Adaptive Handling of Dependence in High-Dimensional Regression Modeling - Supplementary material

Florian Hébert
Institut Agro, Univ. Rennes, CNRS, IRMAR, 35000 Rennes, France
and
David Causeur
Institut Agro, Univ. Rennes, CNRS, IRMAR, 35000 Rennes, France
and
Mathieu Emily
Institut Agro, Univ. Rennes, CNRS, IRMAR, 35000 Rennes, France

May 9, 2022

1 Proofs of Theorems

1.1 Proof of Theorem 1

Let us first formulate $g(\sigma_{\tilde{x}y}, C)$ as a function of γ and λ:

$$\begin{aligned} g(\gamma,\lambda) &= \frac{\gamma' D_\lambda \gamma \gamma' D_\lambda^{-1}\gamma}{(\gamma'\gamma)^2} = \frac{\sum_{j=1}^p \lambda_j \gamma_j^2 \sum_{j=1}^p \frac{\gamma_j^2}{\lambda_j}}{\left(\sum_{j=1}^p \gamma_j^2\right)^2}, \\ g(\gamma,\lambda) &= \frac{\lambda'(\gamma\odot\gamma).(\lambda^{\odot -1})'(\gamma\odot\gamma)}{(\gamma\odot\gamma)'(\gamma\odot\gamma)}, \end{aligned} \tag{1}$$

where $\lambda^{\odot -1} = (1/\lambda_1, \ldots, 1/\lambda_p)'$ and $\odot$ is the term-by-term product of two vectors of equal dimension.

A sharp upper bound for $g(\gamma,\lambda)$ over all possible γ is deduced from expression (1) by equating to zero the derivative of $g(\gamma,\lambda)$ with respect to $\gamma\odot\gamma$:

$$g(\gamma,\lambda) \leq v(\lambda)'\lambda.v(\lambda)'\lambda^{\odot -1} = g_{\max}(\lambda),$$

where $v(\lambda)$ denotes the eigenvector associated to the only positive eigenvalue of the matrix $\lambda\lambda^{\odot -1'} + \lambda^{\odot -1}\lambda'$ with $v(\lambda)'v(\lambda) = 1$.

Indeed, since

$$\lambda\lambda^{\odot -1'} + \lambda^{\odot -1}\lambda' = \frac{1}{2}[\lambda + \lambda^{\odot -1}, \lambda - \lambda^{\odot -1}] \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} [\lambda + \lambda^{\odot -1}, \lambda - \lambda^{\odot -1}]',$$

then $\lambda\lambda^{\odot -1'} + \lambda^{\odot -1}\lambda'$ can only have one positive eigenvalue and the coordinates of the corresponding eigenvector $v(\lambda)$ are all positive.

Moreover, for any vector γ with only one nonzero coordinate, the corresponding value of $g(\gamma, \lambda)$ equals one.

1.2 Proof of Theorem 2

First, let us reformulate expression (10) in the main paper of $\hat{\beta}_\kappa$ by introducing the conditional covariance matrix S:

$$\begin{aligned} D_s^{-1} S_x D_s^{-1} + \kappa \mathrm{I}_p &= D_s^{-1} S D_s^{-1} + \frac{D_s^{-1} s_{xy} s_{xy}' D_s^{-1}}{s_y^2} + \kappa \mathrm{I}_p, \\ &= \hat{C} + \frac{D_s^{-1} s_{xy} s_{xy}' D_s^{-1}}{s_y^2} + \kappa \mathrm{I}_p. \end{aligned}$$

Using the Sherman-Morrison identity (see Hager (1989), equation (2)),

$$(D_s^{-1} S_x D_s^{-1} + \kappa \mathrm{I}_p)^{-1} = (\hat{C} + \kappa \mathrm{I}_p)^{-1} - \frac{(\hat{C} + \kappa \mathrm{I}_p)^{-1} D_s^{-1} s_{xy} s_{xy}' D_s^{-1} (\hat{C} + \kappa \mathrm{I}_p)^{-1}}{s_y^2 + s_{xy}' D_s^{-1} (\hat{C} + \kappa \mathrm{I}_p)^{-1} D_s^{-1} s_{xy}}.$$

Expression (10) of $\hat{\beta}_\kappa$ can therefore be reformulated as follows:

$$\hat{\beta}_\kappa = \frac{s_y^2}{s_y^2 + s_{xy}' D_s^{-1} (\hat{C} + \kappa \mathrm{I}_p)^{-1} D_s^{-1} s_{xy}} (\hat{C} + \kappa \mathrm{I}_p)^{-1} D_s^{-1} s_{xy}.$$

It is deduced that $\hat{\beta}_\kappa$ and $(\hat{C} + \kappa \mathrm{I}_p)^{-1} D_s^{-1} s_{xy}$ are collinear, which also implies that:

$$L_{\mathrm{Ridge}}(X, \kappa) \equiv (X - \bar{X})' D_s^{-1} (\hat{C} + \kappa \mathrm{I}_p)^{-1} D_s^{-1} s_{xy}.$$

Introducing the eigendecomposition of $\hat{C}$ leads to:

$$L_{\mathrm{Ridge}}(X, \kappa) \equiv (X - \bar{X})' D_s^{-1} \hat{U} {D_\kappa}^{-1} \hat{U}' D_s^{-1} s_{xy},$$

where D_κ is the $p \times p$ diagonal matrix which vector of diagonal entries is $\hat{\lambda} + \kappa 1_p$. Finally,

$$L_{\mathrm{Ridge}}(\hat{Z}, \kappa) \equiv \hat{Z}' D_\kappa^{-1} \hat{\gamma}.$$

Therefore, up to a scaling factor, $L_{\mathrm{Ridge}}(\hat{Z}, \kappa)$ belongs to $\mathcal{L}$:

$$L_{\mathrm{Ridge}}(\hat{Z}, \kappa) \equiv h_\kappa' \xi(\hat{Z}),$$

where the first q coordinates of the weighting vector h_κ are

$$h_{i,\kappa} = \frac{\frac{1}{\hat{\lambda}_i+\kappa}}{\sqrt{\sum_{j=1}^q \frac{1}{(\hat{\lambda}_j+\kappa)^2} + \frac{p-q}{\kappa^2}}},\ i=1,\ldots,q,$$

and the last $p-q$ coordinates are all equal to:

$$h_{i,\kappa} = \frac{\frac{1}{\kappa}}{\sqrt{\sum_{j=1}^q \frac{1}{(\hat{\lambda}_j+\kappa)^2} + \frac{p-q}{\kappa^2}}},\ i=q+1,\ldots,p.$$

As a consequence, $\mathcal{L}_{\text{Ridge}} \subset \mathcal{L}$. Moreover, it is straightforwardly checked that $\lim_{\kappa\to+\infty} h_\kappa = (1/\sqrt{p})1_p$ and $\lim_{\kappa\to 0} h_\kappa = \hat{\lambda}^{\odot -1}/\sqrt{\hat{\lambda}^{\odot -1\prime}\hat{\lambda}^{\odot -1}}$.

