Statistically Sound Model Inference with Sparse Regression

Statistical Verification of Analog Circuits under Process Variations

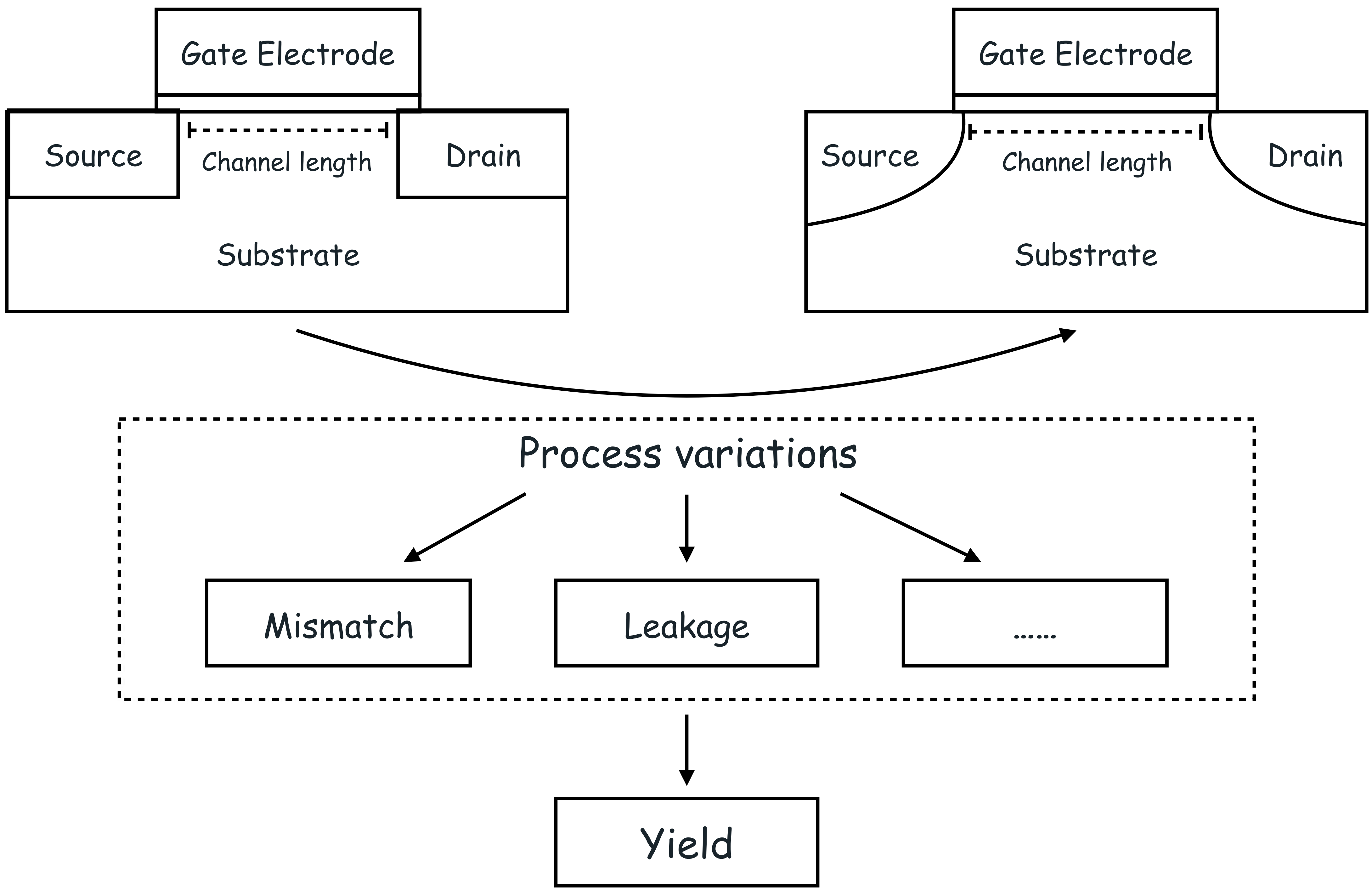
Yan Zhang

Department of ECEE, University of Colorado at Boulder

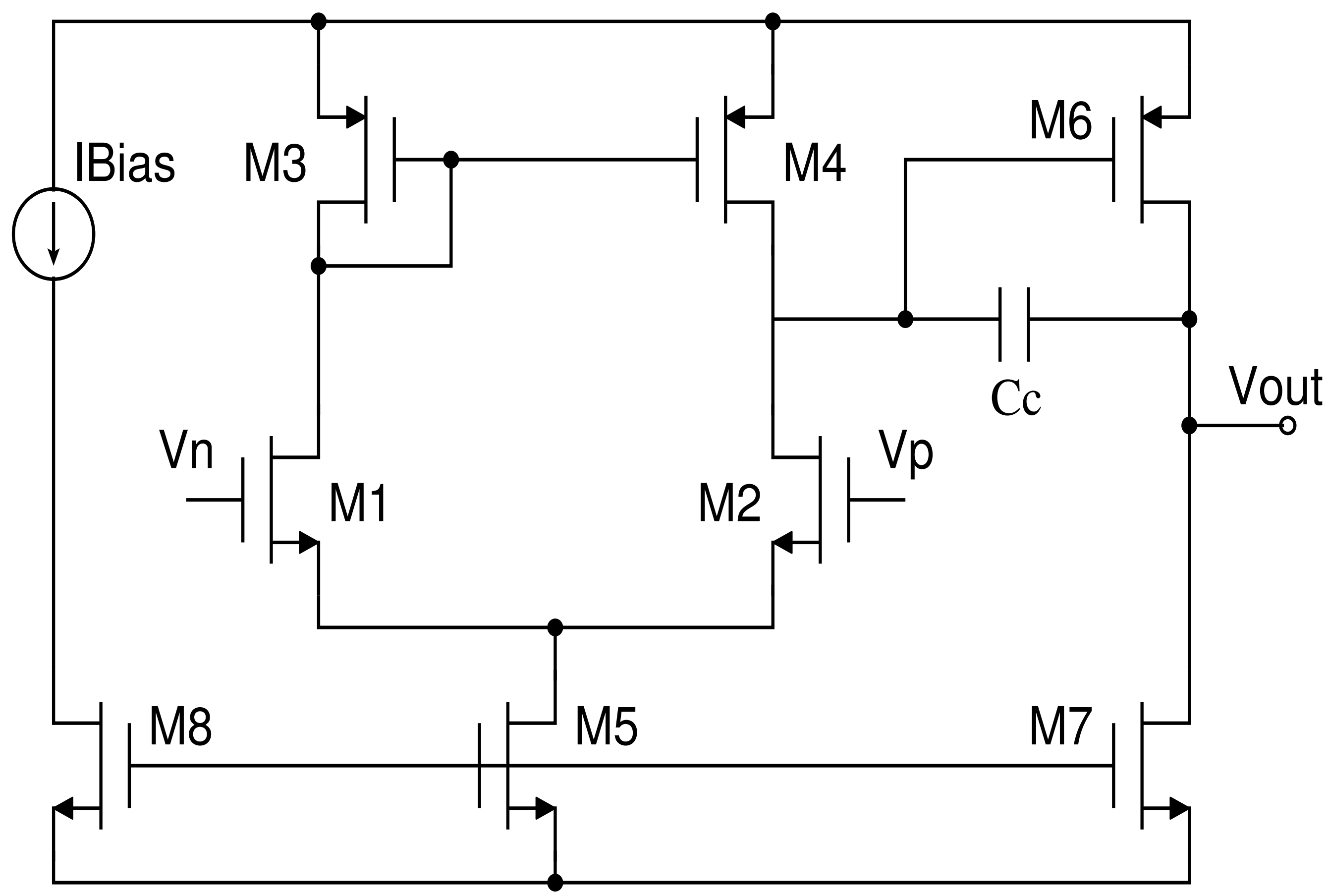
yan.zhang@colorado.edu

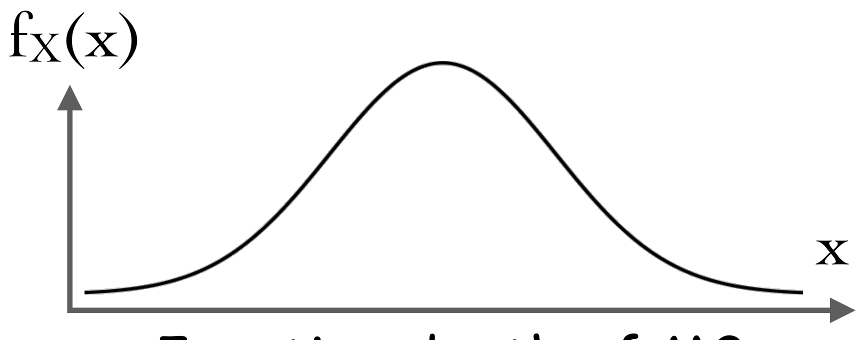
Coauthored with
Sriram Sankaranarayanan and Fabio Somenzi

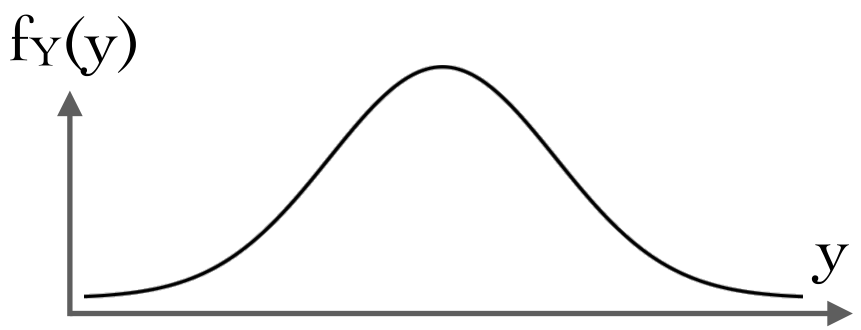
Process Variations



Verification under Process Variations







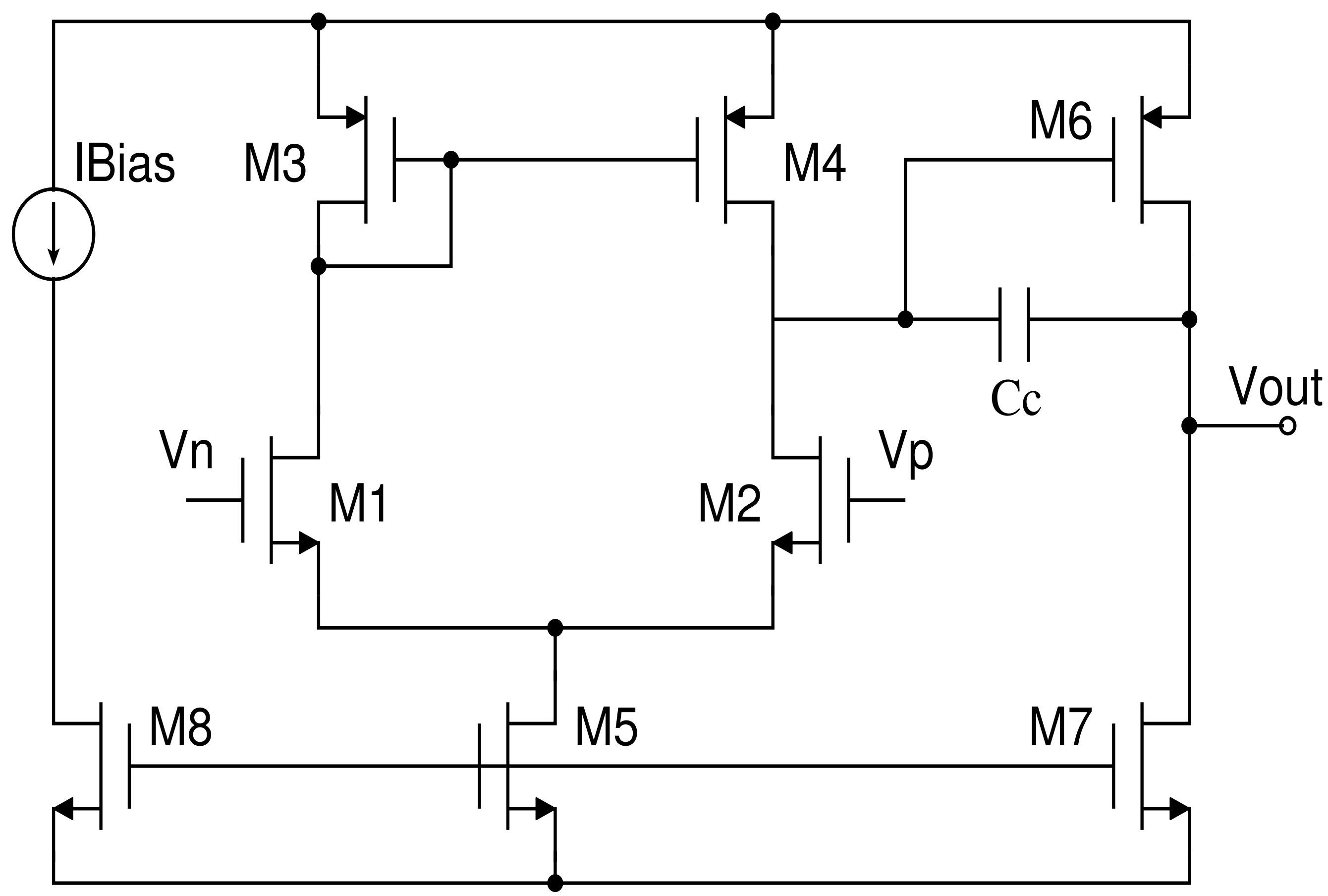
Slew Rate > 10V/μs ?

