Outline: (Linear) Models in Ultra-Large Feature Spaces
Feature Screening

(F. Chiaromonte)

- A key reference: Fan J., Lv J. (2008) Sure Independence Screening for ultrahigh dimensional feature spaces. JRSS-B, 70(5) 849-911.
- Additional references: throughout the slides.

Stochastic mechanism:

$Y, \varepsilon\ \ r.vbls\ in\ R^1\ \ X\ r.vct\ in\ R^p$

$E(X) = 0 \quad Cov(X) = \Sigma_X$

$\varepsilon\ indep\ X \quad \varepsilon \sim N(0, \sigma^2)$

Independent sampling:

$Y_{nx1} = X_{nxp}\beta_{px1} + \varepsilon_{nx1}$

Estimation (fitting)

$\hat{\beta}_{LS} = (X'X)^{-1}X'Y$

- Take the marginal (linear) associations between Y and each of the X's; X'Y
- "Re-map" it based on the (linear) associations among the X's; X'X ~ S_x .

How would you rank the X's and chose the d << p most likely to have an effect on Y?

On the basis of the effects as estimated by LS; rank the coordinates of the LS vector and pick the largest d.

Does this work?

Yes, if the re-mapping is effective; the ranking comprises two ingredients, marginal associations with Y and re-mapping based on associations among the X's. No problem at all if "no re-mapping" i.e. S_x proportional to I_p .

What can go wrong?

λ_{max} (S_x) is large and λ_{min} (S_x) is small…

- Linear “concentration” of the data cloud in feature space, collinearity in the sample. Because Σ_x contains collinearity, and/or because n is not large enough wrt p.
- An ineffective re-mapping inflates the sampling variability in the LS estimates of the feature effects, possibly makes them poorly determined or non-unique on any given sample.

What happens as p grows wrt n; p~n, p>n, p>>n?

- Σ_x is likely to comprise more collinearity; we are considering more and more features which may carry linear associations, and
- S_x becomes progressively more collinear than its population counterpart, just because we have insufficient replication

What do we do?

$$\hat{\beta}_{MP} = (X'X)^{+}X'Y$$

More-Penrose generalized inverse

$$\hat{\beta}_{Ridge}(\lambda) = (X'X + \lambda I)^{-1}X'Y$$

Regularize with a size constraint, L2

$$\hat{\beta}_{LASSO}(\lambda) = argmin\,\{\left|\left|Y - X\beta\right|\right|^{2} + \lambda\, L1(\beta)\}$$

Regularize and sparsify with a size constraint, L1

… and more sophisticated approaches; SCAD, Adaptive LASSO, Danzig Selector. Also old fashioned Partial Least Squares.

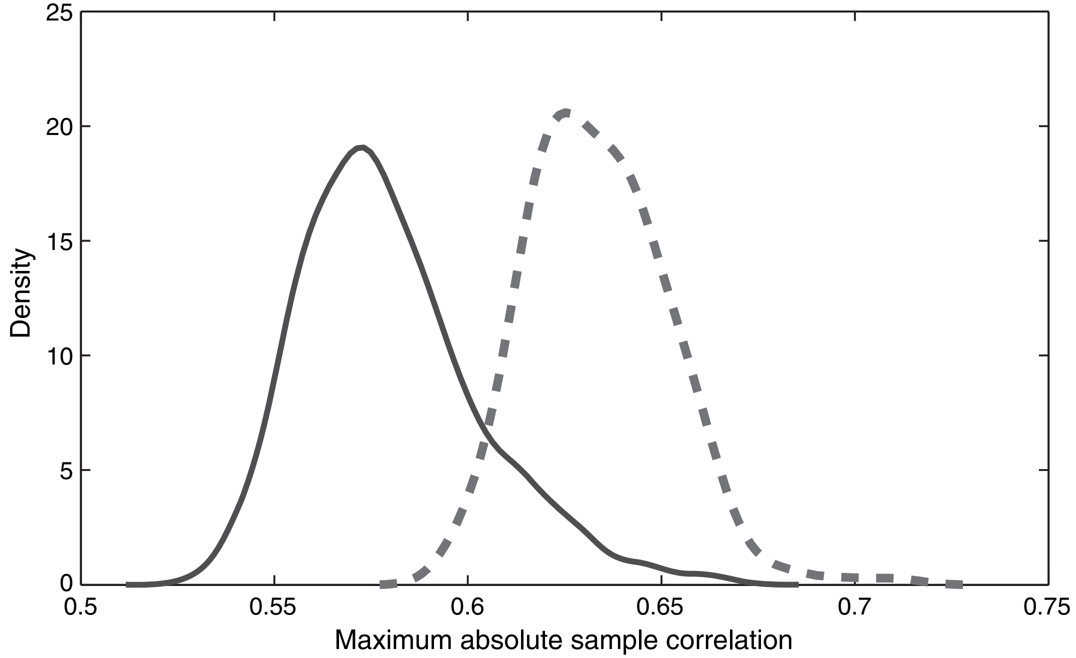


Fig. 1. Distributions of the maximum absolute sample correlation coefficient when $n=60$ and $p=1000$ (———) and $n=60$ and $p=5000$ (- - - - -), based on 500 simulations

X data simulated with Σ_x proportional to I_p.

No correlations at the population level, yet if p >> n the correlations in S_x are large!

What if we ignored the re-mapping all together?

Base the ranking and choice of d features on the component-wise regression, i.e., on the *marginal correlations*

$$\begin{tikzcd}
\hat{\beta}_{LS} = (X'X)^{-1}X'Y & & \hat{\beta}_{Ridge}(\lambda) = (X'X + \lambda I)^{-1}X'Y \\
& \omega = X'Y \propto r_{yx} \arrow[ul, leftrightarrow, "\hat{\beta}_{LS} = \omega \ \ if\ X'X \propto I_p"] \arrow[ur, leftrightarrow, "\hat{\beta}_{Ridge}(\lambda) \xrightarrow{\lambda\to\infty} 0 \quad \lambda\hat{\beta}_{Ridge}(\lambda) \xrightarrow{\lambda\to\infty} \omega"'] &
\end{tikzcd}$$

as the penalty increases the Ridge vector shrinks to 0, but if we multiply it by λ (which does not change the ranking of the components) … it converges to ω!!

