Biometrics 000, 000–000 DOI: 000
000 0000

Simulation–Selection–Extrapolation: Estimation in High–Dimensional Errors–in–Variables Models

Linh Nghiem[1]* and Cornelis Potgieter[2,3]**

[1]College of Business and Economics, Australian National University, Acton ACT 2601, Australia

[2] Department of Mathematics, Texas Christian University, Fort Worth, TX 76129, USA

[3]Department of Statistics, University of Johannesburg, Johannesburg, South Africa

**email:* linh.nghiem@anu.edu.au

***email:* c.potgieter@tcu.edu

Summary: Errors-in-variables models in high-dimensional settings pose two challenges in application. Firstly, the number of observed covariates is larger than the sample size, while only a small number of covariates are true predictors under an assumption of model sparsity. Secondly, the presence of measurement error can result in severely biased parameter estimates, and also affects the ability of penalized methods such as the lasso to recover the true sparsity pattern. A new estimation procedure called SIMSELEX (SIMulation-SELection-EXtrapolation) is proposed. This procedure makes double use of lasso methodology. Firstly, the lasso is used to estimate sparse solutions in the simulation step, after which a group lasso is implemented to do variable selection. The SIMSELEX estimator is shown to perform well in variable selection, and has significantly lower estimation error than naive estimators that ignore measurement error. SIMSELEX can be applied in a variety of errors-in-variables settings, including linear models, generalized linear models, and Cox survival models. It is furthermore shown in the supporting information how SIMSELEX can be applied to spline-based regression models. A simulation study is conducted to compare the SIMSELEX estimators to existing methods in the linear and logistic model settings, and to evaluate performance compared to naive methods in the Cox and spline models. Finally, the method is used to analyze a microarray dataset that contains gene expression measurements of favorable histology Wilms tumors.

Key words: Gene expressions; High-dimensional data; Measurement error; Microarray data; SIMEX; Sparsity.

1. Introduction

Errors-in-variables models arise in settings where some covariates cannot be measured with great accuracy. As such, the observed covariates tend to have larger variance than the true underlying variables, obscuring the relationship between true covariates and outcome. The inflated variances are consistent with the classic additive measurement error framework, which is assumed to hold throughout this paper. The work is motivated by microarray studies in which measurements are taken for a large number of genes, and it is of interest to identify genes related to some outcome of interest. The gene measurements are analyzed after applying a log-transformation to the strictly positive observations, further making the assumption of additive measurement error more realistic. Microarray studies tend to have both noisy measurements and small sample sizes (relative to the number of genes measured). Biological variation in the data is usually of primary interest to investigators, but is obscured by technical variation resulting from sources such as sample preparation, labeling, and hybridization, see Zakharkin et al. (2005). As such, methodology dealing with measurement error in a large-dimensional setting is needed to identify genes related to the outcome of interest. Assuming that only a small number of genes are related to the outcome of interest further imposes a requirement of solution sparsity. One example of a relevant dataset is the favorable histology Wilms tumors analyzed by Sørensen et al. (2015). In this study, Affymetric microarray gene expression measurements are used to identify genes associated with relapse within three years of successful treatment.

Formalizing the problem, let a response variable $Y \in \mathbb{R}$ be related to a function of covariates $\boldsymbol{X} \in \mathbb{R}^p$. However, the observed sample consists of measurements $(\boldsymbol{W}_1, Y_1), \ldots, (\boldsymbol{W}_n, Y_n)$, with $\boldsymbol{W}_i = \boldsymbol{X}_i + \boldsymbol{U}_i$, $i = 1, \ldots, n$ where the measurement error components $\boldsymbol{U}_i \in \mathbb{R}^p$ are *i.i.d.* Gaussian with mean zero and covariance matrix $\boldsymbol{\Sigma}_u$. The $\mathbf{U}_i$ are assumed independent of the true covariates $\mathbf{X}_i$, and the matrix $\boldsymbol{\Sigma}_u$ is assumed known or estimable from auxiliary data. This paper will consider models that specify (at least partially) a distribution for

Y conditional on $\mathbf{X}$ involving unknown parameters $\boldsymbol{\theta}$. Such models include generalized linear models, Cox survival models, and spline-based regression models. Not accounting for measurement error when fitting these models can result in biased parameter estimates, see Carroll et al. (2006). The effects of measurement error have mostly been studied in the low-dimensional setting where the sample size n is larger than the number of covariates p, see Armstrong (1985) for generalized linear models and Prentice (1982) for Cox survival models. Ma and Li (2010) also studied variable selection in the measurement error context using penalized estimating equations.

We consider these models in the high-dimensional setting where p can be much larger than n. The true $\boldsymbol{\theta}$ is assumed sparse, having only $d < \min(n, p)$ non-zero components. Of interest is both recovery of the true sparsity pattern as well as the estimation of the non-zero components of $\boldsymbol{\theta}$. When the covariates $\boldsymbol{X}$ are observed without error, the lasso and its generalizations as proposed by Tibshirani (1996) can be employed for estimating a sparse $\boldsymbol{\theta}$. The lasso adds the ℓ_1 norm of $\boldsymbol{\theta}$ to the loss function $\mathcal{L}(\boldsymbol{\theta}; Y, \mathbf{X})$ being minimized. The estimator $\widehat{\boldsymbol{\theta}}$ is defined to be

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[\mathcal{L}(\boldsymbol{\theta}; Y, \mathbf{X}) + \xi_1 \left\|\boldsymbol{\theta}\right\|_1\right] \tag{1}$$

where ξ_1 is a tuning parameter and $\left\|\boldsymbol{\theta}\right\|_1 = \sum_{j=1}^{p} |\theta_j|$ is the ℓ_1 norm, with θ_j being the jth component of $\boldsymbol{\theta}$. For the generalized linear model, $\mathcal{L}(\boldsymbol{\theta}; Y, \mathbf{X})$ is often chosen as the negative log-likelihood function, while for the Cox survival model, $\mathcal{L}(\boldsymbol{\theta}; Y, \mathbf{X})$ is the negative log of the partial likelihood function, see Hastie et al. (2015) for details.

In high dimensional settings, the presence of measurement error can have severe consequences on the lasso estimator: the number of non-zero estimates can be inflated, sometimes dramatically, and as such the true sparsity pattern is not recovered (Rosenbaum et al., 2010); see section A of the supporting information for an illustration. To correct for measurement error in the high-dimensional setting, Rosenbaum et al. (2010) proposed a matrix uncertainty selector (MU) for linear models. Rosenbaum et al. (2013) proposed an improved version of the MU selector, while Belloni et al. (2017) proved its near-optimal minimax properties and developed a conic programming estimator that can achieve the minimax bound. The conic estimator requires selection of three tuning parameters, a difficult task in practice. Another approach for handling measurement error is to modify the loss function or the conditional score functions used with the lasso, see Loh and Wainwright (2012), Sørensen et al. (2015) and Datta et al. (2017). Additionally, Sørensen et al. (2018) developed the generalized matrix uncertainty selector (GMUS) for generalized linear models. Both the conditional score approach and GMUS require subjective choices of tuning parameters.

This paper proposes a new method of estimation called Simulation-Selection-Extrapolation (SIMSELEX). This method is based on the SIMEX procedure of Cook and Stefanski (1994) which has been well-studied for correcting Normally distributed measurement error in low-dimensional settings, see for example Stefanski and Cook (1995), Küchenhoff et al. (2006) and Apanasovich et al. (2009). A SIMEX procedure for Laplace measurement error was studied by Koul et al. (2014) who considered a single covariate measured with error. Yi et al. (2015) combined SIMEX with a generalized estimating equation approach for variable selection on longitudinal data with covariate measurement error. Their variable selection step is carried out after the extrapolation step and requires a weight matrix to be prespecified.