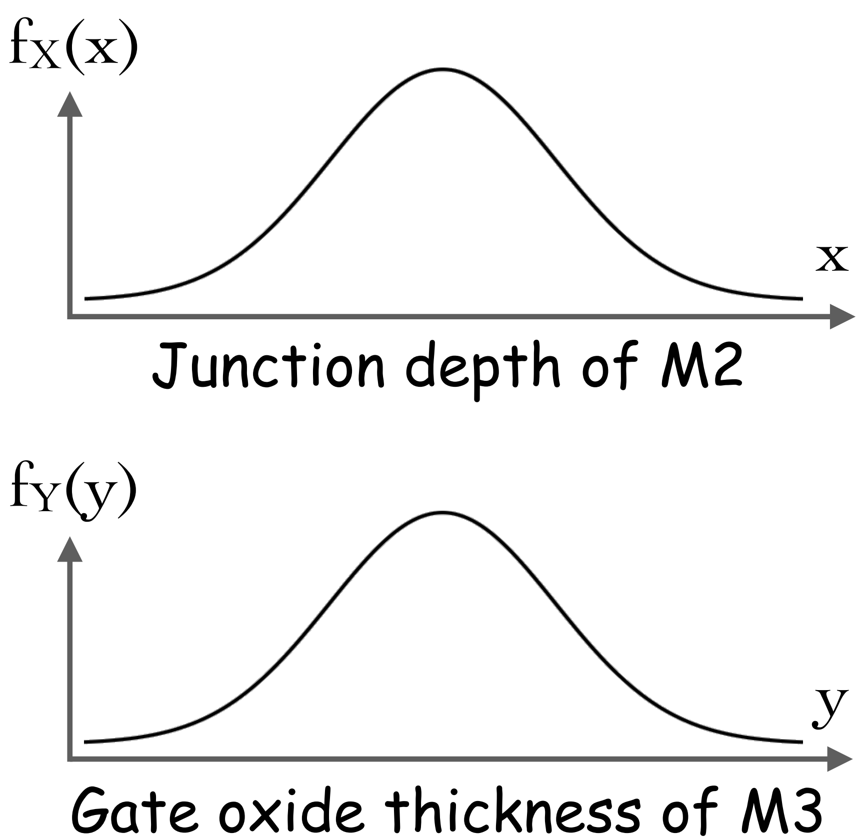
V_{offset} < 15mV ?

Phase Margin > 60° ?

PSRR > 100dB ?

Verification under Process Variations





Slew Rate > 10V/μs ?

V_{offset} < 15mV ?

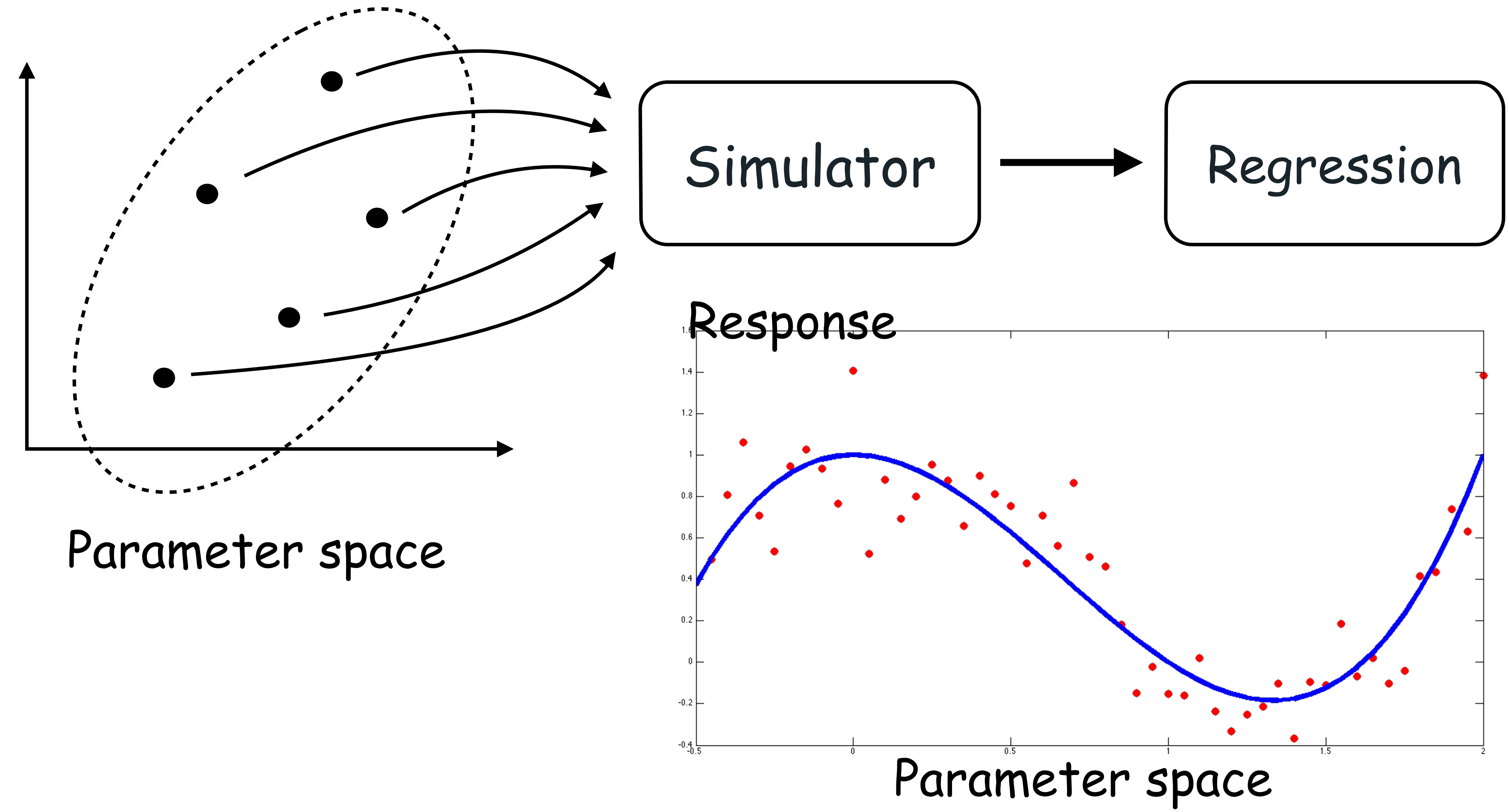
Phase Margin > 60° ?

PSRR > 100dB ?

Model relationship between parameters and response

Performance Modeling

A model inference problem



OLS Regression

Ordinary Least Squares (OLS)

$\min \Sigma_{i \in [1,m]} [y^{(i)} - f(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})]$

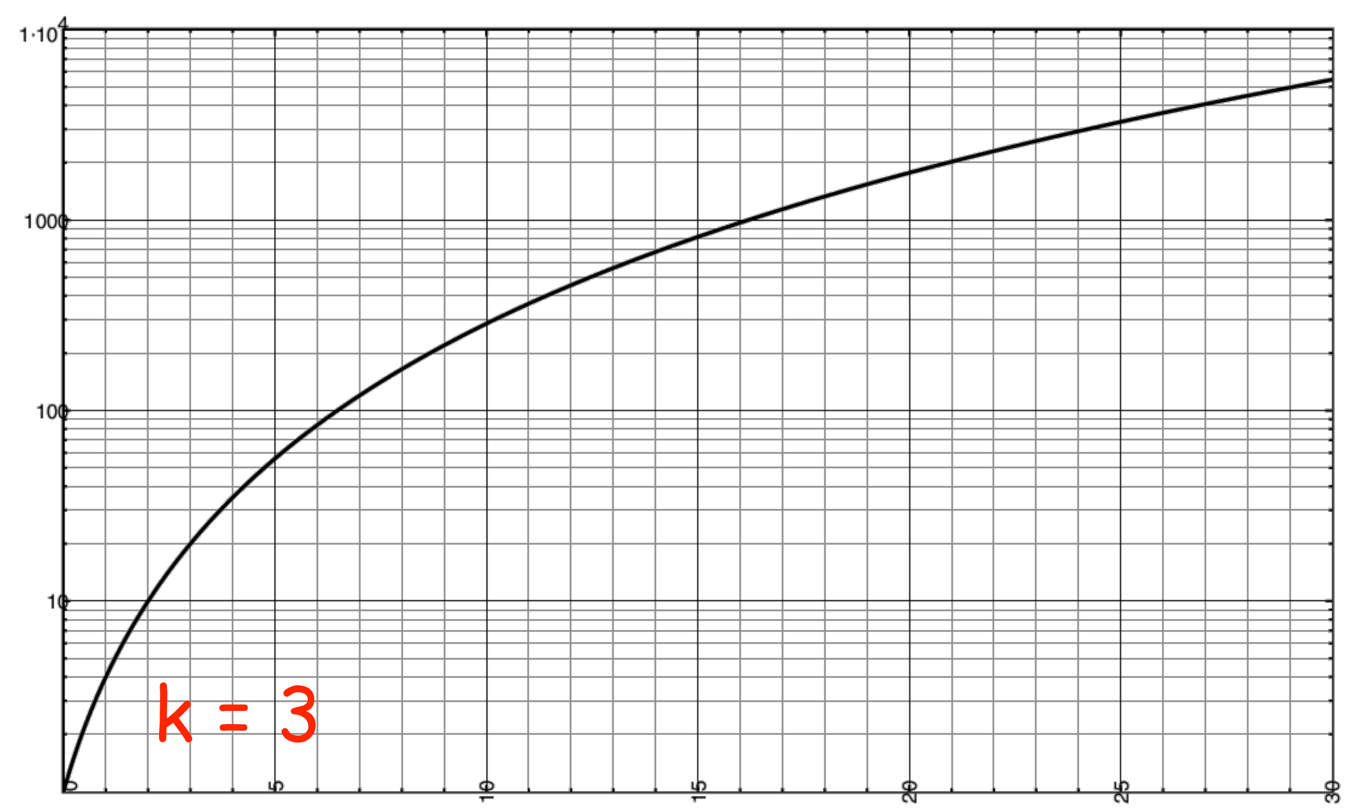
s.t.

f is a degree-k polynomial

Problem: Over-Fitting.

