

10-301/601: Introduction to Machine Learning

Lecture 12 – Regularization

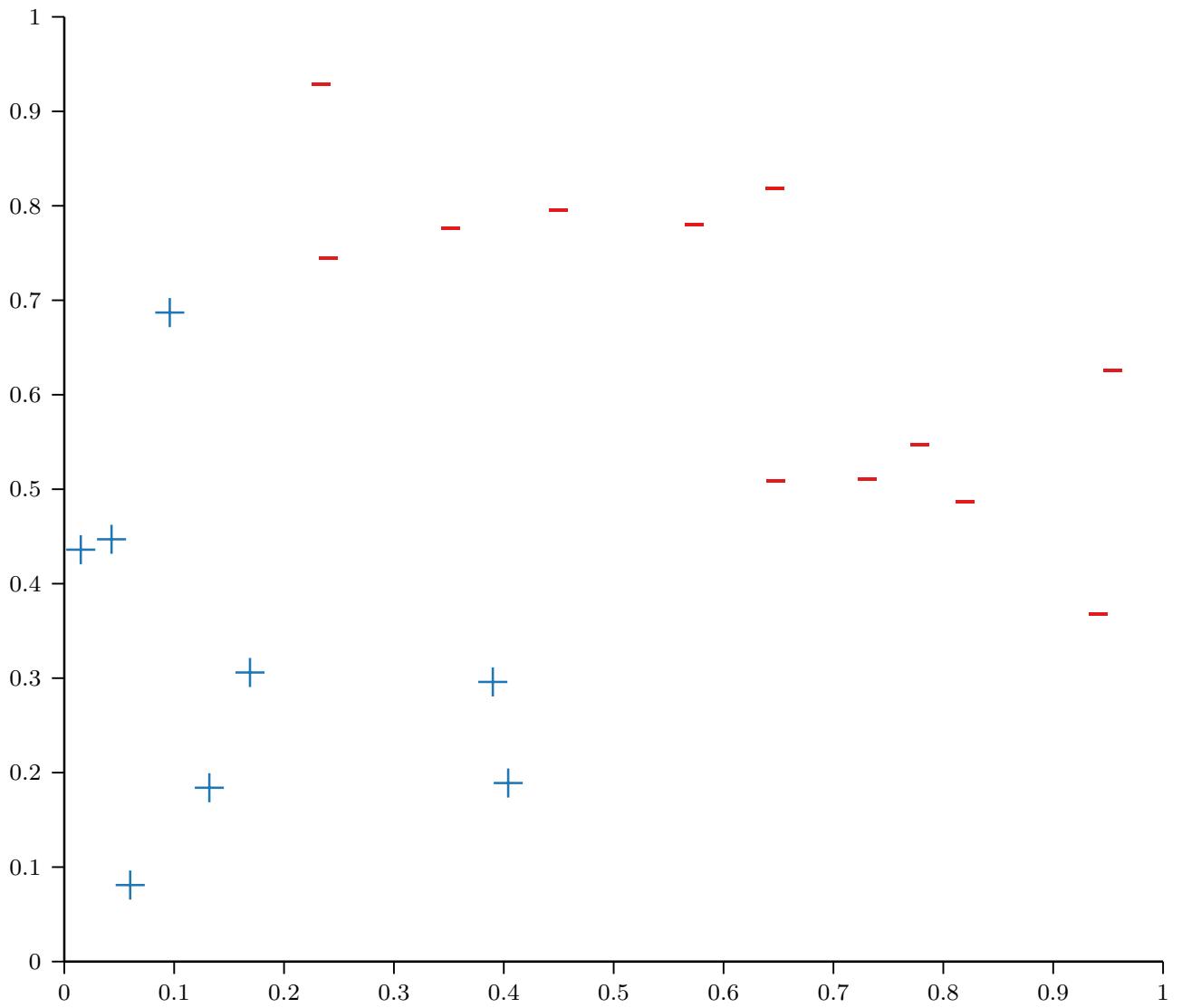
Henry Chai

5/21/25

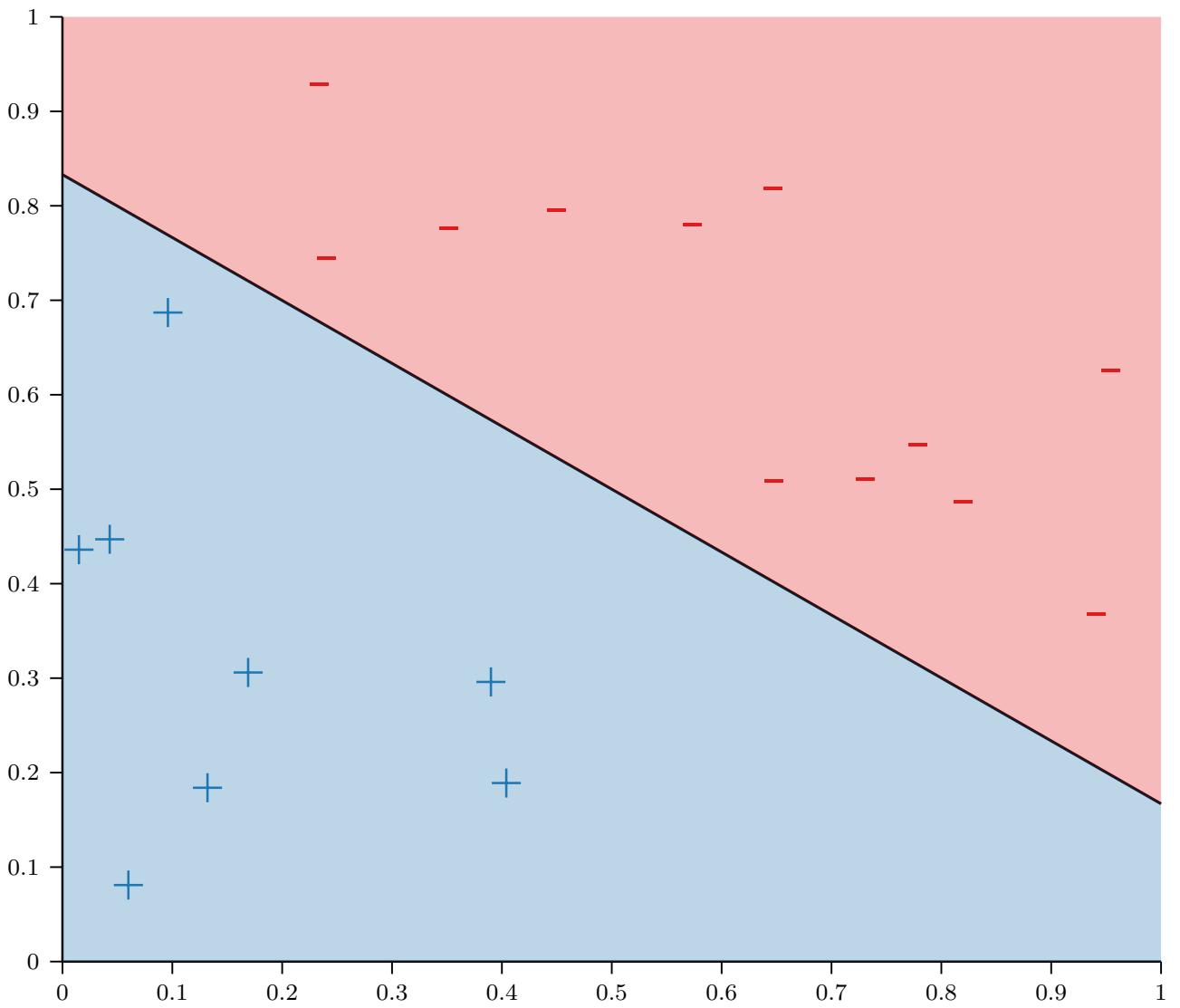
Front Matter

- Announcements:
 - HW3 released on 5/20, due 5/23 at 11:59 PM
 - Quiz 2 on 5/23 at 11:00 AM in BH A36 (here)
 - Study guide solutions to be released this afternoon
 - Midterm on 5/30 at 9:30 AM in BH A36
 - Lectures 1 – 14 are in-scope; **next week's lectures will not be tested on the midterm**