To achieve model sparsity, the SIMSELEX approach proposed in this paper augments SIMEX with a variable selection step (based on the group lasso). Selection is performed after the simulation step and before the extrapolation step. This means that lasso-type methodology is applied twice in SIMSELEX, once to obtain a sparse solution in the simulation step, and then again in the variable selection step. The procedure inherits the flexibility of SIMEX and can be applied to a variety of different high-dimensional errors-in-variables models.

The remainder of this paper is organized as follows. In Section 2, the SIMSELEX procedure for the high-dimensional setting is developed. In Section 3, application of SIMSELEX is illustrated for linear, logistic, and Cox regression models. In Section 4, the methodology is illustrated with the favorable histology Wilms tumor data. Section 5 contains concluding remarks.

2. The SIMSELEX Estimator

Let $\boldsymbol{X}_i$ denote a vector of covariates, let $\boldsymbol{W}_i = \boldsymbol{X}_i + \boldsymbol{U}_i$ denote the covariates contaminated by measurement error $\boldsymbol{U}_i$, and let Y_i denote an outcome variable depending on $\boldsymbol{X}_i$ in a known way through parameter vector $\boldsymbol{\theta}$. The measurement error $\boldsymbol{U}_i$ is assumed independent of $\boldsymbol{X}_i$, and to be multivariate Gaussian with mean zero and known covariance matrix $\boldsymbol{\Sigma}_u$. The observed data are pairs $(\boldsymbol{W}_i, Y_i)$, $i = 1, \dots, n$. While the outcomes Y_i depend on the true covariates $\mathbf{X}_i$, only the observed $\boldsymbol{W}_i$ are available for model estimation. Now, let S denote a method for estimating $\boldsymbol{\theta}$. If the uncontaminated $\boldsymbol{X}_i$ had been observed, we could calculate the *true estimator* $\widehat{\boldsymbol{\theta}}_{\text{true}} = S(\{\mathbf{X}_i, Y_i\}_{i=1,\dots,n})$. The *naive estimator* of $\boldsymbol{\theta}$ based on the observed sample is $\widehat{\boldsymbol{\theta}}_{\text{naive}} = S(\{\mathbf{W}_i, Y_i\}_{i=1,\dots,n})$ and treats the $\mathbf{W}_i$ as if no measurement error is present. Generally, the naive estimator is neither consistent nor unbiased for $\boldsymbol{\theta}$.

A SIMEX estimator of $\boldsymbol{\theta}$ was proposed by Cook and Stefanski (1994). In the *simulation step*, a grid of values $0 < \lambda_1 < \ldots < \lambda_M$ is chosen. For each λ_m, B sets of pseudodata are generated by adding simulated random noise, $\mathbf{W}_i^{(b)}(\lambda_m) = \mathbf{W}_i + \lambda_m^{1/2}\mathbf{U}_i^{(b)},\ b = 1, \dots, B$, with $\boldsymbol{U}^{(b)}$ having the same multivariate Gaussian distribution as $\boldsymbol{U}$. Under this construction, $\text{Cov}[\mathbf{W}_i^{(b)}(\lambda_m)] = (1 + \lambda_m)\boldsymbol{\Sigma}_u$. For each set of pseudodata, the naive estimator is calculated, $\widehat{\boldsymbol{\theta}}^{(b)}(\lambda_m) = S(\{\mathbf{W}_i^{(b)}(\lambda_m), Y_i\}_{i=1,\dots,n})$. These naive estimators are then averaged, $\widehat{\boldsymbol{\theta}}(\lambda_m) = B^{-1}\sum_{b=1}^{B}\widehat{\boldsymbol{\theta}}^{(b)}(\lambda_m)$. In the *extrapolation step* $\widehat{\boldsymbol{\theta}}(\lambda)$ is modeled as a function of λ using a suitable function and extrapolated to $\lambda = -1$, which corresponds to the error-free case and gives estimator $\widehat{\boldsymbol{\theta}}_{\text{simex}}$.

Unfortunately SIMEX as described above should not be applied to the high-dimensional setting without some adjustments. Even if method S enforces sparsity of $\widehat{\boldsymbol{\theta}}^{(b)}(\lambda_m)$ for a given set of pseudodata, this does not guarantee sparsity of the average $\widehat{\boldsymbol{\theta}}(\lambda_m)$, or a consistent sparsity pattern across values of λ_m. Let $(\lambda_m, \widehat{\theta}_j(\lambda_m))$, $m = 1, \ldots, M$, denote the solution path for the θ_j, the jth component of $\boldsymbol{\theta}$, and assume $\theta_j = 0$. If $\widehat{\theta}_j(\lambda_i) \neq 0$ for even a single λ_i, it will result in an extrapolated value $\widehat{\theta}_j(-1) \neq 0$. In this way, many components of the extrapolated solution can be non-zero. The SIMSELEX (SIMulation-SELection-EXtrapolation) algorithm, presented below, addresses solution sparsity. Fundamental to the SIMSELEX approach is a *double-use* of the lasso: it is used for parameter estimation in the simulation step to ensure solution sparsity for a given set of pseudodata, and in the selection step to determine which covariates to include in the model.

2.1 *Simulation step*

The simulation step of SIMSELEX is identical to the simulation step of SIMEX. However, the criterion function being minimized for each set of pseudodata now incorporates a lasso-type penalty on the model parameters. For a given value of λ and corresponding pseudodata $(\boldsymbol{W}_i^{(b)}(\lambda), Y_i)$, $i = 1, \ldots, n$, the estimator $\widehat{\boldsymbol{\theta}}^{(b)}(\lambda)$ is calculated according to a criterion of the form in (1) with the tuning parameter $\xi_1^{(\lambda,b)}$. Note that cross-validation is implemented separately for each set of pseudodata. Two popular choices for the tuning parameter are $\xi_{\min}$, the value that minimizes the estimated prediction risk, and ξ_{1se}, the value that makes the estimated prediction risk fall within one standard error of the minimum (one-se-rule), see Friedman et al. (2001). The simulation step results in pairs $(\lambda_m, \widehat{\boldsymbol{\theta}}(\lambda_m))$, $m = 1, \ldots, M$, which are then used in the selection and extrapolation steps described next.

2.2 *Selection step*

Variable selection is performed by applying a version of the group lasso of Yuan and Lin (2006) to the pairs $(\lambda_m, \widehat{\boldsymbol{\theta}}(\lambda_m))$. It is assumed that the quadratic function serves as a good approximation to this relationship. Now, letting $\widehat{\theta}_{mj} = \widehat{\theta}_j(\lambda_m)$, it follows that

$$\widehat{\theta}_{mj} = \gamma_{0j} + \gamma_{1j}\lambda_m + \gamma_{2j}\lambda_m^2 + e_{mj}, \quad m = 1, \ldots, M, \quad j = 1, \ldots, p, \tag{2}$$

with e_{mj} denoting zero-mean error terms. To achieve model sparsity, it is desirable to shrink (as a group) the parameters $(\gamma_{0j}, \gamma_{1j}, \gamma_{2j})$ to the vector $(0, 0, 0)$ for many of the components θ_j. Extrapolation will then only be applied to the variables with non-zero solutions $(\widehat{\gamma}_{0j}, \widehat{\gamma}_{1j}, \widehat{\gamma}_{2j})$, with all other coefficients being set equal to 0. If the true model is sparse, many of the solutions $(\widehat{\gamma}_{0j}, \widehat{\gamma}_{1j}, \widehat{\gamma}_{2j})$ will be shrunk to the zero vector. Note that the assumed quadratic relationship (2) could easily be replaced with a linear relationship. However, a more "complicated" relationship may be unsuitable for selection as developed here, as such a choice would result in a non-convex loss function below. This would be very expensive computationally when paired with a lasso-type penalty.

