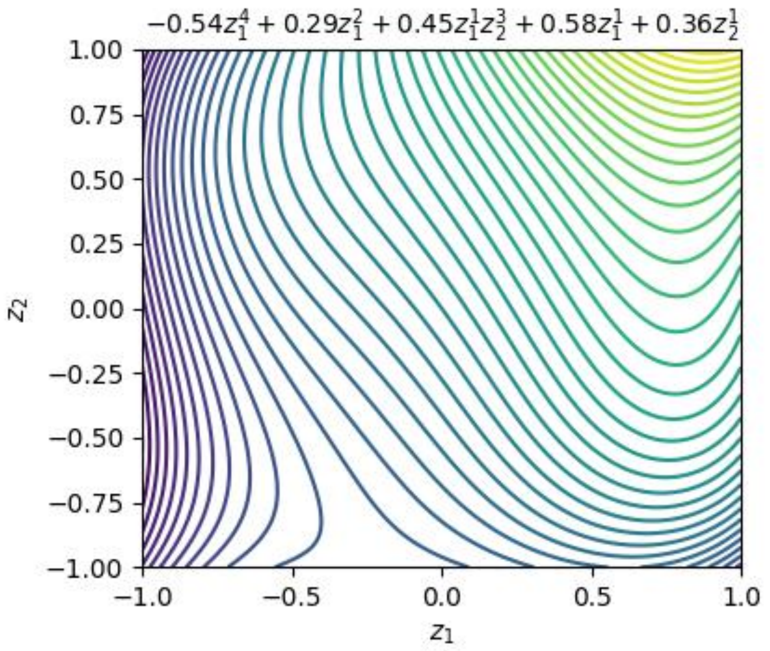
SISSO: Selecting Sparsifying Operators from a Computational and Data Efficiency Perspective

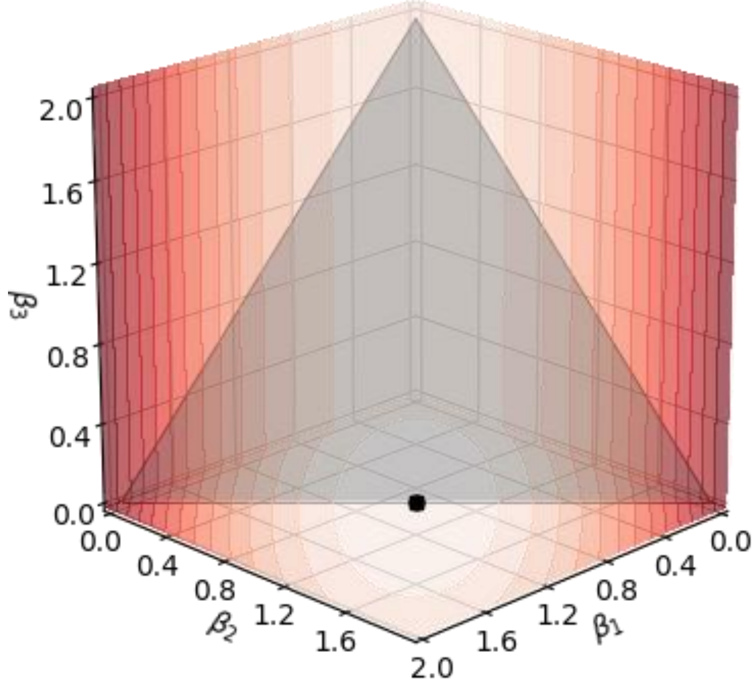
AI[3]-2024, Paphos, Cyprus, 7 Nov 2024

Mario Boley[1,2]
mboley@is.haifa.ac.il

Simon Teshuva[1], Felix Luong[1], Daniel Schmidt[1], Lucas Foppa[3], and Matthias Scheffler[3]

[1]Department of Information Systems, University of Haifa
[2]Department of Data Science and AI, Monash University
[3]NOMAD Laboratory at FHI of Max-Planck-Gesellschaft

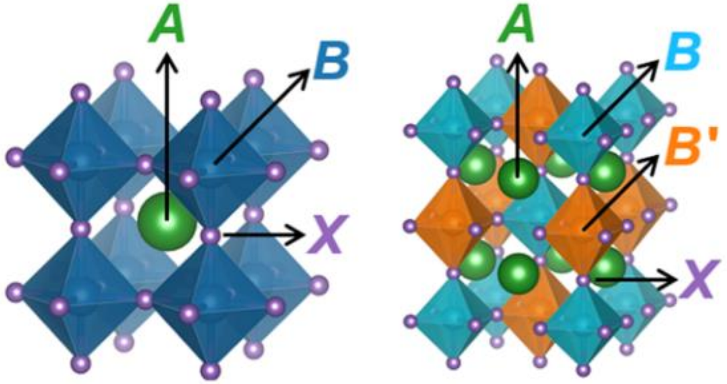




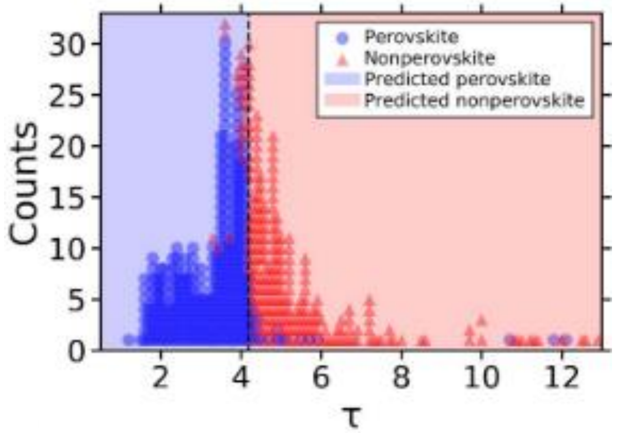
How it all connects

- Closed-loop Materials Discovery Framework
- SISSO Surrogate Model with Uncertainty Estimates
- Statistical Consistency of SISSO
- What SISSO should we use

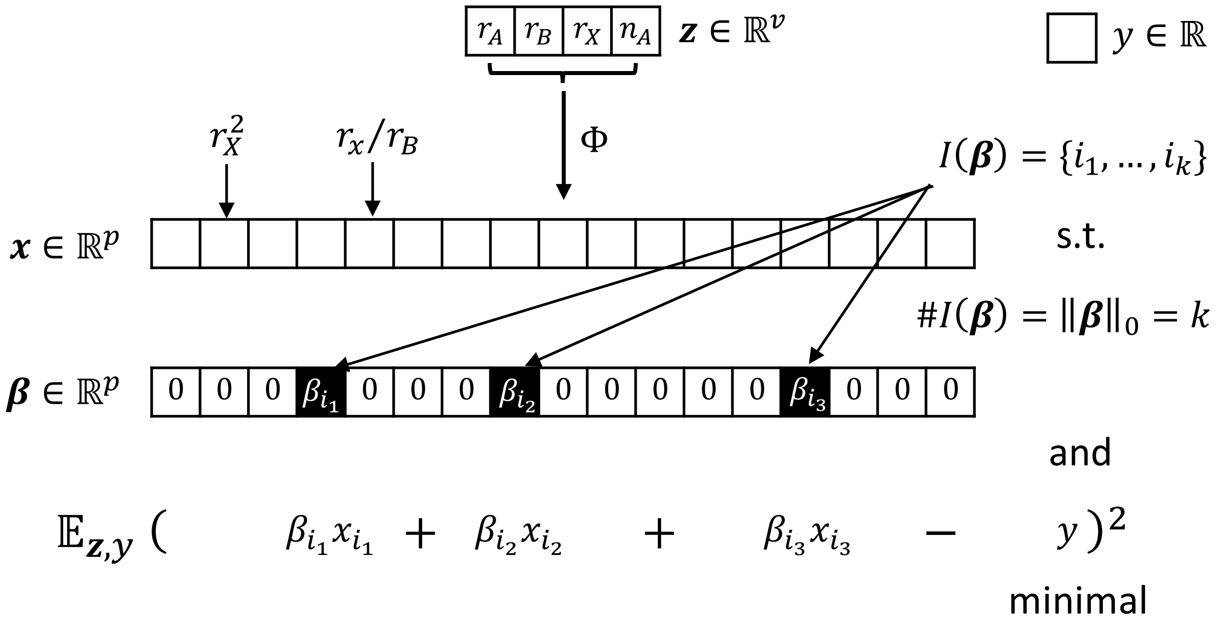
SISSO: Symbolic Regression for Materials Properties