1.3 Proof of Theorem 3

First, the following lemma is needed.

Lemma 1 *Let v be a p-vector, with $p \geq 1$. Let V be a $p \times p$ positive definite matrix. For all $m \geq 1$, $\mathcal{K}_m(V;v) = \mathcal{K}_m(V - vv';v)$.*

Proof: We start by showing using induction that $\mathcal{K}_m(V;v) \subseteq \mathcal{K}_m(V-vv';v)$. First, $Vv \in \text{span}\{v,(V-vv')v\}$, since $Vv = vv'v + (V-vv')v = (v'v)v + (V-vv')v$.

Let us now assume that the claim holds at rank m: there exist linear coefficients $a_0,\ a_1\ldots,\ a_m$ such that $V^m v = \sum_{i=0}^m a_i(V-vv')^i v$. Then,

$$\begin{aligned} V^{m+1}v &= ((V-vv')+vv')\sum_{i=0}^m a_i(V-vv')^i v, \\ &= \sum_{i=0}^m a_i(V-vv')^{i+1}v + v\sum_{i=0}^m a_i v'(V-vv')^i v, \\ &= \sum_{i=0}^m a_i(V-vv')^{i+1}v + (V-vv')^0 v\sum_{i=0}^m b_i, \end{aligned}$$

where $b_i = a_i v'(V-vv')^i v \in \mathbb{R}$. Consequently, the claim still holds at rank $m+1$.

We now show by induction that $\mathcal{K}_m(V;v) \supseteq \mathcal{K}_m(V-vv';v)$. First, $(V-vv')v \in \text{span}\{v,Vv\}$ since $(V-vv')v = Vv - vv'v = Vv - (v'v)v$.

Let us assume that the claim holds at rank m: there exist linear coefficients $a_0,\ a_1\ldots,\ a_m$ such that $(V-vv')^m v = \sum_{i=0}^m a_i V^i v$. Then,

$$\begin{aligned} (V-vv')^{m+1}v &= (V-vv')\sum_{i=0}^m a_i V^i v, \\ &= \sum_{i=0}^m a_i V^{i+1}v - v\sum_{i=0}^m a_i v'V^i v, \\ &= \sum_{i=0}^m a_i V^{i+1}v - V^0 v\sum_{i=0}^m b_i, \end{aligned}$$

where $b_i = a_i v' V^i v \in \mathbb{R}$. Consequently, the claim still holds at rank $m+1$. Finally, $\mathcal{K}_m(V; v) = \mathcal{K}_m(V - vv'; v)$.

We now give a proof for the main result. First, it is deduced from Lemma 1 that $\mathcal{K}_m(D_s^{-1} S_x D_s^{-1}; D_s^{-1} s_{xy}) = \mathcal{K}_m(\hat{C} = D_s^{-1} S D_s^{-1}; D_s^{-1} s_{xy})$. Therefore, there exist linear coefficients $b_1, \ldots, b_m$, such that $\hat{\beta}_{\text{PLS},m}$ can be expressed as follows:

$$\hat{\beta}_{\text{PLS},m} = \sum_{i=1}^{m} b_i \hat{C}^{i-1} D_s^{-1} s_{xy},$$

The corresponding PLS prediction score is deduced:

$$\begin{aligned} L_{\text{PLS}}(X; m) &\equiv (X - \bar{X})' D_s^{-1} \hat{\beta}_{\text{PLS},m}, \\ &\equiv (X - \bar{X})' D_s^{-1} \Big(\sum_{i=1}^{m} b_i \hat{C}^{i-1} \Big) D_s^{-1} s_{xy}. \end{aligned} \tag{2}$$

Introducing the eigendecomposition of $\hat{C}$ leads to:

$$\begin{aligned} L_{\text{PLS}}(X; m) &\equiv (X - \bar{X})' D_s^{-1} \hat{U} D_b \hat{U}' D_s^{-1} s_{xy}, \\ &\equiv \hat{Z}' D_b \hat{\gamma}, \end{aligned}$$

where D_b is the $p \times p$ diagonal matrix whose diagonal entries are the coordinates of the p-vector $h_b = (\sum_{i=1}^{m} b_i \hat{\lambda}_1^{i-1}, \ldots, \sum_{i=1}^{m} b_i \hat{\lambda}_m^{i-1}, 0, \ldots, 0)'$. Finally, up to a scaling factor, $L_{\text{PLS}}(X; m)$ belongs to $\mathcal{L}$:

$$L_{\text{PLS}}(\hat{Z}; m) \equiv h_b' \xi(\hat{Z}).$$

In the special case where $m = 1$, expression (2) simplifies:

$$\begin{aligned} L_{\text{PLS}}(X; m) &\equiv (X - \bar{X})' D_s^{-1} D_s^{-1} s_{xy}, \\ &\equiv L_{\text{N}}(\hat{Z}). \end{aligned}$$

1.4 Proof of Theorem 4

For all $j = 1, \ldots, p$, the jth coordinate $\hat{Z}_j \hat{\gamma}_j$ of the vector $\xi(\hat{Z})$, where $\odot$ stands for the term-by-term product of two vectors with equal dimension, has the following conditional expectation:

$$\begin{aligned} E[\hat{Z}_j \hat{\gamma}_j \mid Y, S, s_{xy}, \bar{X}] &= \hat{\gamma}_j E[\hat{Z}_j \mid Y, S, s_{xy}, \bar{X}], \\ &= \hat{\gamma}_j E[\hat{U}_j' D_s^{-1} (X - \bar{X}) \mid Y, S, s_{xy}, \bar{X}] \\ &= \hat{\gamma}_j \hat{U}_j' D_s^{-1} (\mu_x - \bar{X}) - \tfrac{Y - \mu_y}{\sigma_y^2} \hat{\gamma}_j \hat{U}_j' D_s^{-1} \sigma_{xy}. \end{aligned}$$