CORRELATION LEARNING

More generally, approaches to learn about importance of features ***marginally***; these can be applied with small computational burden and without the curse of dimensionality even when p >> n

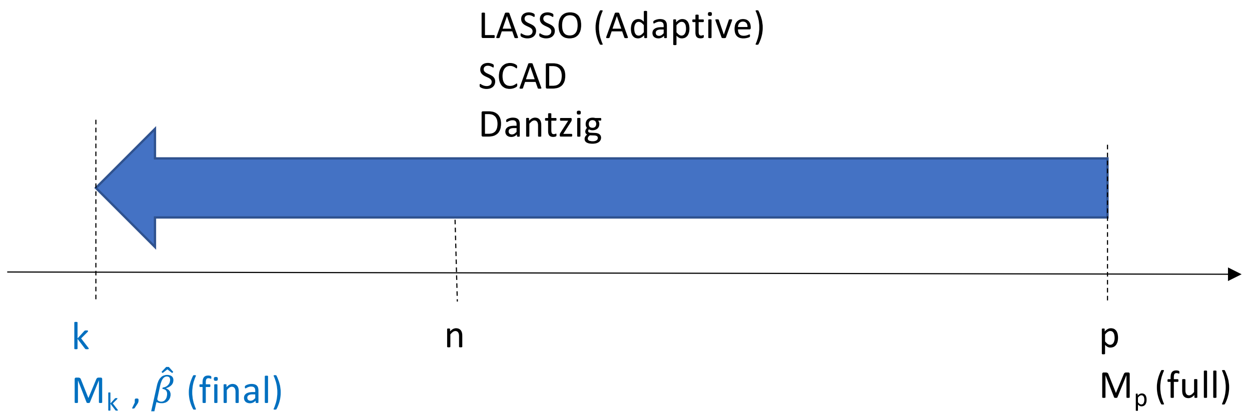
IMPORTANT REMARK: A NEW ASYMPTOTICS

In addition to problems with $p >> n$, starting with the new century we have problems where $p = p(n)$. As n grows, p grows with it, possibly exponentially $\log(p) = O(n^a)$ though perhaps with small a.

The more we sample, observing more units, the more the dimension of our feature space increases. Think of variants (SNPs) as we sequence genomes of more individuals, or recorded product choices/scores as we branch out over a network of individuals on social media. The ”universe” of the observed SNPs (of products) increases as we include more people.

In Statistics this required the formulation of a **new type of asymptotics**; study the performance of statistical procedures as n grows, and p(n) grows with it – possibly not too fast.

The proposal of **Fan and Lv (2008)** : If we believe the underlying true model is sparse, instead of



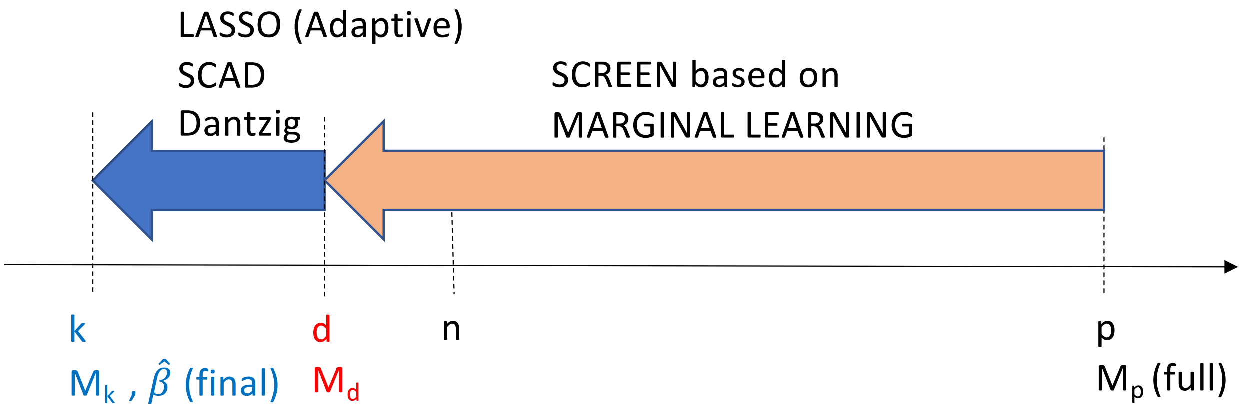
LASSO (Adaptive)
SCAD
Dantzig

k
M_k , $\hat{\beta}$ (final)

n

p
M_p (full)

use a reasonable d < n, e.g., d = n-1, or d = n/log(n) and proceed in two stages



LASSO (Adaptive)
SCAD
Dantzig

SCREEN based on
MARGINAL LEARNING

k
M_k , $\hat{\beta}$ (final)

d
M_d

n

p
M_p (full)

Screening disregards associations among the X's, but when p >> n our ability to account for them and re-map is so poor that it may hurt more than it helps.

To get to the true sparse model M_* with s features, and to estimate its true β, the two-stage strategy (which uses only marginal information when the dimension is still ultra-high) is more effective!!

Not a new idea, practitioners have used it forever, but now formalized. And its performance established by simulations and "new asymptotics" theoretical results.

SURE INDEPENDENCE SCREENING (SIS)

- SURE SCREENING PROPERTY: with a reasonable d = n-1 or n/log(n) and some assumptions on the nature of the stochastic mechanism generating the data and the speed at which p(n)>>n grows with n, when n is large $Pr(M_* \textit{ is contained in } M_d)$ is "overwhelming" (growing to 1).
- INDEPENDENCE: refers to the fact that we operate on marginal information, one X at a time.

The SCREENING algorithm is trivial and computationally inexpensive:

- Compute the marginal correlations $\omega = X'Y \propto r_{yx}$
- Rank the entries in ω and pick the first d to form M_d .

Next, apply feature selection algorithm XXX to the regression of Y on the features contained in M_d to obtain M_k , $\hat{\beta}$ (final).

First, use **simulations** to assess whether, on finite (but large) samples

- the sure screening property holds for SIS, and
- M_k is close to M_* (k close to s) and $\hat{\beta}$ close to the true β for SIS-XXX

In particular

- do we do better than with one pass of XXX? and
- is performance affected by the presence of collinearity in Σ_x (population level)?