$$m \geq \binom{n+k}{k}$$

- m: # data points
- n: # parameters
- k: polynomial degree



OLS Regression

Ordinary Least Squares (OLS)

$$\min \Sigma_{i \in [1,m]} [y^{(i)} - f(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})]$$

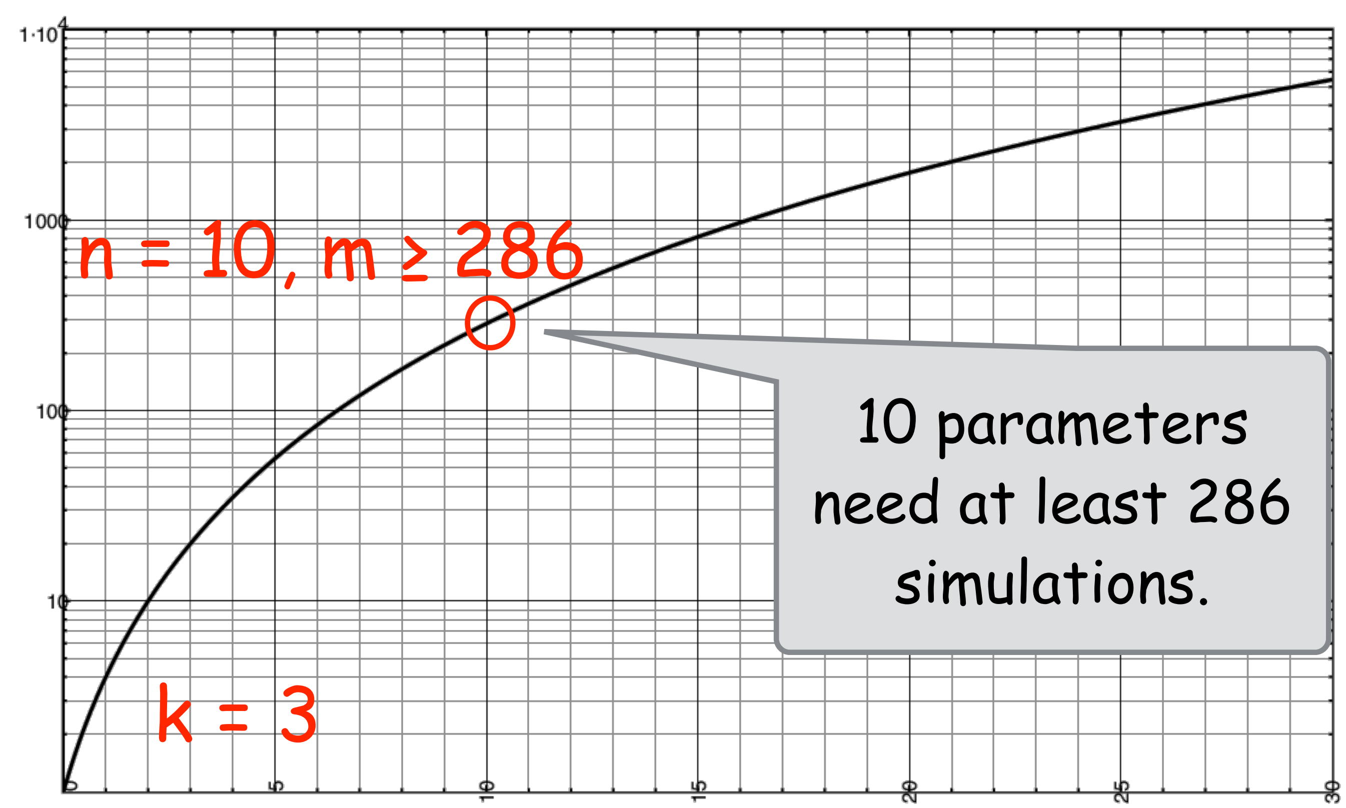
s.t.

f is a degree-k polynomial

Problem: Over-Fitting.

$$m \geq \binom{n+k}{k}$$

- m: # data points
- n: # parameters
- k: polynomial degree



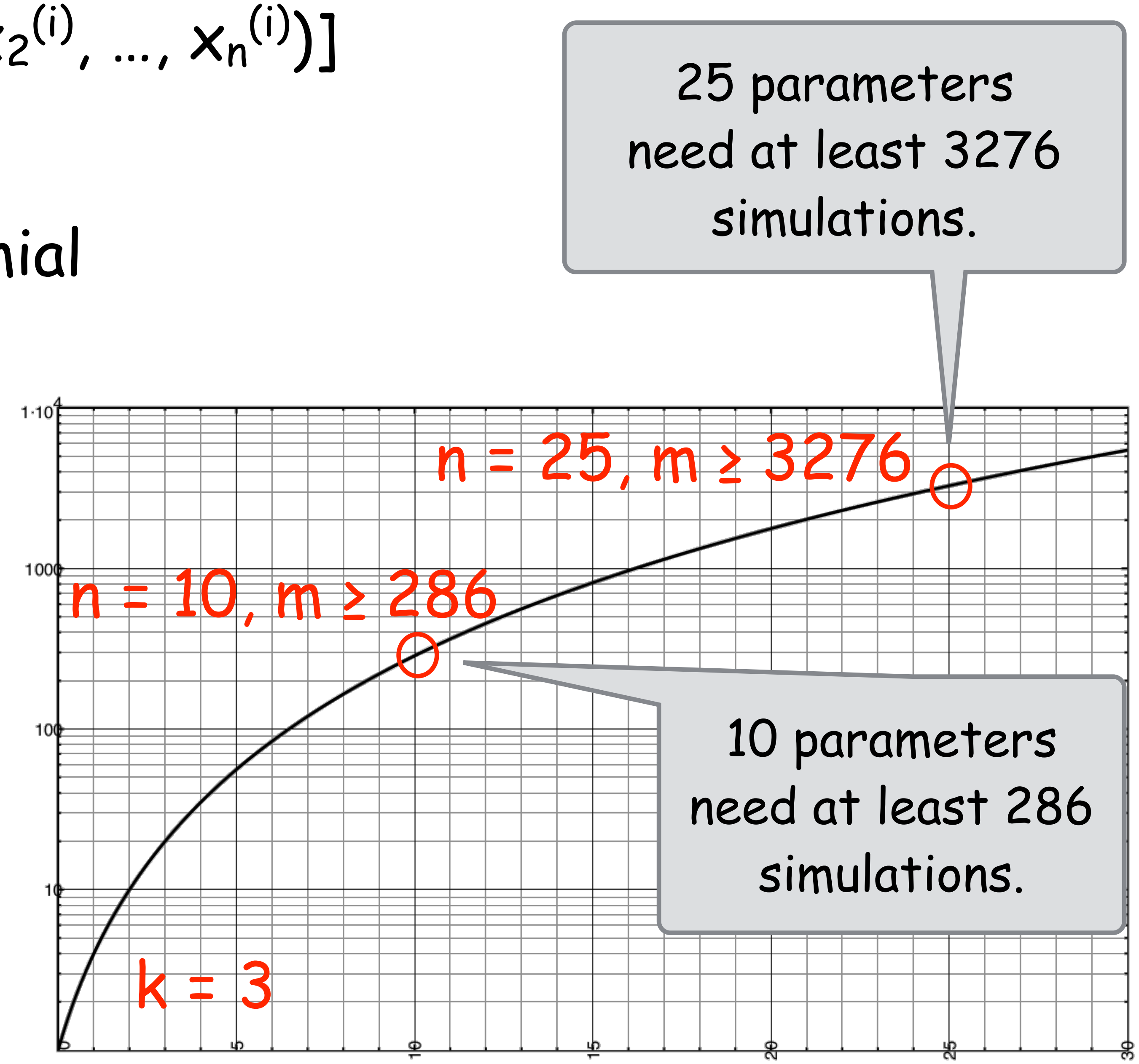
OLS Regression

Ordinary Least Squares (OLS)

$$\min \Sigma_{i \in [1,m]} [y^{(i)} - f(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})]$$

s.t.

f is a degree-k polynomial

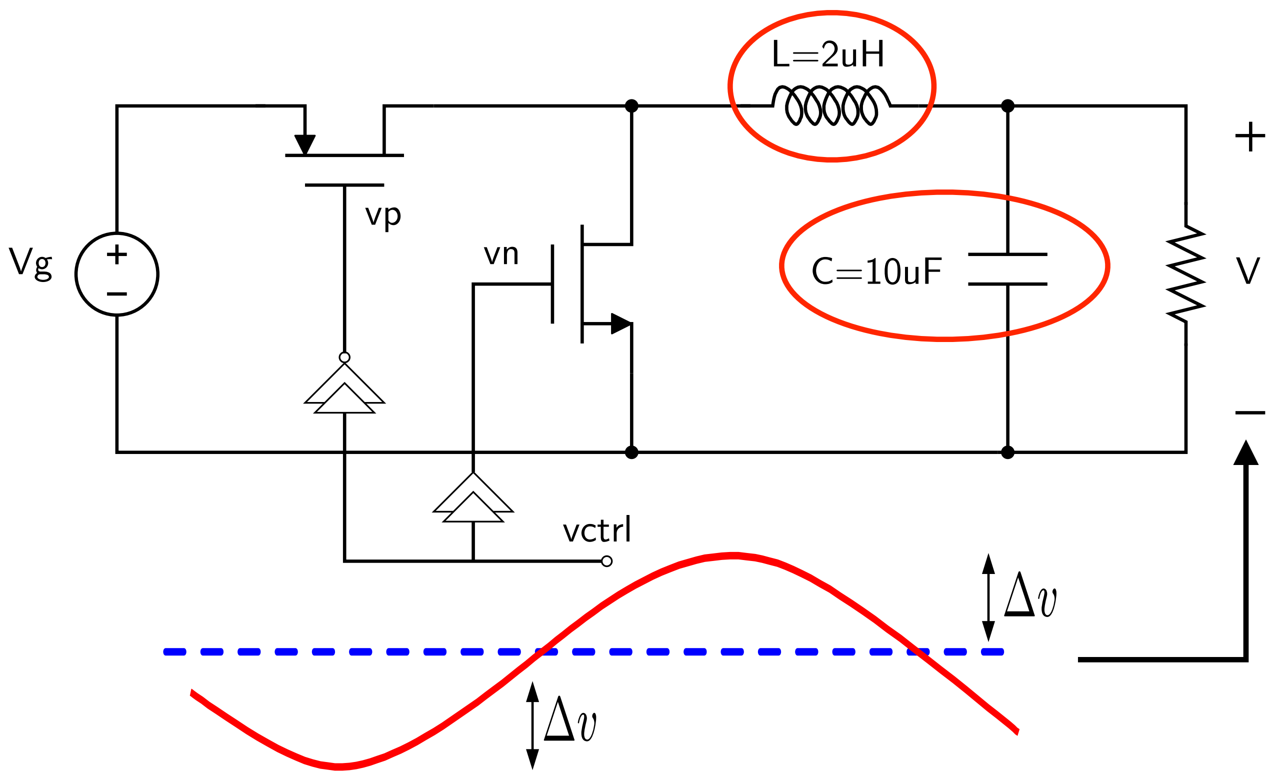


Problem: Over-Fitting.