Hence,

$$E[\xi(\hat{Z}) \mid Y, S, s_{xy}, \bar{X}] = D_{\hat{\gamma}} \hat{U}' D_s^{-1} (\mu_x - \bar{X}) - \frac{Y - \mu_y}{\sigma_y^2} D_{\hat{\gamma}} \hat{U}' D_s^{-1} \sigma_{xy}, \tag{3}$$

where $D_{\hat{\gamma}}$ is the $p \times p$ diagonal matrix which diagonal entries are the coordinates of $\hat{\gamma}$. Then,

$$E[\xi(\hat{Z}) \mid S, s_{xy}, \bar{X}] = D_{\hat{\gamma}} \hat{U}' D_s^{-1} (\mu_x - \bar{X}). \tag{4}$$

Since $\bar{X}$ is independent of both S and s_{xy}, the conditioning can be reduced to S and s_{xy}:

$$E[\xi(\hat{Z}) \mid S, s_{xy}] = 0.$$

We finally get $E[\xi(\hat{Z})] = 0$.

Similarly, the vector $\xi(\hat{Z})$ has the following conditional variance:

$$\mathrm{Var}(\xi(\hat{Z}) \mid Y, S, s_{xy}, \bar{X}) = D_{\hat{\gamma}} \hat{U}' D_s^{-1} \Sigma D_s^{-1} \hat{U} D_{\hat{\gamma}}. \tag{5}$$

It is deduced from expressions (3) and (5) that:

$$\begin{aligned} \mathrm{Var}(\xi(\hat{Z}) \mid S, s_{xy}, \bar{X}) &= E[\mathrm{Var}(\xi(\hat{Z}) \mid Y, S, s_{xy}, \bar{X})] + \mathrm{Var}[E(\xi(\hat{Z}) \mid Y, S, s_{xy}, \bar{X})], \\ &= D_{\hat{\gamma}} \hat{U}' D_s^{-1} \Sigma_x D_s^{-1} \hat{U} D_{\hat{\gamma}}. \end{aligned} \tag{6}$$

Similarly, it is deduced from expressions (4) and (6) that:

$$\begin{aligned} \mathrm{Var}(\xi(\hat{Z}) \mid S, s_{xy}) &= \frac{n+1}{n} D_{\hat{\gamma}} \hat{U}' D_s^{-1} \Sigma_x D_s^{-1} \hat{U} D_{\hat{\gamma}}, \\ &= \frac{n+1}{n} \hat{U}' D_s^{-1} \Sigma_x D_s^{-1} \hat{U} \odot (\hat{\gamma}\hat{\gamma}'). \end{aligned} \tag{7}$$

Since $\mathrm{Var}(\xi(\hat{Z}) \mid S) = E[\mathrm{Var}(\xi(\hat{Z}) \mid S, s_{xy}) \mid S] + \mathrm{Var}(E[\xi(\hat{Z}) \mid S, s_{xy}] \mid S)$ and, as shown previously, $E[\xi(\hat{Z}) \mid S, s_{xy}] = 0$, then it can be deduced from expression (7) that:

$$\mathrm{Var}(\xi(\hat{Z}) \mid S) = \tfrac{n+1}{n} \hat{U}' D_s^{-1} \Sigma_x D_s^{-1} \hat{U} \odot E[\hat{\gamma}\hat{\gamma}' \mid S].$$

Now,

$$\begin{aligned} E[\hat{\gamma}\hat{\gamma}' \mid S] &= E[\hat{U}' D_s^{-1} s_{xy} s_{xy}' D_s^{-1} \hat{U} \mid S] \\ &= \hat{U}' D_s^{-1} E[s_{xy} s_{xy}' \mid S] D_s^{-1} \hat{U} \\ &= \hat{U}' D_s^{-1} E[s_{xy} s_{xy}'] D_s^{-1} \hat{U} \\ &= \hat{U}' D_s^{-1} (\mathrm{Var}(s_{xy}) + \sigma_{xy} \sigma_{xy}') D_s^{-1} \hat{U}. \end{aligned}$$

It is now recalled (Christensen, 2015) that $\mathrm{Var}(s_{xy}) = (\sigma_y^2 \Sigma_x + \sigma_{xy} \sigma_{xy}')/(n-1)$. Consequently:

$$E[\hat{\gamma}\hat{\gamma}' \mid S] = \tfrac{1}{n-1} \sigma_y^2 \hat{U}' D_s^{-1} \Sigma_x D_s^{-1} \hat{U} + \tfrac{n}{n-1} \hat{U}' D_s^{-1} \sigma_{xy} \sigma_{xy}' D_s^{-1} \hat{U}.$$

The final expression for $\mathrm{Var}(\xi(\hat{Z}))$ is obtained by noticing that, since $E[\xi(\hat{Z}) \mid S] = 0$:

$$\mathrm{Var}(\xi(\hat{Z})) = E[\mathrm{Var}(\xi(\hat{Z})) \mid S].$$

Finally, since $\mathrm{Cov}(\xi(\hat{Z}), Y \mid Y, S, s_{xy}, \bar{X}) = 0$, it is deduced from expression (4) that:

$$\begin{aligned} \mathrm{Cov}(\xi(\hat{Z}), Y \mid S, s_{xy}, \bar{X}) &= E[\mathrm{Cov}(\xi(\hat{Z}), Y \mid S, s_{xy}, \bar{X})] + \\ &\quad \mathrm{Cov}[E(\xi(\hat{Z}) \mid S, s_{xy}, \bar{X}), E(Y \mid S, s_{xy}, \bar{X})], \\ &= D_{\hat{\gamma}} \hat{U}' D_s^{-1} \sigma_{xy}. \end{aligned}$$

Consequently,

$$\mathrm{Cov}(\xi(\hat{Z}), Y \mid S, s_{xy}) = D_{\hat{\gamma}} \hat{U}' D_s^{-1} \sigma_{xy}.$$

Similarly, since $E[Y \mid S, s_{xy}] = \mu_y$, then $\mathrm{Cov}[E(\xi(\hat{Z}) \mid S, s_{xy}),\ E(Y \mid S, s_{xy})] = 0$ and:

$$\begin{aligned} \mathrm{Cov}(\xi(\hat{Z}), Y \mid S) &= E[D_{\hat{\gamma}} \big(\hat{U}' D_s^{-1} \sigma_{xy}\big) \mid S] \\ &= \big(\hat{U}' D_s^{-1} \sigma_{xy}\big)^{\odot 2}. \end{aligned}$$