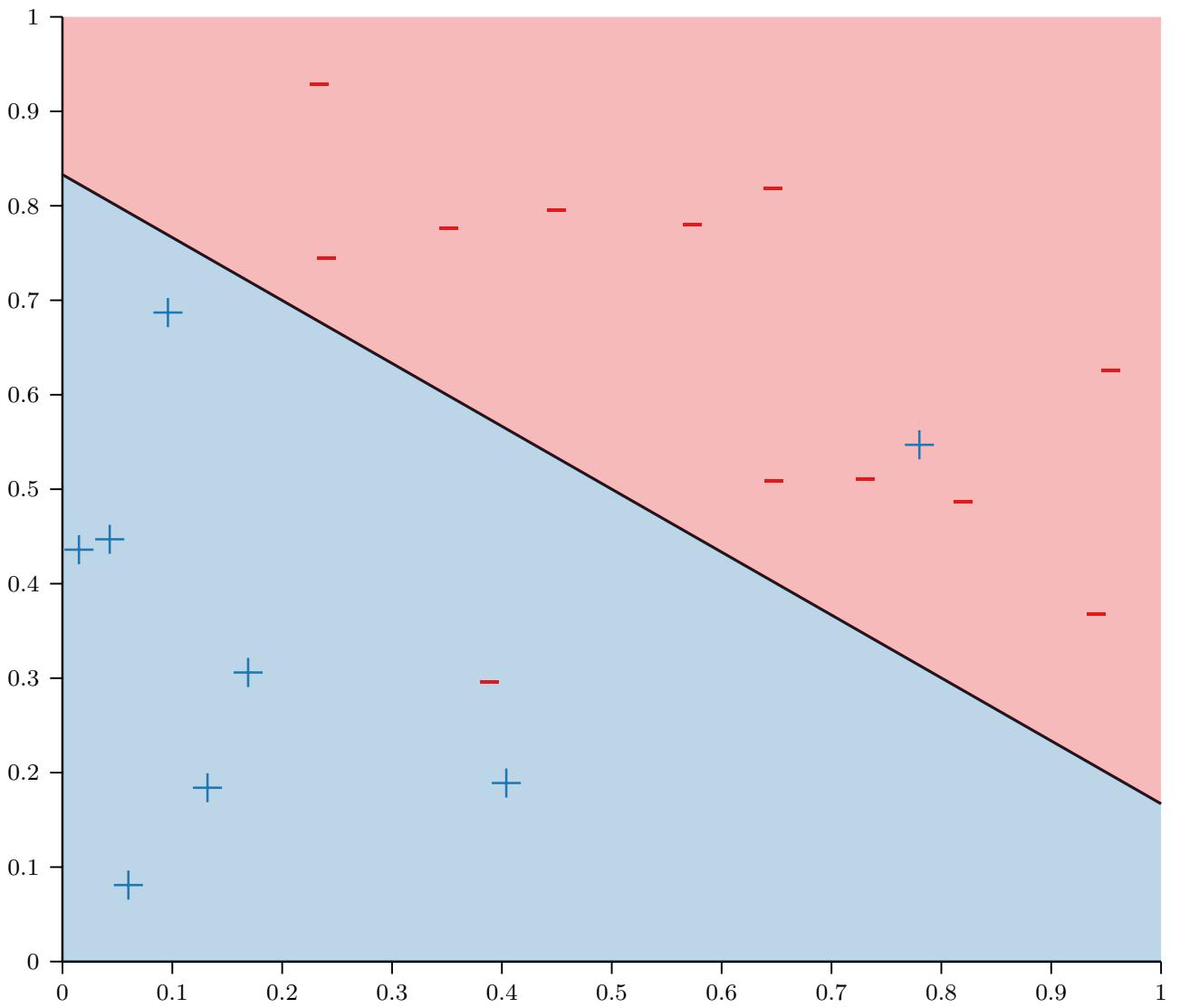
Linear Models



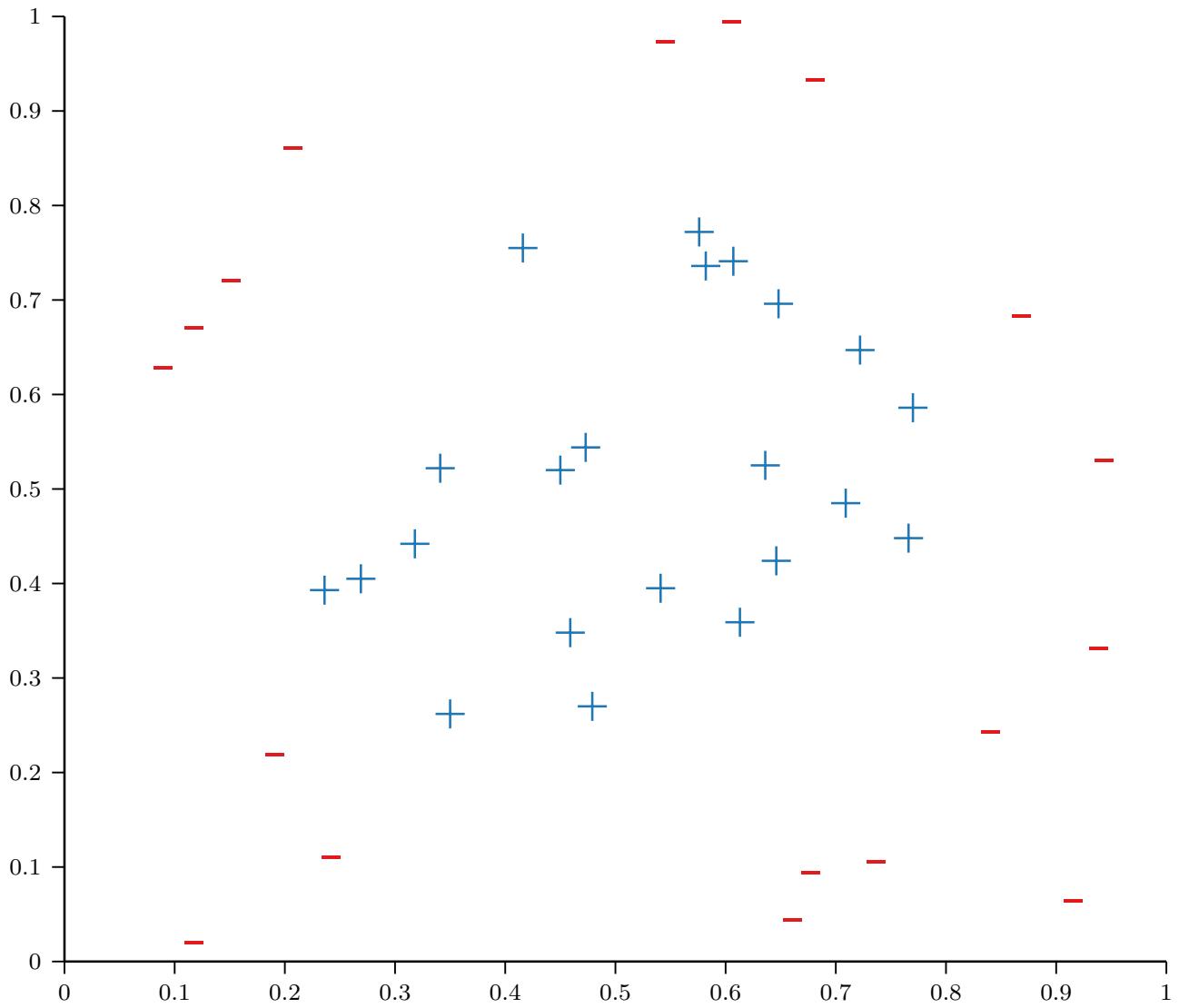
Linear Models



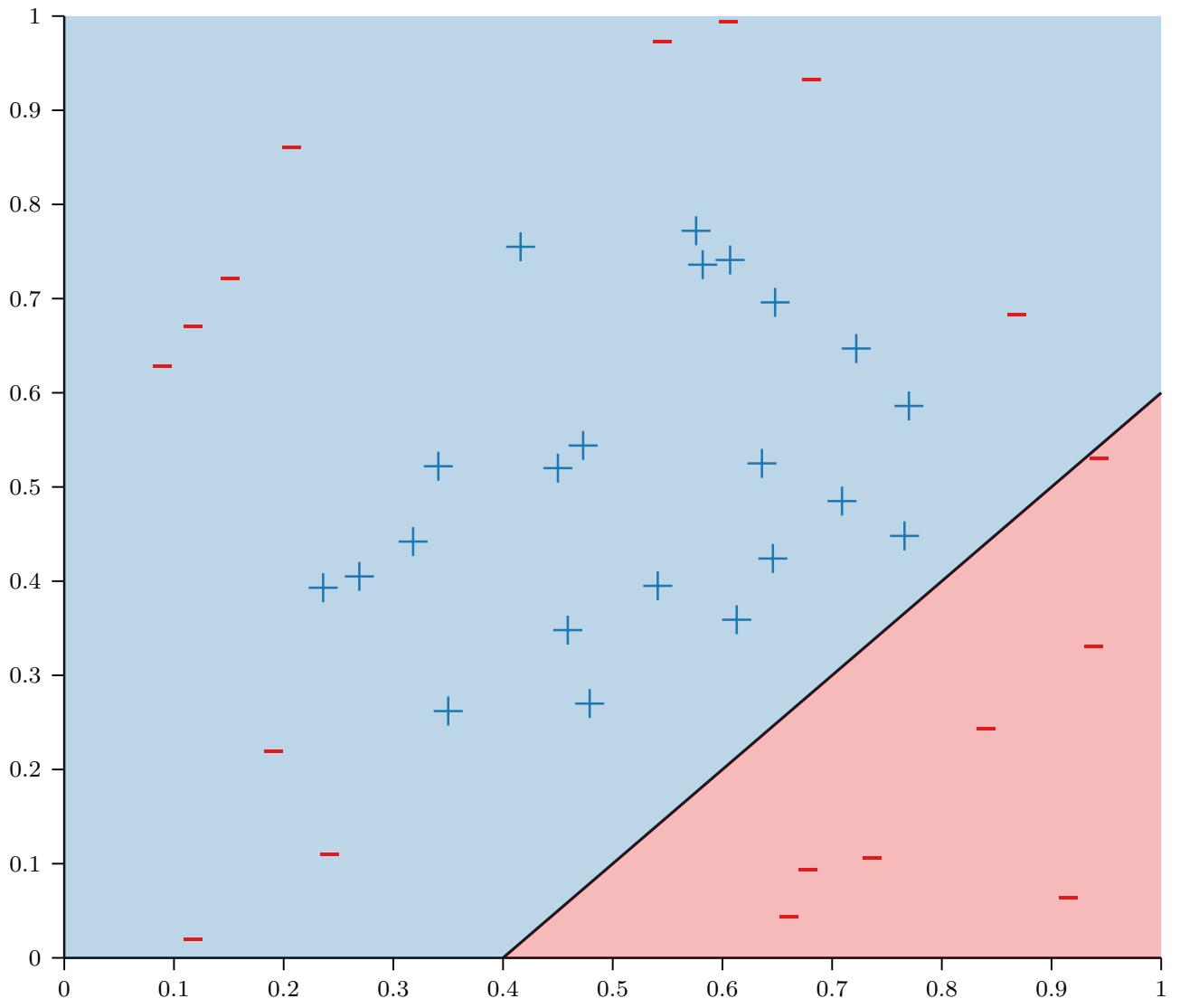
Linear Models



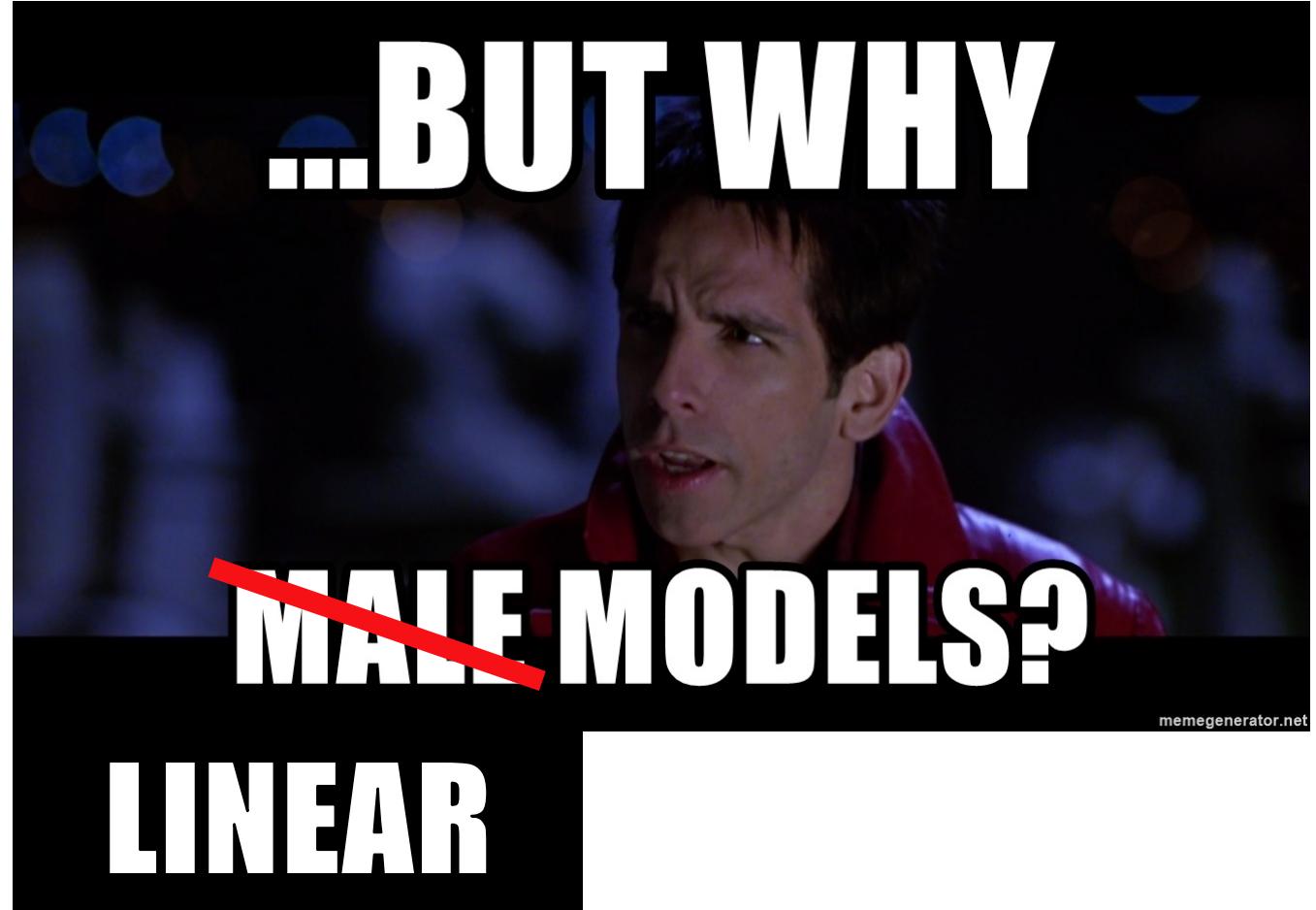
Linear Models?



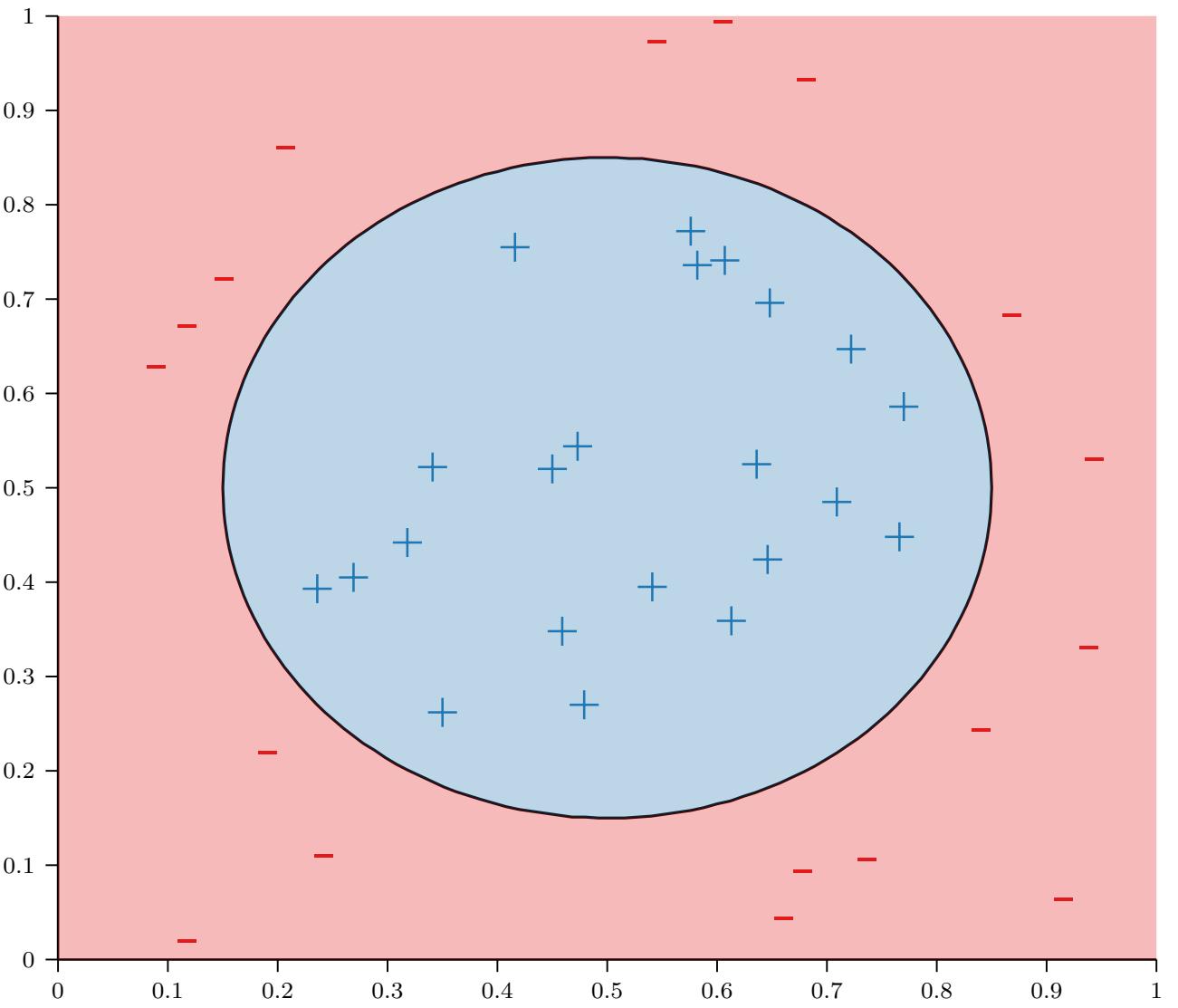
Linear Models?



