0.00
		(4.63)	(3.14)	(0.00)	(6.3)	(3.84)	(0.00)
0.30	True	15.96	3.68	0.00	18.05	12.11	0.00
		(2.99)	(2.75)	(0.00)	(3.28)	(6.47)	(0.00)
	Naive	69.89	9.28	0.01	87.73	13.26	0.08
		(12.31)	(6.42)	(0.12)	(13.2)	(10.84)	(0.28)
	SIMSELEX	38.51	3.74	0.03	54.41	4.06	0.17
		(11.37)	(2.77)	(0.18)	(14.15)	(3.27)	(0.39)

Web Table E.2 demonstrates that SIMSELEX has a significantly lower estimation error (MISE) than the naive estimator in all the configurations considered. Particularly, in the case of $\sigma_u^2 = 0.15$, the SIMSELEX estimator has MISE close to the true estimator. In the case of $\sigma_u^2 = 0.3$, compared to the naive estimator, the SIMSELEX estimator reduces MISE significantly. For example, in the case of $p = 500$, the reduction in MISE resulting from using the SIMSELEX over the naive estimator is more than 38%. Even so, it is clear that measurement error has a significant effect on the recovery of the functions f_j for the case $\sigma_u^2 = 0.3$.

Regarding variable selection, the SIMSELEX estimator performs very well in the case of $\sigma_u^2 = 0.15$. In this case, SIMSELEX is always able to select the true non-zero functions by having false negatives equal 0 in all samples, while having only a slightly higher average number of false positives than the true estimator with $p = 100$ and lowest average number of of false positives with $p = 500$. In the case of $\sigma_u^2 = 0.3$, SIMSELEX gives considerably fewer false positives on averages than both the true and naive estimators. SIMSELEX does have the highest average number of false negatives for this setting, but this is still below 0.5 in all the cases considered.

Web Figure E.1 shows plots of the estimators corresponding to the first, second, and third quantiles (Q_1, Q_2, and Q_3) of ISE for the naive estimator and the SIMSELEX estimator in the case of $\sigma_u^2 = 0.15$ and $p = 500$. The SIMSELEX estimator captures the shape of the functions considerably better, especially around the peaks of f_1 and f_2. Particularly, in the case of $\sigma_u^2 = 0.15$, the SIMSELEX estimator is able to capture the shape of all the nonzero functions very well. Comparable figures for the case $\sigma_u^2 = 0.3$ and $p = 500$ are given in Web Figure E.2. As one would anticipate there, the increase in measurement error variance results in poorer recovery of the underlying functions. Even so, SIMSELEX has notably better performance than the naive approach.

F Post-Selection SIMEX Estimator

When implementing SIMSELEX, a natural question is whether the performance of the method can be improved by implementing standard SIMEX methodology after the variable selection step. That is, a method of simulation–selection–simulation–extrapolation could be implemented. The second simulation step is therefore implemented using only the selected variables, and no penalty method is used since the number of variables in the model has already been reduced. This estimator is referred as the post-selection SIMEX estimator. The section compares the performance of the SIMSELEX and the post-selection SIMEX estimator in the linear and logistic regression settings.

The data were generated as outlined in Section 3.1 and Section 3.2 of the main paper. Only the simulation configurations with Normal measurement error and the coefficients $\boldsymbol{\theta}_1$ were considered. For the post-selection SIMEX estimator, the grid of added measurement error level λ in the simulation step consists of 5 equally spaced values from 0.01 to 2 and $B = 100$ sets of pseudo-data were generated for each value of λ (this corresponds to implementation of SIMSELEX). In the extrapolation step, both the nonlinear means function and quadratic function were considered. The estimators are compared based on ℓ_2 estimation error. The simulation results are presented below in Table F.1 in the supporting information.

It can be seen that the post-selection SIMEX estimator gives smaller ℓ_2 estimation error than the SIMSELEX estimator in all the considered settings. The gain is most considerable in the