The p equations in (2) can be written in matrix form, $\boldsymbol{\Theta} = \boldsymbol{\Lambda}\boldsymbol{\Gamma} + \boldsymbol{E}$, where

$$\boldsymbol{\Lambda} = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 \\ \vdots & \vdots & \vdots \\ 1 & \lambda_M & \lambda_M^2 \end{bmatrix}, \quad \boldsymbol{\Theta} = \begin{bmatrix} \widehat{\theta}_{11} & \dots & \widehat{\theta}_{1p} \\ \vdots & & \vdots \\ \widehat{\theta}_{M1} & \dots & \widehat{\theta}_{Mp} \end{bmatrix},$$

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma_{01} & \dots & \gamma_{0p} \\ \gamma_{11} & \dots & \gamma_{1p} \\ \gamma_{21} & \dots & \gamma_{2p} \end{bmatrix} \quad \text{and} \quad \boldsymbol{E} = \begin{bmatrix} e_{11} & \dots & e_{1p} \\ \vdots & & \vdots \\ e_{M1} & \dots & e_{Mp}. \end{bmatrix}.$$

When the kth column of the estimated matrix $\widehat{\boldsymbol{\Gamma}}$ is a zero vector, the corresponding kth column of $\widehat{\boldsymbol{\Theta}} = \boldsymbol{\Lambda}\widehat{\boldsymbol{\Gamma}}$ will also be a zero vector and the kth variable is not selected for inclusion in the model. In the present context, the group lasso penalized discrepancy function

$$D(\boldsymbol{\Gamma}) = \frac{1}{2} \sum_{m=1}^{M} \sum_{j=1}^{p} \left(\widehat{\theta}_{mj} - \gamma_{0j} - \gamma_{1j}\lambda_m - \gamma_{2j}\lambda_m^2 \right)^2 + \xi_2 \left(\sum_{j=1}^{p} \sqrt{\gamma_{0j}^2 + \gamma_{1j}^2 + \gamma_{2j}^2} \right)$$

is used with ξ_2 a tuning parameter. This function can be written in matrix form,

$$D(\boldsymbol{\Gamma}) = \frac{1}{2} \sum_{j=1}^{p} \left(\|\boldsymbol{\Theta}_j - \boldsymbol{\Lambda}\boldsymbol{\Gamma}_j\|_2^2 + \xi_2 \|\boldsymbol{\Gamma}_j\|_2 \right) \tag{3}$$

where $\boldsymbol{\Theta}_j$ and $\boldsymbol{\Gamma}_j$ denote the jth columns of $\boldsymbol{\Theta}$ and $\boldsymbol{\Gamma}$ respectively, and $\|\Gamma_j\|_2 = \sqrt{\gamma_{0j}^2 + \gamma_{1j}^2 + \gamma_{2j}^2}$ denotes the ℓ_2 norm.

Group lasso variable selection is illustrated in the left plot of Figure 1 where each path represents the ℓ_2 norm of a column of $\widehat{\mathbf{\Gamma}}$ as a function of ξ_2 in the Wilms tumor data example. Note that only eight of 2074 paths are shown. A larger value of ξ_2 sets more coefficients to zero. The cross-validation (one-se rule) value of ξ_2 is also shown.

To find $\widehat{\mathbf{\Gamma}}$ that minimizes D, standard numerical subgradient methods can be used. As equation (3) is block-separable and convex, subgradient methods will converge to the global minimum. The subgradient equations (Hastie et al., 2015, Section 5.2.2) are

$$-\mathbf{\Lambda}^T \left(\mathbf{\Theta}_j - \mathbf{\Lambda}\widehat{\mathbf{\Gamma}}_j\right) + \xi_2 \widehat{\mathbf{s}}_j = 0, \quad j = 1, \ldots, p, \tag{4}$$

where $\widehat{\mathbf{s}}_j \in \mathbb{R}^3$ is an element of the subdifferential of the norm $||\widehat{\mathbf{\Gamma}}_j||_2$. As a result, if $\widehat{\mathbf{\Gamma}}_j \neq \mathbf{0}$, then $\widehat{\mathbf{s}}_j = \widehat{\mathbf{\Gamma}}_j / ||\widehat{\mathbf{\Gamma}}_j||_2$. On the other hand, if $\widehat{\mathbf{\Gamma}}_j = \mathbf{0}$, then $\widehat{\mathbf{s}}_j$ is any vector with $\left\|\widehat{\mathbf{s}}_j\right\|_2 \leq 1$. Therefore, $\widehat{\mathbf{\Gamma}}_j$ must satisfy

$$\widehat{\mathbf{\Gamma}}_j = \begin{cases} \mathbf{0} & \text{if } \left\|\mathbf{\Lambda}^\top \mathbf{\Theta}_j\right\|_2 \leq \xi_2 \\ \left(\mathbf{\Lambda}^\top \mathbf{\Lambda} + \dfrac{\xi_2}{\left\|\widehat{\mathbf{\Gamma}}_j\right\|_2} \mathbf{I}\right)^{-1} \mathbf{\Lambda}^\top \mathbf{\Theta}_j & \text{otherwise.} \end{cases} \tag{5}$$

The first equation of (5) gives a simple rule for when to set the columns of $\widehat{\mathbf{\Gamma}}$ equal to $\mathbf{0}$ for a specific value of ξ_2. Therefore $\widehat{\mathbf{\Gamma}}$ can be computed using proximal gradient descent (Hastie et al., 2015, Section 5.3). At the kth iteration, each column $\widehat{\mathbf{\Gamma}}_j$ can be updated by first calculating $\omega_j^{(k)} = \widehat{\mathbf{\Gamma}}_j^{(k-1)} + \nu \mathbf{\Lambda}^\top (\mathbf{\Theta}_j - \mathbf{\Lambda}\widehat{\mathbf{\Gamma}}_j^{(k-1)})$ and then using this quantity to update

$$\widehat{\mathbf{\Gamma}}_j^{(k)} = \left(1 - \frac{\nu \xi_2}{\left\|\omega_j^{(k)}\right\|_2}\right)_+ \omega_j^{(k)}$$

for all $j = 1, \ldots, p$. Here, ν is the step size that needs to be specified for the algorithm and $(z)_+ = \max(z, 0)$. The convergence of the algorithm is guaranteed for step size $\nu \in (0, 1/L)$ where L is the maximum eigenvalue of the matrix $\mathbf{\Lambda}^\top \mathbf{\Lambda}/M$. The parameter ξ_2 can be chosen using cross-validation. The algorithm stops when the distance between the current estimate

$\widehat{\mathbf{\Gamma}}^{(k)}$ and the previous estimate $\widehat{\mathbf{\Gamma}}^{(k-1)}$ is smaller than some tolerance level, say 10^{-4}. The jth variable is selected for inclusion in the model if $\widehat{\mathbf{\Gamma}}_j^{(\text{final})}$ is non-zero.

2.3 *Extrapolation step*

The extrapolation step of SIMSELEX is identical to that of SIMEX, but with extrapolation only applied to the selected variables. Thus, if the jth variable has been selected for inclusion in the model, an extrapolation function $\Gamma_{\text{ex}}(\lambda)$ is fit to the simulation-step pairs $(\lambda_m, \widehat{\theta}_j(\lambda_m))$. Let $\hat{\Gamma}_{\text{ex},j}(\lambda)$ denote the extrapolation function fit obtained for the coefficient path of variable j. The SIMSELEX estimate is then given by $\hat{\theta}_j = \hat{\Gamma}_{\text{ex},j}(-1)$. Two common extrapolation functions are the quadratic and nonlinear means models, respectively $\Gamma_{\text{quad}}(\lambda) = \gamma_0 + \gamma_1\lambda + \gamma_2\lambda^2$ and $\Gamma_{\text{nonlin}}(\lambda) = \gamma_0 + \gamma_1/(\gamma_2 + \lambda)$. Note that the extrapolation step does not directly incorporate any model penality, but the coefficient paths being used for extrapolation did result from fitting a penalized model in the simulation step.

The right plot in the Figure 1 illustrates the simulation and extrapolation steps of SIMSELEX. For four genes selected in the Wilms tumor example, the plotted points represents the coefficients resulting from added measurement error level λ, and the dotted lines illustrate quadratic extrapolation to $\lambda = -1$.