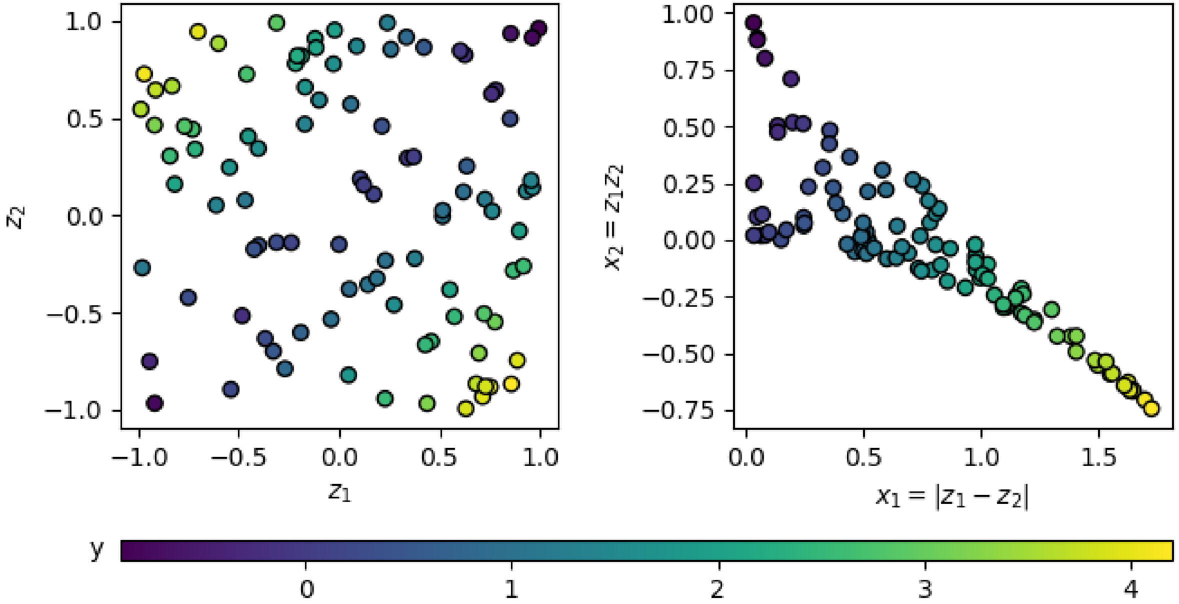


$$\log \frac{P(\text{stable})}{1 - P(\text{stable})} = \beta_1 \frac{r_X}{r_B} + \beta_2 n_A^2 - \beta_3 \frac{n_A r_A / r_B}{\ln(r_A / r_B)}$$



[Bartel, C. J., et al. (2019). *New tolerance factor to predict the stability of perovskite oxides and halides.* Sci. Adv. 5(2).]





[Ouyang et al. (2018). *SISSO: A compressed-sensing method for low-dimensional descriptors*. Phys. Rev. Mater. 2(8)]

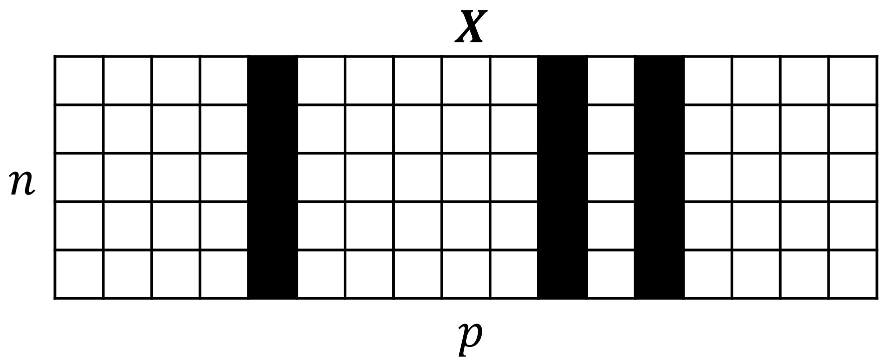
Need to Select Subset via Data Sample

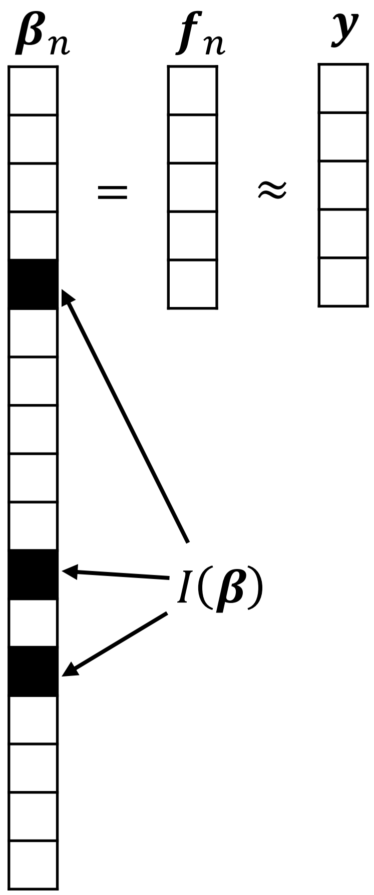
Given:

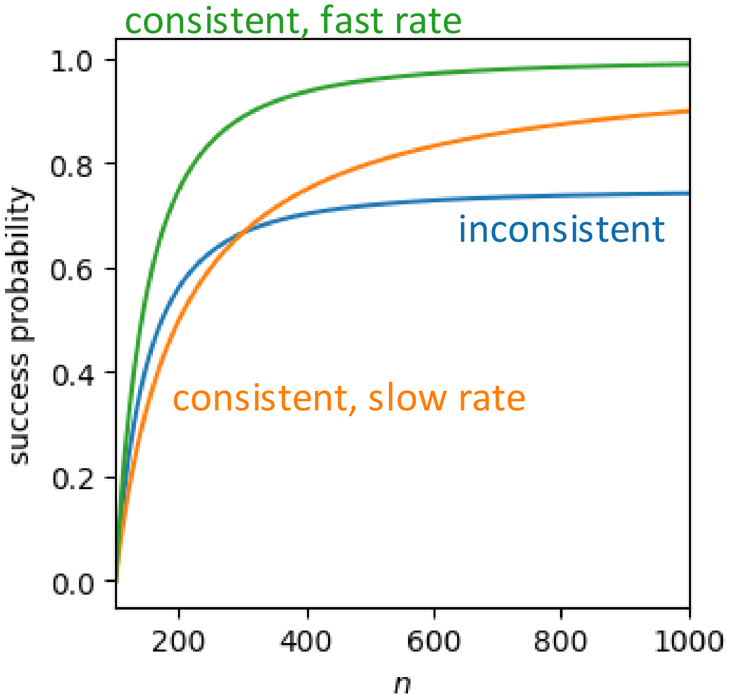
- input matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, output vector $\boldsymbol{y} \in \mathbb{R}^n$ with rows sampled w.r.t. joint $\boldsymbol{x}, y$ distribution
- prescribed sparsity/complexity $k \in \mathbb{N}$
- typically assume $k < n \ll p$

Goal:

- identify $\boldsymbol{\beta}_* = \mathrm{argmin}\{\mathbb{E}(y - x^T\boldsymbol{\beta})^2 : \#I(\boldsymbol{\beta}) = k\}$
- via sparse estimate $\boldsymbol{\beta}_n$ with $\#I(\boldsymbol{\beta}_n) = k$
- computationally efficiently, i.e., in time $O(knp)$
- consistently, i.e., $\lim_{n\to\infty} P\big(I(\boldsymbol{\beta}_n) = I(\boldsymbol{\beta}_*)\big) = 1$
- with as fast a rate as possible







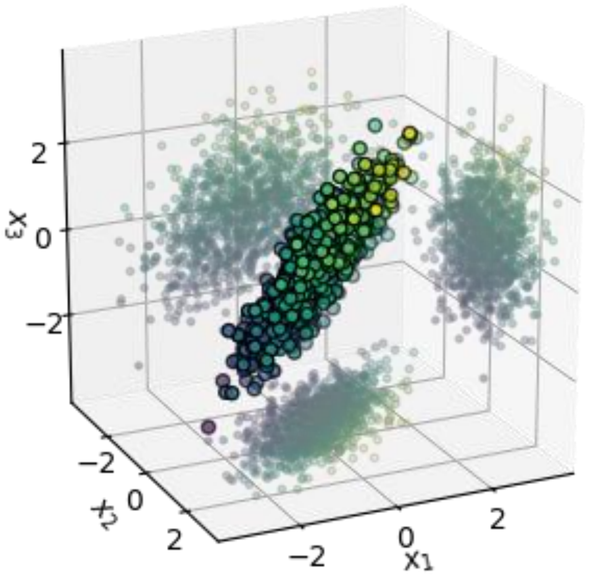
There are many methods… that fail to reach goals

Best-subset-search:

find $\boldsymbol{\beta}_n^{\mathrm{BSS}} = \mathrm{argmin}\{\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 : \#I(\boldsymbol{\beta}) = k\}$

consistent (ordinary least squares parameter consistency)