SIMULATIONS WITH UNCORRELATED X'S

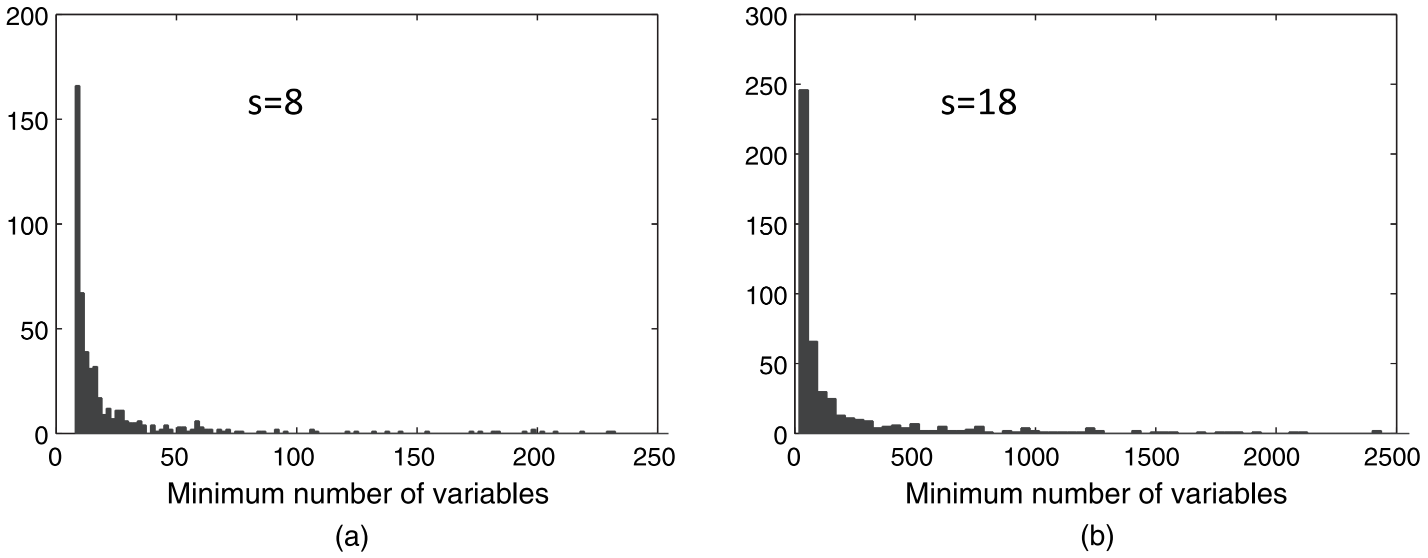


Fig. 5. Distribution of the minimum number of selected variables that is required to include the true model by using SIS when (a) $n = 200$ and $p = 1000$ and (b) $n = 800$ and $p = 20000$ in simulation I

OK!

Table 1. Results of simulation I: medians of the selected model sizes and estimation errors (in parentheses)

k $\|\widehat{\beta} - \beta\|^2$

one pass

p	*Results for the following methods:*					
	Dantzig selector	*Lasso*	*SIS–SCAD*	*SIS–DS*	*SIS–DS–SCAD*	*SIS–DS–AdaLasso*
1000 s=8	10^3 (1.381)	62.5 (0.895)	15 (0.374)	37 (0.795)	27 (0.614)	34 (1.269)
20000 s=18	— —	— —	37 (0.288)	119 (0.732)	60.5 (0.372)	99 (1.014)

One pass does very poorly at imposing sparsity and at estimating effects. And for p=20,000 could NOT be run in reasonable time (in 2008).

SIMULATIONS WITH CORRELATED X'S

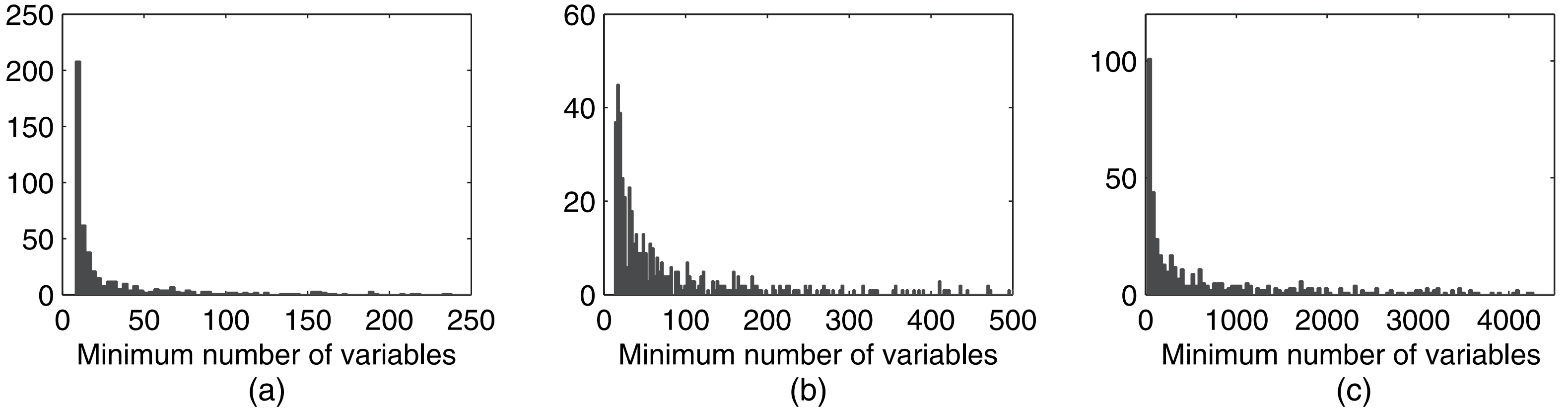
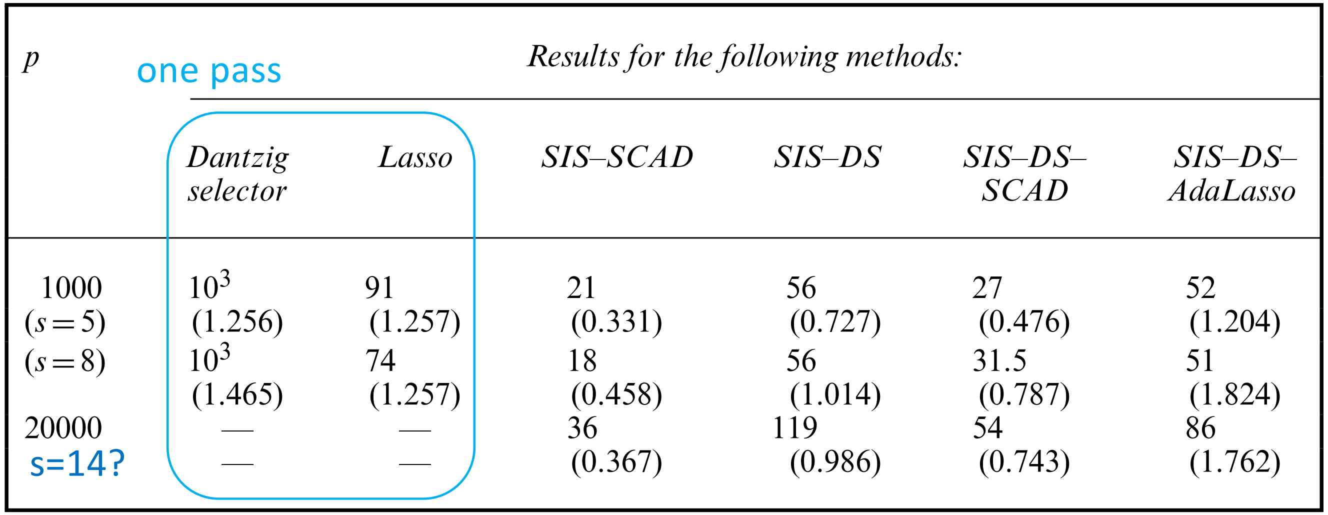


Fig. 6. Distribution of the minimum number of selected variables that is required to include the true model by using SIS when (a) $n = 200$, $p = 1000$ and $s = 5$, (b) $n = 200$, $p = 1000$ and $s = 8$ and (c) $n = 800$, $p = 20000$ and $s = 8$ in simulation II s=14?