[Figure 1 about here.]

It is important to note that the extrapolation step does not make use of the estimate $\widehat{\mathbf{\Gamma}}$ calculated by minimizing (3) in the selection step. Doing so will result in a final estimator $\hat{\boldsymbol{\theta}}$ that performs poorly due to aggressive shrinkage of the columns of $\widehat{\mathbf{\Gamma}}$ from the group lasso. This does give SIMSELEX the feel of model refitting post-selection as discussed by Lederer (2013). However, this is not the case. By using the estimate coefficient paths from the selection step, we are avoiding *double shrinkage* before arriving at our final estimator. A true post-selection SIMSELEX approach is discussed in section F of the supporting information.

3. Model Illustration and Simulation Results

SIMSELEX performance in high-dimensional errors-in-variables models is discussed in this section for linear, logistic, and Cox regression models and in section E of the supporting information for spline-based regression models. Where applicable, the performance of existing estimators is also included. Performance was assessed through extensive simulation studies. The metrics used for the comparison of estimators relate to the recovery of the sparsity pattern and the estimation error of the parameter estimates. Throughout the simulations, the measurement error covariance matrix was assumed known.

3.1 *Linear Regression*

Three solutions have been proposed in the literature for linear models with high-dimensional covariates subject to measurement error. Rosenbaum et al. (2010) proposed the Matrix Uncertainty Selection (MUS), which does not require knowledge of the measurement error covariance matrix $\mathbf{\Sigma}_u$. Sørensen et al. (2015) developed a corrected scores lasso, while Belloni et al. (2017) proposed a conic programming estimator. The latter two approaches require knowledge of $\mathbf{\Sigma}_u$. Furthermore, the corrected scores lasso requires the selection of one tuning parameter, and the conic programming estimator requires the selection of three tuning parameters. See section B.1 of the supporting information for a brief overview of these approaches.

For the simulation study, data pairs $(\boldsymbol{W}_i, Y_i)$, $i = 1, \ldots, n$, were generated assuming $Y_i = \boldsymbol{X}_i^\top \boldsymbol{\theta} + \varepsilon_i$ and $\boldsymbol{W}_i = \boldsymbol{X}_i + \boldsymbol{U}_i$. The true covariates $\boldsymbol{X}_i$ were generated to be *i.i.d.* p-variate Gaussian with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}$, the latter having entries $\Sigma_{ij} = \rho^{|i-j|}$ with $\rho = 0.25$. The p components of each measurement error vector $\boldsymbol{U}_i$ were generated to be either *i.i.d.* Gaussian or Laplace with mean 0 and variance σ_u^2, so that $\boldsymbol{U}_i$ has mean $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}_u = \sigma_u^2 \boldsymbol{I}_{p\times p}$. Two values were considered for the measurement error variance, $\sigma_u^2 \in \{0.15, 0.30\}$. As SIMSELEX assumes normality of the measurement error, the Laplace distribution setting was chosen in part to evaluate model robustness.

The error components ε_i were simulated to be *i.i.d.* univariate normal, $\varepsilon_i \sim N(\mathbf{0}, \sigma_\varepsilon^2)$ with $\sigma_\varepsilon^2 = 0.256^2$. The sample size was fixed at $n = 300$, and simulations were done for the number of covariates $p \in \{500, 1000, 2000\}$. Two choice of the true $\boldsymbol{\theta}$ were considered, namely $\boldsymbol{\theta}_1 = (2, 1.75, 1.5, 1.25, 1.0, 0, \ldots, 0)^\top$ and $\boldsymbol{\theta}_2 = (1, 1, 1, 1, 1, 0, \ldots, 0)^\top$. Both cases have $d = 5$ non-zero coefficients. Under each simulation configuration considered, $N = 500$ samples were generated.

The above simulation settings correspond to noise-to-signal ratios of approximately 15% and 30% for each individual covariate. However, in multivariate space a metric such as the proportional increase in total variability, $\Delta V = (\det(\boldsymbol{\Sigma}_{\mathbf{W}}) - \det(\boldsymbol{\Sigma}))\,/\det(\boldsymbol{\Sigma})$, is more informative. When $\sigma_u^2 = 0.15$, if one were to only observe the $d = 5$ non-zero covariates, $\Delta V = 1.145$, while for $p = 500$, this metric becomes $\Delta V = 6.79 \times 10^{33}$. When $\sigma_u^2 = 0.3$, the equivalent values are $\Delta V = 3.132$ for $d = 5$ and $\Delta V = 5.79 \times 10^{62}$ for $p = 500$. The dramatic increase of ΔV emphasizes the severe consequences of measurement error in high-dimensional space.

In the simulation study, five different estimators were computed: the true lasso using the uncontaminated $\boldsymbol{X}$-data, the naive lasso treating the $\boldsymbol{W}$-data as if it were uncontaminated, the conic estimator with tuning parameters as proposed by Belloni et al. (2017), the corrected lasso with the tuning parameter chosen based on 10-fold cross-validation, and SIMSELEX.

SIMSELEX used $M = 5$ equi-spaced λ values ranging from 0.01 to 2. For each λ, $B = 100$ sets of pseudodata were generated. The tuning parameter of the lasso was chosen using the one-se rule and 10-fold cross-validation. For group lasso selection, $\nu = (20L)^{-1}$ was used as step size with L the maximum eigenvalue of $\boldsymbol{\Lambda}^\top\boldsymbol{\Lambda}/M$. The lasso was implemented using the `glmnet` function in MATLAB, see Qian et al. (2013). The group lasso was implementing using our own code, available online with this paper. Extrapolation was performed using both the quadratic and nonlinear means functions. Only results for quadratic extrapolation are reported in the main paper, as this approach consistently resulted in smaller square estimation error that nonlinear extrapolation. The nonlinear means extrapolation results can be found in section D of the supporting information.

The five estimators were compared using average estimation error, $\ell_2 = \left\{N^{-1}\sum_{i=1}^{N}\sum_{j=1}^{p}(\widehat{\theta}_j^{(i)} - \theta_j)^2\right\}^{1/2}$ where $\widehat{\theta}_j^{(i)}$ denotes the estimate of θ_j obtained in the ith simulated dataset. Furthermore, each method's ability to recover the true sparsity pattern was evaluated using the average number of false positive (FP) and false negative (FN) estimates across the N simulated datasets. Note that the conic estimator does not set any estimates exactly equal to 0 and cannot be used for variable selection. The simulation results for parameter vector $\boldsymbol{\theta}_1$ are presented in Table 1, while the results for $\boldsymbol{\theta}_2$ are presented in Table C.1 in the supporting information.

[Table 1 about here.]

As seen in Table 1, the naive estimator performs worst — it has ℓ_2 error often twice that of either the conic or SIMSELEX methods. The conic estimator has comparable performance to the SIMSELEX estimators, with SIMSELEX having slightly smaller ℓ_2 error for $\boldsymbol{\theta}_1$, and the conic estimator having slightly smaller ℓ_2 error for $\boldsymbol{\theta}_2$. Both the conic and SIMSELEX estimators have smaller ℓ_2 error than the corrected scores lasso. Interestingly, the ℓ_2 error corresponding to the Normal and Laplace measurement error settings is quite similar. This suggests that SIMSELEX is robust to at least moderate departures from normality (for the simulation settings considered).

When considering the recovery of true sparsity pattern, the average number of false negatives are negligible for all methods. For the average number of false positives, the corrected lasso generally performs the worst, while SIMSELEX does not result in any false positive for the parameter specifications considered. Overall, Table 1 demonstrates that SIMSELEX can have performance superior to existing methods in the literature with regards to the performance metrics considered.

It is also worth mentioning that SIMSELEX has lower average number false positive than the true estimator. We believe this can be attributed to these methods having different FP-FN trade-off levels. When considering the Logistic and Cox Regression simulations in Sections 3.2 and 3.3, it can be seen that SIMSELEX tends to have a higher average number of false positives compared to the true estimator, while SIMSELEX still has lower average number of false positives.