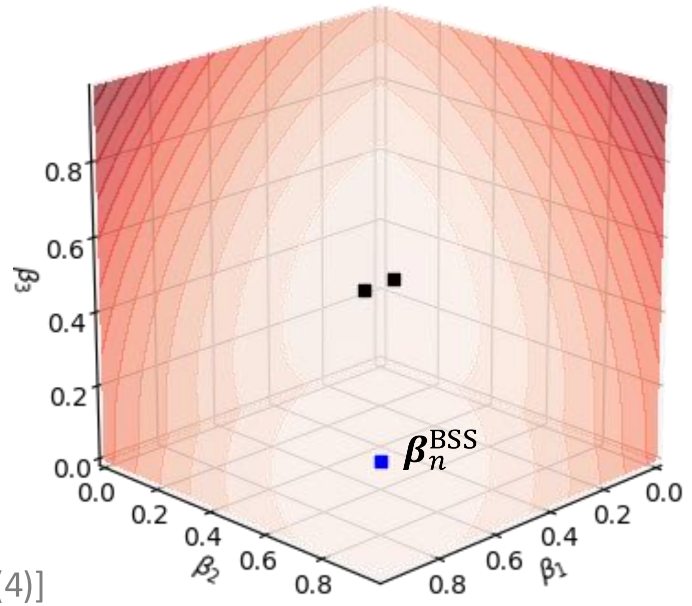
but computationally inefficient $O(C_{p,k}(nk^2 + k^3))$



$y = 0.5x_1 + 0.5x_2$

$x \sim N_3(0, C)$

$$C = \begin{bmatrix} 1 & -3/4 & 0.3 \\ -3/4 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix}$$



[Hastie et al. (2020) *Best Subset, Forward Stepwise or Lasso?* Statist. Sci. 35(4)]

There are many methods… that fail to reach goals

Best-subset-search:

find $\boldsymbol{\beta}_n^{\mathrm{BSS}} = \mathrm{argmin}\{\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 \colon \#I(\boldsymbol{\beta}) = k\}$

consistent (ordinary least squares parameter consistency)

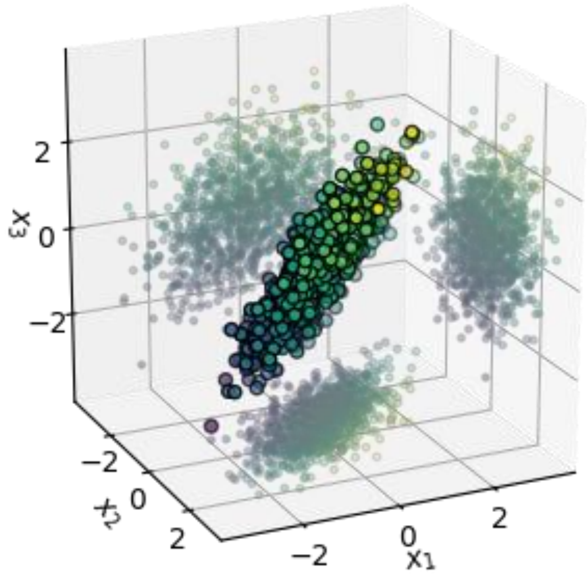
but computationally inefficient $O(C_{p,k}(nk^2 + k^3))$

LASSO:

find $\boldsymbol{\beta}_n^{\mathrm{LAS}} = \mathrm{argmin}\{\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 \colon \|\boldsymbol{\beta}\|_1 \leq c_k\}$

computationally efficient $O(knp + k^3)$

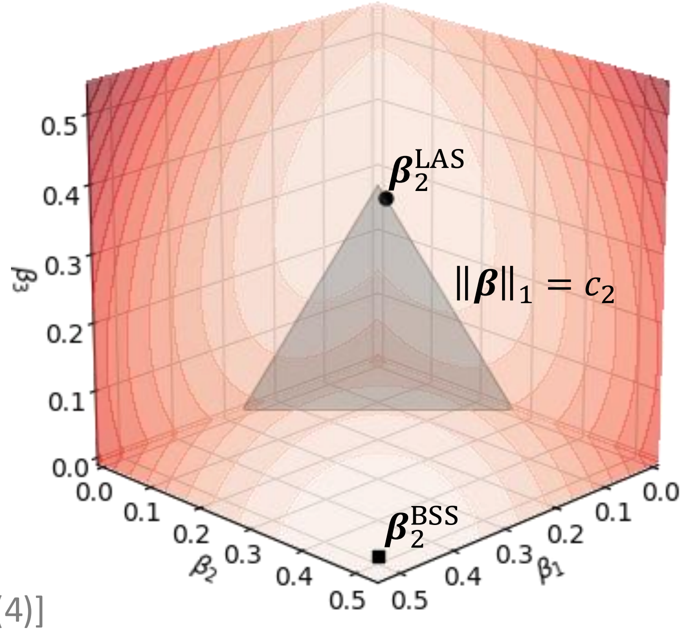
but inconsistent for non-trivial correlation structure



$y = 0.5x_1 + 0.5x_2$

$x \sim N_3(0, C)$

$$C = \begin{bmatrix} 1 & -3/4 & 0.3 \\ -3/4 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix}$$



[Hastie et al. (2020) *Best Subset, Forward Stepwise or Lasso?* Statist. Sci. 35(4)]

There are many methods… that fail to reach goals

Best-subset-search:

find $\boldsymbol{\beta}_n^{\text{BSS}} = \operatorname{argmin}\{\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 : \#I(\boldsymbol{\beta}) = k\}$

consistent (ordinary least squares parameter consistency)

but computationally inefficient $O(C_{p,k}(nk^2 + k^3))$

LASSO:

find $\boldsymbol{\beta}_n^{\text{LAS}} = \operatorname{argmin}\{\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 : \|\boldsymbol{\beta}\|_1 \le c_k\}$

computationally efficient $O(knp + k^3)$

but inconsistent for non-trivial correlation structure

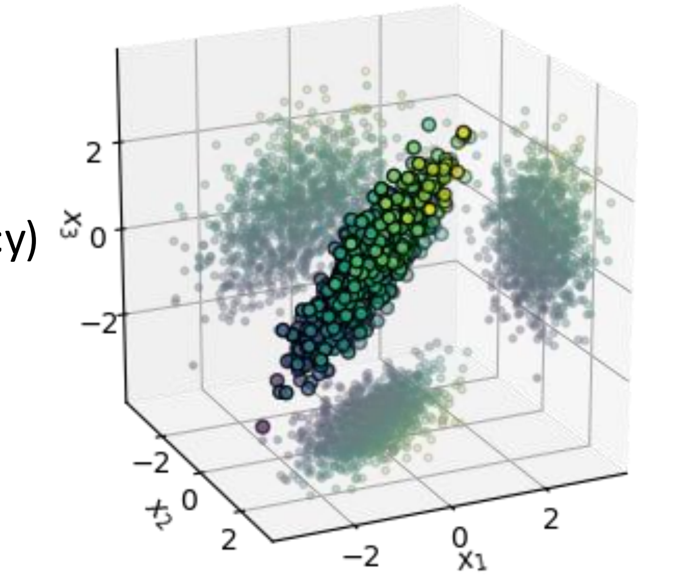
Thresholded Minimum-Norm Least Squares:

find $\boldsymbol{\beta} = \operatorname{argmin}\left\{\lim_{\lambda \to 0_+} \|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_2^2\right\}$

and set $\beta_j^{\text{TLS}} = \begin{cases} \beta_j, & \text{if } |\beta_j| \text{ among } k \text{ largest} \\ 0, & \text{otherwise.} \end{cases}$

consistent (although rate can be slow)

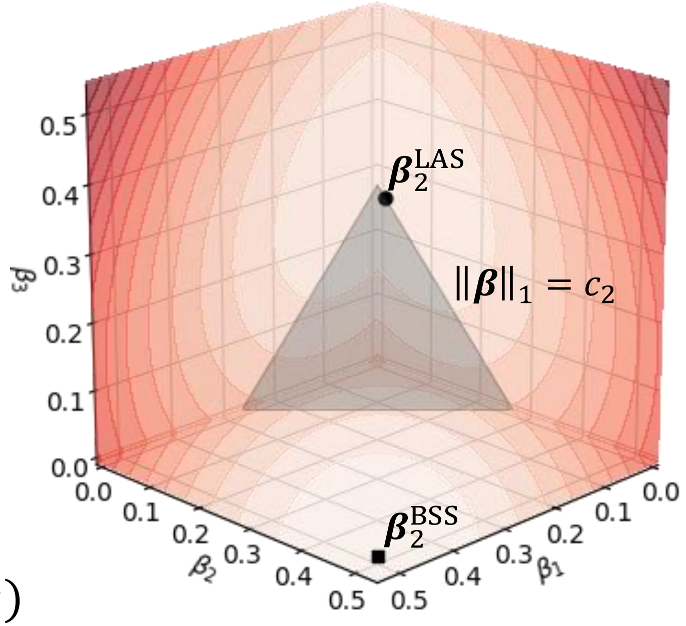
computationally inefficient $O(np^2 + p^3)$ or $O(n^2p + n^3)$



$y = 0.5x_1 + 0.5x_2$

$x \sim N_3(0, C)$

$$C = \begin{bmatrix} 1 & -3/4 & 0.3 \\ -3/4 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix}$$