Linear Models?



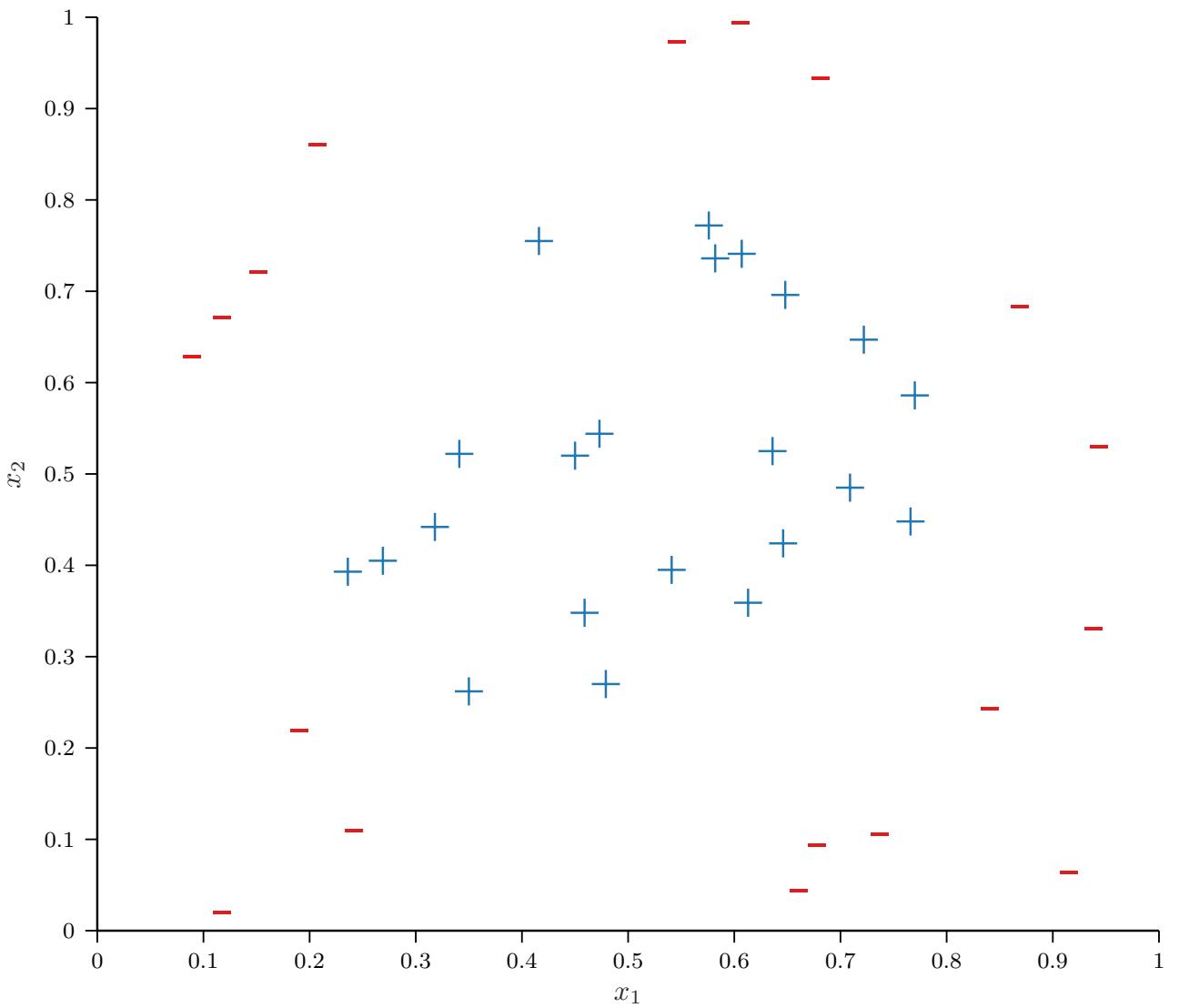
Nonlinear Models



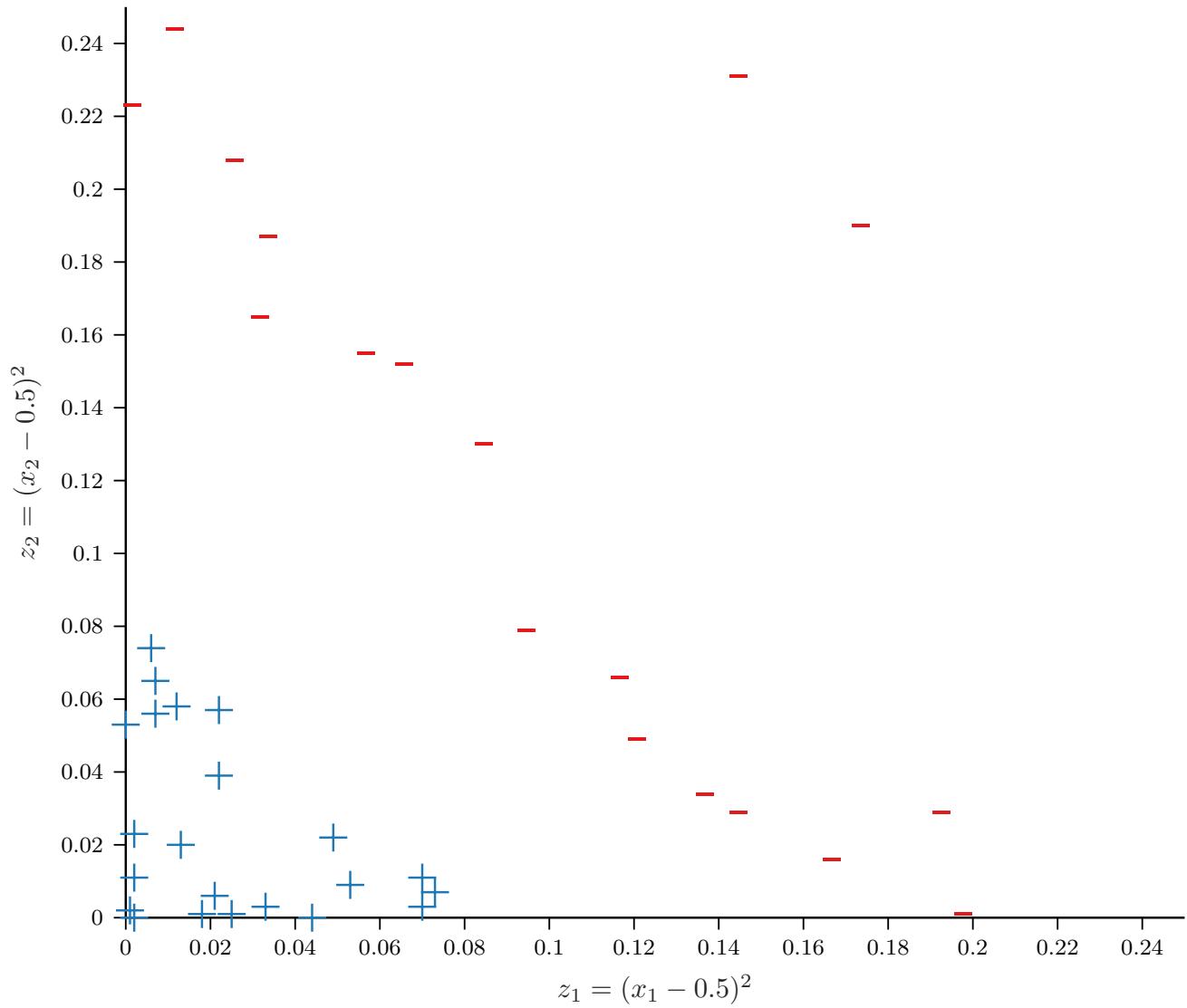
Feature Transforms

- Given D -dimensional inputs $\mathbf{x} = [x_1, \dots, x_D]^T$, first compute some transformation of our input, e.g.,
$$\phi([x_1, x_2]^T) = [z_1 = (x_1 - 0.5)^2, z_2 = (x_2 - 0.5)^2]$$