3.2 *Logistic Regression*

Two solutions for performing logistic regression in a high-dimensional errors-in-variables setting have been proposed in the literature. The conditional scores lasso approach of Sørensen et al. (2015) can be applied to GLMs. This method requires that the covariance matrix $\boldsymbol{\Sigma}_u$ be known or estimable. Sørensen et al. (2018) proposed a Generalized Matrix Uncertainty Selector (GMUS) for sparse high-dimensional GLM models with measurement error. The GMUS estimator does not make use of $\boldsymbol{\Sigma}_u$. These methods are reviewed in section B.2 of the supporting information.

For the simulation study, data pairs $(\boldsymbol{W}_i, Y_i)$ were generated using $Y_i|\boldsymbol{X}_i \sim \text{Bernoulli}(p_i)$ with $\text{logit}(p_i) = \boldsymbol{X}_i^\top\boldsymbol{\theta}$. The true covariates $\boldsymbol{X}_i$, measurement error components $\boldsymbol{U}_i$, coefficient vectors $\boldsymbol{\theta}$, and sample size were exactly as outlined for the linear model simulation, see Section 3.1. The true estimator, naive estimator, conditional scores lasso, and SIMSELEX estimator using both quadratic and nonlinear extrapolation were computed for each simulated dataset for $p \in \{500, 1000, 2000\}$. The GMUS estimator was only computed for the case $p = 500$; Sørensen et al. (2018) note that GMUS becomes too computationally expensive for large p. We attempted implementation for $p = 1000$ using the `hdme` package in R, but a run time exceeding 12 hours for one sample demonstrated the impracticality of this method. For the conditional scores lasso, Sørensen et al. (2015) recommend using an elbow method to choose the tuning parameter. For the simulation study, an adapted elbow described in section B.2 of the supporting information was used to select the tuning parameter. This adapted method isn't usable in practice and does tend to give over-optimistic results for the corrected scores approach than one is likely to otherwise obtain. The performance metrics ℓ_2 error, and average number of false positives (FP) and false negatives (FN) were calculated to compare the estimators. The results for $\boldsymbol{\theta}_1$ are presented in Table 2, while the results for $\boldsymbol{\theta}_2$ are presented in Table C.2 in the supporting information.

[Table 2 about here.]

Table 2 shows that in terms of ℓ_2 error, the SIMSELEX estimator always performs better than the naive estimator. In many configurations, SIMSELEX has performance close to the true estimator. The conditional scores lasso has the smallest ℓ_2 error of the methods that control for measurement error, sometimes even outperforming the true estimator. We believe this to be an artifact of how the tuning parameter is selected in the simulation study, and does not correspond to “real world” performance. Furthermore, in terms of variable selection, the conditional scores lasso has both the highest average number of false positives and false negatives in all the considered settings. On the other hand, the SIMSELEX estimator performs variable selection well. SIMSELEX has the lowest average number of false positives in all the cases considered, and has only slightly higher average number of false negatives than the true and naive estimator. In the case of $p = 500$, GMUS has larger ℓ_2 error than both SIMSELEX and the conditional scores lasso. However, it has smallest average number of false negatives among all the estimators and a slightly larger number of average false positive than SIMSELEX. As in the linear model, performance of the estimators do not differ markedly for the Normal and Laplace measurement error settings. Again, this suggests some robustness to departure from the assumed normality of measurement error in SIMSELEX.

3.3 *Cox Proportional Hazard Model*

The Cox proportional hazard model is commonly used for the analysis of survival data. It is assumed that the random failure time T has conditional hazard function $h(t|\boldsymbol{X}) = h_0(t)\exp(\boldsymbol{X}^\top\boldsymbol{\theta})$ where $h_0(t)$ is the baseline hazard function. As survival data is frequently subject to censoring, it is assumed that the observed data are of the form $(\boldsymbol{W}_i, Y_i, I_i)$, $i = 1, \ldots, n$, where $Y_i = \min(T_i, C_i)$ with C_i being the censoring time for observation i, and $I_i = \mathcal{I}(T_i < C_i)$ being an indicator of whether failure occurred in subject i before the censoring time.

For the simulation study, the true covariates $\boldsymbol{X}_i$ and the measurement error $\boldsymbol{U}_i$ were simulated as in the linear model simulation (see Section 3.1), but with only Normally distributed measurement error being considered. The survival times T_i were simulated using the Weibull hazard as baseline, $h_0(t) = \lambda_T \rho t^{\rho-1}$ with shape parameter $\rho = 1$ and scale parameter $\lambda_T = 0.01$. The censoring times C_i were randomly drawn from an exponential distribution with rate $\lambda_C = 0.001$. Two choice of the true $\boldsymbol{\theta}$ were considered, $\boldsymbol{\theta}_1 = (1, 1, 1, 1, 1, 0, \ldots, 0)^\top$ and $\boldsymbol{\theta}_2 = (2, 1.75, 1.50, 1.25, 1, 0, \ldots, 0)^\top$. For $\boldsymbol{\theta}_1$, the model configuration resulted in samples with between 20% and 25% of the observations being censored, while for $\boldsymbol{\theta}_2$, between 25% and 30% of the observations were censored. The sample size was fixed at $n = 300$, and simulations were done for number of covariates $p \in \{500, 1000, 2000\}$.

For the Cox model, implementation of SIMSELEX is much more computationally intensive than the linear and logistic models. This can be attributed to computation of the generalized lasso for the Cox model, see Section 3.5 of Hastie et al. (2015). As such, only $B = 40$ replicates were used for each λ value in the extrapolation step of the SIMSELEX algorithm. It should further be noted that, to the best of our knowledge, the Cox model with high-dimensional data subject to measurement error has not been considered by any other authors. As such, there is no competitor method for use in the simulation study. However, the model using the true covariates not subject to measurement error can be viewed as a gold standard measure of performance. Finally, the naive model was also implemented. The simulation results for the case of $\boldsymbol{\theta}_1$ are reported in Table 3, while the results for the case of $\boldsymbol{\theta}_2$ are presented in Table C.3 in the supporting information.

[Table 3 about here.]

Table 3 shows that the SIMSELEX has a significantly lower ℓ_2 error than the naive estimator. With regards to recovery of the sparsity pattern, SIMSELEX has negligible average number of false positives in all the considered settings, while the naive estimator and the true estimator respectively result in the selection of more than 10 and 2 false positives on average. Neither the true nor the naive estimator results in false negatives, while the SIMSELEX estimator has average number of false negatives around 0.05 for the case $\sigma_u^2 = 0.15$ and around 0.6 for the case $\sigma_u^2 = 0.3$.

3.4 *Computational Time*

The nature of SIMSELEX may lead one to suspect that it is a computationally inefficient method. We have investigated how SIMSELEX scales with increasing sample size and present here a comparison with other existing methods for linear and logistic regression.

The bulk of SIMSELEX computational time is taken up by generating pseudodata and model fitting in the simulation step. Even so, if algorithms exist for fast computation of the true estimator, then implementation for the pseudodata is equally fast. Furthermore, the generation of the psuedodata only requires the simulation of normal random vectors, for which fast algorithms exist. Consider the linear model as an example. In the simulation study, the median implementation time of the simulation step with 5 values of λ and $B = 100$ replicates per λ was approximately 350, 480, and 760 seconds for $p = 500, 1000$, and 2000 respectively. For logistic regression, thes equivalent times were 510, 680, and 1010 seconds. The simulation step for the Cox survival model takes much longer time: Even with only $B = 40$ replicates with $p = 500$, the median time is approximately 5380 seconds. For all three models, the median implementation time for selection and extrapolation combined was less than 270 seconds.