Table 2. Results of simulation II: medians of the selected model sizes and estimation errors (in parentheses)

k $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2$



p	one pass		*Results for the following methods:*			
	Dantzig selector	*Lasso*	*SIS–SCAD*	*SIS–DS*	*SIS–DS–SCAD*	*SIS–DS–AdaLasso*
1000 ($s = 5$)	10^3 (1.256)	91 (1.257)	21 (0.331)	56 (0.727)	27 (0.476)	52 (1.204)
($s = 8$)	10^3 (1.465)	74 (1.257)	18 (0.458)	56 (1.014)	31.5 (0.787)	51 (1.824)
20000 s=14?	— —	— —	36 (0.367)	119 (0.986)	54 (0.743)	86 (1.762)

OK! also with correlated X's

One pass does very poorly at imposing sparsity and at estimating effects. And for p=20,000 could NOT be run in reasonable time (in 2008).

Theoretical Assessments

SURE SCREENING PROPERTY OF SIS:

- Let M_d = SIS(M_p) with d = n/log(n)
- *Impose assumptions on the nature of the stochastic mechanism generating the data, and on the speed at which p(n)>>n grows with n*
- Prove

$$\Pr(M_* \subseteq M_d) \geq 1 - (fast\ vanishing\ as\ n \to \infty)$$

Also, again under appropriate assumption, prove

- CONSISTENCY FOR SIS-DANTZIG $\hat{\beta} \xrightarrow[n\to\infty]{} \beta$
- ORACLE PROPERTY FOR SIS-SCAD (i) $\hat{\beta}_j = 0\ for\ any\ j \notin M_*$

  (ii) $\widehat{\beta}$ does as well for β as if we had run standard LS **knowing** the true M_*

Assumptions

Stochastic mechanism:

$$Y, \varepsilon\ \ r.vbls\ in\ R^1\ \ X, Z = \Sigma_X^{-1/2} X\ r.vct\ in\ R^p$$

$$E(Z) = 0 \quad Cov(Z) \propto I_p$$

$$\varepsilon\ indep\ Z \quad \varepsilon \sim N(0, \sigma^2)$$

Independent sampling:

$$Y_{nx1} = X_{nxp}\beta_{px1} + \varepsilon_{nx1} = Z_{nxp}\Sigma_X^{1/2}\beta_{px1} + \varepsilon_{nx1}$$

(I) Var(Y) does not grow, and the effects for the relevant features are strong enough, as n grows; for some κ ≥ 0

$$\min_{j \in M_*} |\beta_j| \ge \frac{pos\ const}{n^{\kappa}}$$

(II) The relevant features have a strong enough “trace” in their marginal linear associations with Y

$$\min_{j \in M_*} \left| cov\left(\frac{Y}{\beta_J}, X_j\right) \right| \ge pos\ const$$

p > n grows exponentially, but slow enough relative to the strength of the signals; for some $0 < \alpha <$ (1-2κ)

$$\log(p) = O(n^{\alpha})$$

(III) The feature collinearity (population level) remains weak enough as p(n) grows ; for some τ ≥ 0

$$\lambda_{max(\Sigma_X)} =\le pos\ const \times n^{\tau}$$

Z is spherically symmetric and, for any submatrix with more than n columns $\tilde{Z}_{nxh}, n < h \le p,\ \tilde{Z}_{nxh}\tilde{Z}_{nxh}{}'$ has all n eigenvalues of the same order – not too dissimilar (adds to the required symmetry for the features). Nothing much beyond Σ_x. This certainly holds if X is Gaussian (Z standard Gaussian).

SIS-XXX can perform poorly though (even for fairly large n) if:

- The effects of some relevant features are weaker than the spurious marginal traces carried by non-relevant features correlated with strong relevant ones; failure in (I), (III).
- Some relevant features have weak marginal linear traces – they may act only jointly with other features, or act with little linear trend; failure in (II).

Performance can be rescued through another very old and effective idea: iterate the procedure, working with residuals as to magnify weaker signals: **ITERATIVE SIS**, ISIS-XXX. The algorithm:

- SIS for Y on M_p => $M^{(1)}_{dIT}$
- XXX for Y on $M^{(1)}_{dIT}$ => $(M^{(1)}_{k(1)}, \hat{\beta}_{(1)})$
- $A^{(1)} = M^{(1)}_{k(1)}$
- $e^{(1)} = Y - X_{A(1)} \hat{\beta}_{(1)}$

Iteration 1

- SIS for $e^{(1)}$ on $M_p \setminus A^{(1)}$ => $M^{(2)}_{dIT}$
- XXX for $e^{(1)}$ on $M^{(2)}_{dIT}$ => $(M^{(2)}_{k(2)}, \hat{\beta}_{(2)})$
- $A^{(2)} = A^{(1)}$ U $M^{(1)}_{k(1)}$
- $e^{(2)} = Y - X_{A(2)} \hat{\beta}_{(2)}$

Iteration 2

… etc. (usually just a few times). Stop iterating when #($A^{(h)}$) reaches n-1 or n/log(n)

XXX for Y on $A^{(h)}$ => $(M_k, \hat{\beta}$) the *final result*.

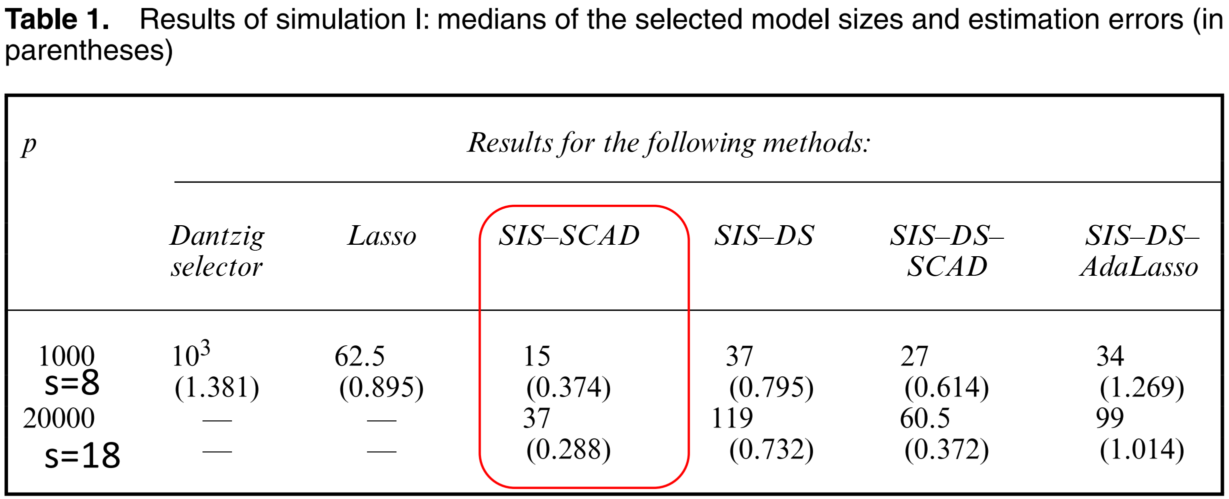


Table 1. Results of simulation I: medians of the selected model sizes and estimation errors (in parentheses)

p	*Results for the following methods:*					
	Dantzig selector	*Lasso*	*SIS–SCAD*	*SIS–DS*	*SIS–DS–SCAD*	*SIS–DS–AdaLasso*
1000 s=8	10^3 (1.381)	62.5 (0.895)	15 (0.374)	37 (0.795)	27 (0.614)	34 (1.269)
20000 s=18	— —	— —	37 (0.288)	119 (0.732)	60.5 (0.372)	99 (1.014)