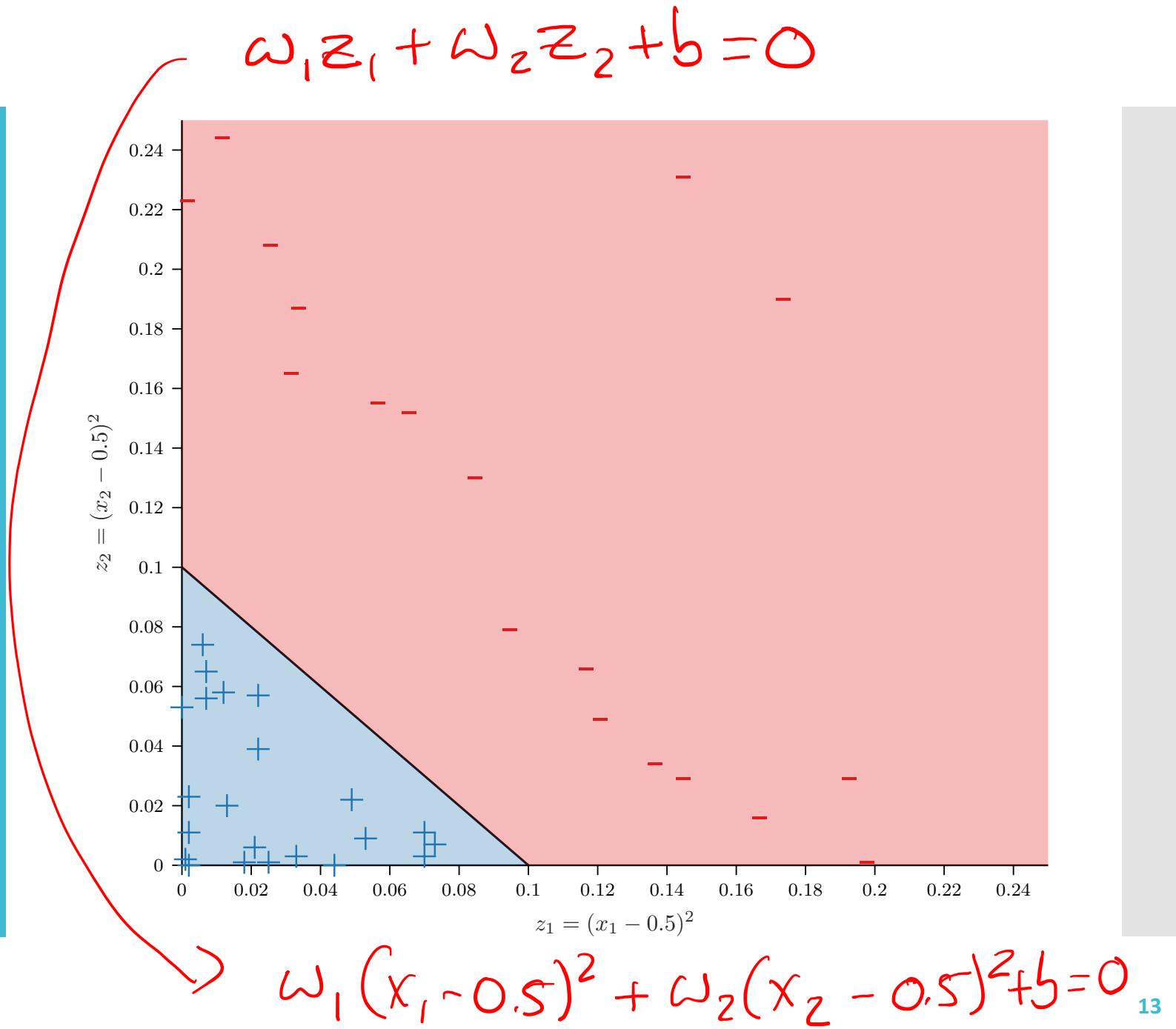
Nonlinear Models



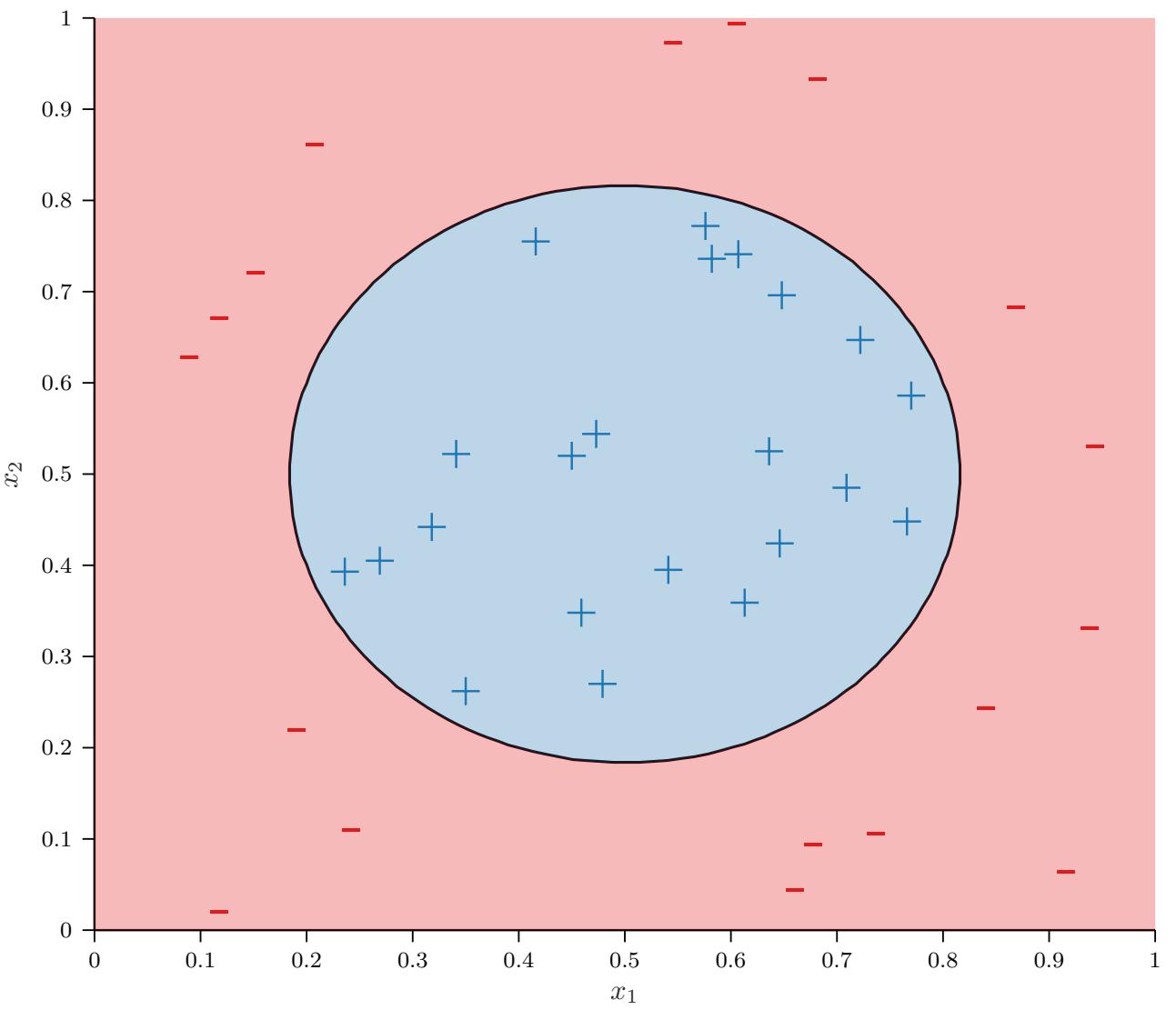
Nonlinear Models



Nonlinear Models



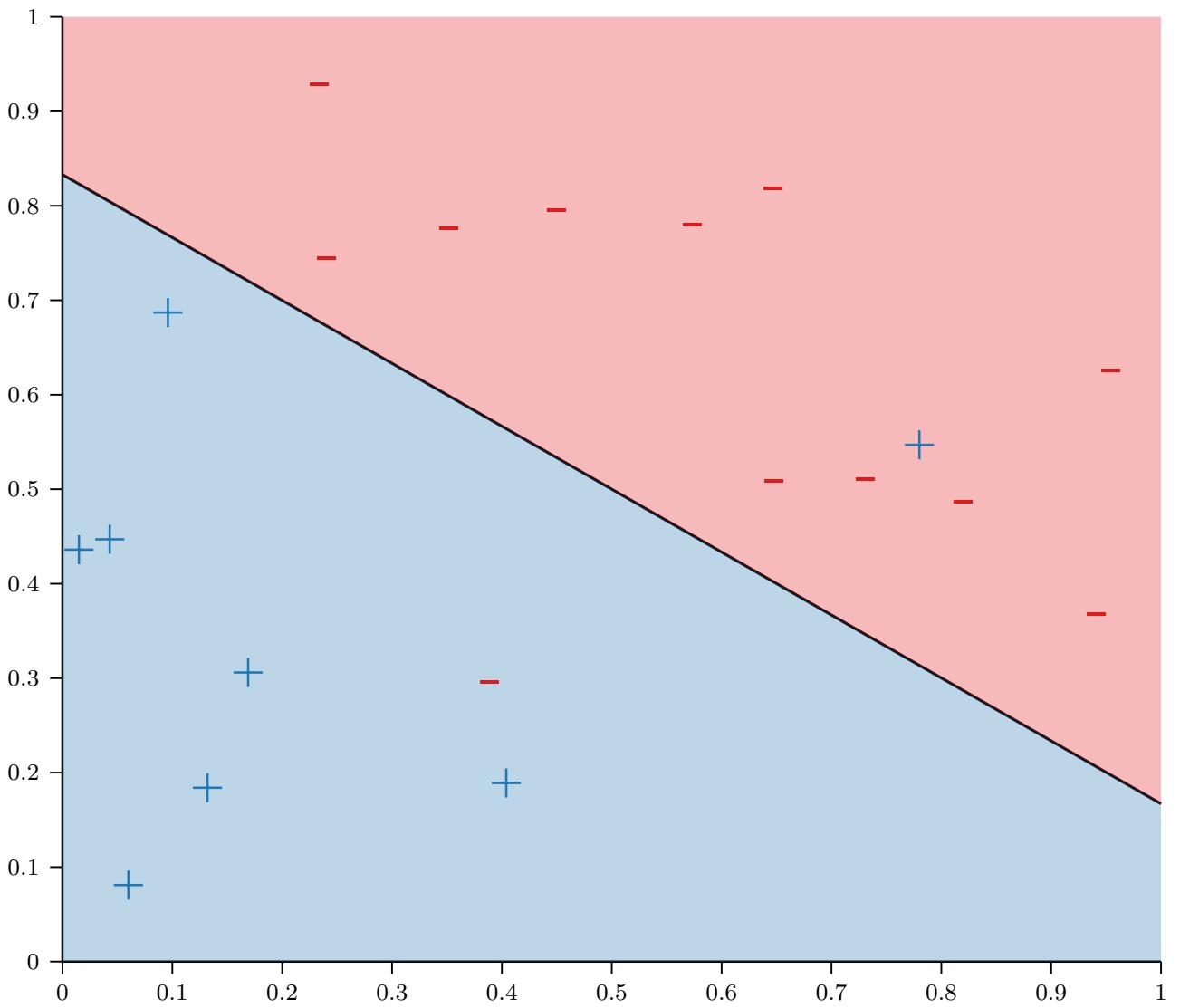
Nonlinear Models



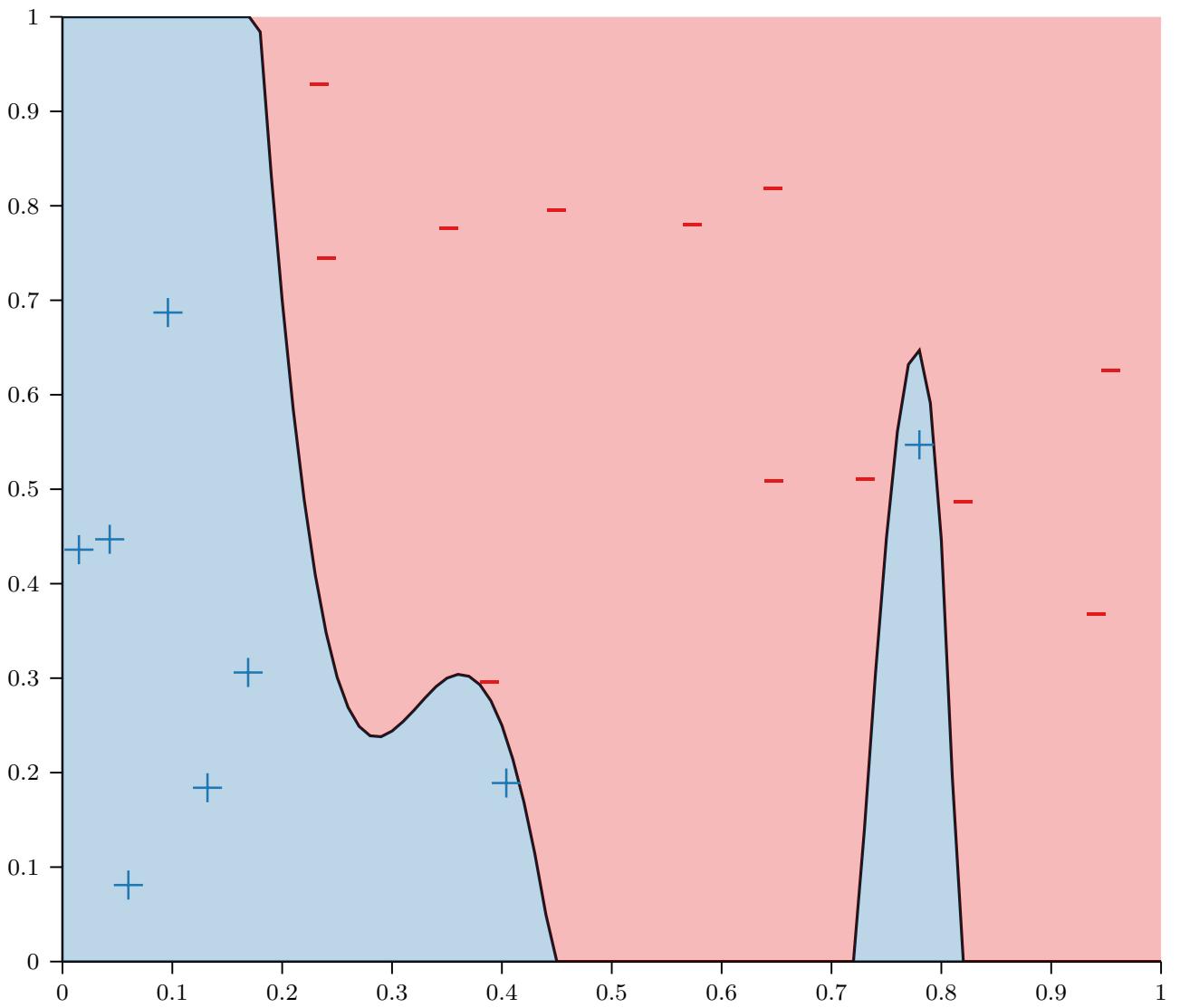
General Q^{th} -order Transforms

- $\phi_{2,2}([x_1, x_2]^T) = [x_1, x_2, x_1^2, x_1 x_2, x_2^2]^T$
- $\phi_{2,3}([x_1, x_2]^T) = [x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3]^T$
- $\phi_{2,4}([x_1, x_2]^T) = [x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3, x_1^4, x_1^3 x_2, x_1^2 x_2^2, x_1 x_2^3, x_2^4]^T$
- $\phi_{2,Q}$ maps a 2-dimensional input to a $\frac{Q(Q+3)}{2}$ -dimensional output
- Scales even worse for higher-dimensional inputs...