There are many methods… that fail to reach goals

Best-subset-search:

find $\boldsymbol{\beta}_n^{\mathrm{BSS}} = \operatorname{argmin}\{\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 : \#I(\boldsymbol{\beta}) = k\}$

consistent (ordinary least squares parameter consistency)

but computationally inefficient $O(C_{p,k}(nk^2 + k^3))$

LASSO:

find $\boldsymbol{\beta}_n^{\mathrm{LAS}} = \operatorname{argmin}\{\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 : \|\boldsymbol{\beta}\|_1 \leq c_k\}$

computationally efficient $O(knp + k^3)$

but inconsistent for non-trivial correlation structure

Adaptive LASSO

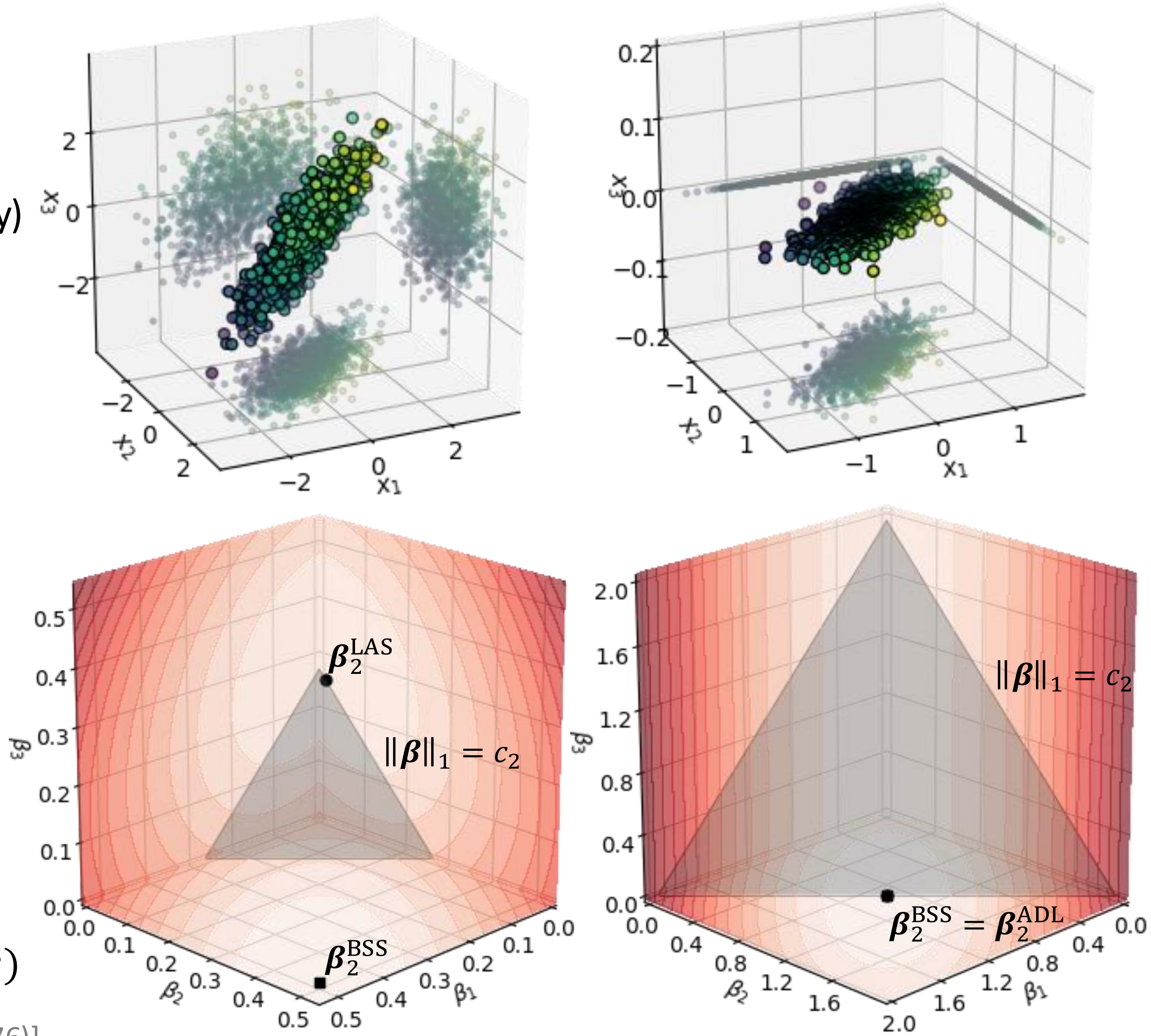
find $\boldsymbol{\alpha} = \operatorname{argmin}\left\{\lim_{\lambda \to 0_+} \|\boldsymbol{y} - \boldsymbol{X\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_2^2\right\}$

and $\boldsymbol{\beta}' = \operatorname{argmin}\|\boldsymbol{y} - \boldsymbol{Z\beta}'\|^2 + \lambda_k\|\boldsymbol{\beta}'\|_1$

and $\beta_j = |\alpha_j|\beta'_j$ where $z_{i,j} = |\alpha_j| x_{i,j}$

consistent (oracle rate in parameter reconstruction)

computationally inefficient $O(np^2 + p^3)$ or $O(n^2p + n^3)$



[Zou, H. (2006). *The Adaptive Lasso and Its Oracle Properties*. JASA, 101(476)]

SIS or "Correlation Learning" Reduces Complexity

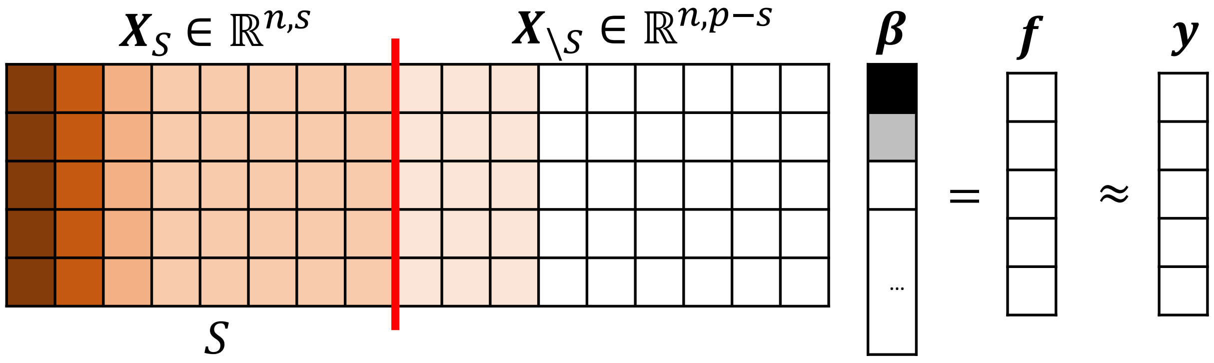
SIS+SO:

find $S = \{j_1, \dots, j_s\}$ where $|\tilde{\boldsymbol{x}}_j^T \boldsymbol{y}| \geq |\tilde{\boldsymbol{x}}_{j+1}^T \boldsymbol{y}|$ for $1 \leq j < p$

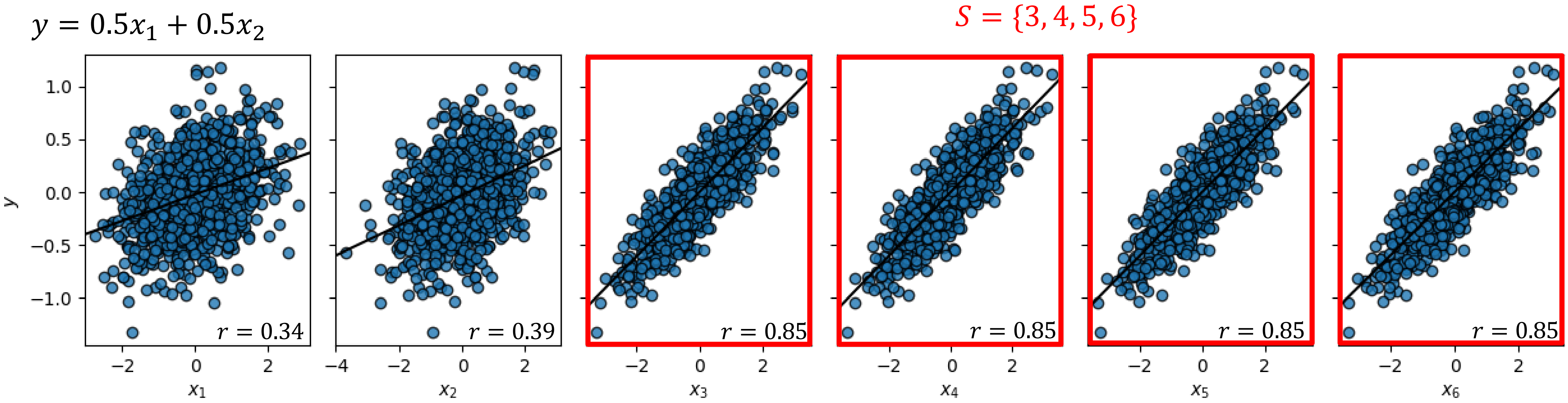
and apply SO to sub-matrix $\boldsymbol{\beta}_n^{\mathrm{SO}}(\boldsymbol{X}_S, \boldsymbol{y})$

computationally efficient for small s: $O(np + T_{\mathrm{SO}}(k, n, s))$

but inconsistent if s too small



$y = 0.5x_1 + 0.5x_2$



[Fan, J., Lv, J. (2008) *Sure independence screening* J. R. Stat. Soc. Ser. B 70(5)]