The final expression for $\mathrm{Cov}(\xi(\hat{Z}), Y)$ is obtained by noticing that $E[\xi(\hat{Z}) \mid S] = 0$:

$$\mathrm{Cov}(\xi(\hat{Z}), Y) = E[\mathrm{Cov}(\xi(Z), Y \mid S)].$$

2 Simple comparative study in Witten and Tibshirani (2009)'s simulation setup

Witten and Tibshirani (2009) introduce a toy simulation setup to demonstrate the estimation accuracy of linear regression model parameters by the `scout` method. A similar setting is used hereafter to compare the prediction performance of five prediction scores within $\mathcal{L}$: the OLS, Naive, Ridge, PLS prediction scores and the proposed adaptive regression method. Additionally, two alternative methods are introduced in the comparative study: `scout(1,1)`, with ℓ_1-penalized estimation of both the partial correlation matrix and the regression coefficients, as in Witten and Tibshirani (2009) for the same simulation setting, and Principal Component Regression (PCR), identified in Section 3 of the main document as being out of $\mathcal{L}$. When needed, hyperparameters (Ridge, PLS, PCR, Scout and proposed method) are optimized using a 10-fold cross validation procedure. The PLS and PCR methods are implemented using the R package `pls` (Mevik et al., 2020), the Ridge method using the R package `glmnet` (Friedman et al., 2010) and the Scout method using the R package `scout` (Witten and Tibshirani, 2015). The proposed `Adaptive` method is implemented using the R package `AdaptiveRegression`, available at `https://github.com/fhebert`.

All simulated datasets have $n = 20$ observations on $p = 19$ normally distributed explanatory variables with mean 0 and standard deviations 1. The first 10 variables have correlation 0.9 with each other (in Witten and Tibshirani (2009) this equicorrelation parameter is 0.5); the rest are uncorrelated. The response variable Y is generated under the model $Y = X\beta + \varepsilon$, where:

- $\beta_j = j$ for $j \leq 10$ and $\beta_j = 0$ for $j > 10$ (scenario 1 as in Witten and Tibshirani (2009)),
- $\beta_j = 0$ for $j \leq 10$ and $\beta_j = p - j$ for $j > 10$ (scenario 2),

and where $\varepsilon \sim \mathcal{N}(0; 25)$. In addition, β is multiplied by a constant so that the asymptotic squared correlation between the response and the prediction score is 0.8 (arbitrarily chosen).

For each simulation scenario, 1,000 training datasets are randomly generated. For each training dataset, a test dataset with 1,000 individuals is generated following the same

Table 1: Simulation study results: mean squared correlations between the response observed on the test dataset and seven prediction scores under each scenario (values between parentheses are the corresponding standard deviations)

		Scenario 1	Scenario 2
Within $\mathcal{L}$	OLS	0.30 (0.17)	0.28 (0.17)
	Naive	0.79 (0.01)	0.23 (0.18)
	Ridge	0.73 (0.08)	0.55 (0.15)
	PLS	0.66 (0.30)	0.22 (0.27)
	Adaptive	0.76 (0.08)	0.52 (0.16)
Out of $\mathcal{L}$	Scout	0.76 (0.05)	0.54 (0.13)
	PCR	0.68 (0.27)	0.21 (0.25)

simulation scheme. Table 1 reproduces the mean squared correlations between the response observed on the test dataset and the seven prediction scores.

In the first scenario, the seven prediction methods can be grouped as follows, regarding their prediction performance: a first group composed of `Naive`, `Adaptive` and `Scout` showing the best prediction performance ($0.76 \leq R^2 \leq 0.79$), `Ridge` (with $R^2 = 0.73$), a group composed of `PCR` and `PLS` showing markedly lower prediction performance ($0.66 \leq R^2 \leq 0.68$) and `OLS` being clearly outperformed.

In the second scenario, the composition of the groups of methods regarding their prediction performance is changed: a group composed of `Ridge`, `Scout` and `Adaptive` showing the best prediction performance ($0.52 \leq R^2 \leq 0.55$) and a group composed of `OLS`, `Naive`, `PLS` and `PCR` with poor prediction performance ($0.21 \leq R^2 \leq 0.28$).

This simple example, with a limited simulation setup, demonstrates that, with the same dependence across explanatory variables, some prediction methods can show very different prediction performance, here `Naive`, `PLS`, `PCR`, depending on the pattern of association with the response variable. Others, including `Adaptive` and `Scout` remain among the best methods in the two scenarios.

The simulation study above is completed in Section 5 of the main document by more intensive simulations and a comparison using two public datasets.

3 Data-driven simulation study in Section 5.1

Detailed results of the data-driven simulation study introduced in Section 5.1 of the main paper are given below. Tables 2 to 5 give the average MSEP of the nine prediction methods over 1,000 generations of training and test datasets in each of the sixteen simulation scenario (four marginal distributions of explanatory variables and conditional distributions of response and four vectors of regression parameters).

Table 2: Average MSEP of each method for each distribution under scenario 1 over 1,000 simulations (with corresponding standard deviations in brackets). Values in bold indicate the best prediction performance.

		Normal	$\mathcal{T}_5$	χ^2_5	$\mathcal{F}_{10,10}$
Within $\mathcal{L}$	OLS	42.90	63.21	37.36	158.85
		[11.88]	[22.29]	[10.22]	[251.58]
	Naive	342.1	370.08	385.2	572.87
		[27.80]	[41.79]	[40.75]	[192.03]
	Ridge	142.16	176.48	188.79	329.81
		[12.12]	[17.92]	[20.91]	[80.33]
	PLS	14.18	32.41	22.09	68.97
		[1.89]	[9.81]	[4.72]	[45.77]
	Adaptive	**12.02**	**20.60**	**16.28**	**38.19**
		[2.62]	[7.67]	[4.24]	[20.38]
Out of $\mathcal{L}$	Lasso	28.21	33.99	27.47	58.46
		[5.32]	[10.32]	[7.68]	[42.57]
	PCR	15.21	33.87	23.88	70.79
		[2.82]	[10.12]	[5.15]	[37.58]
	SLM	30.36	62.19	57.81	175.3
		[3.80]	[12.70]	[9.87]	[77.34]
	Scout	29.76	50.17	45.79	98.30
		[3.73]	[8.97]	[8.18]	[39.59]

Table 3: Average MSEP of each method for each distribution under scenario 2 over 1,000 simulations (with corresponding standard deviations in brackets). Values in bold indicate the best prediction performance.