Linear Models



Nonlinear Models?



Feature Transforms: Tradeoffs

	Low-Dimensional Input Space	High-Dimensional Input Space
Training Error	High	Low
Generalization	Good	Bad

overfitting

Feature Transforms: Experiment

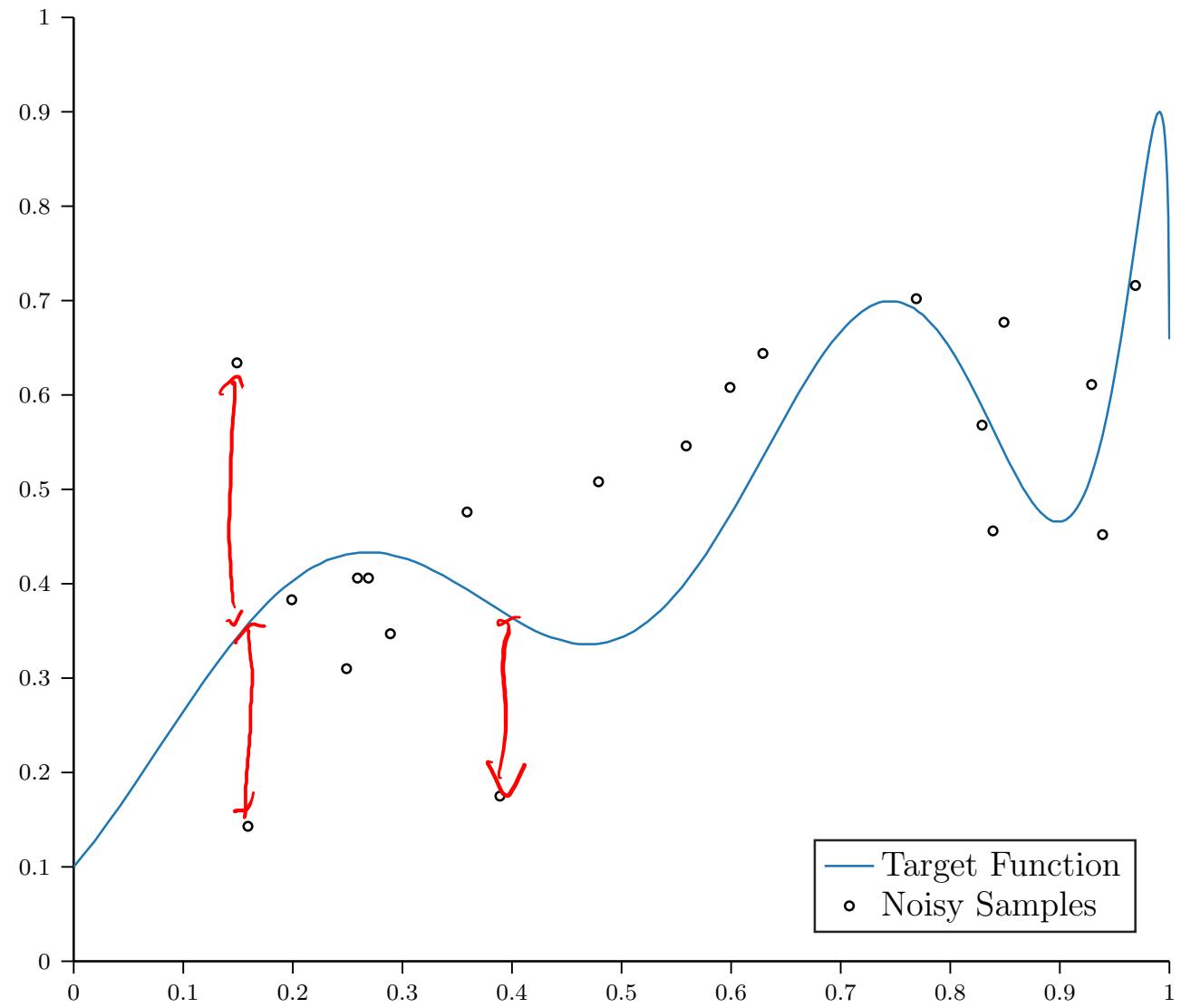
- $x \in \mathbb{R}$, $y \in \mathbb{R}$ and $N = 20$
- Targets are generated by a 10th-order polynomial in x with additive Gaussian noise:

$$y = \sum_{d=0}^{10} a_d x^d + \epsilon \text{ where } \epsilon \sim N(0, \sigma^2)$$

- $\mathcal{H}_2 = 2^{\text{nd}}$ -order polynomials
 - $\phi_{1,2}(x) = [x, x^2]^T$
- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomials
 - $\phi_{1,10}(x) = [x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}]^T$

Noisy Targets

- 10-dimensional target function with additive Gaussian noise
- $\mathcal{H}_2 = 2^{\text{nd}}\text{-order polynomial}$
- $\mathcal{H}_{10} = 10^{\text{th}}\text{-order polynomial}$



0 surveys completed



0 surveys underway

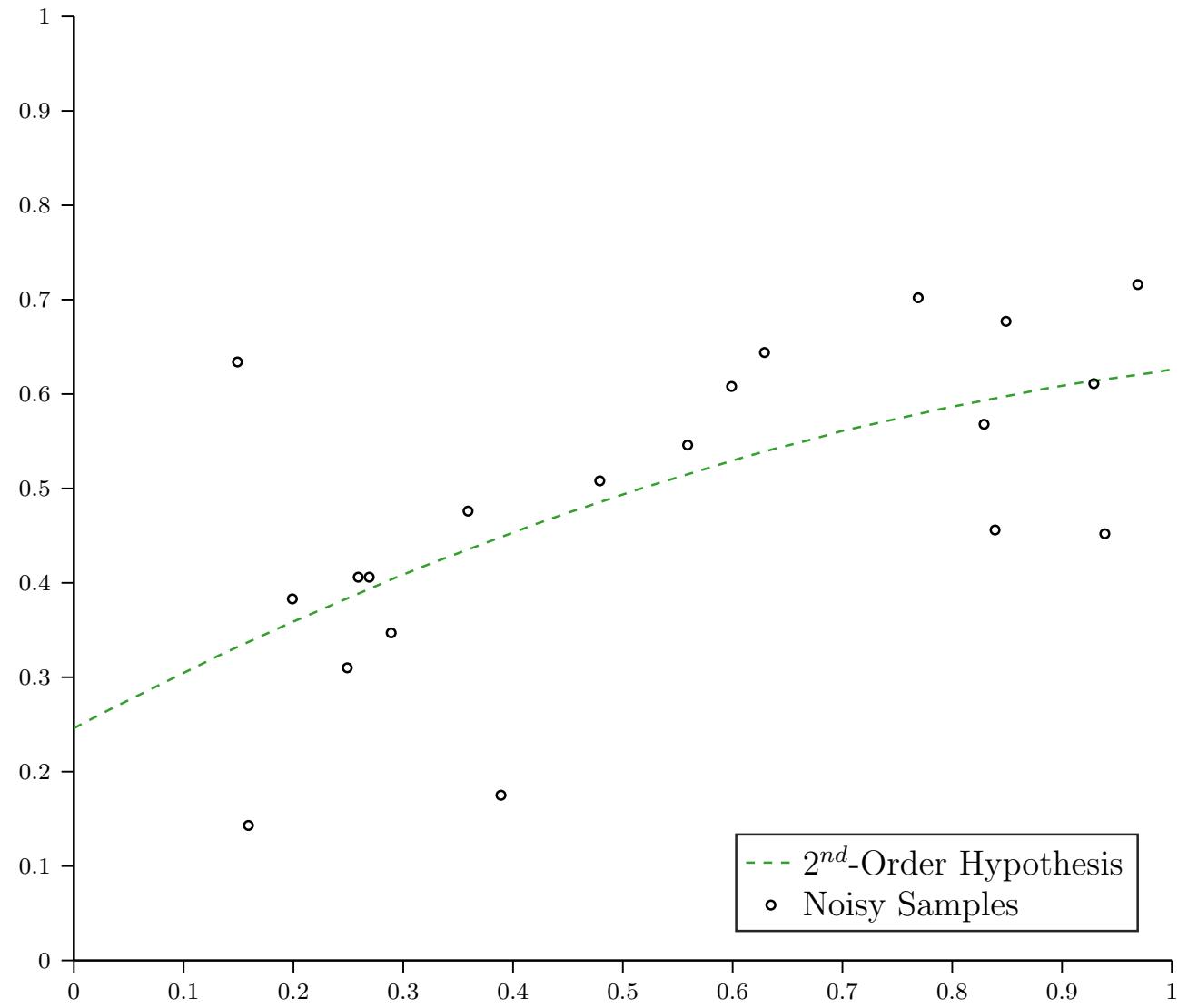
Which model do you think will have a lower true error in this setting ($N = 20$)?

\mathcal{H}_2

\mathcal{H}_{10}

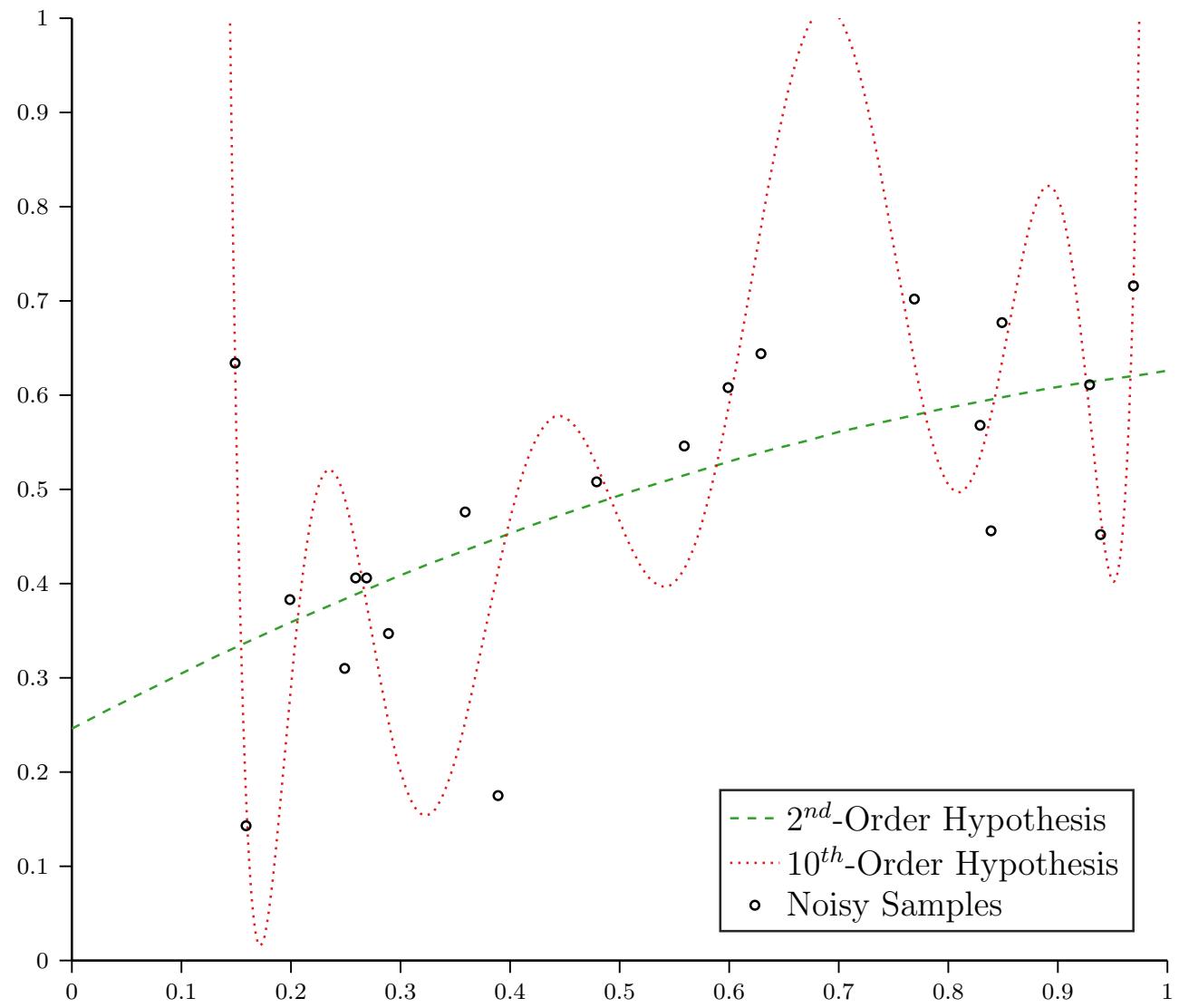
Noisy Targets

- 10-dimensional target function with additive Gaussian noise
- $\mathcal{H}_2 = 2^{\text{nd}}$ -order polynomial
- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomial



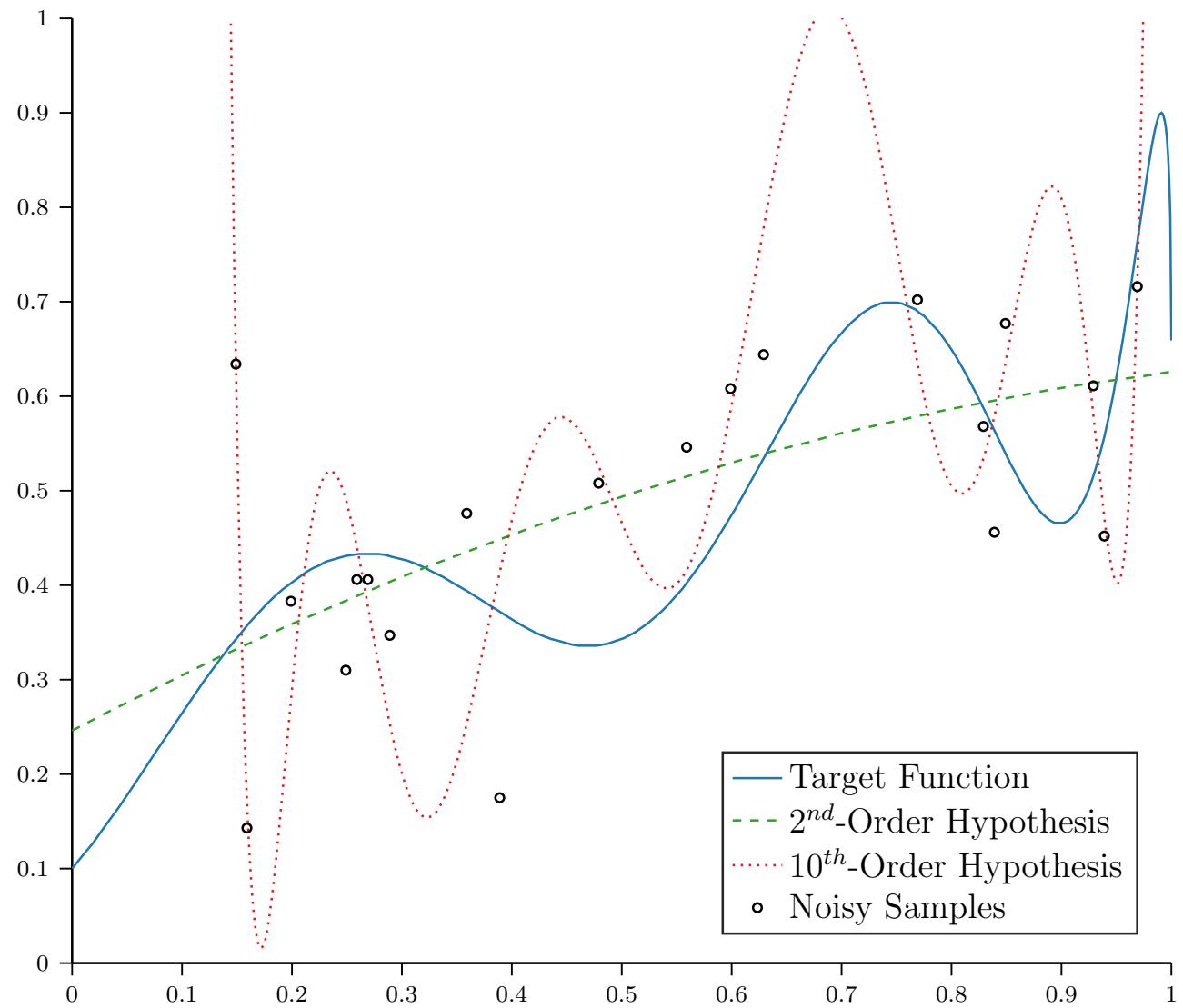
Noisy Targets

- 10-dimensional target function with additive Gaussian noise
- $\mathcal{H}_2 = 2^{\text{nd}}$ -order polynomial
- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomial



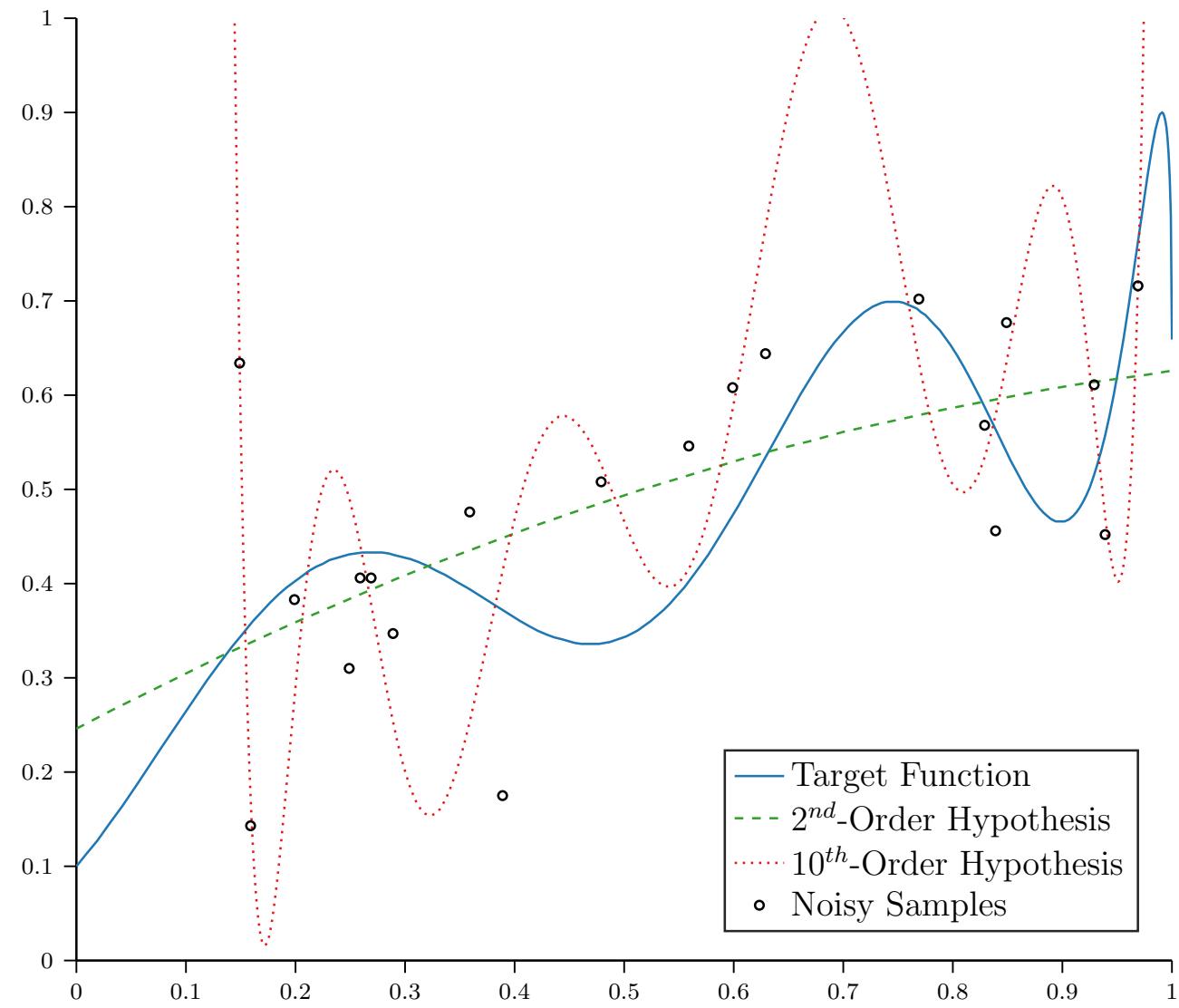
Noisy Targets

- 10-dimensional target function with additive Gaussian noise
- $\mathcal{H}_2 = 2^{\text{nd}}$ -order polynomial
- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomial



Noisy Targets

	\mathcal{H}_2	\mathcal{H}_{10}
Training Error	0.016	0.011
True Error	0.009	3797



Feature Transforms: Experiment

- $x \in \mathbb{R}$, $y \in \mathbb{R}$ and $N = 100$
- Targets are generated by a 10th-order polynomial in x with additive Gaussian noise:

$$y = \sum_{d=0}^{10} a_d x^d + \epsilon \text{ where } \epsilon \sim N(0, \sigma^2)$$

- $\mathcal{H}_2 = 2^{\text{nd}}$ -order polynomials
 - $\phi_{1,2}(x) = [x, x^2]^T$
- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomials
 - $\phi_{1,10}(x) = [x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}]^T$

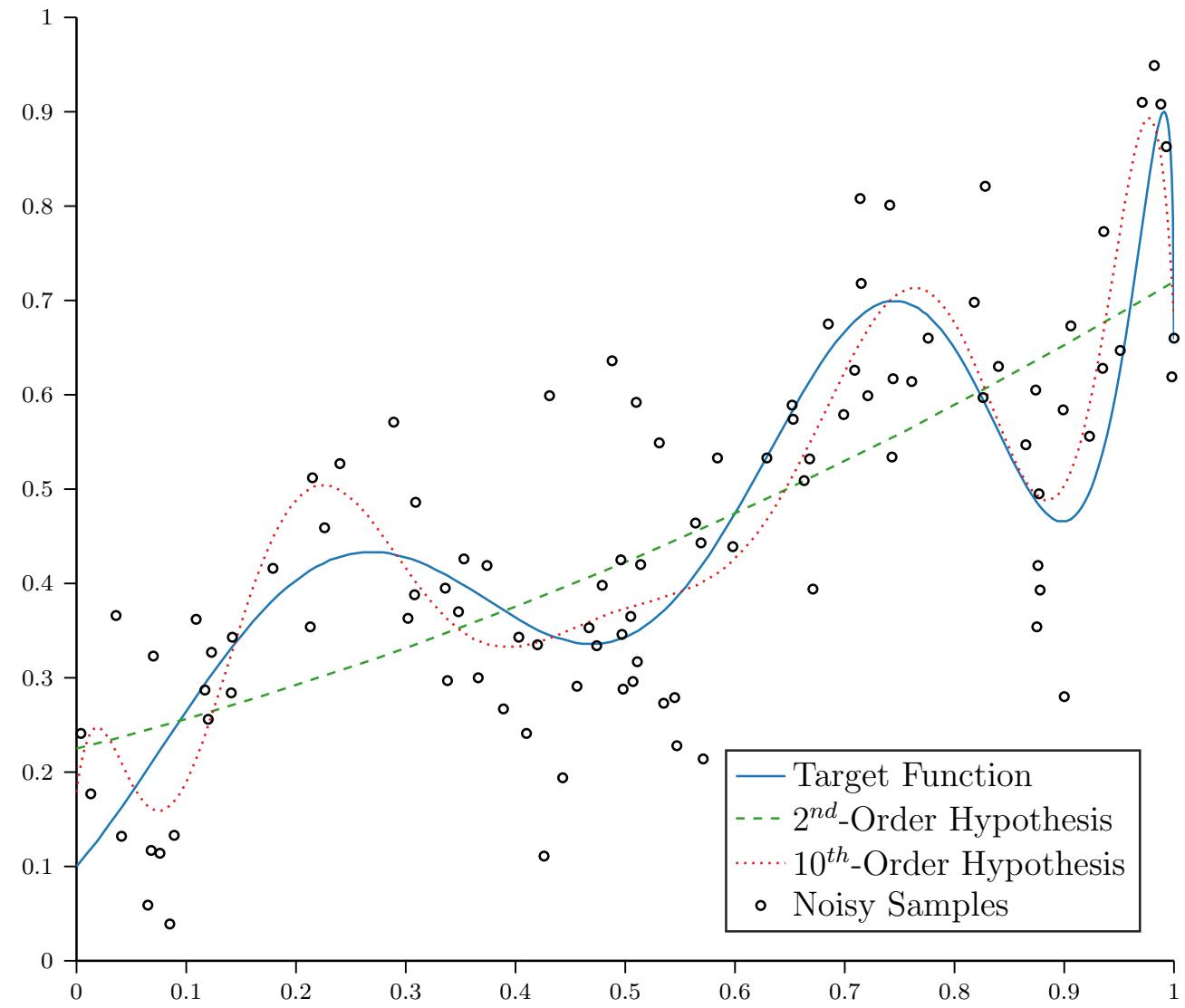
Which model do you think will have a lower true error in this setting ($N = 100$)?