SISSO is an Iterative Correlation Learning Procedure

SISSO:

set $\boldsymbol{r}_1 = \boldsymbol{y}$

for $l = 1, \ldots, k$:

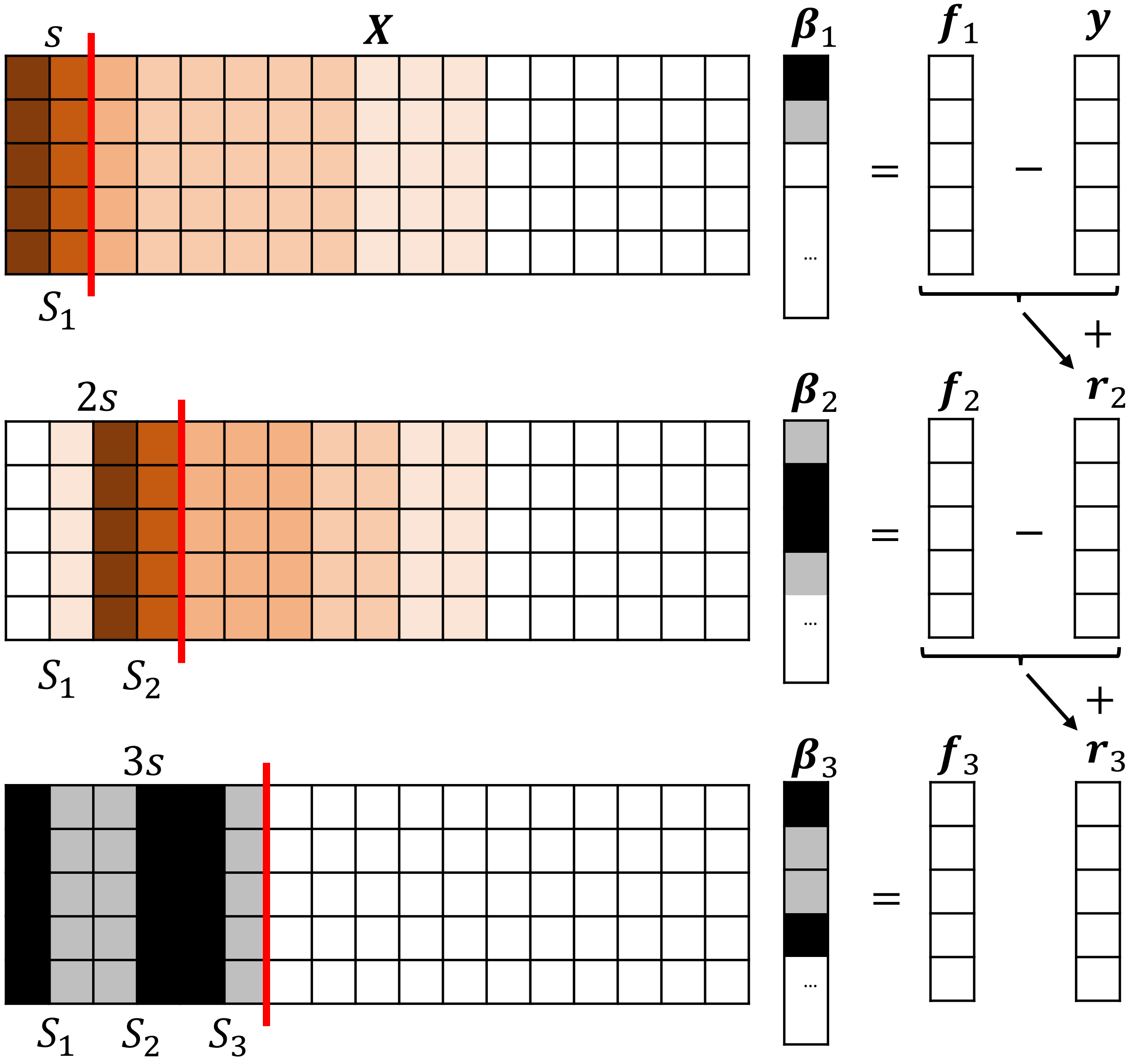
find $S_l = \{j_1, \ldots, j_s\}$ s.t. $|\tilde{\boldsymbol{x}}_j^T \boldsymbol{r}_l| \geq |\tilde{\boldsymbol{x}}_{j+1}^T \boldsymbol{r}_l|$ for $1 \leq j < p$

set $\boldsymbol{\beta}_{l,n}^{\mathrm{SISSO}} = \boldsymbol{\beta}_{l,n}^{\mathrm{SO}}(\boldsymbol{X}_S, \boldsymbol{y})$ with $S = S_1 \cup \cdots \cup S_l$

and $\boldsymbol{r}_{l+1} = \boldsymbol{y} - \boldsymbol{X}_S \boldsymbol{\beta}_{l,n}^{\mathrm{SISSO}}$

Fundamental Questions:

1. What s computationally efficient, i.e., what is $s_{\max}$ st $T_{\mathrm{ICL}}^{\mathrm{SO}} \in O\left(knp + \sum_{l=1}^{k} T_{\mathrm{SO}}(l, n, ls_{\max})\right) \leq c_0 + c_1 knp$?
2. What SO is consistent / performs best when choosing optimal $s \leq s_{\max}$?
3. Can performance be retained when choosing s data-driven?

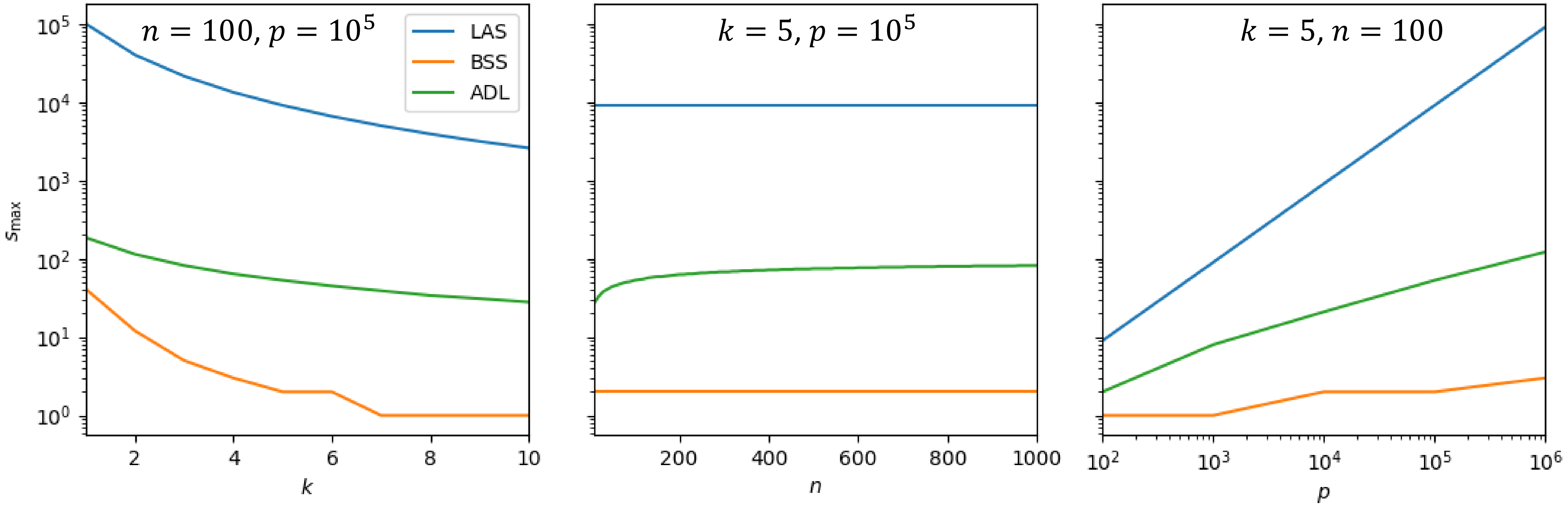


[Barut et al. (2016) *Conditional sure independence screening* JASA 111(515)]

[Fan, J., Lv, J. (2008) *Sure independence screening* J. R. Stat. Soc. Ser. B 70(5)]

[Ouyang et al. (2018) *SISSO: A compressed-sensing method for low-dimensional descriptors* Phys. Rev. Mater. 2(8)]

Computationally Feasible Pool Increment Values



General definition: $s_{\max}(k, n, p) = \max\{s \in \mathbb{N}: T_{\mathrm{ICL}}(k, n, p, s) \leq c_0 + c_1 knp\}$

Lasso: $s_{\max}^{\mathrm{LAS}} \in \Theta(p/k^2)$

Best-subset-search: $s_{\max}^{\mathrm{BSS}} \in \Theta\left(\sqrt[k]{p}\right)$

Adaptive Lasso: $s_{\max}^{\mathrm{BSS}} \in O\left(\min\left(\left(\sqrt{p}, \sqrt[3]{np}\right)\right)/k\right) \cap \Omega\left(\sqrt[3]{p/k^2}\right)$