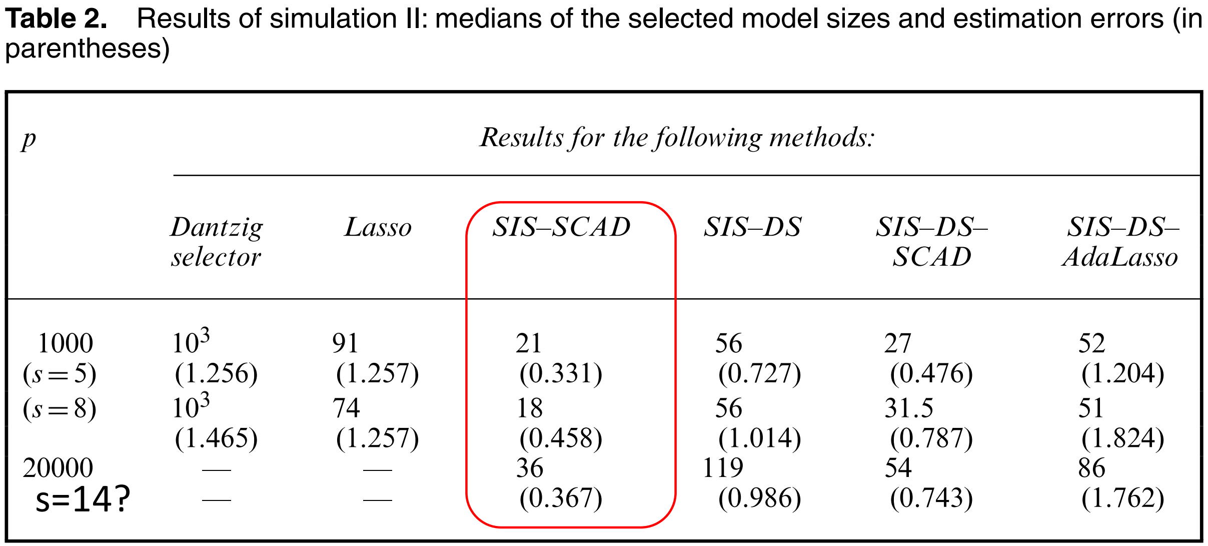


Table 2. Results of simulation II: medians of the selected model sizes and estimation errors (in parentheses)

p	*Results for the following methods:*					
	Dantzig selector	*Lasso*	*SIS–SCAD*	*SIS–DS*	*SIS–DS–SCAD*	*SIS–DS–AdaLasso*
1000 $(s=5)$	10^3 (1.256)	91 (1.257)	21 (0.331)	56 (0.727)	27 (0.476)	52 (1.204)
$(s=8)$	10^3 (1.465)	74 (1.257)	18 (0.458)	56 (1.014)	31.5 (0.787)	51 (1.824)
20000 s=14?	— —	— —	36 (0.367)	119 (0.986)	54 (0.743)	86 (1.762)

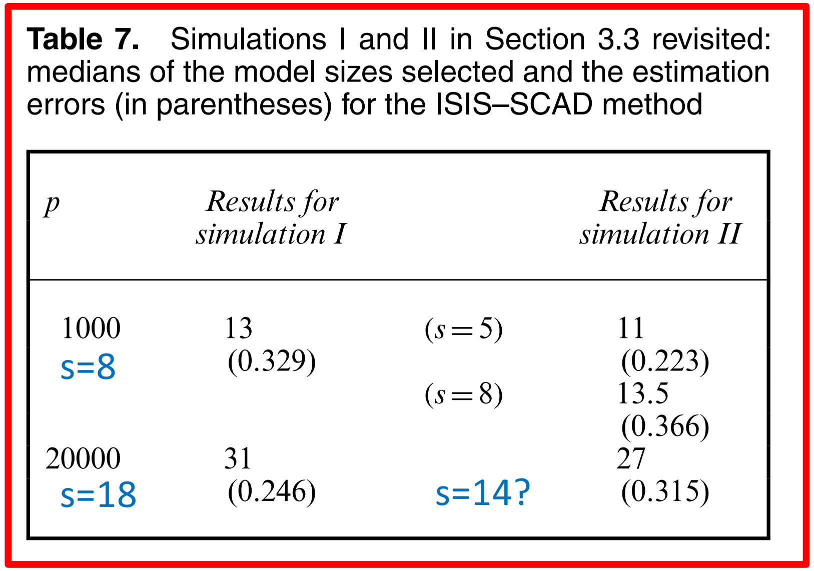


Table 7. Simulations I and II in Section 3.3 revisited: medians of the model sizes selected and the estimation errors (in parentheses) for the ISIS–SCAD method

p	*Results for simulation I*		*Results for simulation II*
1000 s=8	13 (0.329)	$(s=5)$	11 (0.223)
		$(s=8)$	13.5 (0.366)
20000 s=18	31 (0.246)	s=14?	27 (0.315)

Iterations work!

In the article more simulation results to demonstrate effectiveness against more specific failure scenarios

After the seminal paper by Fan and Lv (2008), there has been a deluge of literature on screening algorithms for ultra-high dimensional supervised problems.

- **Extension to GLMs**: Fan J., Samworth R., Wu Y. (2009) Ultrahigh dimensional feature selection: beyond the linear model. Journal of Machine Learning Research, 10 2013-2038.

- **Extension to model-free settings**: Zhu L., Li L., Li R., Zhu L. (2011). Model-free feature screening for ultra-high dimensional data. JASA 106(496) 1464-1475.

… and many others. Still a very active area of research.

What changes is the implementation of marginal learning; **ω is defined differently** (based on a GLM framework, or non-parametrically). Also, theoretical results can be different and require alternative proof strategies.