\mathcal{H}_2

\mathcal{H}_{10}

Noisy Targets

	\mathcal{H}_2	\mathcal{H}_{10}
Training Error	0.018	0.010
True Error	0.009	0.003



Regularization

- Constrain models to prevent them from overfitting
- Learning algorithms are optimization problems and regularization imposes constraints on the optimization

Hard Constraints

- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomials
 - $\phi_{1,10}(x) = [x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}]^T$
- Given $X = \begin{bmatrix} 1 & \phi_{1,10}(x^{(1)})^\top \\ 1 & \phi_{1,10}(x^{(2)})^\top \\ \vdots & \vdots \\ 1 & \phi_{1,10}(x^{(N)})^\top \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ find $\boldsymbol{\omega} = [\omega_0, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}]^T$ that minimizes
$$\frac{1}{N} (X\boldsymbol{\omega} - \mathbf{y})^T (X\boldsymbol{\omega} - \mathbf{y})$$
- Subject to
$$\omega_3 = \omega_4 = \omega_5 = \omega_6 = \omega_7 = \omega_8 = \omega_9 = \omega_{10} = 0$$

Hard Constraints

- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomials
 - $\phi_{1,10}(x) = [x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}]^T$
- Given $X = \begin{bmatrix} 1 & \phi_{1,10}(x^{(1)})^T \\ 1 & \phi_{1,10}(x^{(2)})^T \\ \vdots & \vdots \\ 1 & \phi_{1,10}(x^{(N)})^T \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ find $\boldsymbol{\omega} = [\omega_0, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}]^T$ that minimizes
$$\frac{1}{N} \sum_{n=1}^N \left(\left(\sum_{d=0}^{10} x_d^{(n)} \omega_d \right) - y^{(n)} \right)^2$$
- Subject to
$$\omega_3 = \omega_4 = \omega_5 = \omega_6 = \omega_7 = \omega_8 = \omega_9 = \omega_{10} = 0$$

Hard Constraints

- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomials
 - $\phi_{1,10}(x) = [x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}]^T$
- Given $X = \begin{bmatrix} 1 & \phi_{1,10}(x^{(1)})^T \\ 1 & \phi_{1,10}(x^{(2)})^T \\ \vdots & \vdots \\ 1 & \phi_{1,10}(x^{(N)})^T \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ find $\boldsymbol{\omega} = [\omega_0, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}]^T$ that minimizes
$$\frac{1}{N} \sum_{n=1}^N \left(\left(\sum_{d=0}^2 x_d^{(n)} \omega_d \right) - y^{(n)} \right)^2$$
- Subject to nothing!

Hard Constraints

- $\mathcal{H}_2 = 2^{\text{nd}}\text{-order polynomials}$
 - $\phi_{1,2}(x) = [x, x^2]^T$
- Given $X = \begin{bmatrix} 1 & \phi_{1,2}(x^{(1)})^T \\ 1 & \phi_{1,2}(x^{(2)})^T \\ \vdots & \vdots \\ 1 & \phi_{1,2}(x^{(N)})^T \end{bmatrix}$ and $y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ find $\omega = [\omega_0, \omega_1, \omega_2]^T$ that minimizes
$$\frac{1}{N} (X\omega - y)^T (X\omega - y)$$
- Subject to nothing!

Soft Constraints

- More generally, ϕ can be any nonlinear transformation, e.g., exp, log, sin, sqrt, etc...

- Given $X = \begin{bmatrix} 1 & \phi_1(\mathbf{x}^{(1)}) & \dots & \phi_m(\mathbf{x}^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_1(\mathbf{x}^{(N)}) & \dots & \phi_m(\mathbf{x}^{(N)}) \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$,
find $\boldsymbol{\omega}$ that minimizes

$$\frac{1}{N} (\mathbf{X}\boldsymbol{\omega} - \mathbf{y})^T (\mathbf{X}\boldsymbol{\omega} - \mathbf{y})$$

- Subject to:

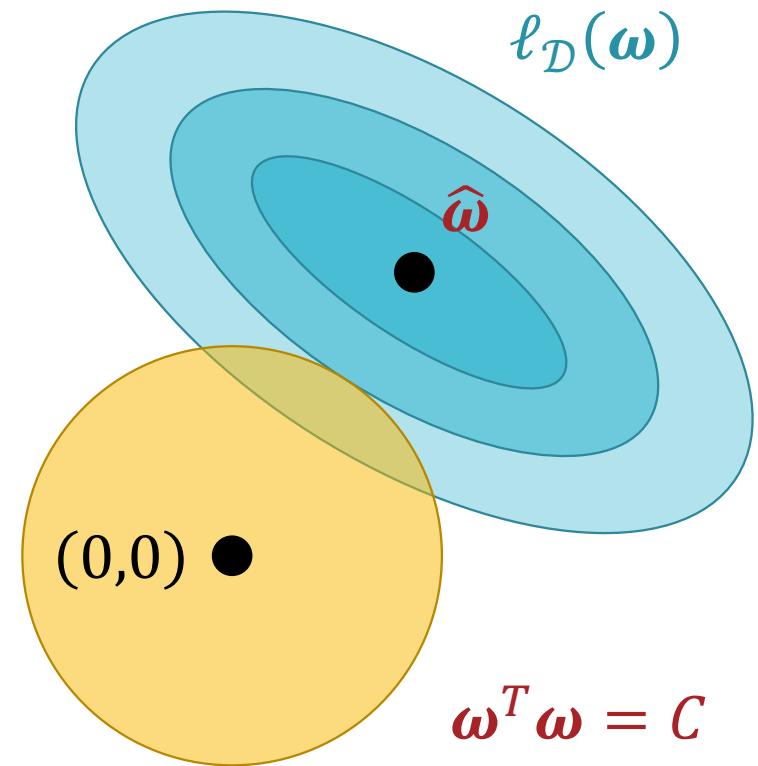
$$\|\boldsymbol{\omega}\|_2^2 = \boldsymbol{\omega}^T \boldsymbol{\omega} = \sum_{d=0}^D \omega_d^2 \leq C$$

Soft Constraints

minimize $\ell_{\mathcal{D}}(\omega) = (X\omega - y)^T(X\omega - y)$

$$\omega_1^2 + \omega_2^2 \leq C$$

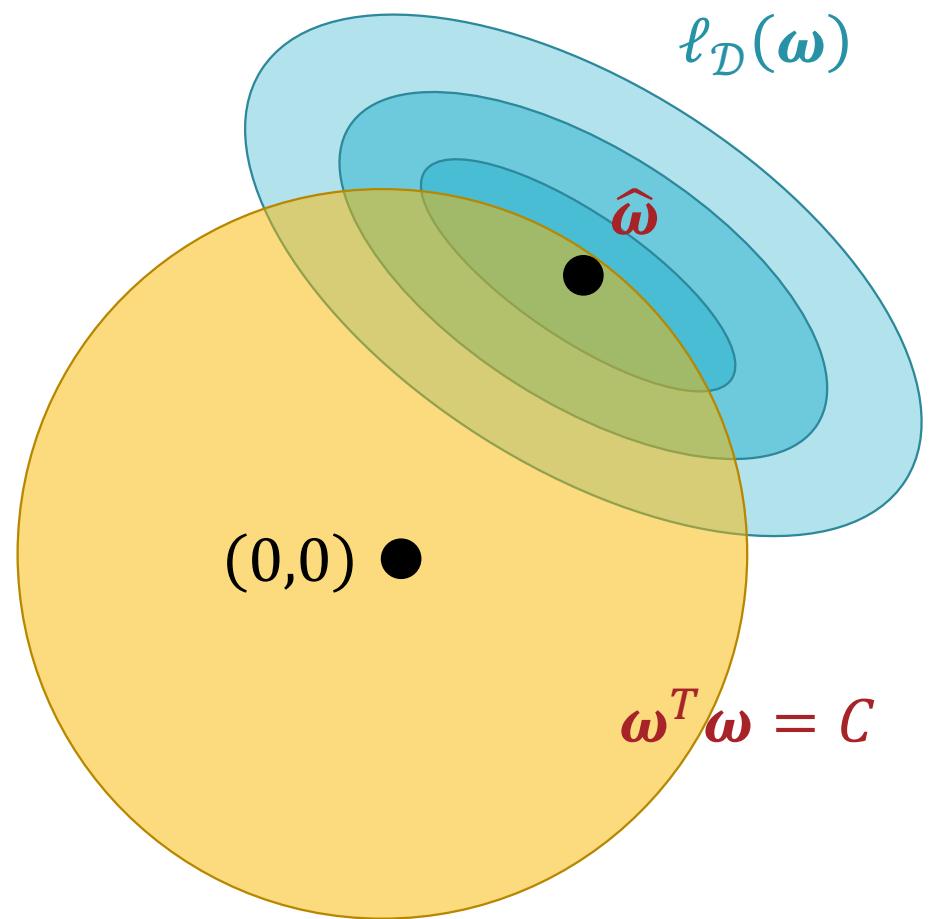
subject to $\omega^T \omega \leq C$



Soft Constraints

minimize $\ell_{\mathcal{D}}(\omega) = (X\omega - y)^T(X\omega - y)$

subject to $\omega^T \omega \leq C$



Soft Constraints

$$\left\{ \begin{array}{l} \text{minimize } \ell_D(\omega) = (X\omega - y)^T(X\omega - y) \\ \text{subject to } \omega^T \omega \leq C \end{array} \right.$$

$$\hat{\vec{\omega}}_{REG} \leftarrow \nabla_{\vec{\omega}} \ell_D(\hat{\vec{\omega}}_{REG})$$

$$2\lambda_C \hat{\vec{\omega}}_{REG} = -\nabla_{\vec{\omega}} \ell_D(\hat{\vec{\omega}}_{REG})$$

$$\nabla_{\vec{\omega}} \ell_D(\hat{\vec{\omega}}_{REG}) + 2\lambda_C \hat{\vec{\omega}}_{REG} = 0$$

$$\nabla_{\vec{\omega}} \left(\ell_D(\hat{\vec{\omega}}_{REG}) + \lambda_C \hat{\vec{\omega}}_{REG}^T \hat{\vec{\omega}}_{REG} \right) = 0$$

$\ell_D(\omega)$

$\hat{\vec{\omega}}$

$\hat{\vec{\omega}}_{REG}$

$(0,0)$

$\omega^T \omega = C$

Soft Constraints: Solving for $\hat{\omega}_{REG}$

$$\text{minimize } \ell_{\mathcal{D}}(\boldsymbol{\omega}) = (X\boldsymbol{\omega} - \mathbf{y})^T(X\boldsymbol{\omega} - \mathbf{y})$$

$$\text{subject to } \boldsymbol{\omega}^T \boldsymbol{\omega} \leq C$$

\Updownarrow

$$\text{minimize } \ell_{\mathcal{D}}^{AUG}(\boldsymbol{\omega}) = \ell_{\mathcal{D}}(\boldsymbol{\omega}) + \lambda_C \boldsymbol{\omega}^T \boldsymbol{\omega}$$

Ridge Regression

minimize $\ell_D^{AUG}(\omega) = \ell_D(\omega) + \lambda_C \omega^T \omega$

$$\nabla_{\vec{\omega}} \ell_D^{AUG}(\vec{\omega}) = 2(X^T X \vec{\omega} - X^T \vec{y}) + 2\lambda_C \vec{\omega}$$

$$\Rightarrow 2X^T X \vec{\omega}_{REG} - 2X^T \vec{y} + 2\lambda_C \vec{\omega}_{REG} = 0$$

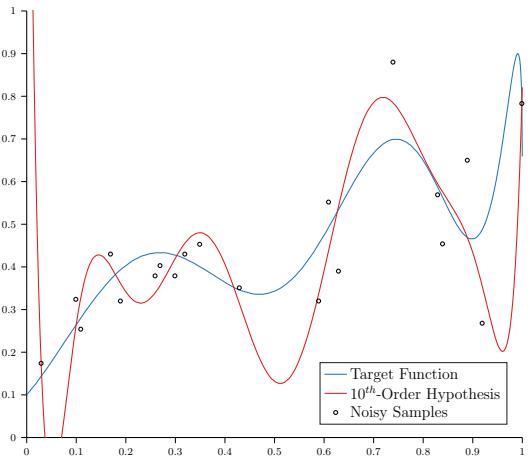
$$\Rightarrow X^T X \vec{\omega}_{REG} + \lambda_C \vec{\omega}_{REG} = X^T \vec{y}$$

$$\Rightarrow (X^T X + \lambda_C I_{D+1}) \vec{\omega}_{REG} = X^T \vec{y}$$

$(D+1) \times (D+1)$ identity

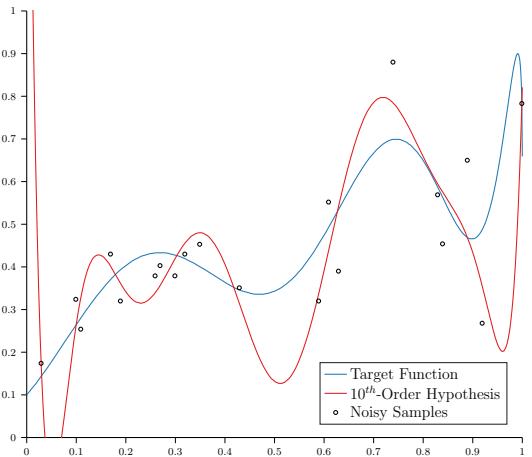
$$\Rightarrow \vec{\omega}_{REG} = (X^T X + \lambda_C I_{D+1})^{-1} X^T \vec{y}$$