		Normal	$\mathcal{T}_5$	χ^2_5	$\mathcal{F}_{10,10}$
Within $\mathcal{L}$	OLS	45.33	63.27	36.80	102.53
		[13.64]	[24.90]	[9.62]	[51.67]
	Naive	111.81	137.58	132.12	199.59
		[5.98]	[12.70]	[7.29]	[23.15]
	Ridge	73.78	94.32	87.94	139.57
		[4.79]	[8.03]	[6.37]	[16.26]
	PLS	**9.96**	**18.27**	**14.40**	29.12
		[1.26]	[4.30]	[3.38]	[10.86]
	Adaptive	12.14	19.10	15.62	**27.27**
		[2.21]	[5.80]	[3.54]	[8.74]
Out of $\mathcal{L}$	Lasso	25.31	41.16	38.67	78.51
		[2.70]	[8.26]	[3.89]	[14.74]
	PCR	10.44	18.67	14.71	28.70
		[1.28]	[4.08]	[2.48]	[10.04]
	SLM	13.83	28.94	25.67	65.26
		[1.23]	[5.56]	[3.73]	[17.28]
	Scout	15.12	27.86	25.03	48.01
		[1.48]	[4.09]	[2.90]	[9.41]

Table 4: Average MSEP of each method for each distribution under scenario 3 over 1,000 simulations (with corresponding standard deviations in brackets). Values in bold indicate the best prediction performance.

		Normal	$\mathcal{T}_5$	χ^2_5	$\mathcal{F}_{10,10}$
Within $\mathcal{L}$	OLS	219.97	305.29	177.09	486.53
		[74.81]	[100.33]	[53.37]	[313.82]
	Naive	519.34	551.6	578.52	765.47
		[23.85]	[33.63]	[35.96]	[243.59]
	Ridge	362.49	397.27	405.23	529.62
		[19.66]	[29.29]	[29.21]	[118.93]
	PLS	40.28	56.45	48.69	72.42
		[4.35]	[19.67]	[8.66]	[21.28]
	Adaptive	33.11	37.16	35.48	38.97
		[6.5]	[9.28]	[6.43]	[10.14]
Out of $\mathcal{L}$	Lasso	**26.34**	**26.70**	**26.44**	**26.68**
		[1.07]	[1.51]	[1.37]	[2.26]
	PCR	41.85	57.11	51.43	72.79
		[5.15]	[15.33]	[11.09]	[21.48]
	SLM	49.01	77.11	69.89	153.51
		[6.04]	[24.38]	[16.64]	[74.05]
	Scout	34.57	34.94	34.67	37.67
		[4.16]	[4.46]	[5.44]	[12.20]

Table 5: Average MSEP of each method for each distribution under scenario 4 over 1,000 simulations (with corresponding standard deviations in brackets). Values in bold indicate the best prediction performance.

		Normal	$\mathcal{T}_5$	χ^2_5	$\mathcal{F}_{10,10}$
Within $\mathcal{L}$	OLS	121.89 [40.82]	216.65 [75.46]	126.39 [33.61]	124.95 [71.22]
	Naive	841.64 [37.88]	1080.69 [98.60]	1142.36 [51.27]	499.88 [85.45]
	Ridge	482.81 [27.93]	708.57 [56.82]	787.11 [44.30]	385.22 [68.95]
	PLS	**44.57** [8.62]	119.84 [35.92]	**76.06** [14.51]	**52.59** [19.95]
	Adaptive	47.54 [11.01]	**110.49** [31.35]	80.89 [16.92]	53.67 [16.67]
Out of $\mathcal{L}$	Lasso	87.14 [13.01]	189.41 [33.04]	149.99 [21.09]	84.85 [18.09]
	PCR	47.49 [9.29]	121.43 [30.00]	80.69 [16.28]	55.06 [14.96]
	SLM	86.97 [9.49]	223.57 [45.27]	190.66 [33.19]	137.13 [44.18]
	Scout	80.10 [8.31]	183.38 [36.19]	152.6 [18.32]	82.13 [16.02]

4 Additional simulation study with synthetic dependence patterns

A complementary simulation study is conducted in order to compare the prediction performance of the same nine methods as introduced in Section 5.1. of the main paper, in a wide scope of situations regarding the dimension of the regression parameter (n/p), the vector of regression coefficients and various synthetic patterns of dependence across explanatory variables. Six scenarios are considered, with a single response variable, normally distributed with mean $x'\beta$ and standard deviation 1. The p–profile x of explanatory variables is normally distributed with mean 0 and variance-covariance matrix Σ_x with all diagonal entries equal to one. For each scenario, the training sample size n and the nonzero entries in β and Σ_x are given below:

- **Scenario 1**: $n = 25$, $p = 500$ ($n/p = 0.05$)
  - For $j = 1, \ldots, 20$, $\beta_j = -j$ (sparsity rate: 0.96).
  - For $i, j = 1, \ldots, 250$, $i \neq j$, $\left[\Sigma_x\right]_{ij} = 0.9$. For $i, j = 251, \ldots, 400$, $i \neq j$, $\left[\Sigma_x\right]_{ij} = 0.5$.
- **Scenario 2**: $n = 25$, $p = 300$ ($n/p = 0.08$)
  - For $j = 131, \ldots, 170$, $\beta_j = 1$. (sparsity rate: 0.87)
  - Let B stand for the $p \times 5$ matrix whose nonzero coefficients are as follows: for $i = 1, \ldots, 100$, $B_{i1} = 1$, for $i = 51, \ldots, 150$, $B_{i2} = -1$, for $i = 101, \ldots, 200$, $B_{i3} = 1$, for $i = 151, \ldots, 250$, $B_{i4} = -1$, for $i = 201, \ldots, 300$, $B_{i5} = 1$. Σ_x is obtained by scaling rows and columns of $0.01 I_p + BB'$, so that the diagonal entries of Σ_x are all equal to one.
- **Scenario 3**: $n = 75$, $p = 100$ ($n/p = 0.75$)
  - For $j = 1, \ldots, 50$, $\beta_j = 1$, for $j = 51, \ldots, 100$, $\beta_j = -1$ (sparsity rate: 0)
  - For $i, j = 1, \ldots, 100$, $\left[\Sigma_x\right]_{ij} = 0.9^{|i-j|}$.
- **Scenario 4**: $n = 100$, $p = 200$ ($n/p = 0.50$)
  - For $j = 1, \ldots, 50$, $\beta_j = j - 1$, for $j = 51, \ldots, 100$, $\beta_j = 101 - j$ (sparsity rate: 0.50)
  - Let Σ_1 denote the following 5×5 matrix: for $i, j = 1, \ldots, 5$, $\left[\Sigma_1\right]_{ij} = 0.9^{|i-j|}$. Let Σ_2 denote the following 40×40 matrix: for $i, j = 1, \ldots, 40$, $i \neq j$, $\left[\Sigma_2\right]_{ij} = 0.5$ and for all $i = 1, \ldots, 40$, $\left[\Sigma_2\right]_{ii} = 1$. Then, $\Sigma_x = \Sigma_1 \otimes \Sigma_2$, where $\otimes$ stands for the Kronecker product of matrices.
- **Scenario 5**: $n = 25$, $p = 300$ ($n/p = 0.08$)
  - For $j = 121, \ldots, 180$, $\beta_j = j - 120$ (sparsity rate: 0.80)