$$m \geq \binom{n+k}{k}$$

- m: # data points
- n: # parameters
- k: polynomial degree

A Simple Buck Converter



Process variations

L ~ U(1.8,2.2)μH, C ~ U(9,11)μF

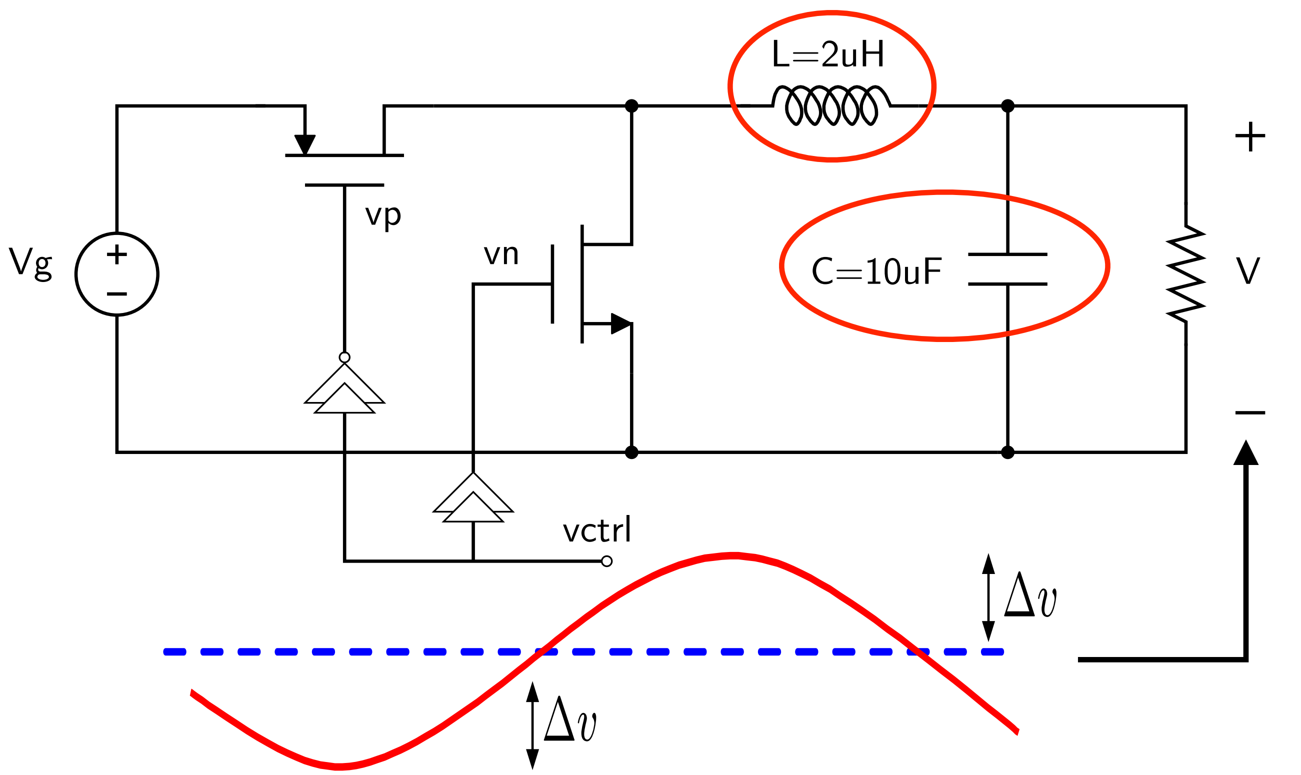
Specification

Δv ≤ 30mV

Regression using cubic polynomial

$$\Delta v \approx f(L,C) = c_{00} + c_{10}L + c_{01}C + c_{11}LC + \ldots + c_{30}L^3 + c_{03}C^3$$

A Simple Buck Converter



Process variations

$L \sim U(1.8, 2.2)\mu H$, $C \sim U(9, 11)\mu F$

Specification

$\Delta v \leq 30mV$

Regression using cubic polynomial

$$\Delta v \approx f(L,C) = c_{00} + c_{10}L + c_{01}C + c_{11}LC + \ldots + c_{30}L^3 + c_{03}C^3$$

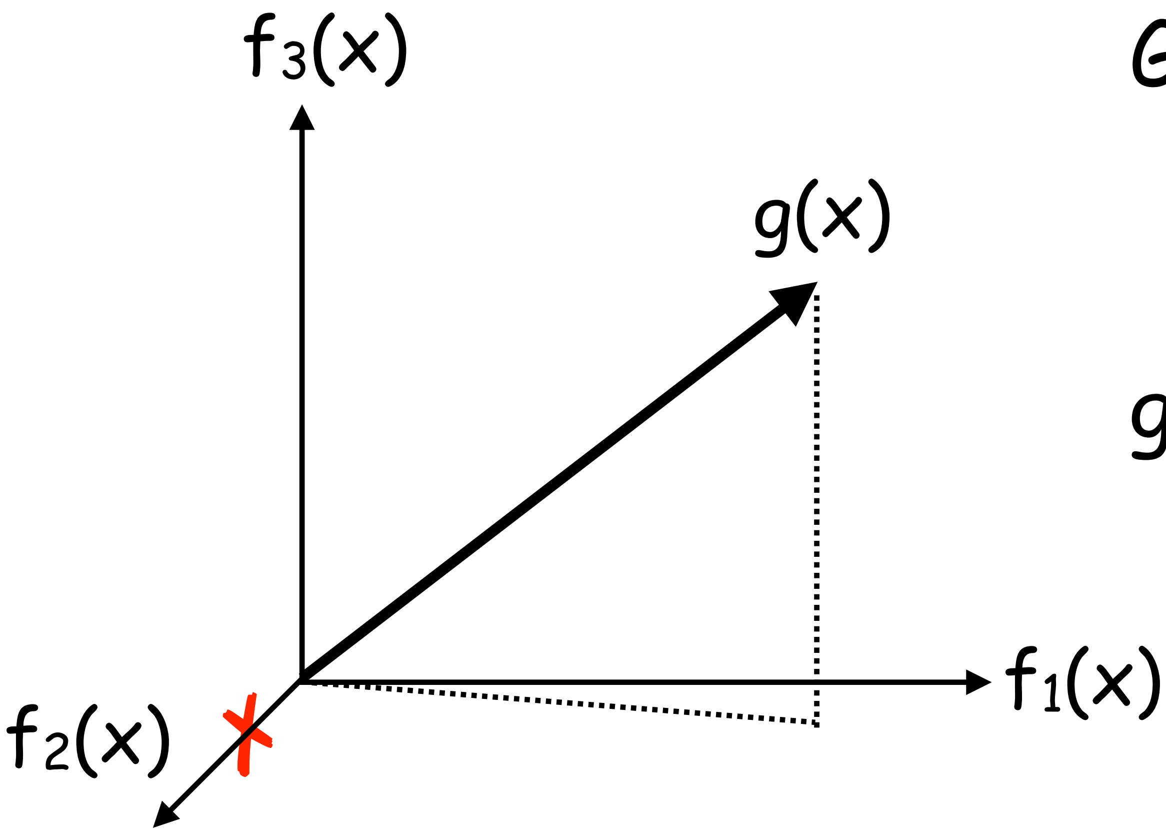
- Ordinary Least squares

$$\Delta v \approx 0.031 - 0.002L - 0.009C - 0.002L^2 - 0.005C^2 + 0.003LC + 0.004L^2C - 0.002LC^2 - 0.000L^3 + 0.010C^3$$

- Proposed sparse method

$$\Delta v \approx 0.028 - 0.001L - 0.001C$$

Sparse Regression



Given a set of basis functions

$$\{f_1(x), f_2(x), f_3(x), \ldots\}$$

$$g(x) = c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x) + \ldots$$

$c_1 f_1(x)$: Important

$c_2 f_2(x)$: Has little contribution

$c_3 f_3(x)$: Important

Sparsity = Predict and drop unimportant terms

Does Sparsity Alone Suffice?

Numerous techniques exist for sparse regression

- For example, LASSO, basis pursuit, etc.
- But is sparsity enough?

Which one of the following two models is better?

Model 1	Model 2
$c_{00} + c_{21}x^2y - c_{12}xy^2 + c_{30}x^3 - c_{04}y^4$	$c_{00} - c_{10}x - c_{01}y$

Sparse Regression: Algorithm Overview

Data matrix

Input: X (m x n), Y(m x 1)

Residual R = Y

Degree	Candidate basis functions	Important basis functions	$g_d(x)$

Construct a set of basis functions of degree-d

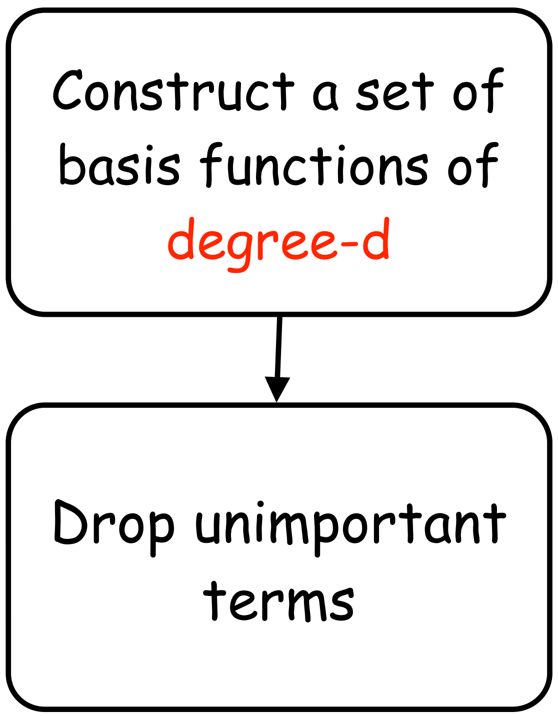
Sparse Regression: Algorithm Overview

Data matrix

Input: X (m x n), Y(m x 1)

Residual R = Y

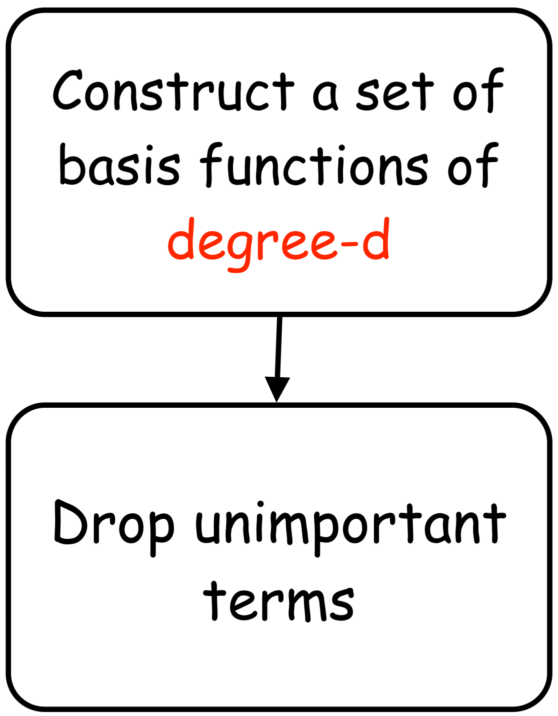
Degree	Candidate basis functions	Important basis functions	$g_d(x)$



Sparse Regression: Algorithm Overview

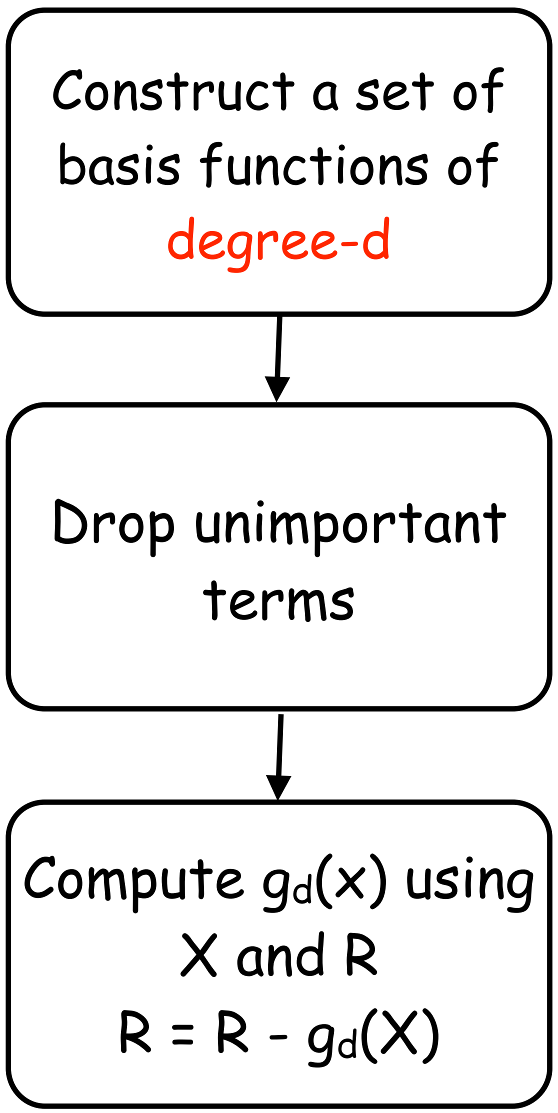
Data matrix
Input: X (m x n), Y(m x 1)
Residual R = Y