When compared to other estimation methods, SIMSELEX scales well with the dimension of the problem. For the linear model, the conic estimator is slow to compute for a large number of covariates. For $p = 2000$, the median computation time of the conic estimator was around

6600 seconds, roughly six times longer than SIMSELEX for the same dimension size. The corrected lasso tends to be faster than SIMSELEX for $p = 500$ and 1000, but takes roughly the same amount of time for $p = 2000$. For logistic regression, the conditional scores lasso takes less time to compute than the SIMSELEX procedure. However, the relevant tuning parameter is selected using a subjective rule and not in a data-driven way. As previously stated, computation of the GMUS estimator is does not scale well. Further details and tabulated computation times can be found in section G of the supporting information.

4. Microarray Analysis

We analyzed an Affymetrix microarray dataset containing gene expression measurements of 144 favorable histology Wilms tumors. The data are publicly available on the ArrayExpress website under access number E-GEOD-10320. In these Wilms tumors, the cancer cell's nuclei is not very large or distorted, so a high proportion of patients are successfully treated. However, relapse is a possibility after treatment. It is of interest to identify any genes associated with relapse. A total of 53 patients experienced a relapse, while 91 patients had no relapse over a three year follow-up. Replicate data are available for each patient as multiple probes were collected per patient. This allows for the estimation of gene-specific measurement error variances. The analysis is performing after applying a logarithmic transformation.

To make our analysis comparable with that previously done by Sørensen et al. (2015), data preprocessing was done as described by them. The raw data were processed using the Bayesian Gene Expression (BGX) Bioconductor of Hein et al. (2005) creating a posterior distribution for the log-scale expression level of each gene in each sample. For gene j in patient i, the posterior mean $\widehat{\mu}_{ij}$ was then taken as an estimates of the true gene expression level.

Now, let $\widehat{\boldsymbol{\mu}}_j = (\widehat{\mu}_{1j}, \ldots, \widehat{\mu}_{nj})^\top$ denote the estimated vector of gene expression levels for gene $j = 1, \ldots, p$ for the n patients. Furthermore, let $\bar{\mu}_j = (1/n)\sum_{i=1}^{n} \widehat{\mu}_{ij}$ and $\widehat{\sigma}_j^2 =$

$(1/n)\sum_{i=1}^{n}(\widehat{\mu}_{ij}-\bar{\mu}_j)^2$ denote the estimated mean and variance of gene j. Standardized measurements $\mathbf{W}_i=(W_{i1},\ldots,W_{ip})$, $i=1,\ldots,n$ were then calculated as $W_{ij}=(\widehat{\mu}_{ij}-\bar{\mu}_j)/\widehat{\sigma}_j,\ i=1,\ldots,n,\ j=1,\ldots,p$. To estimate $\mathbf{\Sigma}_u$, it was assumed that measurement error is independent of the patient's true gene expression levels and that the associated variance is constant across patients for a given gene. Let $\text{var}(\widehat{\mu}_{ij})$ denote the posterior variance of the estimated distribution of gene j in patient i. These estimates were then combined, $\widehat{\sigma}^2_{u,j}=(1/n)\sum_{i=1}^{n}\text{var}(\widehat{\mu}_{ij})$, and the measurement error covariance matrix associated with $\boldsymbol{W}$ was estimated by the diagonal matrix with elements $(\widehat{\mathbf{\Sigma}}_u)_{j,j}=\widehat{\sigma}^2_{uj}/\widehat{\sigma}^2_j, j=1,\ldots,p$. Only the $p=2074$ genes with $\widehat{\sigma}^2_{u,j}<(1/2)\widehat{\sigma}^2_j$, i.e. estimated noise-to-signal ratio less than 1, were retained for analysis.

Using the data $(\boldsymbol{W}_i,Y_i)$, $i=1,\ldots,n$, with Y_i an indicator of relapse, four different procedures were used to fit a logistic regression model to the data. These procedures are a naive model with lasso penalty, the conditional scores lasso of Sørensen et al. (2015), the SIMSELEX model, and a SIMEX model without variable selection. For the naive, SIMSELEX and SIMEX models, 10-fold cross-validation using the one-standard-error rule was used to select the tuning parameter. For SIMEX and SIMSELEX, a grid of 16 equally spaced λ-values from 0.01 to 2 and $B=100$ replicates were used in the simulation step. The elbow method was used for tuning parameters selection in the conditional scores lasso. SIMEX without selection identified 1699 out of 2074 genes for inclusion in the model. Though many of the estimated coefficients are close to zero, 17 estimated coefficients exceed 0.1, and a further 41 exceed 0.01. Few would consider the results from this analysis to be congruent with a sparse model. Results of the other three analyses are in Table 4.

[Table 4 about here.]

The naive approach identified 26 non-zero genes, while conditional scores identified 13 non-zero genes. SIMSELEX identified only 4 non-zero genes. Note that one of the genes chosen by SIMSELEX was not chosen by the conditional scores method (although it was chosen by the naive estimator). However, the magnitude of the estimated coefficients were much larger for SIMSELEX compared to the naive and conditional scores estimators. The large number of genes selected by the naive and conditional scores approaches are potentially a consequence of the false positive rates seen in the simulation studies. While SIMSELEX does suffer from the occasional false negative, this rate was lower in our simulation studies than the equivalent rate for the conditional scores lasso.

5. Discussion

The paper presents a modified SIMEX algorithm with a selection step for sparse models estimation in high-dimensional settings with covariates subject to measurement error. This SIMSELEX algorithm is explored in linear and logistic regression models as well as the Cox proportional hazards model. Spline-based regression is considered in section E of the supporting information. In the linear model, SIMSELEX has performance comparable to the corrected lasso. In the logistic model, it has much better performance than the corrected scores lasso. In the Cox model and spline-model settings, no other estimators have been proposed in the literature. For these, it is shown that the method leads to much better performance than a naive approach that ignores measurement error, and compares favorably to estimators obtained using uncontaminated data.

It was noted that SIMSELEX requires the measurement error covariance matrix be known or estimable. In our data application, an estimation method based on the BGX Bioconductor of Hein et al. (2005) was used. The development and comparison of other methods for estimating measurement error covariance matrices will be explored in future work. Further work around reducing the number of false negatives in SIMSELEX will also be conducted. For example, the group lasso used for variable selection provides an ordering for the inclusion/exclusion of variables in the model (see, for example, Figure 1). As such, a decision can be made beforehand to include an additional number of variables, say q, after selection.

Thus, if selection recommends keeping $\hat{p}$ variables, then the practitioner keeps $\hat{p}+q$ variables for extrapolation. The performance of this idea was not explored here.

References

Apanasovich, T. V., Carroll, R. J., and Maity, A. (2009). Simex and standard error estimation in semiparametric measurement error models. Electronic journal of statistics **3,** 318.

Armstrong, B. (1985). Measurement error in the generalised linear model. Communications in Statistics-Simulation and Computation **14,** 529–544.

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.

Carroll, R. J., Ruppert, D., Stefanski, L. A., and Crainiceanu, C. M. (2006). Measurement error in nonlinear models: a modern perspective. CRC press.

Cook, J. R. and Stefanski, L. A. (1994). Simulation-extrapolation estimation in parametric measurement error models. Journal of the American Statistical association **89,** 1314–1328.

Datta, A., Zou, H., et al. (2017). Cocolasso for high-dimensional error-in-variables regression. The Annals of Statistics **45,** 2400–2426.

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

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). Statistical learning with sparsity: the lasso and generalizations. CRC press.

Hein, A.-M. K., Richardson, S., Causton, H. C., Ambler, G. K., and Green, P. J. (2005). Bgx: a fully bayesian integrated approach to the analysis of affymetrix genechip data. Biostatistics **6,** 349–373.

Koul, H. L., Song, W., et al. (2014). Simulation extrapolation estimation in parametric models with laplace measurement error. Electronic Journal of Statistics **8,** 1973–1995.

Küchenhoff, H., Mwalili, S. M., and Lesaffre, E. (2006). A general method for dealing with misclassification in regression: the misclassification simex. Biometrics **62,** 85–96.