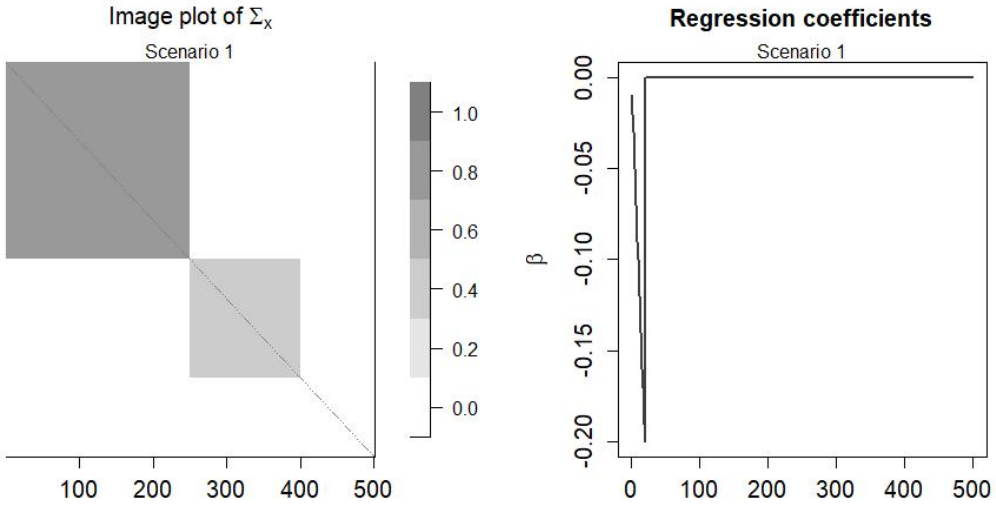

Figure 1: Simulation scenario 1: image plot of the variance-covariance matrix of the explanatory variables (left plot) and vector of regression coefficients (right plot).

- Let Σ_1 denote the following 15×15 matrix: for $i, j = 1, \ldots, 15$, $\left[\Sigma_1\right]_{ij} = 0.95^{|i-j|}$. Let Σ_2 denote the following 20×20 matrix: for $i, j = 1, \ldots, 20$, $i \neq j$, $\left[\Sigma_2\right]_{ij} = 0.9$ and for all $i = 1, \ldots, 20$, $\left[\Sigma_2\right]_{ii} = 1$. Then, $\Sigma_x = \Sigma_1 \otimes \Sigma_2$.

- **Scenario 6**: $n = 25$, $p = 400$ $(n/p = 0.06)$
  - For $j = 1, \ldots, 75$, $\beta_j = 1$ (sparsity rate: 0.81)
  - Let B stand for the $p \times 5$ matrix whose nonzero coefficients are as follows: for $i = 1, \ldots, 150$, $B_{i1} = 1$, for $i = 51, \ldots, 200$, $B_{i2} = -1$, for $i = 101, \ldots, 250$, $B_{i3} = 2$, for $i = 151, \ldots, 300$, $B_{i4} = -2$, for $i = 301, \ldots, 400$, $B_{i5} = 3$. Σ_x is obtained by scaling rows and columns of $0.1I_p + BB'$, so that the diagonal entries of Σ_x are all equal to one.

Furthermore, all vectors of regression coefficients are scaled so that the ratio $\beta'\Sigma_x\beta/\sigma_y^2$ between the variance of $\mathbb{E}(Y \mid X)$ and the variance of Y is 0.8.

Figures 1 to 6 display for each scenario an image plot of Σ_x and a plot of β.

For each scenario, a large dataset of 100,000 profiles of explanatory variables and corresponding response variable are generated. Training datasets are obtained by randomly choosing n profiles within this population and for each training dataset, a test dataset is obtained by randomly choosing 10,000 profiles in the rest of the population. Table 6 gives the average MSEP of all methods over 1,000 simulations.

Similarly as in the data-driven simulation study reported in Section 5.1 of the main paper, the proposed adaptive prediction method shows either the best prediction performance

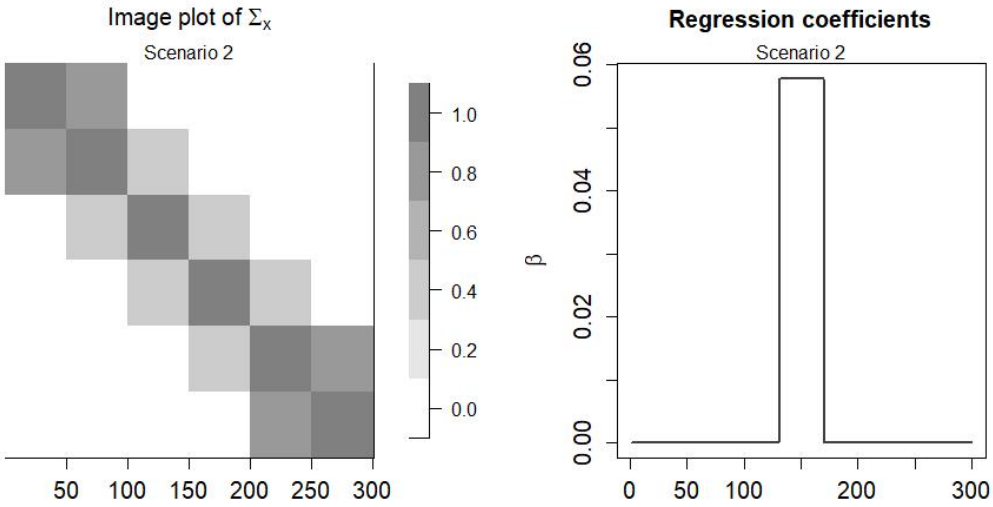

Figure 2: Simulation scenario 2: image plot of the variance-covariance matrix of the explanatory variables (left plot) and vector of regression coefficients (right plot).