Evaluation over Wide Range of Functions

Ten correlated normal primary inputs

$\boldsymbol{z} \sim \mathrm{N}_{10}(\boldsymbol{0}, \boldsymbol{C}),\ C_{i,j} = 0.8^{|i-j|}$

Degree $d = 1,2,\dots,7$ multinomial feature maps

$\Phi_d = \{\boldsymbol{\varphi} \in \mathbb{N}^{10} \colon \|\boldsymbol{\varphi}\|_1 \leq d\}$

$x_{\boldsymbol{\varphi}} = \boldsymbol{z}^{\boldsymbol{\varphi}} = z_1^{\varphi_1} z_2^{\varphi_2} \dots z_{10}^{\varphi_{10}}$

$\boldsymbol{x} = \left(z_1^d, z_1^{d-1} z_2, z_1^{d-2} z_2 z_3, \dots, z_{10}^2, z_1, \dots, z_9, z_{10}\right)$

Random sparse polynomials

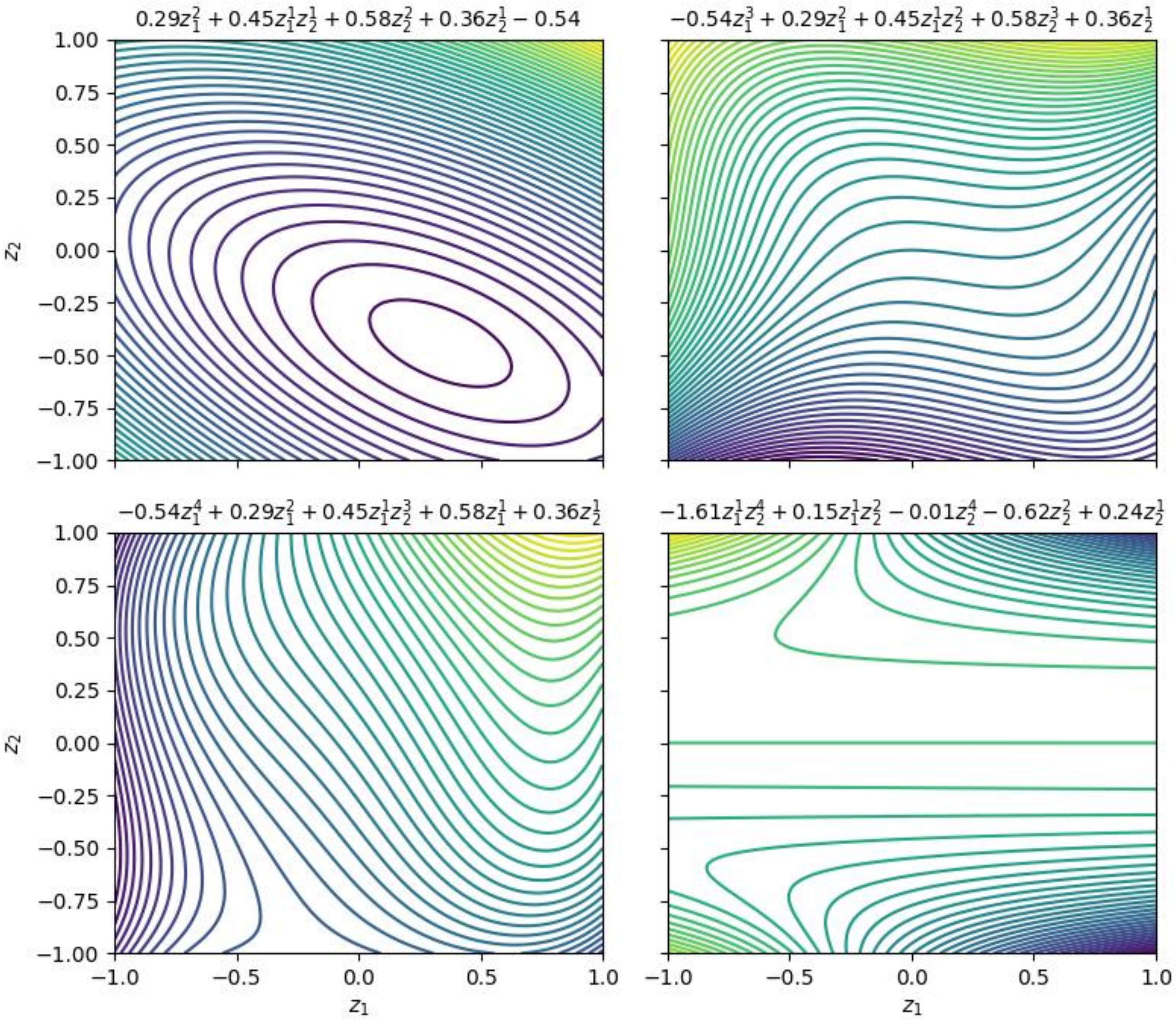
$R = \{\boldsymbol{\varphi} \in \Phi \colon \varphi_6 = \cdots = \varphi_{10} = 0\}$

$I^* \sim \mathrm{Unif}(\{I \subseteq R \colon \#I = 5\})$

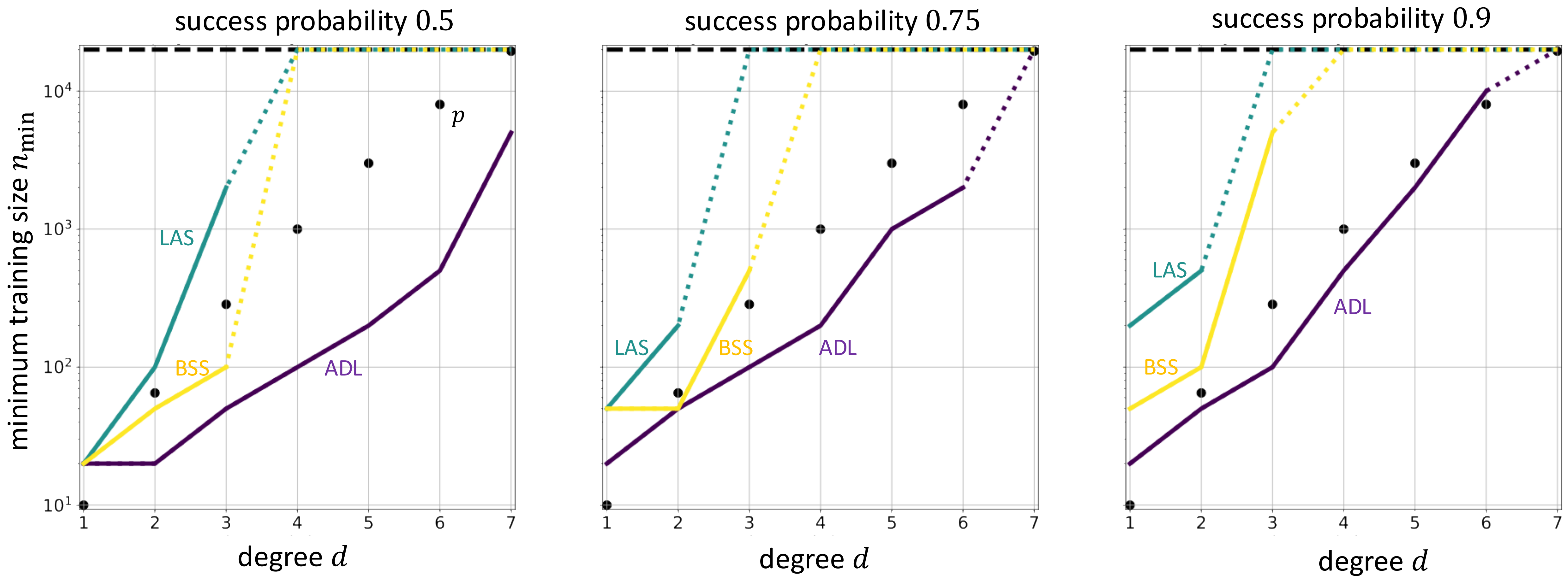
$\beta_j^* \sim N\left(0, \sigma_j^{-1}\right)$ for $j \in I^*$ and $\beta_j^* = 0$ for $j \notin I^*$