Degree	Candidate basis functions	Important basis functions	$g_d(x)$
0	{1}	{1}	c_{00}



Sparse Regression: Algorithm Overview

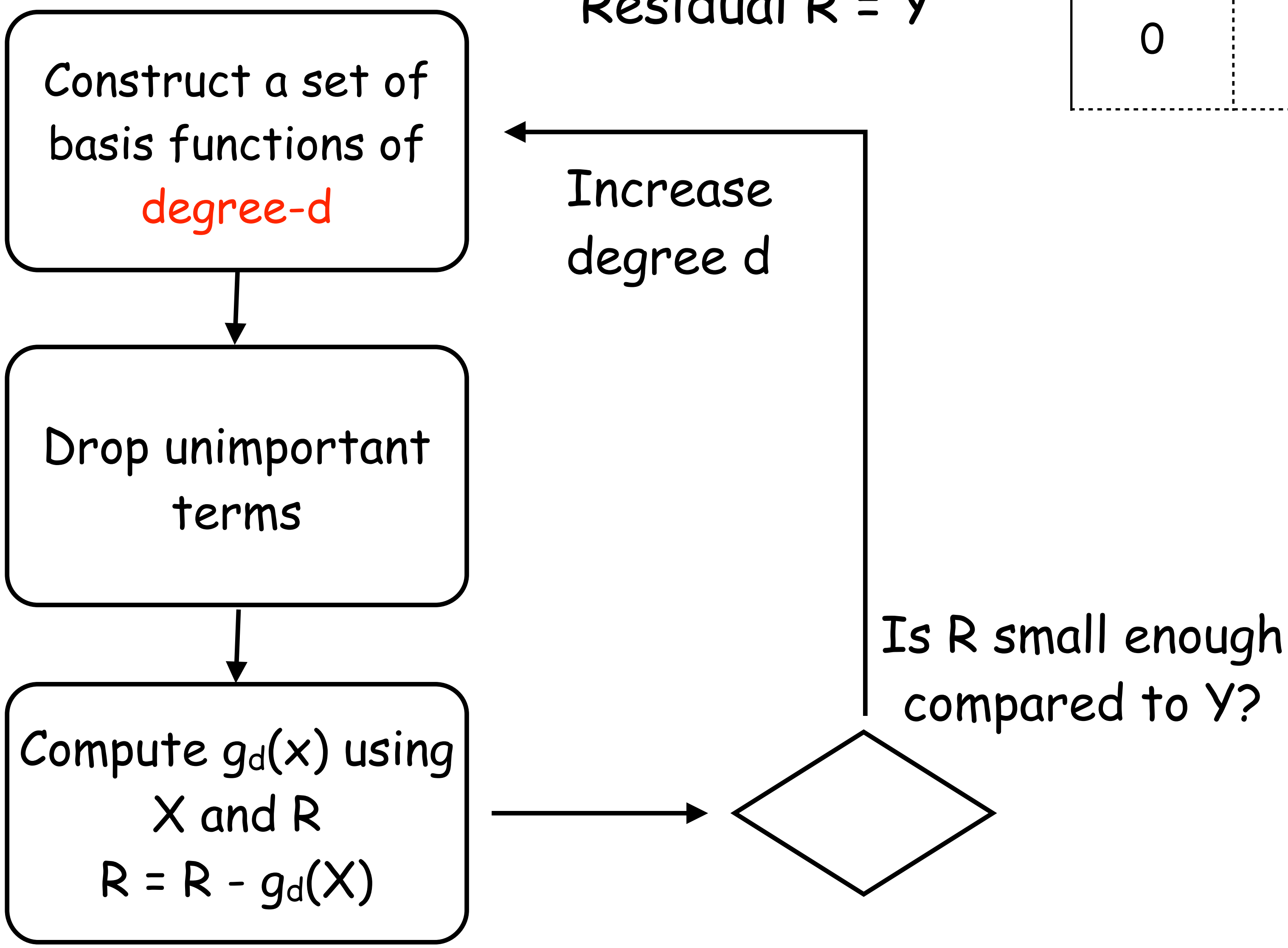
Data matrix
Input: X (m x n), Y(m x 1)
Residual R = Y



Degree	Candidate basis functions	Important basis functions	$g_d(x)$
0	{1}	{1}	c_{00}

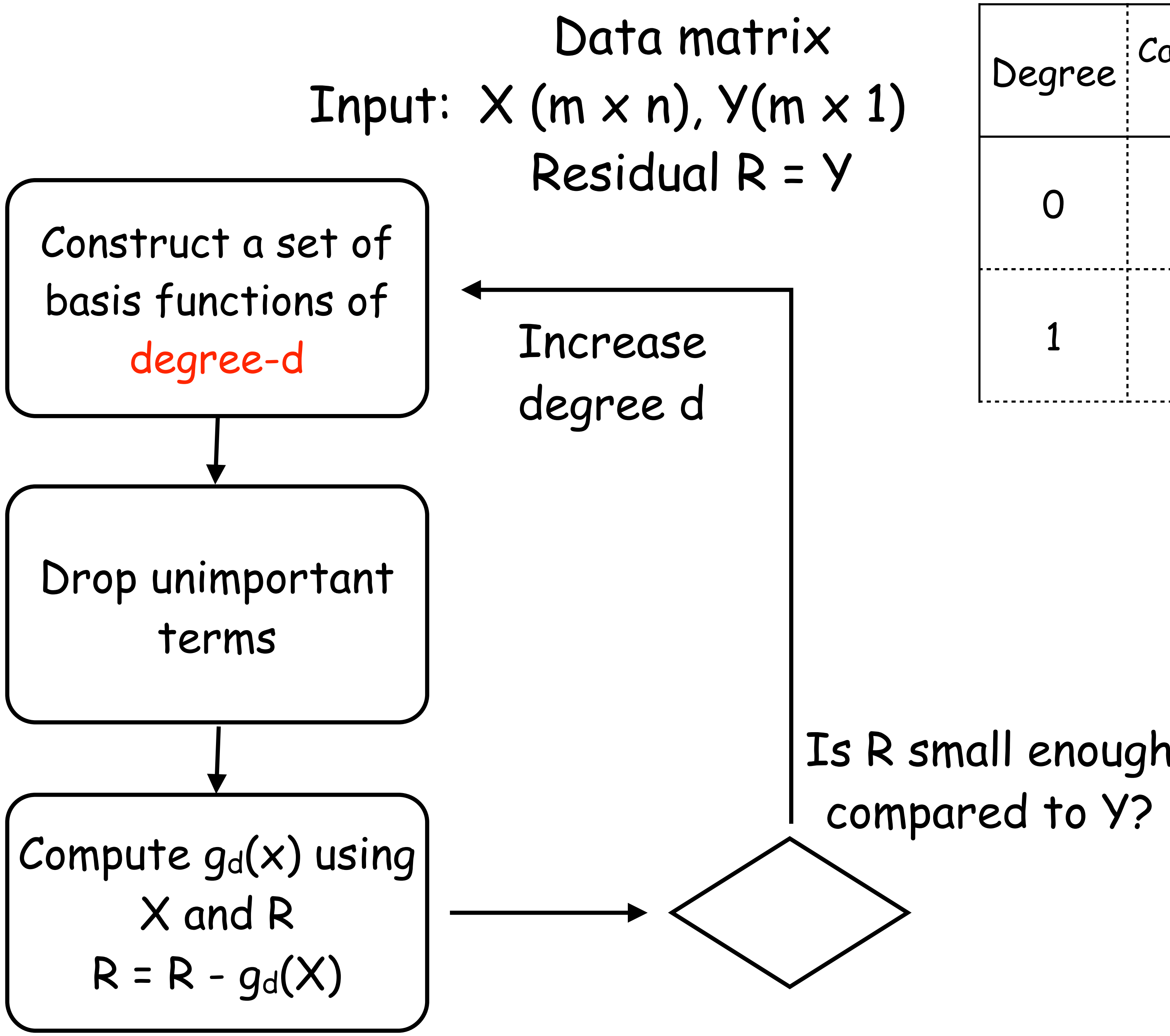
Sparse Regression: Algorithm Overview

Data matrix
Input: X (m x n), Y(m x 1)
Residual R = Y



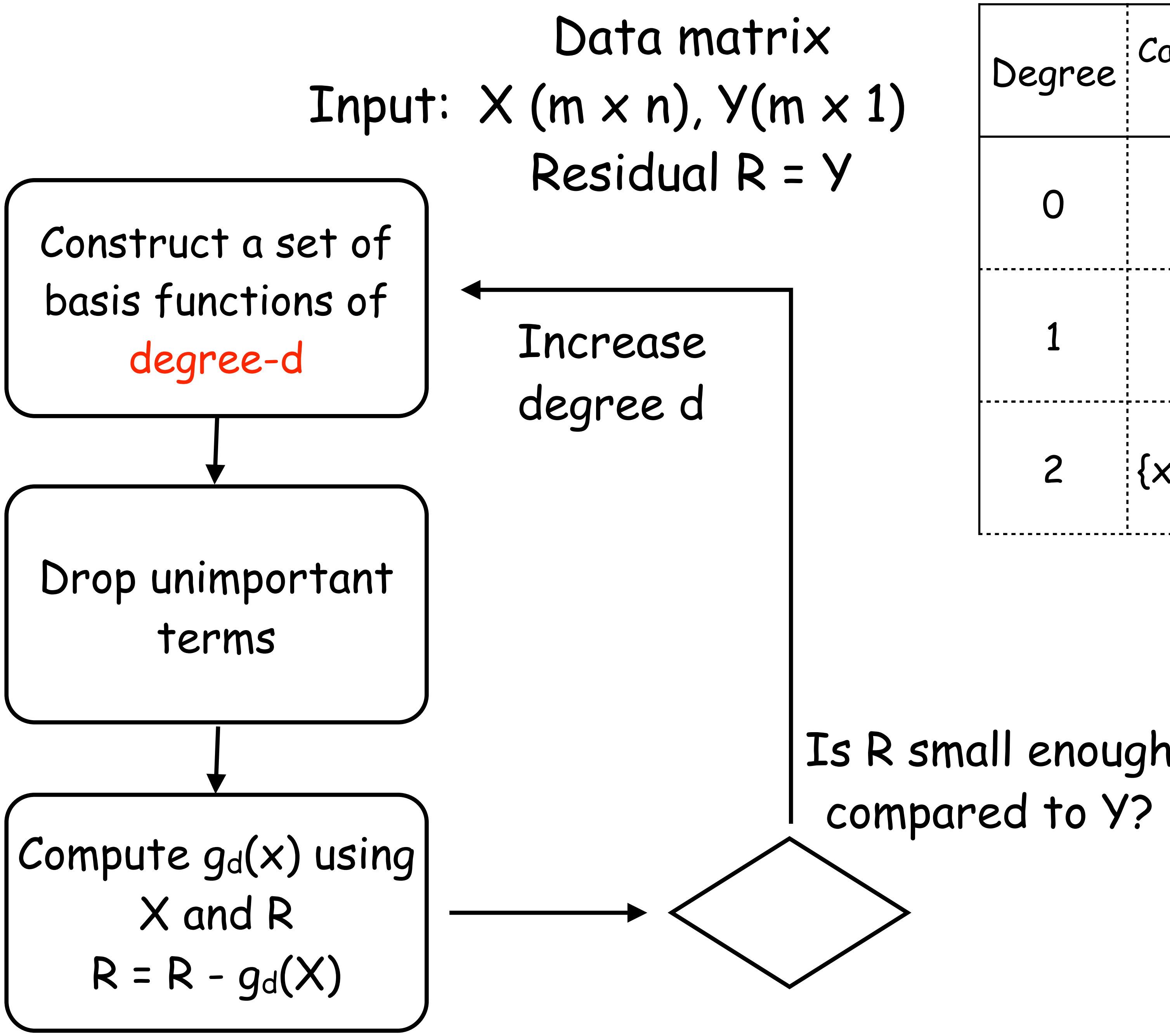
Degree	Candidate basis functions	Important basis functions	$g_d(x)$
0	{1}	{1}	c_{00}