Lederer, J. (2013). Trust, but verify: benefits and pitfalls of least-squares refitting in high dimensions. arXiv preprint arXiv:1306.0113 .

Loh, P.-L. and Wainwright, M. J. (2012). High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. The Annals of Statistics **40,** 1637–1664.

Ma, Y. and Li, R. (2010). Variable selection in measurement error models. Bernoulli: official journal of the Bernoulli Society for Mathematical Statistics and Probability **16,** 274.

Prentice, R. (1982). Covariate measurement errors and parameter estimation in a failure time regression model. Biometrika **69,** 331–342.

Qian, J., Hastie, T., Friedman, J., Tibshirani, R., and Simon, N. (2013). Glmnet for matlab 2013. URL http://www. stanford. edu/˜ hastie/glmnet_matlab .

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

Rosenbaum, M., Tsybakov, A. B., et al. (2013). Improved matrix uncertainty selector. In From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner, pages 276–290. Institute of Mathematical Statistics.

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 .

Stefanski, L. A. and Cook, J. R. (1995). Simulation-extrapolation: the measurement error jackknife. Journal of the American Statistical Association **90,** 1247–1256.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pages 267–288.

Yi, G. Y., Tan, X., and Li, R. (2015). Variable selection and inference procedures for marginal analysis of longitudinal data with missing observations and covariate measurement error. Canadian Journal of Statistics **43,** 498–518.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **68,** 49–67.

Zakharkin, S. O., Kim, K., Mehta, T., Chen, L., Barnes, S., Scheirer, K. E., Parrish, R. S., Allison, D. B., and Page, G. P. (2005). Sources of variation in affymetrix microarray experiments. BMC bioinformatics **6,** 214.

Supporting Information

Web Appendices, Tables and Figures referenced in Sections 1, 2, and 3, are available with this paper at the Biometrics website on Wiley Online Library. MATLAB code that implements the SIMSELEX procedure is available at `https://github.com/lnghiemum/SIMSELEX`.

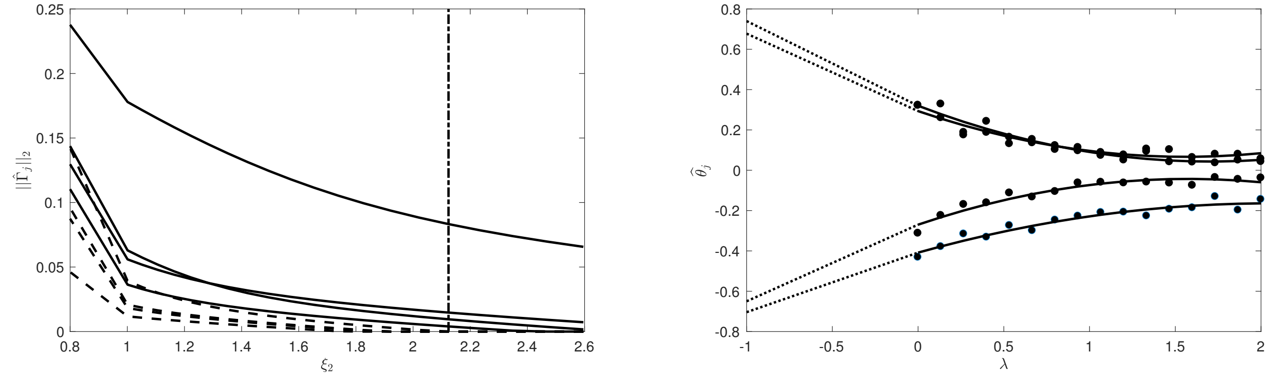

Figure 1: SIMSELEX illustration using microarray data (Section 4). Left figure: solid and dashed lines represent the norms $||\widehat{\mathbf{\Gamma}}_j||_2$ of the selected and (some) unselected genes respectively; the vertical dash-dot line is the one-se cross-validation tuning parameter. Right figure: coefficients of selected genes are modeled quadratically in λ and then extrapolated to $\lambda = -1$.

Table 1: Comparison of estimators for linear regression ~~with~~ with the case of $\boldsymbol{\theta_1}$ based on ℓ_2 estimation error, average number of false positives (FP) and false negatives (FN). Standard errors in parentheses.

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	0.98	0.00	0.09	0.81	0.00	0.09	0.82	0.00	0.09	0.83	0.00
		(0.02)	(2.06)	(0.00)	(0.02)	(1.56)	(0.00)	(0.02)	(1.86)	(0.00)	(0.02)	(1.65)	(0.00)
	Naive	0.73	1.36	0.00	0.74	0.99	0.00	1.11	1.48	0.00	1.12	1.12	0.00
		(0.08)	(3.3)	(0.00)	(0.08)	(2.21)	(0.00)	(0.1)	(3.29)	(0.00)	(0.1)	(2.24)	(0.00)
	SIMSELEX	0.32	0.00	0.00	0.34	0.00	0.00	0.5	0.00	0.00	0.52	0.00	0.00
		(0.1)	(0.00)	(0.00)	(0.11)	(0.00)	(0.00)	(0.14)	(0.00)	(0.00)	(0.16)	(0.04)	(0.00)
	Conic	0.37	-	-	0.38	-	-	0.52	-	-	0.53	-	-
		(0.07)	-	-	(0.06)	-	-	(0.1)	-	-	(0.1)	-	-
	Corrected	0.43	2.3	0.00	0.44	1.76	0.00	0.62	2.74	0.00	0.63	2.1	0.00
		(0.08)	(5.27)	(0.00)	(0.08)	(3.51)	(0.00)	(0.11)	(4.93)	(0.00)	(0.11)	(3.88)	(0.00)
1000	True	0.09	1.27	0.00	0.09	1.06	0.00	0.09	1.01	0.00	0.09	1.06	0.00
		(0.02)	(2.55)	(0.00)	(0.02)	(2.18)	(0.00)	(0.02)	(2.04)	(0.00)	(0.02)	(2.22)	(0.00)
	Naive	0.75	1.69	0.00	0.76	1.18	0.00	1.14	1.38	0.00	1.15	1.39	0.00
		(0.08)	(3.29)	(0.00)	(0.08)	(2.72)	(0.00)	(0.1)	(3.03)	(0.00)	(0.11)	(3.16)	(0.00)
	SIMSELEX	0.33	0.00	0.00	0.35	0.00	0.00	0.51	0.00	0.00	0.53	0.00	0.00
		(0.11)	(0.00)	(0.00)	(0.12)	(0.00)	(0.00)	(0.15)	(0.00)	(0.04)	(0.16)	(0.00)	(0.04)
	Conic	0.39	-	-	0.4	-	-	0.55	-	-	0.56	-	-
		(0.07)	-	-	(0.07)	-	-	(0.1)	-	-	(0.1)	-	-
	Corrected	0.44	3.48	0.00	0.46	3.11	0.00	0.63	3.57	0.00	0.65	3.14	0.00
		(0.09)	(6.37)	(0.00)	(0.08)	(6.26)	(0.00)	(0.12)	(5.97)	(0.00)	(0.13)	(5.26)	(0.00)
2000	True	0.1	1.29	0.00	0.1	1.45	0.00	0.1	1.56	0.00	0.1	1.32	0.00
		(0.02)	(2.68)	(0.00)	(0.02)	(3)	(0.00)	(0.02)	(3.41)	(0.00)	(0.02)	(2.62)	(0.00)
	Naive	0.77	1.76	0.00	0.78	1.59	0.00	1.17	1.89	0.00	1.17	2.06	0.00
		(0.08)	(3.52)	(0.00)	(0.09)	(5.06)	(0.00)	(0.1)	(4.57)	(0.00)	(0.11)	(5.72)	(0.00)
	SIMSELEX	0.34	0.00	0.00	0.36	0.00	0.00	0.53	0.00	0.00	0.55	0.00	0.01
		(0.1)	(0.00)	(0.00)	(0.11)	(0.00)	(0.00)	(0.15)	(0.04)	(0.00)	(0.17)	(0.00)	(0.09)
	Conic	0.41	-	-	0.41	-	-	0.59	-	-	0.59	-	-
		(0.07)	-	-	(0.07)	-	-	(0.1)	-	-	(0.11)	-	-
	Corrected	0.45	4.91	0.00	0.47	3.88	0.00	0.64	5.42	0.00	0.66	3.83	0.00
		(0.08)	(7.66)	(0.00)	(0.09)	(7.12)	(0.00)	(0.12)	(8.11)	(0.00)	(0.13)	(5.99)	(0.00)

Table 2: Comparison of estimators for logistic regression ~~with~~ with the case of $\boldsymbol{\theta_1}$ based on ℓ_2 estimation error, average number of false positives (FP) and false negatives (FN). Standard errors in parentheses.