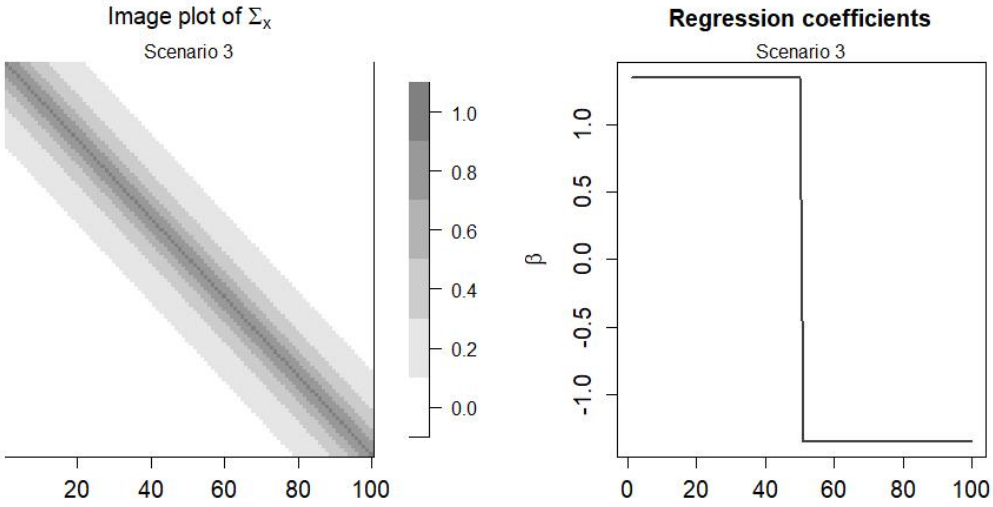

Figure 3: Simulation scenario 3: image plot of the variance-covariance matrix of the explanatory variables (left plot) and vector of regression coefficients (right plot).

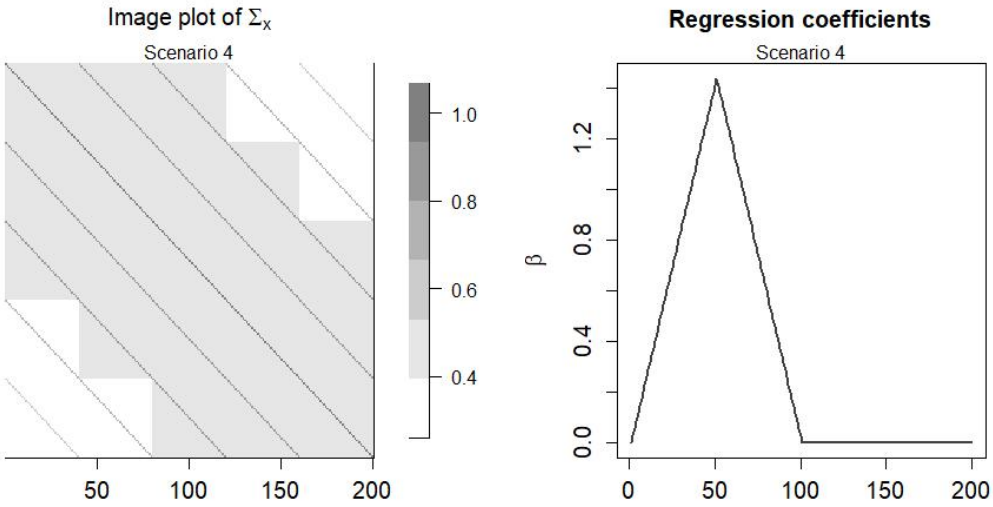

Figure 4: Simulation scenario 4: image plot of the variance-covariance matrix of the explanatory variables (left plot) and vector of regression coefficients (right plot).

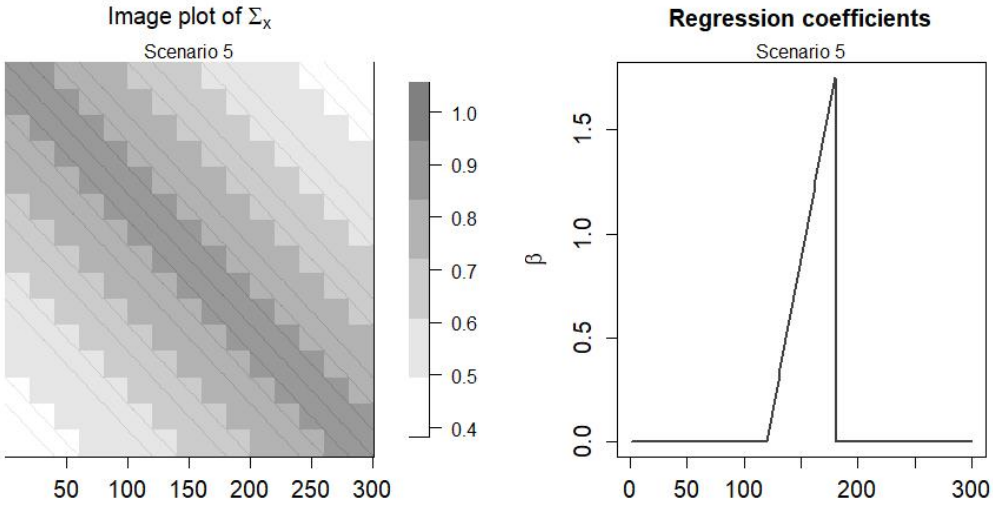

Figure 5: Simulation scenario 5: image plot of the variance-covariance matrix of the explanatory variables (left plot) and vector of regression coefficients (right plot).