Sparse Regression: Algorithm Overview



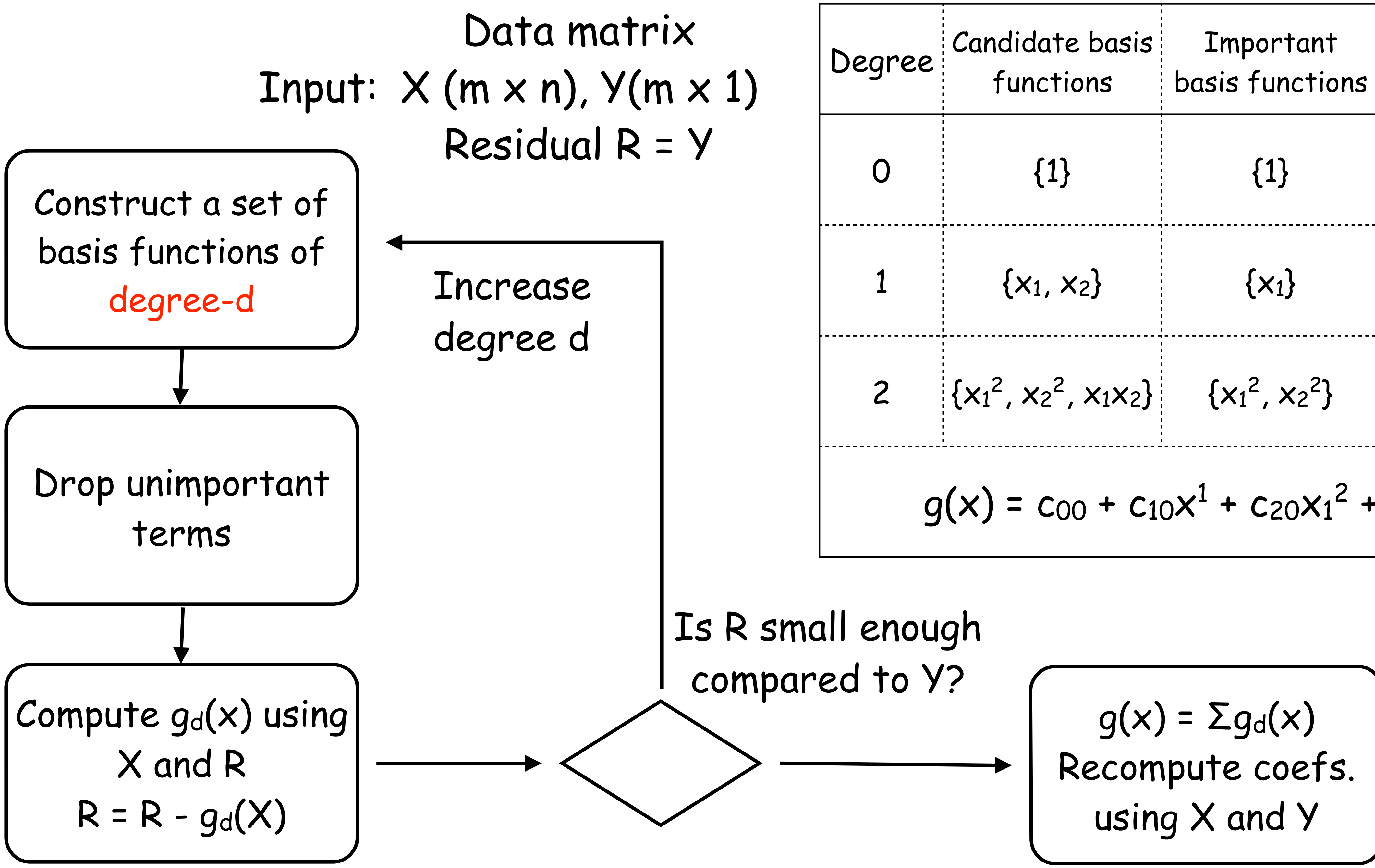
Degree	Candidate basis functions	Important basis functions	$g_d(x)$
0	{1}	{1}	c_{00}
1	$\{x_1, x_2\}$	$\{x_1\}$	$c_{10}x_1$

Sparse Regression: Algorithm Overview



Degree	Candidate basis functions	Important basis functions	$g_d(x)$
0	{1}	{1}	c_{00}
1	$\{x_1, x_2\}$	$\{x_1\}$	$c_{10}x_1$
2	$\{x_1^2, x_2^2, x_1x_2\}$	$\{x_1^2, x_2^2\}$	$c_{20}x_1^2 + c_{02}x_2^2$

Sparse Regression: Algorithm Overview



Degree	Candidate basis functions	Important basis functions	$g_d(x)$
0	$\{1\}$	$\{1\}$	c_{00}
1	$\{x_1, x_2\}$	$\{x_1\}$	$c_{10}x_1$
2	$\{x_1^2, x_2^2, x_1x_2\}$	$\{x_1^2, x_2^2\}$	$c_{20}x_1^2 + c_{02}x_2^2$
$g(x) = c_{00} + c_{10}x^1 + c_{20}x_1^2 + c_{02}x_2^2$			

Sparse Regression: Lower Degree Goes First

$$g(x) = c_{00} + c_{10}x^1 + c_{20}x_1^2 + c_{02}x_2^2$$

Degree	Candidate basis functions	Important basis functions	$g_d(x)$
0	{1}	{1}	c_{00}
1	$\{x_1, x_2\}$	$\{x_1\}$	$c_{10}x_1$
2	$\{x_1^2, x_2^2, x_1x_2\}$	$\{x_1^2, x_2^2\}$	$c_{20}x_1^2 + c_{02}x_2^2$
...	...		

Sparse Regression: Lower Degree Goes First

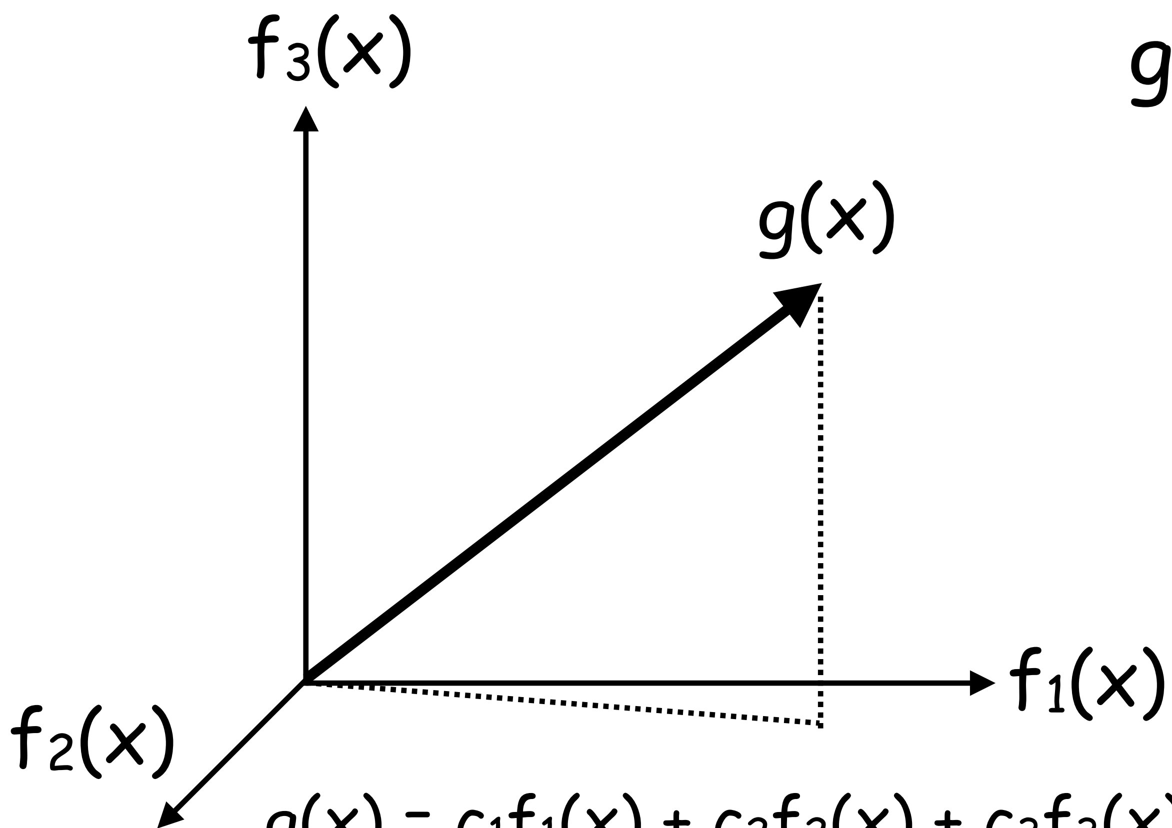
$$g(x) = c_{00} + c_{10}x^1 + c_{20}x_1^2 + c_{02}x_2^2$$

Degree	Candidate basis functions	Important basis functions	$g_d(x)$
0	$\{1\}$	$\{1\}$	c_{00}
1	$\{x_1, x_2\}$	$\{x_1\}$	$c_{10}x_1$
2	$\{x_1^2, x_2^2, x_1x_2\}$	$\{x_1^2, x_2^2\}$	$c_{20}x_1^2 + c_{02}x_2^2$
...	...		

Include higher-degree terms only when necessary

- Easier to interpret
- More efficient to evaluate

Sparse Regression: Importance Estimation via gPC



gPC: generalized polynomial chaos

Distribution	Polynomial	Example
Normal	Hermite	$f_1(x) = x$, $f_2(x) = x^2 - 1$ $f_3(x) = x^3 - 3x$
Uniform	Legendre	$f_1(x) = x$, $f_2(x) = x^2 - 1/3$ $f_3(x) = x^3 - 3/5x$

$g(x) = c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x) + \ldots$

$$\langle f_i(x), f_j(x) \rangle = \int f_i(x) f_j(x) dF_X(x) = \delta_{ij}\gamma_i$$

γ_i → Constant

δ_{ij} → Kronecker delta

$$\langle f_i(x), g(x) \rangle = c_i \gamma_i$$

* Xiu *Numerical Methods for Stochastic Computation: A Spectral Method Approach* '10