Ten polynomials per degree

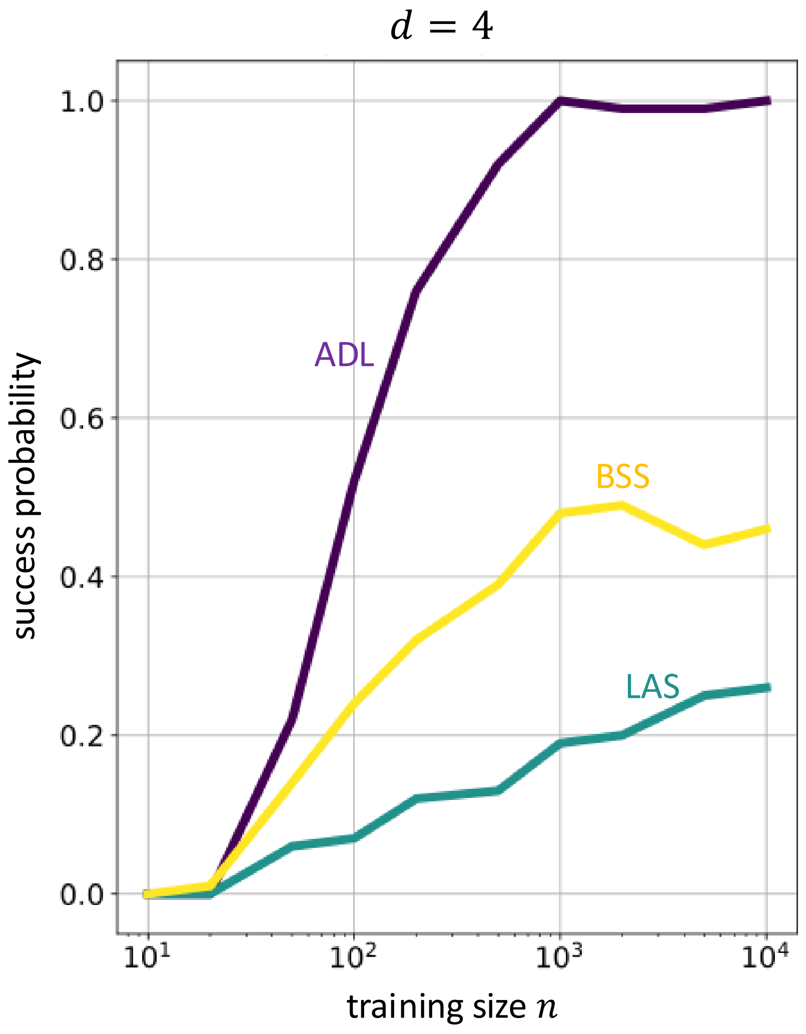
Ten datasets per polynomial

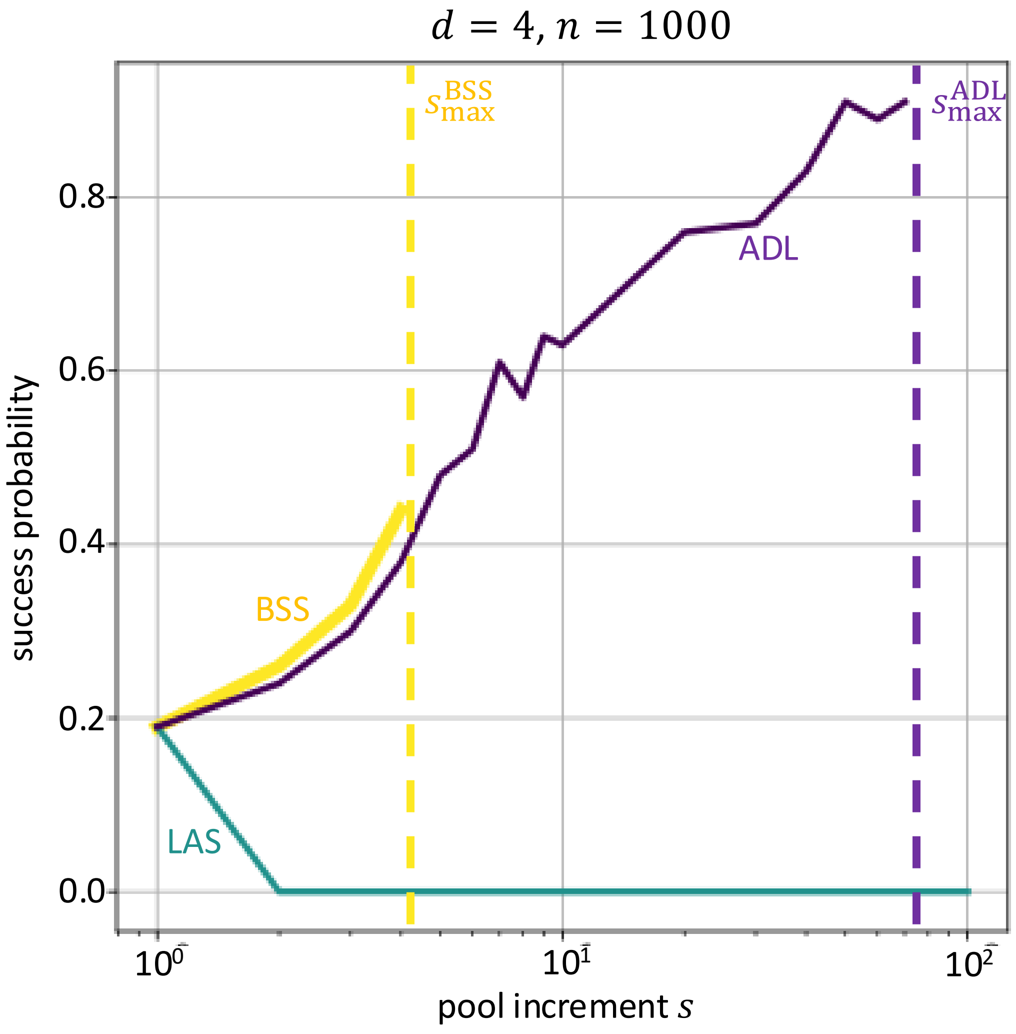


Adaptive Lasso Best-performing Sparsifying Operator

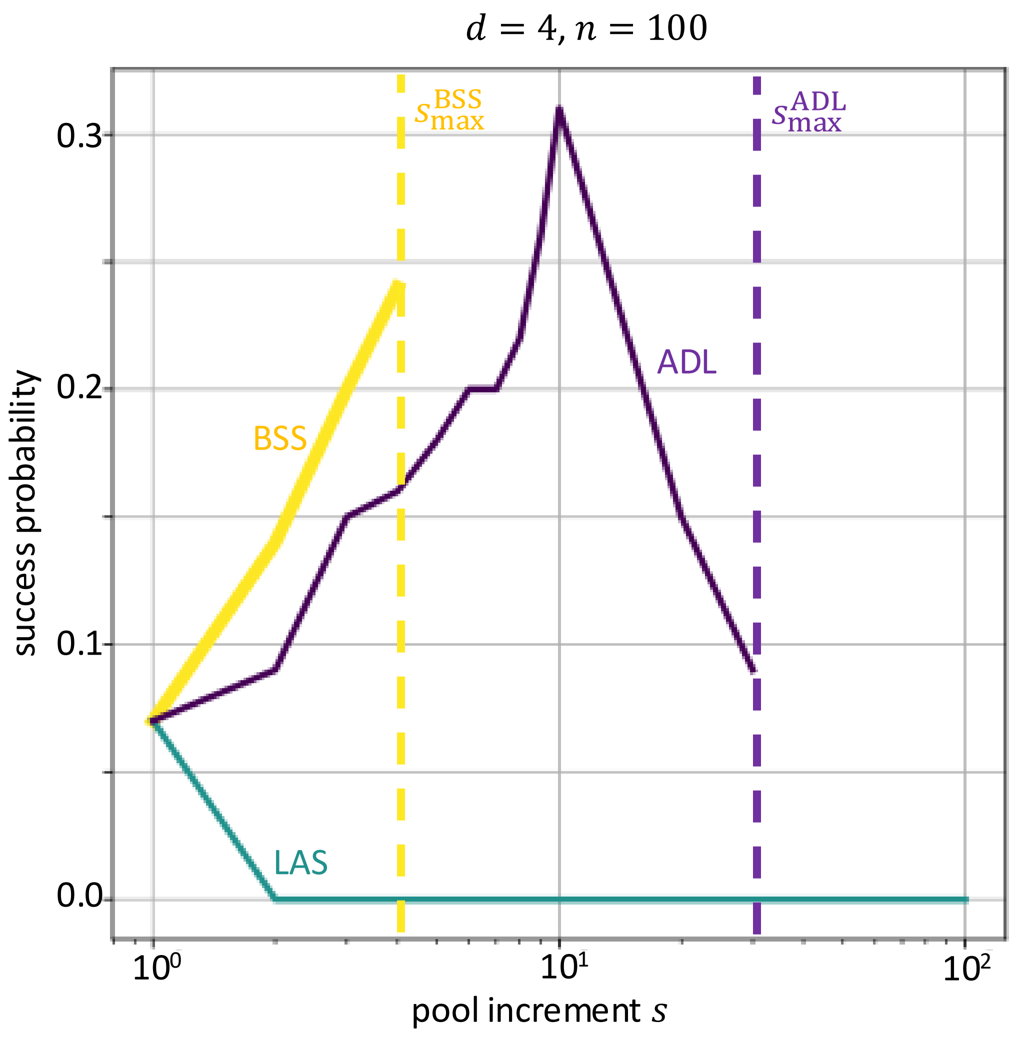


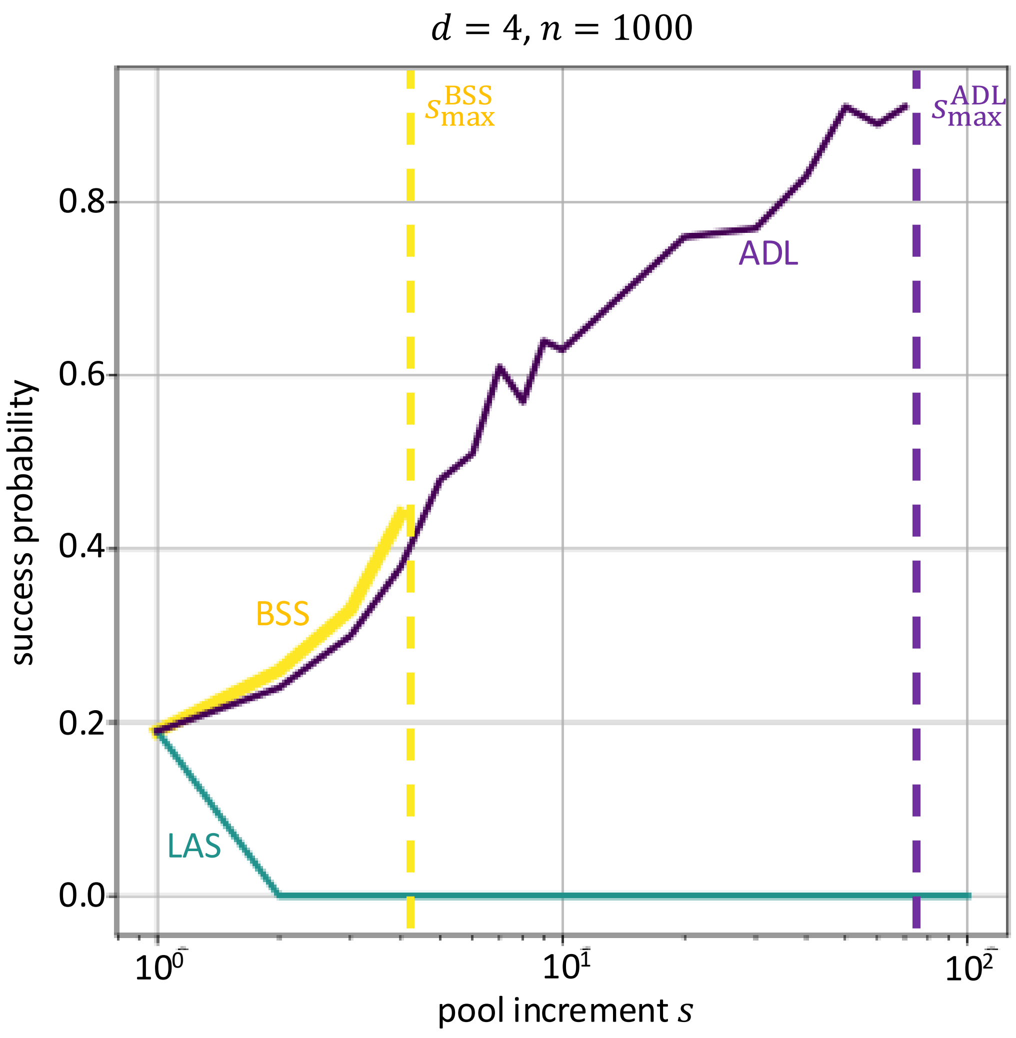
Advantage due Larger Range of Available s values





Maximum Pool Increment is not Always Optimal





Advantage Retained with Data-driven Selection

In practice: s_* unknown and s needs to be selected based on fixed rule or data, e.g., via cross validation:

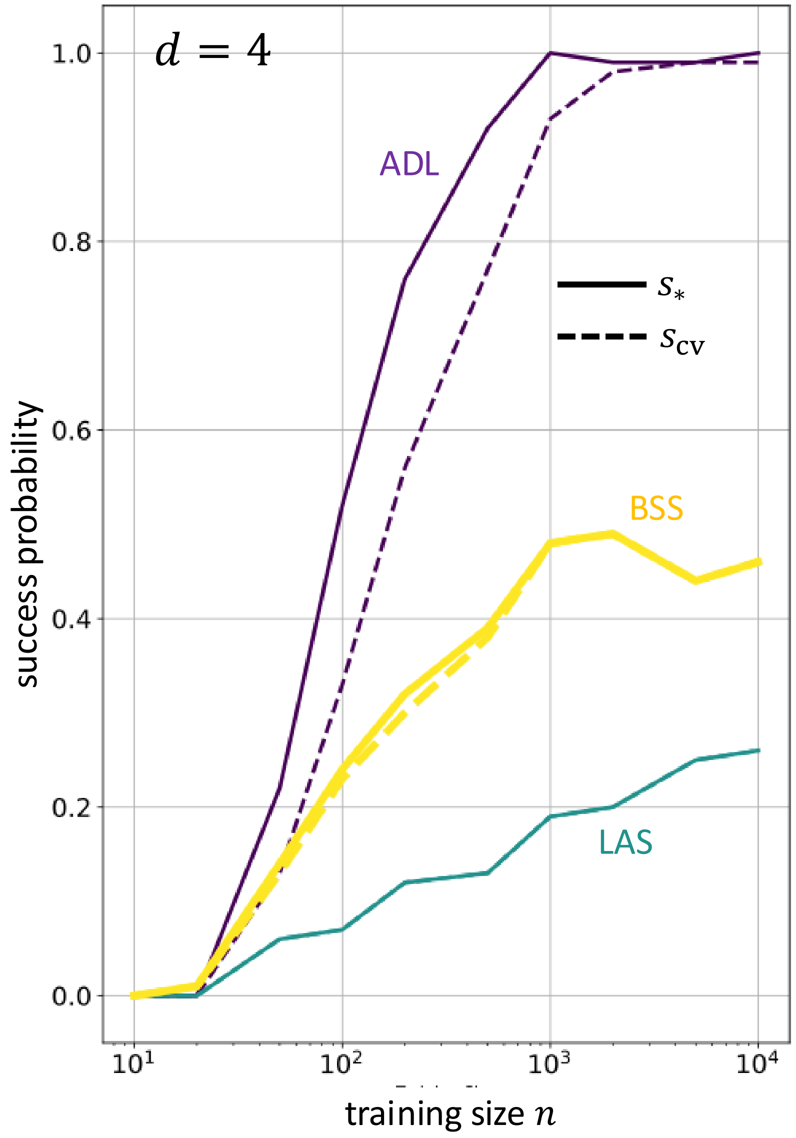
- $s_{\mathrm{cv}} = \mathrm{argmin}\{\sum_{l=1}^{10} \|\boldsymbol{X}_l \boldsymbol{\beta}_l - \boldsymbol{y}_l\|^2 : 1 \leq s \leq s_{\max}\}$
- $\boldsymbol{\beta}_l = \boldsymbol{\beta}(\boldsymbol{X}_{\setminus l}, \boldsymbol{y}_{\setminus l}, s)$

Note: selection problem hardest for adaptive Lasso

- **BSS:** only few feasible s and $s = s_{\max}$ tends to work well
- **Lasso:** generally want very small s (1 or 2), i.e., slightly relaxed matching pursuit works better than Lasso