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	2.62	0.39	0.23	2.61	0.39	0.23	2.61	0.54	0.25	2.59	0.54	0.25
		(0.21)	(2.54)	(0.43)	(0.21)	(2.54)	(0.43)	(0.23)	(3)	(0.59)	(0.2)	(3)	(0.59)
	Naive	2.83	0.59	0.57	2.83	0.59	0.57	2.99	0.42	1.08	2.99	0.42	1.08
		(0.22)	(3.79)	(1.03)	(0.22)	(3.79)	(1.03)	(0.23)	(1.59)	(1.6)	(0.24)	(1.59)	(1.6)
	SIMSELEX	2.65	0.01	0.65	2.63	0.01	0.68	2.73	0.00	1.38	2.74	0.00	1.57
		(0.43)	(0.09)	(0.58)	(0.39)	(0.09)	(0.61)	(0.46)	(0.00)	(1.09)	(0.45)	(0.06)	(1.19)
	Cond Scores	2.36	7.02	1.15	2.33	7.02	1.15	2.58	5.67	1.7	2.53	5.67	1.7
		(0.65)	(9.77)	(0.95)	(0.61)	(9.77)	(0.95)	(0.56)	(7.94)	(1.09)	(0.57)	(7.94)	(1.09)
	GMUS	2.67	0.21	0.21	2.87	0.21	0.21	2.75	0.66	0.22	2.74	0.66	0.22
		(0.08)	(0.52)	(0.41)	(0.07)	(0.52)	(0.41)	(0.08)	(0.98)	(0.42)	(0.08)	(0.98)	(0.42)
1000	True	2.65	0.36	0.26	2.64	0.36	0.26	2.64	0.61	0.3	2.64	0.61	0.3
		(0.19)	(1.38)	(0.48)	(0.21)	(1.38)	(0.48)	(0.21)	(3.38)	(0.55)	(0.21)	(3.38)	(0.55)
	Naive	2.86	0.7	0.63	2.85	0.7	0.63	3.01	0.78	1.25	3.01	0.78	1.25
		(0.22)	(3.62)	(1.11)	(0.22)	(3.62)	(1.11)	(0.24)	(4.53)	(1.74)	(0.23)	(4.53)	(1.74)
	SIMSELEX	2.67	0.00	0.72	2.65	0.00	0.71	2.76	0.00	1.59	2.77	0.00	1.61
		(0.44)	(0.06)	(0.64)	(0.41)	(0.06)	(0.64)	(0.46)	(0.00)	(1.14)	(0.42)	(0.00)	(1.21)
	Cond Scores	2.44	8.82	1.18	2.46	8.82	1.18	2.62	7.53	1.75	2.64	7.53	1.75
		(0.63)	(11.22)	(0.99)	(0.66)	(11.22)	(0.99)	(0.59)	(11.2)	(1.05)	(0.57)	(11.2)	(1.05)
2000	True	2.66	0.75	0.33	2.65	0.75	0.33	2.65	0.89	0.35	2.65	0.89	0.35
		(0.22)	(3.7)	(0.66)	(0.21)	(3.7)	(0.66)	(0.22)	(3.39)	(0.6)	(0.23)	(3.39)	(0.6)
	Naive	2.88	0.56	0.68	2.87	0.56	0.68	3.02	0.84	1.23	3.03	0.84	1.23
		(0.2)	(2.68)	(1.1)	(0.22)	(2.68)	(1.1)	(0.23)	(4.63)	(1.71)	(0.23)	(4.63)	(1.71)
	SIMSELEX	2.70	0.01	0.78	2.68	0.01	0.80	2.77	0.00	1.76	2.80	0.00	1.79
		(0.41)	(0.08)	(0.67)	(0.42)	(0.06)	(0.68)	(0.44)	(0.04)	(1.25)	(0.44)	(0.00)	(1.20)
	Cond Scores	2.52	12.04	1.29	2.52	12.04	1.29	2.75	10.58	1.85	2.71	10.58	1.85
		(0.63)	(14.38)	(0.94)	(0.65)	(14.38)	(0.94)	(0.62)	(15.59)	(1.11)	(0.61)	(15.59)	(1.11)

Table 3: Comparison of estimators for Cox survival models for the case $\boldsymbol{\theta}_1$ based on ℓ_2 estimation error, average number of false positives (FP) and false negatives (FN). Standard errors in parentheses.

σ_u^2	p	ℓ_2			FP			FN		
		True	Naive	SIM-SELEX	True	Naive	SIM-SELEX	True	Naive	SIM-SELEX
0.15	500	1.36	2.25	1.8	8.59	3.66	0.00	0.00	0.00	0.03
		(0.15)	(0.11)	(0.23)	(6.34)	(3.99)	(0.00)	(0.00)	(0.00)	(0.18)
	1000	1.41	2.27	1.82	10.8	4.77	0.00	0.00	0.00	0.05
		(0.15)	(0.11)	(0.23)	(7.63)	(5.18)	(0.00)	(0.00)	(0.00)	(0.22)
	2000	1.47	2.31	1.89	12.92	5.68	0.00	0.00	0.00	0.07
		(0.15)	(0.12)	(0.24)	(8.97)	(6.32)	(0.00)	(0.00)	(0.00)	(0.26)
0.30	500	1.37	2.58	2.19	8.03	2.33	0.00	0.00	0.00	0.52
		(0.16)	(0.1)	(0.21)	(6.02)	(3.09)	(0.00)	(0.00)	(0.06)	(0.51)
	1000	1.43	2.6	2.22	10.31	3.24	0.00	0.00	0.00	0.55
		(0.15)	(0.09)	(0.2)	(7.6)	(4.1)	(0.00)	(0.00)	(0.00)	(0.53)
	2000	1.46	2.63	2.26	13.71	3.84	0.00	0.00	0.00	0.7
		(0.16)	(0.09)	(0.19)	(9.5)	(4.97)	(0.00)	(0.00)	(0.04)	(0.5)

Table 4: Gene symbols and estimated coefficients from the naive lasso, the conditional scores lasso, and the SIMSELEX estimator applied to the Wilms tumors data. Genes selected by SIMSELEX are printed in bold.

Gene	Naive	Conditional scores	SIMSELEX
202016_at	**-0.2216**	**-0.0348**	**-0.7038**
205132_at	**-0.1997**	**-0.2127**	**-0.6498**
213089_at	**0.2096**	**0.0575**	**0.6775**
207761_at	**0.0691**	-	**0.7399**
209466_x_at	-0.0310	-0.2425	
218678_at	-0.1256	-0.1600	
209259_s_at	-0.1038	-0.1599	
209281_s_at	-0.0511	-0.1054	
204710_s_at	-0.2004	-0.0958	
202766_s_at	-	-0.0740	
208905_at	-	-0.0463	
201194_at	-	-0.0448	
211737_x_at	-	-0.0279	
203156_at	-0.1090	-0.0128	
213779_at	0.1142		
201859_at	-0.1087		
208965_s_at	0.1388		
205933_at	0.0913		
(11 more non-zero genes)	$\lvert\cdot\rvert < 0.06$		