Sparse Regression: Importance Estimation via gPC

$$\langle f_i(x), g(x) \rangle = \int f_i(x) g(x) dF_X(x)$$

— Impossible b/c analytical form of g(x) is unknown

Estimation of c_i via Monte-Carlo sampling

$$\int f_i(x) g(x) dF_X(x) \approx \sum_{j=1}^{N} f_i(x^{(j)}) g(x^{(j)})$$

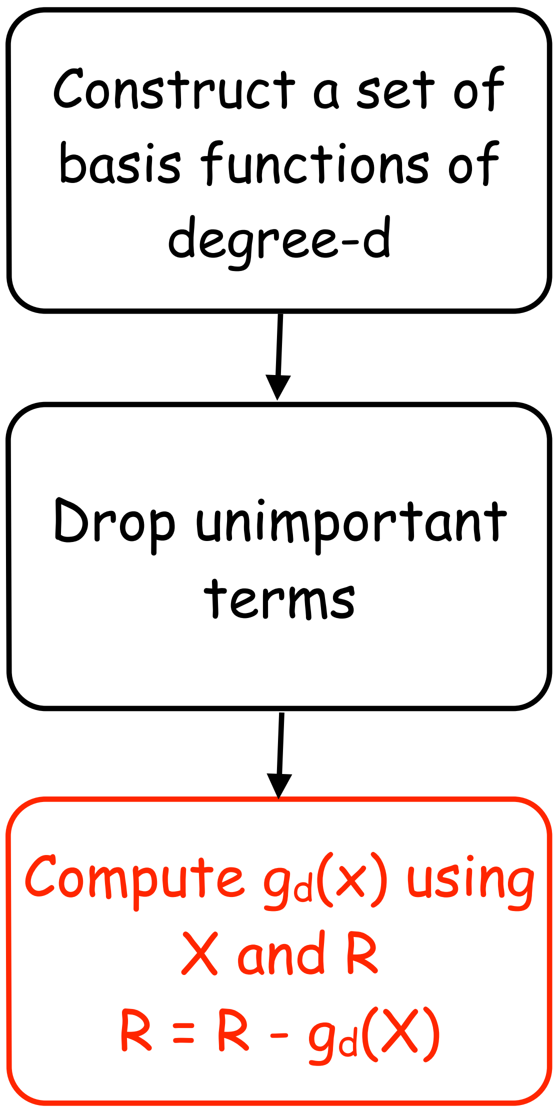
$$c_i \approx \hat{c}_i = \frac{1}{\gamma_i} \sum_{j=1}^{N} f_i(x^{(j)}) g(x^{(j)})$$

$$\left(x^{(1)}, \ldots, x^{(N)}\right) \sim F_X(x)$$

Use the estimation to drop terms if

$$\hat{c}_i \leq \alpha \max(\hat{c}_1, \hat{c}_2, \ldots)$$

Sparse Regression: Single Degree Approximation



Estimated coefficients are not accurate enough

Use regression to improve accuracy, i.e.,

$$\min \Sigma_{i \in [1,m]} (y^{(i)} - g_2(x_1^{(i)}, x_2^{(i)}))$$

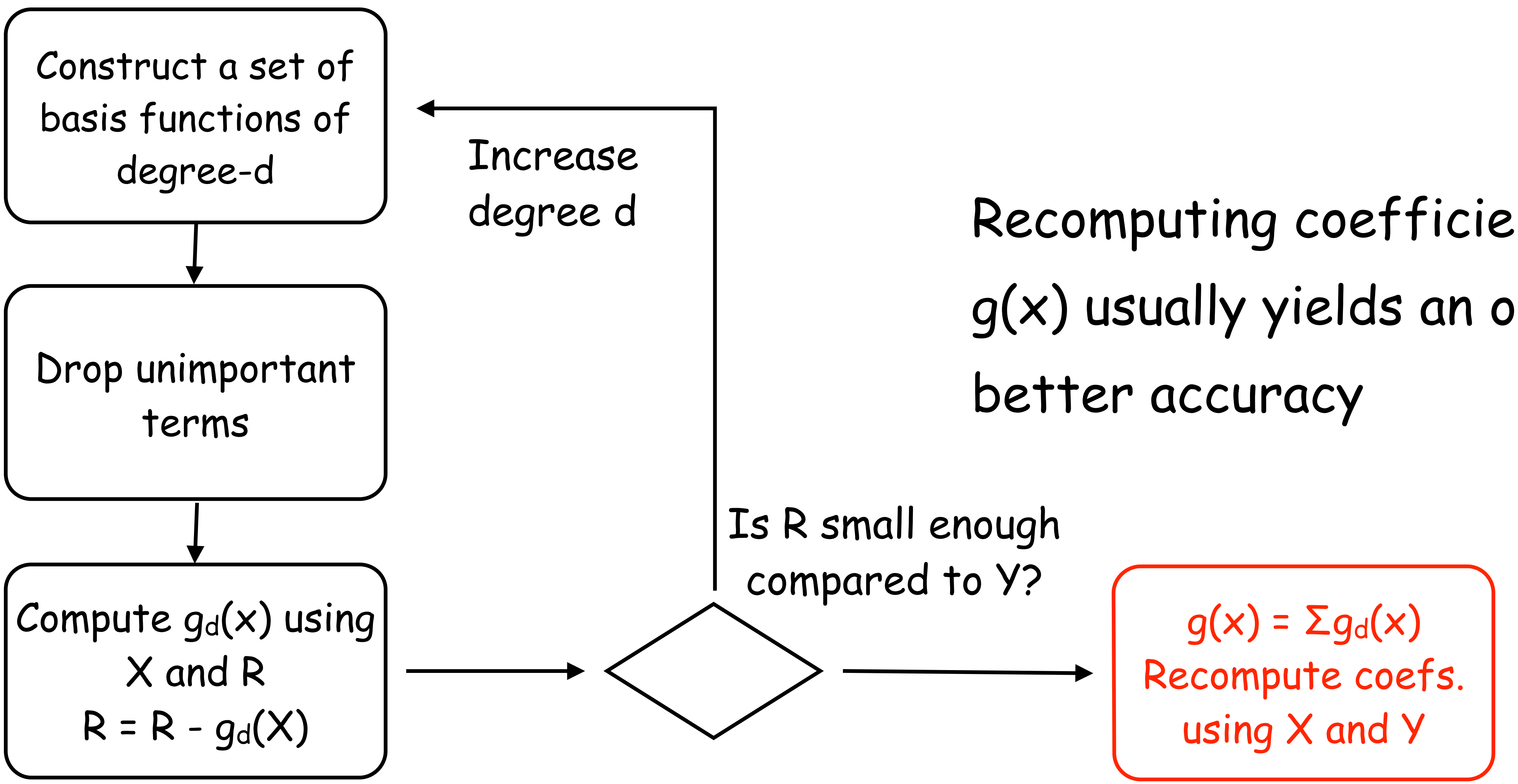
s.t.

$$g_2(x) = c_{20} x_1^2 + c_{02} x_2^2$$

- OLS if there are enough data to avoid over-fitting
- Otherwise, use L1 regularization, e.g., LASSO

* Tibshirani *J. of Royal Stats. Society* '96

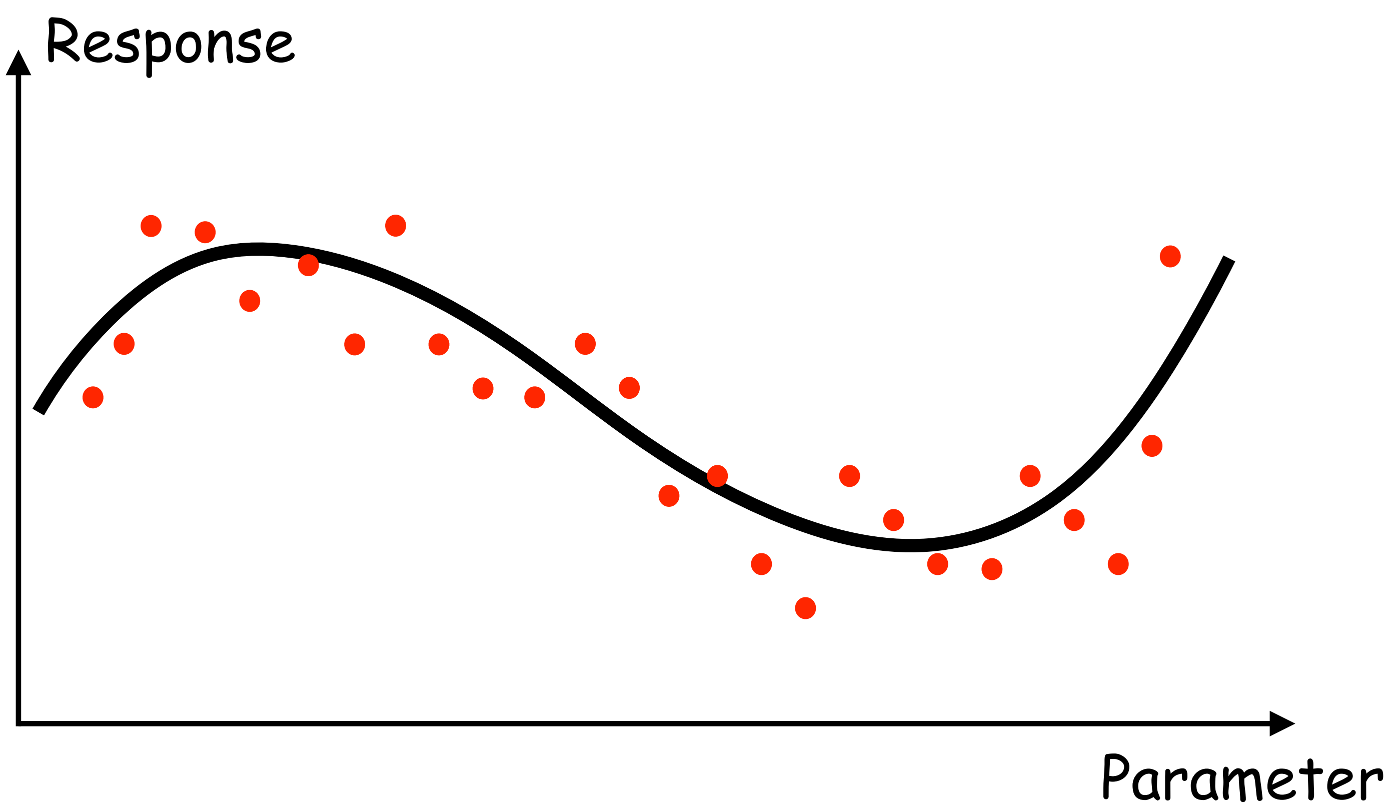
Sparse Regression: Final Approximation



Recomputing coefficients of g(x) usually yields an overall better accuracy

Application: Statistically Sound Model Inference

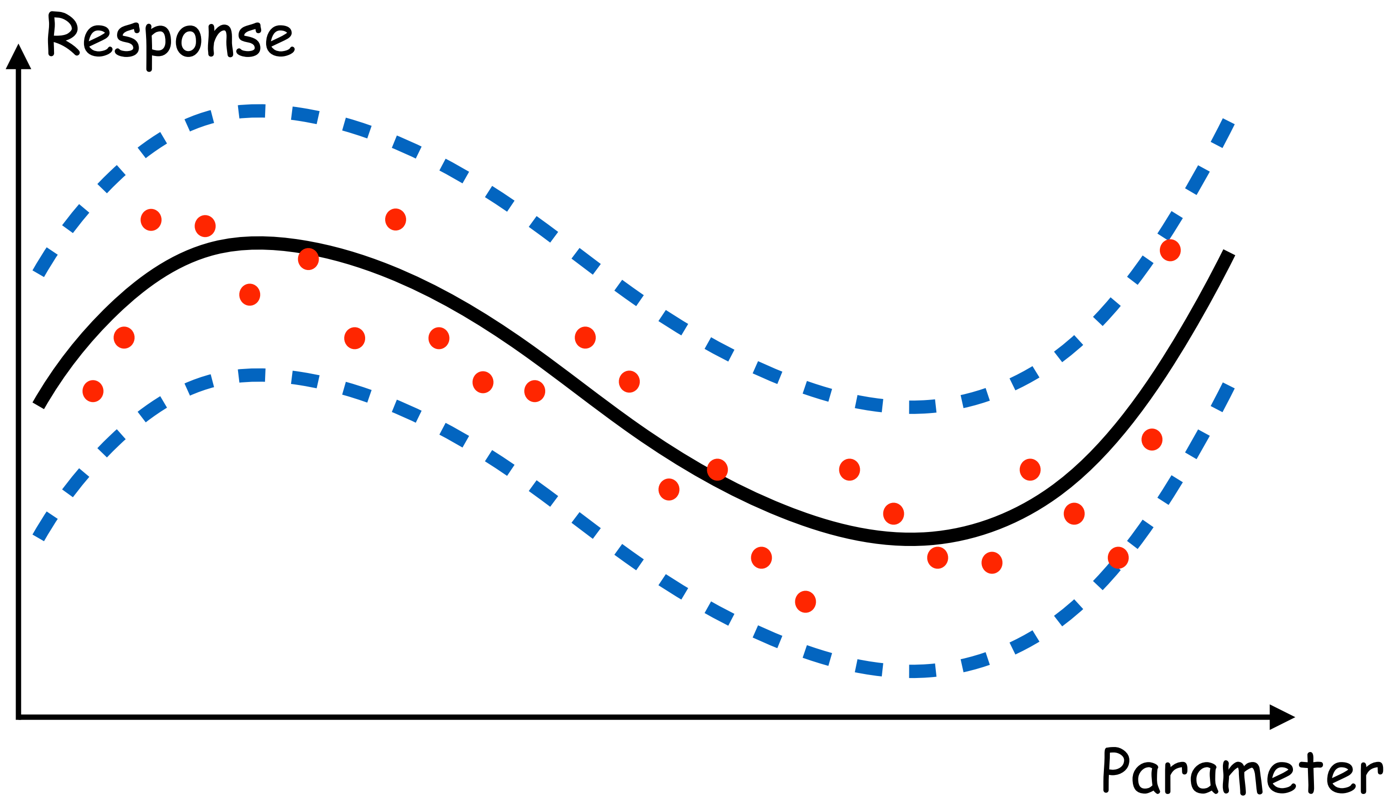
- A simulation-based model inference technique
- Used for statistical verification of analog circuits
- Three phases
  1. Regression
  2. Bloating
  3. Verification