- **Adaptive Lasso:** relatively wide range available and need to trade off selection of relevant versus irrelevant variables

Result

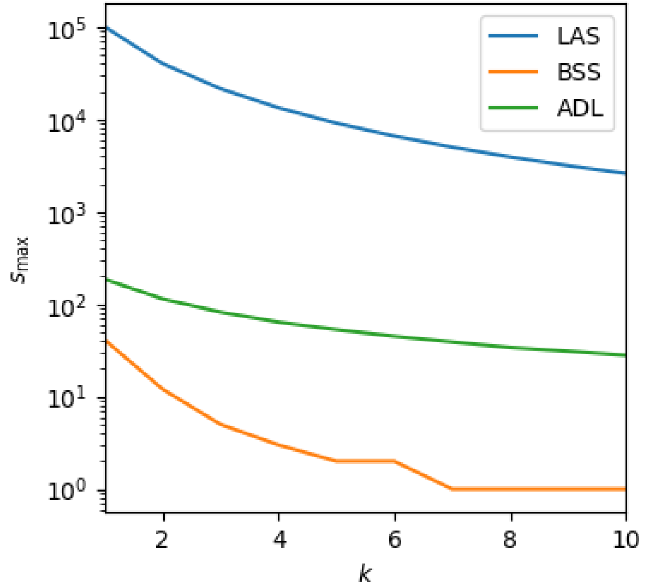
- While data-driven selection reduces adaptive Lasso performance, marked advantage retained over BSS
- …at least for degree 4 polynomials (limit due to 10x comp. cost)



Conclusion

Summary

- Investigate identification consistency and convergence rates of SISSO methods under explicit computational constraint
- Adaptive Lasso appears to be attractive SO, combining consistency with relative computational efficiency
- Indeed, outperforms BSS and Lasso in wide range of practical problems and retained when using cross validation to choose pool increment



Future

- Theoretical bounds for SISSO success probability
- Translation to materials properties modelling
- Sparse regression estimators with computational cost between ADL and BSS, e.g., SCAD, Dantzig Selector, iterative thresholding?