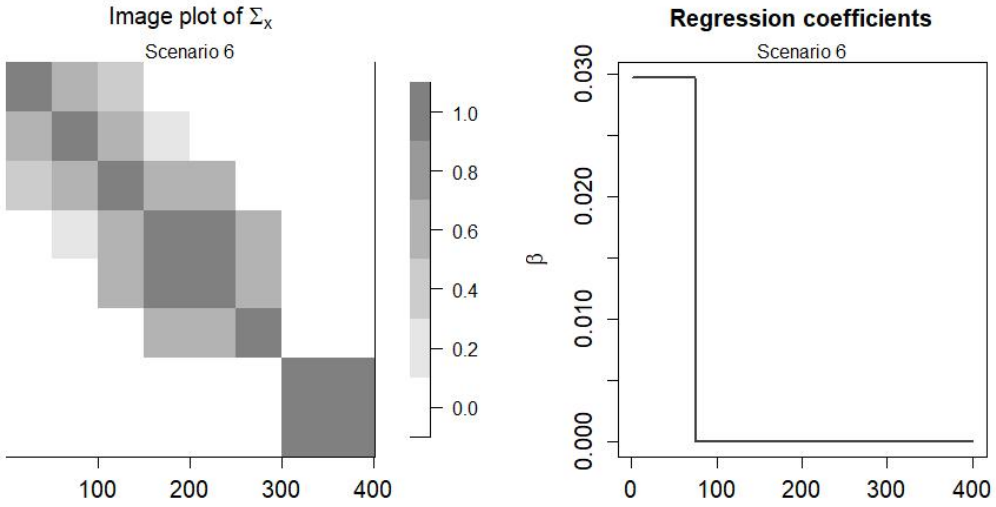

Figure 6: Simulation scenario 6: image plot of the variance-covariance matrix of the explanatory variables (left plot) and vector of regression coefficients (right plot).

over the nine methods or is close to the best performance, depending on the simulation scenario. In scenario 1, with a very sparse vector of regression coefficients and two large independent blocks of equicorrelated explanatory variables, the adaptive regression method and the naive prediction score outperform all the other methods. Surprisingly, OLS, Ridge and PCR show better prediction performance than the ℓ_1–penalized Scout and Lasso methods and even more clearly outperform the double-shrinkage estimation procedure SLM.

In scenario 2, where the vector of regression coefficient is also sparse, Scout, Lasso and Ridge have the best prediction performance (with $1.26 \leq \text{MSEP} \leq 1.29$). The OLS and the adaptive regression method also show good prediction performance (with $1.31 \leq \text{MSEP} \leq 1.32$), much better than the rank-reduced regression methods PLS and PCR (with $2.01 \leq \text{MSEP} \leq 2.03$).

In scenario 3, where the explanatory variables are strongly autocorrelated and the vector of regression coefficient is dense (only nonzero coefficients), the PLS, naive and adaptive regression methods outperform the other method. Among the regression methods based on penalized estimation, the Scout method does clearly better that Ridge and Lasso.

In scenarios 4, 5 and 6, with moderately sparse vectors of regression coefficients and dependent blocks of correlated explanatory variables, either obtained by a 5-factor model (scenario 6), or Kronecker products of within-block equi- and auto-correlation variance-covariance matrices (scenarios 4 and 5), the prediction performance of Ridge and the adaptive regression methods are the best. The doubly-penalized estimation method Scout also shows a good prediction performance in scenarios 4 whereas the OLS method also reaches the best prediction performance in scenario 6. The PLS and SLM methods are clearly outperformed in scenarios 5 and 6.

As observed in the data-driven simulation study presented in Section 5.1 of the main

Table 6: Average MSEP of each method for each scenarios over 1,000 simulations (values between brackets are the corresponding standard deviations). Values in bold indicate the best prediction performance.

		Simulation scenario					
		1	2	3	4	5	6
Within $\mathcal{L}$	OLS	1.19	1.31	3.28	2.12	1.54	**1.34**
		[0.13]	[0.21]	[0.73]	[0.25]	[0.23]	[0.22]
	Naive	**1.15**	1.69	**1.28**	1.32	1.46	2.25
		[0.11]	[0.44]	[0.15]	[0.04]	[0.13]	[0.51]
	Ridge	1.21	1.29	1.43	**1.16**	**1.33**	**1.34**
		[0.14]	[0.19]	[0.13]	[0.05]	[0.16]	[0.22]
	PLS	1.41	2.03	**1.27**	1.28	1.91	2.43
		[1.04]	[1.21]	[0.14]	[0.10]	[1.23]	[1.22]
	Adaptive	**1.15**	1.32	**1.28**	**1.16**	1.37	1.36
		[0.11]	[0.21]	[0.15]	[0.14]	[0.20]	[0.27]
Out of $\mathcal{L}$	Lasso	1.41	1.29	1.45	1.27	1.49	1.53
		[0.25]	[0.21]	[0.14]	[0.09]	[0.30]	[0.29]
	PCR	1.18	2.01	1.32	**1.15**	1.48	2.33
		[0.15]	[0.94]	[0.19]	[0.14]	[0.15]	[1.08]
	SLM	1.99	1.81	1.38	1.79	2.18	1.94
		[0.44]	[0.45]	[0.11]	[0.16]	[0.43]	[0.45]
	Scout	1.27	**1.26**	1.30	1.22	1.44	1.46
		[0.17]	[0.22]	[0.13]	[0.09]	[0.33]	[0.27]

paper, the relative performance of the prediction methods are highly variable depending on the patterns of regression coefficients and dependence across the explanatory variables. Indeed, even under assumption of a sparse regression model, penalized methods can show poor prediction performance. Also, for strong dependence patterns across explanatory variables, with block or factor structure, rank-reduced methods can also be outperformed. Over the scenarios considered in the present simulation study, the adaptive regression method turns out to show stable and among the best prediction performance, whereas all other methods are, at least in one scenario, clearly outperformed.

Bibliography

References

Christensen, R. (2015). Covariance of the Wishart distribution with applications to regression. http://www.stat.unm.edu/ fletcher/Wishart.pdf.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1–22.

Hager, W. W. (1989). Updating the inverse of a matrix. *SIAM review 31*(2), 221–239.

Mevik, B.-H., R. Wehrens, and K. H. Liland (2020). *pls: Partial Least Squares and Principal Component Regression*. R package version 2.7-3.

Witten, D. M. and R. Tibshirani (2009). Covariance-regularized regression and classification for high dimensional problems. *Journal of the Royal Statistical Society: Series B (Methodological) 71*(3), 615–636.

Witten, D. M. and R. Tibshirani (2015). *scout: Implements the Scout Method for Covariance-Regularized Regression*. R package version 1.0.4.