* Zhang et al. *ICCAD*'13

Application: Statistically Sound Model Inference

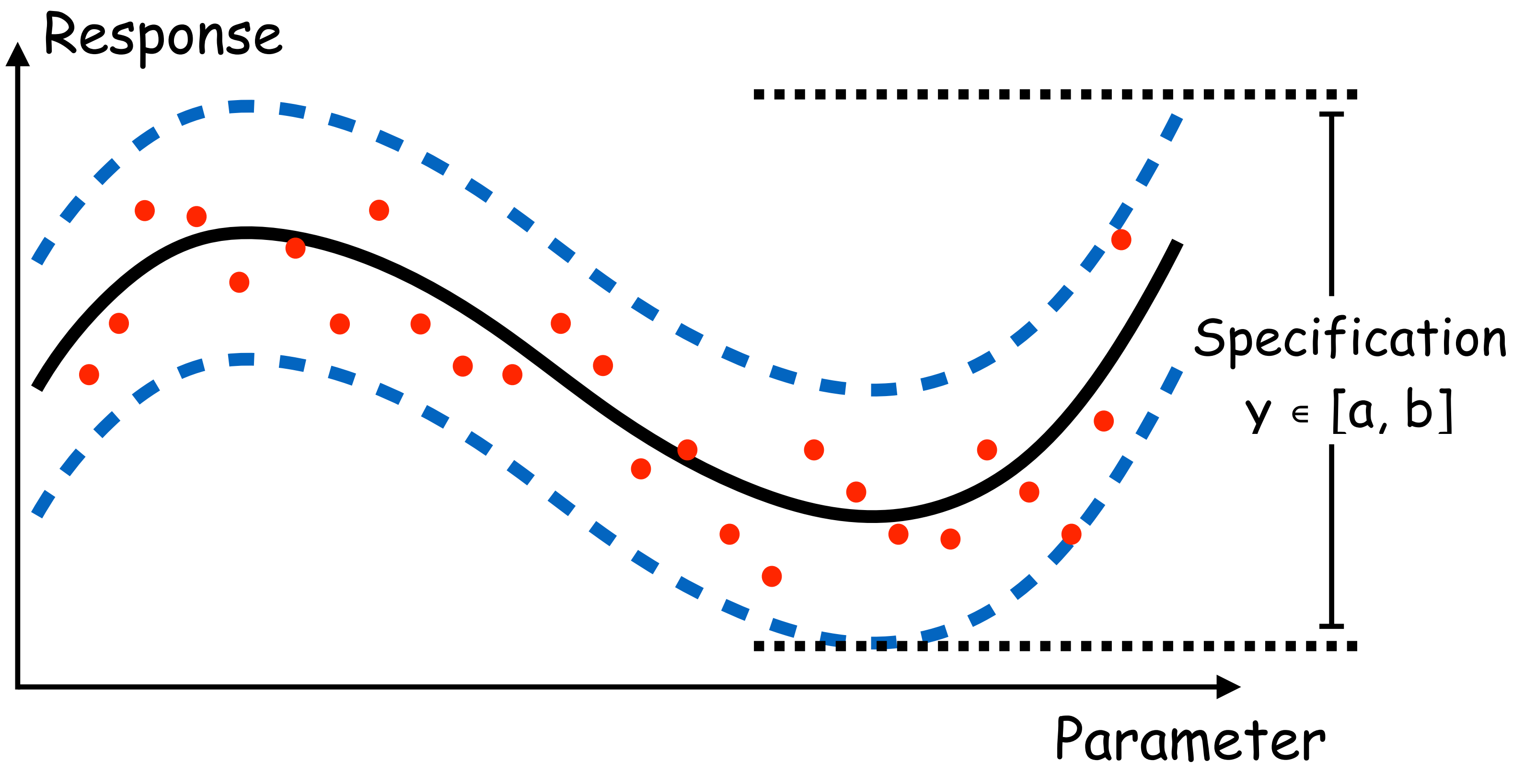
- A simulation-based model inference technique
- Used for statistical verification of analog circuits
- Three phases
  1. Regression
  2. Bloating
  3. Verification



* Zhang et al. *ICCAD'13*

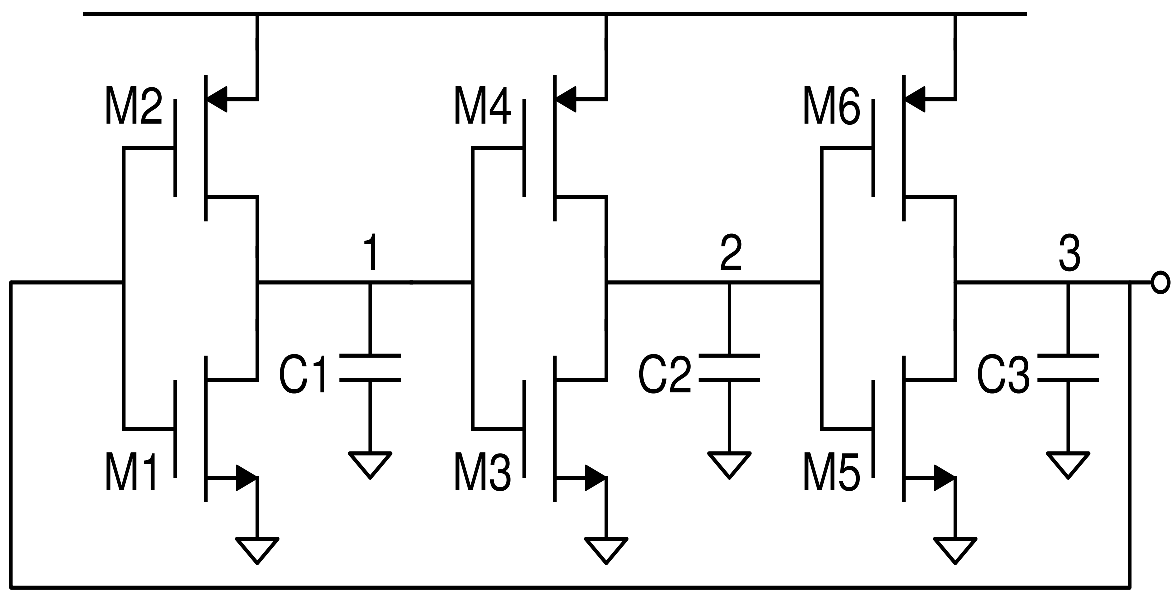
Application: Statistically Sound Model Inference

- A simulation-based model inference technique
- Used for statistical verification of analog circuits
- Three phases
  1. Regression
  2. Bloating
  3. Verification



* Zhang et al. *ICCAD*'13

Experimental Results



Process variations

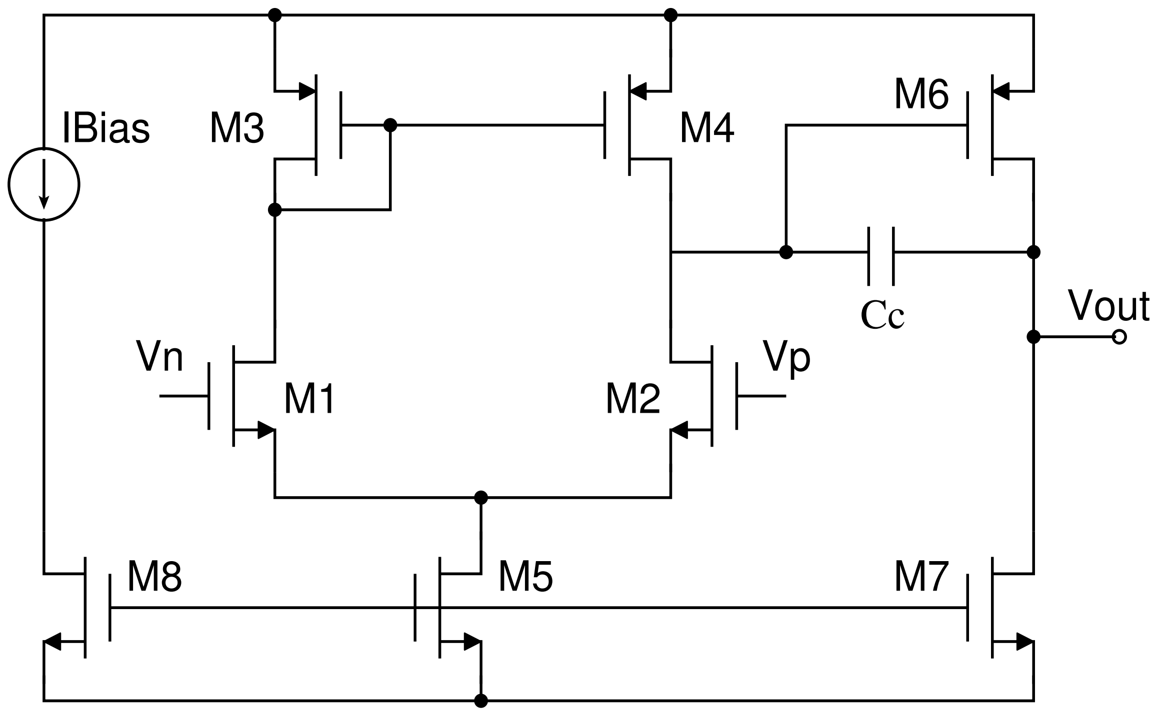
24 normally distributed parameters

Specification

$f_{osc} \in 0.98GHz \pm 50MHz$

10^4-MC	SMI (Degree-3 poly)			Sparse-SMI			
Yield	#Sims	SMI Time	Predicted Yield	#Sims	SMI Time	Degree	Predicted Yield
51%	3213	408s	46%	397	19s	3	45%

Experimental Results



Process variations

32 normally distributed parameters

Specification

1. $V_{os} \leq 50mV$
2. DC-Gain ≥ 60dB
3. Bandwidth ≥ 5MHz

10^4-MC	SMI (Degree-3 poly)			Sparse-SMI			
Yield	#Sims	SMI Time	Predicted Yield	#Sims	SMI Time	Degree	Predicted Yield
61%	8701	4.2h	58%	393	8s	1	58%
65%	8618	4.9h	61%	380	101s	3	60%
100%	8741	5.1h	100%	361	7s	1	100%

Conclusion

- A sparse regression algorithm based on gPC
    - Use limited data to fit response surfaces with many parameters
    - Produce low-degree approximation
- Applied to our statistically sound model inference framework