Remark: a screening algorithm must be ***conservative, the focus is on controling false negatives***. We want to make sure we do not leave out any relevant feature; more features can be eliminated as needed by feature selection after screening. If we consider SIS-XXX as a whole, it makes sense to also assess performance in terms of false positives (we do not want to finish with a k much larger than s; the size of the true sparse model) and in terms of estimation quality (how close $\hat{\beta}$ comes to the true β).

A package published in 2018 that implements various versions of sure independence screening:

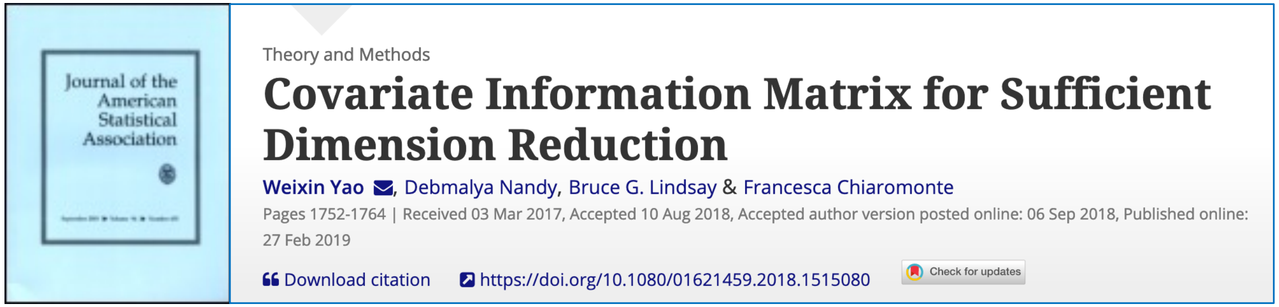
D.F. Saldana, Y. Feng (2018) SIS: An R Package for Sure Independence Screening in Ultrahigh-Dimensional Statistical Models. *Journal of Statistical Software*, *83*(2), 1–25. https://doi.org/10.18637/jss.v083.i02

Abstract

We revisit sure independence screening procedures for variable selection in generalized linear models and the Cox proportional hazards model. Through the publicly available R package SIS, we provide a unified environment to carry out variable selection using iterative sure independence screening (ISIS) and all of its variants. For the regularization steps in the ISIS recruiting process, available penalties include the LASSO, SCAD, and MCP while the implemented variants for the screening steps are sample splitting, data-driven thresholding, and combinations thereof. Performance of these feature selection techniques is investigated by means of real and simulated data sets, where we find considerable improvements in terms of model selection and computational time between our algorithms and traditional penalized pseudo-likelihood methods applied directly to the full set of covariates.

Some additions: covariate information matrices and numbers for supervised dimension reduction and feature screening

SUFFICIENT DIMENSION REDUCTION BASED ON INFORMATION MATRICES



Theory and Methods

Covariate Information Matrix for Sufficient Dimension Reduction

Weixin Yao, Debmalya Nandy, Bruce G. Lindsay & Francesca Chiaromonte

Pages 1752-1764 | Received 03 Mar 2017, Accepted 10 Aug 2018, Accepted author version posted online: 06 Sep 2018, Published online: 27 Feb 2019

Download citation   https://doi.org/10.1080/01621459.2018.1515080   Check for updates

Covariate Information Matrix (CIM): combine “local” non-parametric assessment of densities and “global” Eigen-decomposition, in an ***Information Matrix framework***.

… prior applications to:

- Projection Pursuit
- Spherical Symmetry, Multivariate Structures
- Independent Components Analysis
- Graphical Models

Hui G., Lindsay B. (2010). Projection pursuit via white noise matrices . Sankhya B, 72(2): 123–153 .
Lindsay B., Yao W. (2012). Fisher information matrix: A tool for dimension reduction, projection pursuit, independent component analysis, and more. Canadian Journal of Statistics, 40(4): 712–730.

What is the theory behind this?

$$U_{\mathbf{x}}(y) = \nabla_{\mathbf{x}} \log f(y \mid \mathbf{x})$$

score vector

$$\mathbb{F}_{\mathbf{x}} = \int U_{\mathbf{x}}(y) U_{\mathbf{x}}(y)^T f(y \mid \mathbf{x}) dy$$

Fisher Information Matrix for "parameter" $\boldsymbol{x}$
How $\boldsymbol{Y} \mid \boldsymbol{X}{=}\boldsymbol{x}$ changes with $\boldsymbol{x}$

$$\mathbb{C}_{\mathbf{X}} = \int \mathbb{F}_{\mathbf{x}} f(\mathbf{x}) d\mathbf{x}.$$

average on X … ***Covariate Information Matrix***

<u>Key result</u>: under very general conditions one has $\boldsymbol{Span}[\mathbb{C}_{\boldsymbol{X}}] \equiv \boldsymbol{\Sigma_X S_{Y|X}}$ (the Central Subspace of SDR)

$$U_f(\mathbf{x}) = \nabla_{\mathbf{x}} \log f(\mathbf{x})$$

$$\mathbb{J}_{\mathbf{X}} = \int U_f(\mathbf{x}) U_f(\mathbf{x})^T f(\mathbf{x}) d\mathbf{x}$$

Density Information Matrix for $\boldsymbol{X}$
characterization of the $\boldsymbol{X}$ distribution

$$U_{f^{(y)}}(\mathbf{x}) = \nabla_{\mathbf{x}} \log f^{(y)}(\mathbf{x})$$

$$\mathbb{J}_{\mathbf{X}|Y=y} = \int U_{f^{(y)}}(\mathbf{x}) U_{f^{(y)}}(\mathbf{x})^T f^{(y)}(\mathbf{x}) d\mathbf{x},$$

Density Information Matrix for $\boldsymbol{X} \mid \boldsymbol{Y}{=}\boldsymbol{y}$

$$\mathbb{J}_{\mathbf{X}|Y} = \int \mathbb{J}_{\mathbf{X}|Y=y} f(y) dy.$$

average on Y … ***Inverse Regression***

<u>Key result</u>: $\boxed{\mathbb{C}_{\boldsymbol{X}} = \mathbb{J}_{\boldsymbol{X}|Y} - \mathbb{J}_{\boldsymbol{X}}}$

rewrite CIM in terms of X-related DIMs
Inverse Regression, "adjusted" for $\boldsymbol{X}$

What is the implementation?