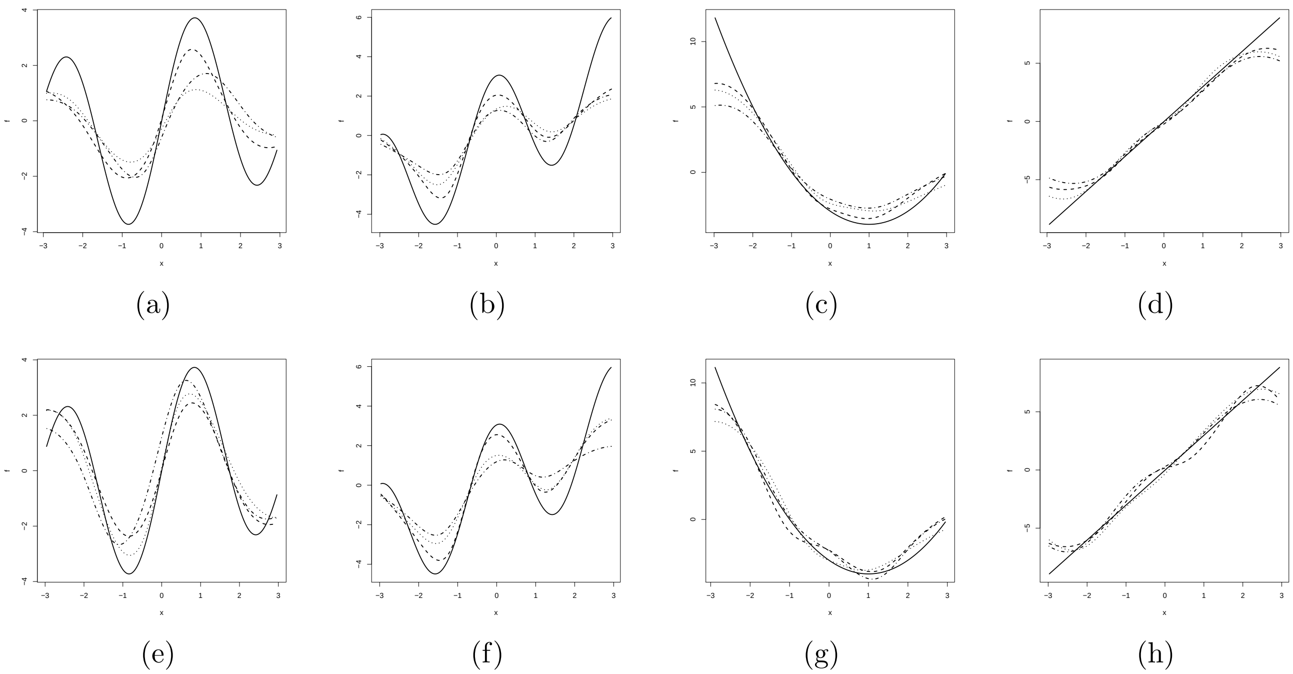


Figure E.1: Curves Q_1 (- - - - -), Q_2 (⋯⋯⋯), Q_3 (-·-·-·-), and true function (———) for the esimated functions from the naive estimators (top) and the SIMSELEX estimators (bottom) corresponding to $p = 600$ and $\sigma_u^2 = 0.15$. For (a),(e): $f_1(x) = 3\sin(2x) + \sin(x)$; for (b),(f): $f_2(x) = 3\cos(2\pi x/3) + x$; for (c), (g): $f_3(x) = (1 - x)^2 - 4$; for (d), (h): $f_4(x) = 3x$.