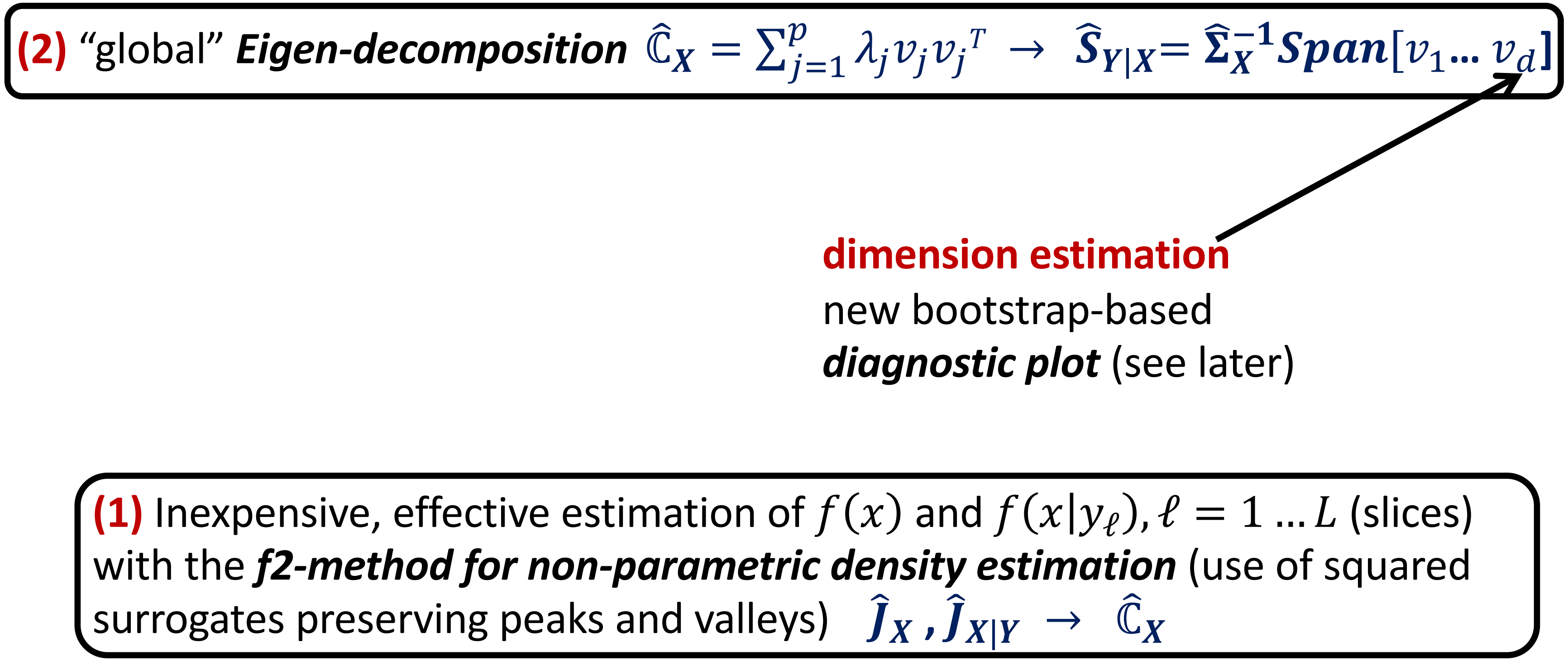
(2) "global" ***Eigen-decomposition*** $\hat{\mathbb{C}}_{\boldsymbol{X}} = \sum_{j=1}^{p} \lambda_j v_j {v_j}^T \rightarrow \hat{\boldsymbol{S}}_{\boldsymbol{Y}|\boldsymbol{X}} = \hat{\boldsymbol{\Sigma}}_{\boldsymbol{X}}^{-1} \boldsymbol{Span}[v_1 \ldots v_d]$

Key result: without (L) and (C), one has $\boldsymbol{Span}[\mathbb{C}_{\boldsymbol{X}}] \equiv \boldsymbol{\Sigma}_{\boldsymbol{X}} \boldsymbol{S}_{\boldsymbol{Y}|\boldsymbol{X}}$

(1) Inexpensive, effective estimation of $f(x)$ and $f(x|y_\ell), \ell = 1 \ldots L$ (slices) with the ***f2-method for non-parametric density estimation*** (use of squared surrogates preserving peaks and valleys) $\hat{\boldsymbol{J}}_{\boldsymbol{X}}, \hat{\boldsymbol{J}}_{\boldsymbol{X}|\boldsymbol{Y}} \rightarrow \hat{\mathbb{C}}_{\boldsymbol{X}}$

Key result: $\boxed{\mathbb{C}_{\boldsymbol{X}} = \mathbb{J}_{\boldsymbol{X}|Y} - \mathbb{J}_{\boldsymbol{X}}}$

What is the implementation?



(2) “global” ***Eigen-decomposition*** $\widehat{\mathbb{C}}_{\boldsymbol{X}} = \sum_{j=1}^{p} \lambda_j v_j v_j{}^{T} \rightarrow \boldsymbol{\widehat{S}_{Y|X}} = \boldsymbol{\widehat{\Sigma}_X^{-1} Span}[v_1 \dots v_d]$

dimension estimation

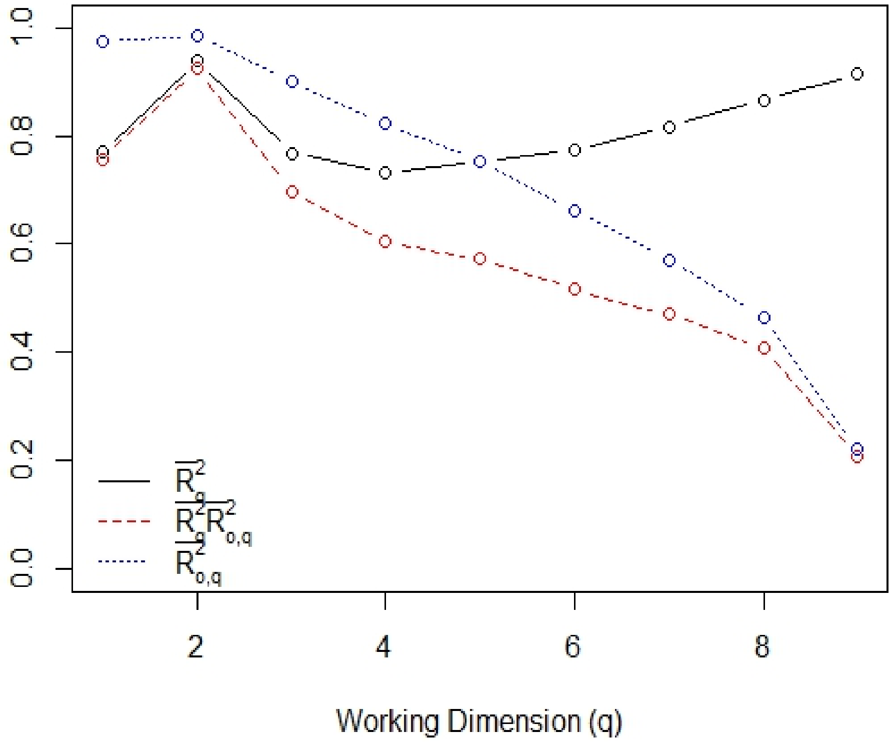
new bootstrap-based

diagnostic plot (see later)

(1) Inexpensive, effective estimation of $f(x)$ and $f(x|y_\ell), \ell = 1 \dots L$ (slices) with the ***f2-method for non-parametric density estimation*** (use of squared surrogates preserving peaks and valleys) $\boldsymbol{\hat{J}_X}, \boldsymbol{\hat{J}_{X|Y}} \rightarrow \widehat{\mathbb{C}}_{\boldsymbol{X}}$

Diagnostic plot for dimension estimation

(applicable with any SDR method)



Heteroscedastic Y scenario

$\sigma = 0.2$, *Independent* $\mathbf{X}$, $n = 400$

CIM with 5 slices

$R^2_q(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{q}\mathrm{tr}(P_{\mathcal{S}_1} P_{\mathcal{S}_2})$ ***Squared Trace Correlation***, 1 for q=p

$R^2_{o,q}(\mathcal{S}_1, \mathcal{S}_2) = R^2_q(\mathcal{S}_1^\perp, \mathcal{S}_2^\perp)$ complement version, 1 for q=0

Bootstrap Scheme: For each "working dimension" $1 \le q \le p - 1$:

- Estimate $\hat{\mathcal{S}}_q$
- For $j = 1, \ldots, B$ bootstrap replicates, estimate $\hat{\mathcal{S}}_q^{(j)}$ and compute $R_q^{2(j)} = R^2_q(\hat{\mathcal{S}}_q, \hat{\mathcal{S}}_q^{(j)})$, $R_{o,q}^{2(j)} = R^2_{o,q}(\hat{\mathcal{S}}_q, \hat{\mathcal{S}}_q^{(j)})$, and $R_q^{2(j)} R_{o,q}^{2(j)}$.
- Calculate the averages $\bar{R}^2_q$, $\bar{R}^2_{o,q}$, and $\overline{R^2_q R^2_{o,q}}$.

$\bar{R}^2_q$, $\bar{R}^2_{o,q}$ and $\overline{R^2_q R^2_{o,q}}$ all measure 'stability' in estimating $\mathcal{S}_q$.

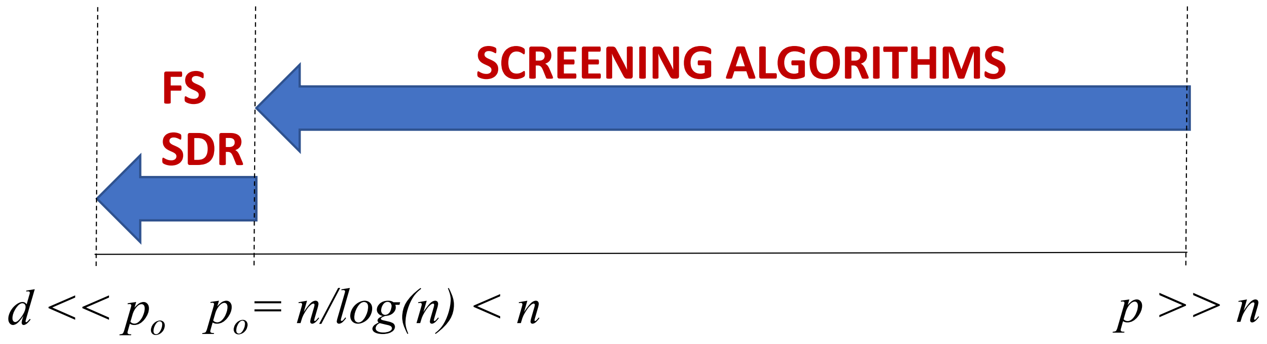
Select $\hat{d}$ where $\overline{R^2_q R^2_{o,q}}$ has a "peak".

What to do when the feature space becomes very high dimensional?

Cut p based on ***marginal utilities***, e.g.

$$\omega_j = corr(Y, X_j) \quad j = 1,2 \ldots p$$

Disregard associations among the X's but with $p >> n$ our ability to account for them is too poor!



SCREENING ALGORITHMS

FS

SDR

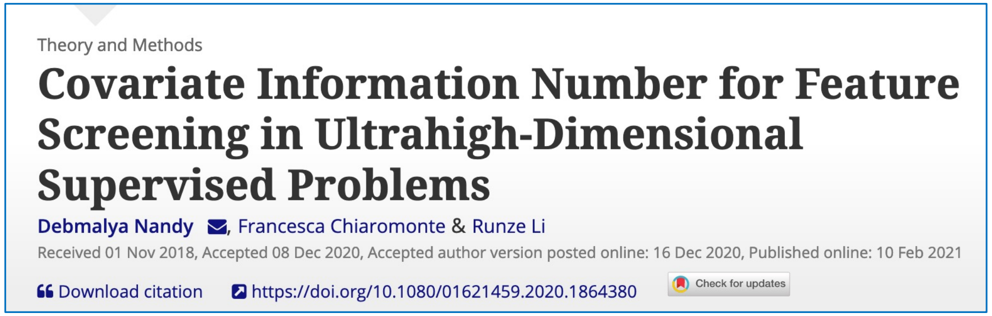
$d << p_o \quad p_o = n/\log(n) < n \qquad p >> n$

Covariate Information Number (one feature at a time)

$$\omega_j = \mathbb{C}_{X_j} = \int \mathbb{F}_{x_j} f(x_j) dx_j = \mathbb{J}_{X_j|Y} - \mathbb{J}_{X_j} \quad j = 1,2 \ldots p$$

$$\uparrow$$

$$\boxed{\int \left[\frac{\partial}{\partial x_j} \log f(y \mid x_j) \right]^2 f(y \mid x_j) \ dy}$$



Theory and Methods

Covariate Information Number for Feature Screening in Ultrahigh-Dimensional Supervised Problems

Debmalya Nandy ✉, Francesca Chiaromonte & Runze Li

Received 01 Nov 2018, Accepted 08 Dec 2020, Accepted author version posted online: 16 Dec 2020, Published online: 10 Feb 2021

Download citation https://doi.org/10.1080/01621459.2020.1864380 Check for updates

Covariate Information Number for Feature Screening in Ultrahigh-Dimensional Supervised Problems

Abstract: Contemporary high-throughput experimental and surveying techniques give rise to ultrahigh-dimensional supervised problems with sparse signals; that is, a limited number of observations (n), each with a very large number of covariates ($p \gg n$), only a small share of which is truly associated with the response. In these settings, major concerns on computational burden, algorithmic stability, and statistical accuracy call for substantially reducing the feature space by eliminating redundant covariates before the use of any sophisticated statistical analysis. Along the lines of *Pearson's correlation coefficient-based sure independence screening* and other model- and correlation-based feature screening methods, we propose a model-free procedure called *covariate information number-sure independence screening* (CIS). CIS uses a marginal utility connected to the notion of the traditional Fisher information, possesses the sure screening property, and is applicable to any type of response (features) with continuous features (response). Simulations and an application to transcriptomic data on rats reveal the comparative strengths of CIS over some popular feature screening methods.