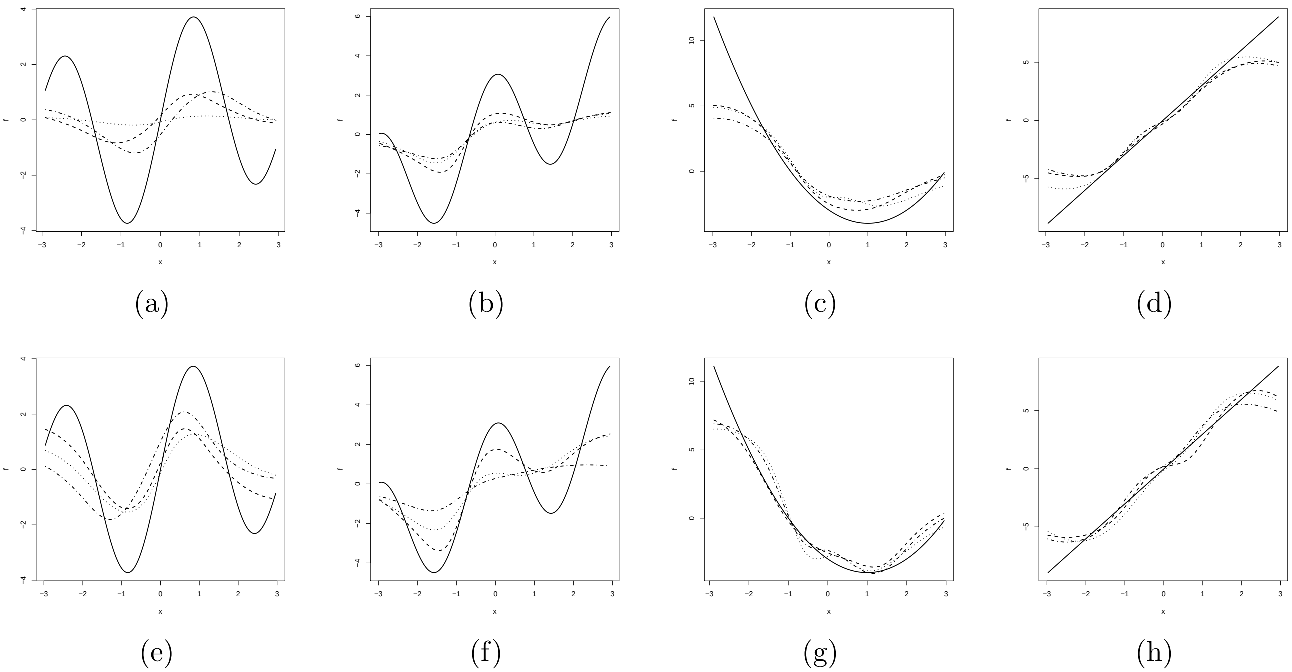


Figure E.2: Curves Q_1 (- - - - -), Q_2 (··········), Q_3 (-·-·-·-), and true function (———) for the esimated functions from the naive estimators (top) and the SIMSELEX estimators (bottom) corresponding to $p = 600$ and $\sigma_u^2 = 0.30$. For (a),(e): $f_1(x) = 3\sin(2x) + \sin(x)$; for (b),(f): $f_2(x) = 3\cos(2\pi x/3) + x$; for (c), (g): $f_3(x) = (1-x)^2 - 4$; for (d), (h): $f_4(x) = 3x$.

case of logistic regression, especially when large measurement error exists. The nonlinear and the quadratic extrapolation function have roughly the same performance in the linear model, while the quadratic function has better performance in the logistic model.

G Computation Time

Web Table G.1 presents the median computation times for the different estimators in the linear and logistic models as considered in the simulation studies of Section 3 of the main paper. In the case of the linear model, the median computation time for SIMSELEX increased by approximately 150% when going from 500 to 2000 variables, whereas the corrected scores lasso increased by around 1500% and the conic estimator increased by around 1800%. For logistic regression, the median computation time for SIMSELEX increased by 120%, while GMUS computation time increased by over 5000%. As noted in Sørensen et al. (2018), GMUS is not feasible for implementation with a large number of variables. The computation times for the conditional scores lasso for logistic regression are misleading and appear overly optimistic; the computation time here is very low as there is no sample-specific selection of tuning parameter taking place in the simulation study. In practice, this will be done using the elbow method as discussed in section B.2.

Table F.1: Comparison of SIMSELEX and post-selection SIMEX estimators using mean ℓ_2 error for linear and logistic model. Nonlinear (Nonlin) and quadratic (Quad) extrapolation were considered.

	σ_u^2	p	SIMSELEX		Post-sel. SIMEX	
			Nonlin	Quad	Nonlin	Quad
Linear	0.15	500	0.34	0.32	0.20	0.20
			(0.24)	(0.1)	(0.07)	(0.06)
		1000	0.37	0.33	0.20	0.19
			(0.65)	(0.11)	(0.07)	(0.06)
		2000	0.34	0.34	0.20	0.20
			(0.31)	(0.1)	(0.07)	(0.07)
	0.30	500	0.50	0.50	0.30	0.28
			(0.35)	(0.14)	(0.10)	(0.09)
		1000	0.55	0.51	0.30	0.28
			(0.60)	(0.15)	(0.11)	(0.10)
		2000	1.14	0.53	0.3	0.28
			(8.12)	(0.15)	(0.11)	(0.10)
Logistic	0.15	500	2.64	3.20	1.05	0.90
			(2.32)	(0.47)	(0.58)	(0.39)
		1000	6.42	3.20	0.99	0.88
			(86.1)	(0.47)	(0.52)	(0.38)
		2000	2.61	3.21	1.07	0.95
			(0.25)	(0.44)	(0.45)	(0.36)
	0.30	500	2.73	3.21	1.34	1.15
			(0.42)	(0.50)	(1.16)	(0.39)
		1000	2.75	3.21	1.37	1.24
			(0.28)	(0.52)	(0.54)	(0.37)
		2000	2.76	3.20	1.36	1.25
			(0.22)	(0.49)	(0.54)	(0.43)

Table G.1: Median computation time (in second) for different estimators. For the conditional score lasso and GMUS it is the median time to generate a coefficient path with 25 values of the tuning parameter.

Model	p	SIMSELEX	Corrected / Conditional	Conic	GMUS
Linear	500	428	58	349	-
	1000	631	264	888	-
	2000	1064	1016	6597	-
Logistic	500	572	7	-	330
	1000	798	15	-	>4.5 hours
	2000	1248	43	-	>4.5 hours
Survival	500	5435	-	-	-
	1000	7924	-	-	-
	2000	10461	-	-	-

References

Belloni, A., Rosenbaum, M., and Tsybakov, A. B. (2017). Linear and conic programming estimators in high dimensional errors-in-variables models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79,** 939–956.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Rosenbaum, M., Tsybakov, A. B., et al. (2010). Sparse recovery under matrix uncertainty. *The Annals of Statistics* **38,** 2620–2651.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics* **22,** 231–245.

Sørensen, Ø., Frigessi, A., and Thoresen, M. (2015). Measurement error in lasso: Impact and likelihood bias correction. *Statistica Sinica* pages 809–829.

Sørensen, Ø., Hellton, K. H., Frigessi, A., and Thoresen, M. (2018). Covariate selection in high-dimensional generalized linear models with measurement error. *Journal of Computational and Graphical Statistics* .

Xue-Kun Song, P. (2000). Multivariate dispersion models generated from gaussian copula. *Scandinavian Journal of Statistics* **27,** 305–320.