40



Ridge Regression

- 10-dimensional target function with additive Gaussian noise
- $\mathcal{H}_{10} = 10^{\text{th}}$ -order polynomial



Ridge Regression

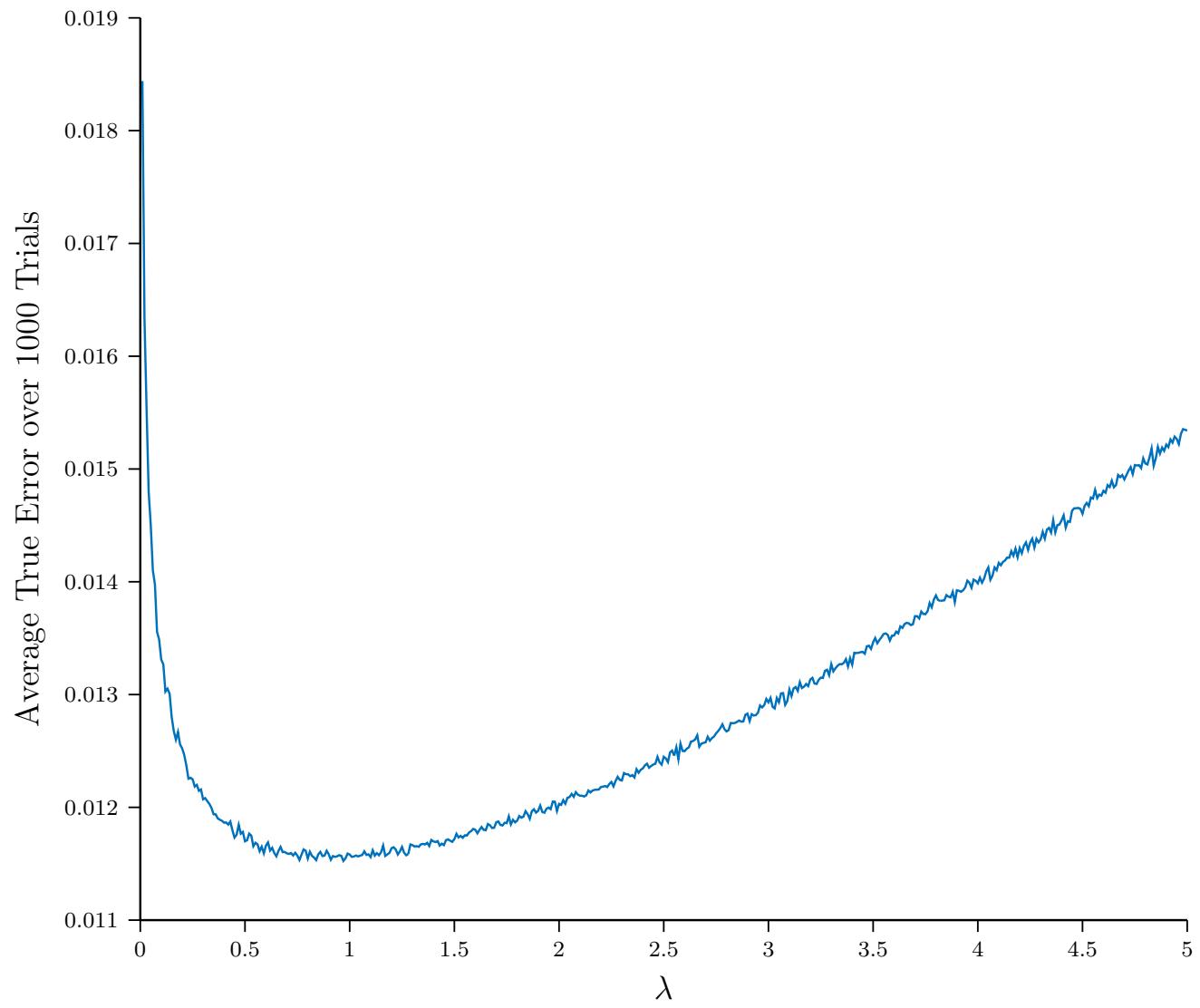
$$\lambda_c = 0$$

True
Error

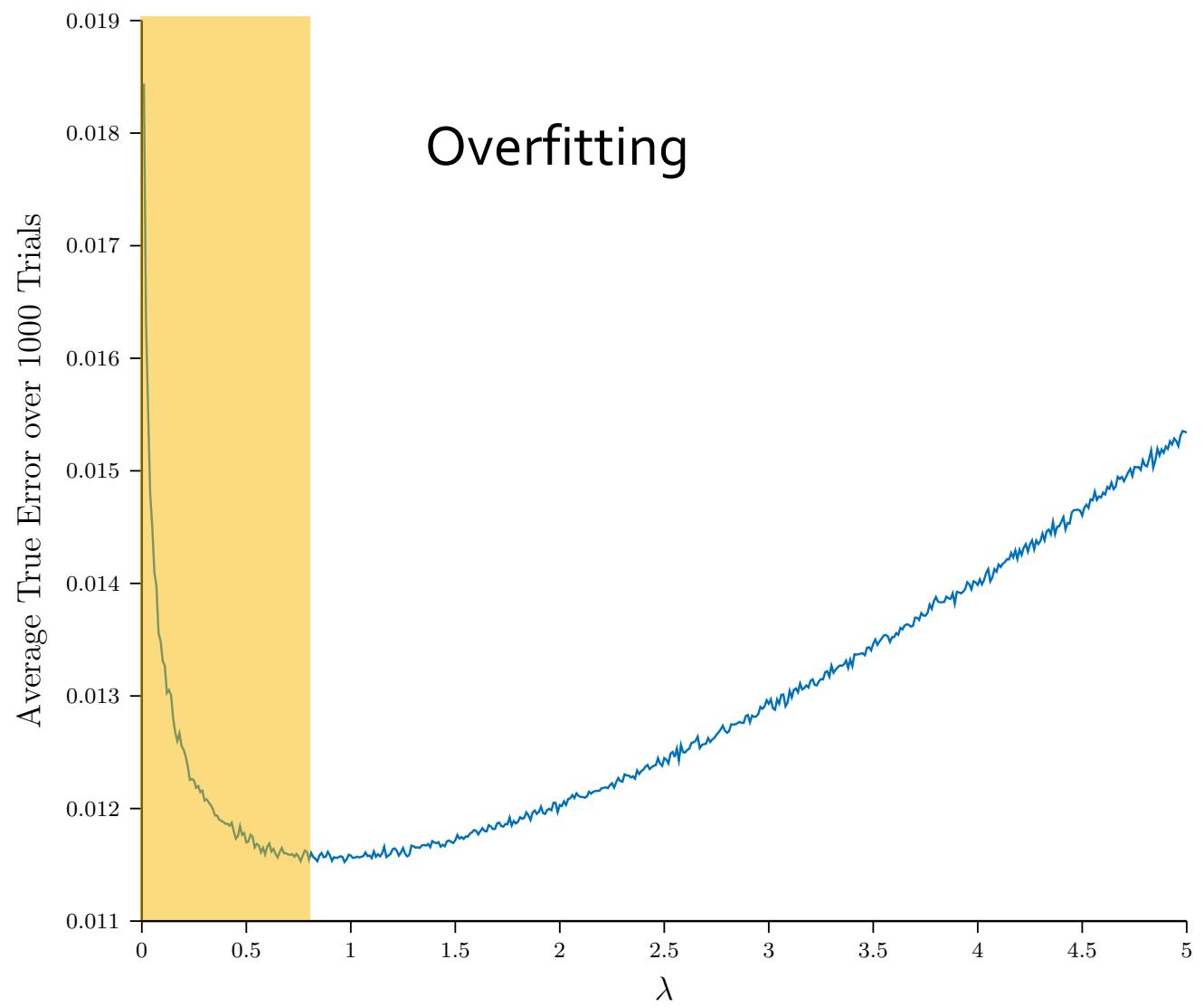
0.059

Overfit

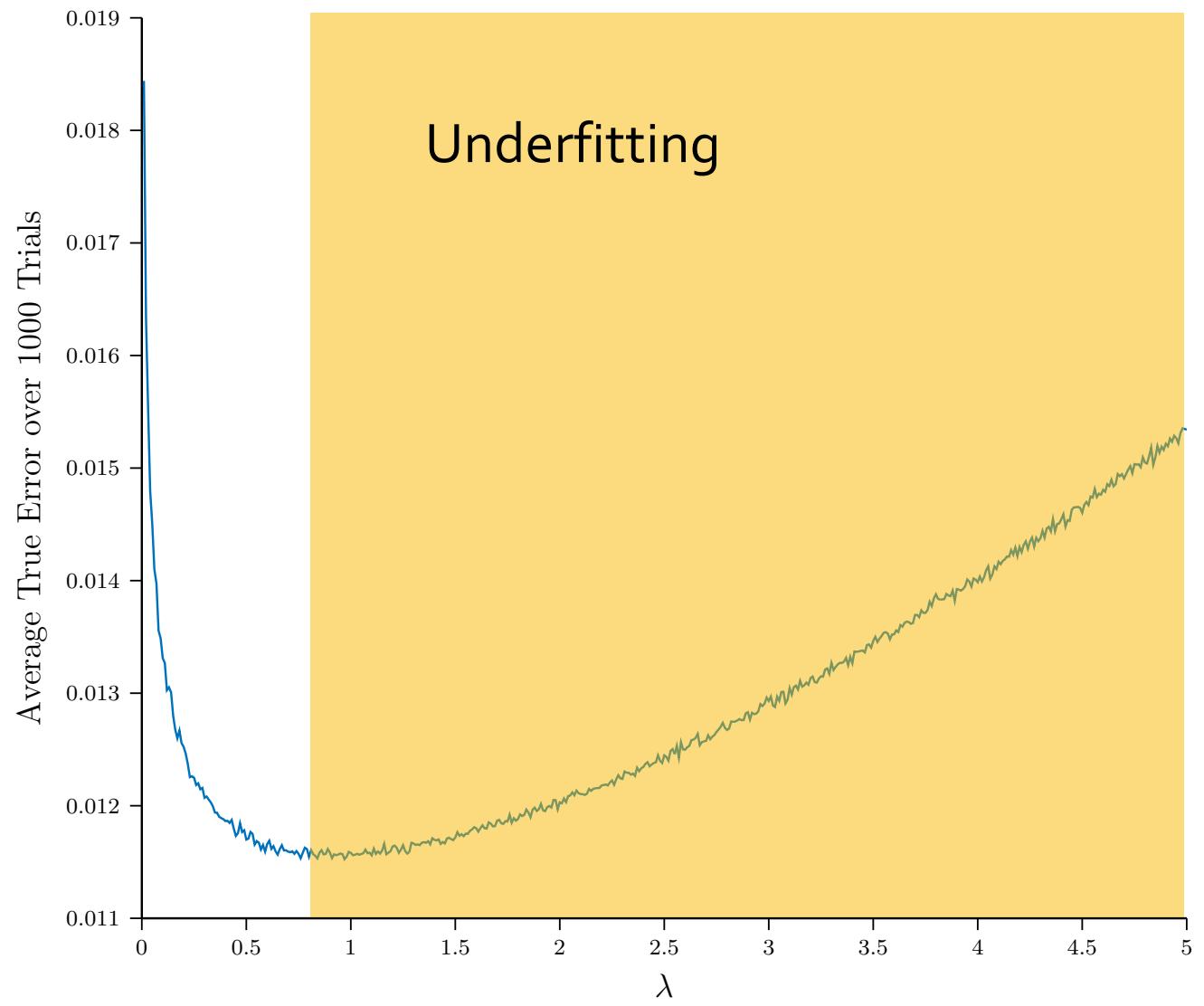
Setting λ



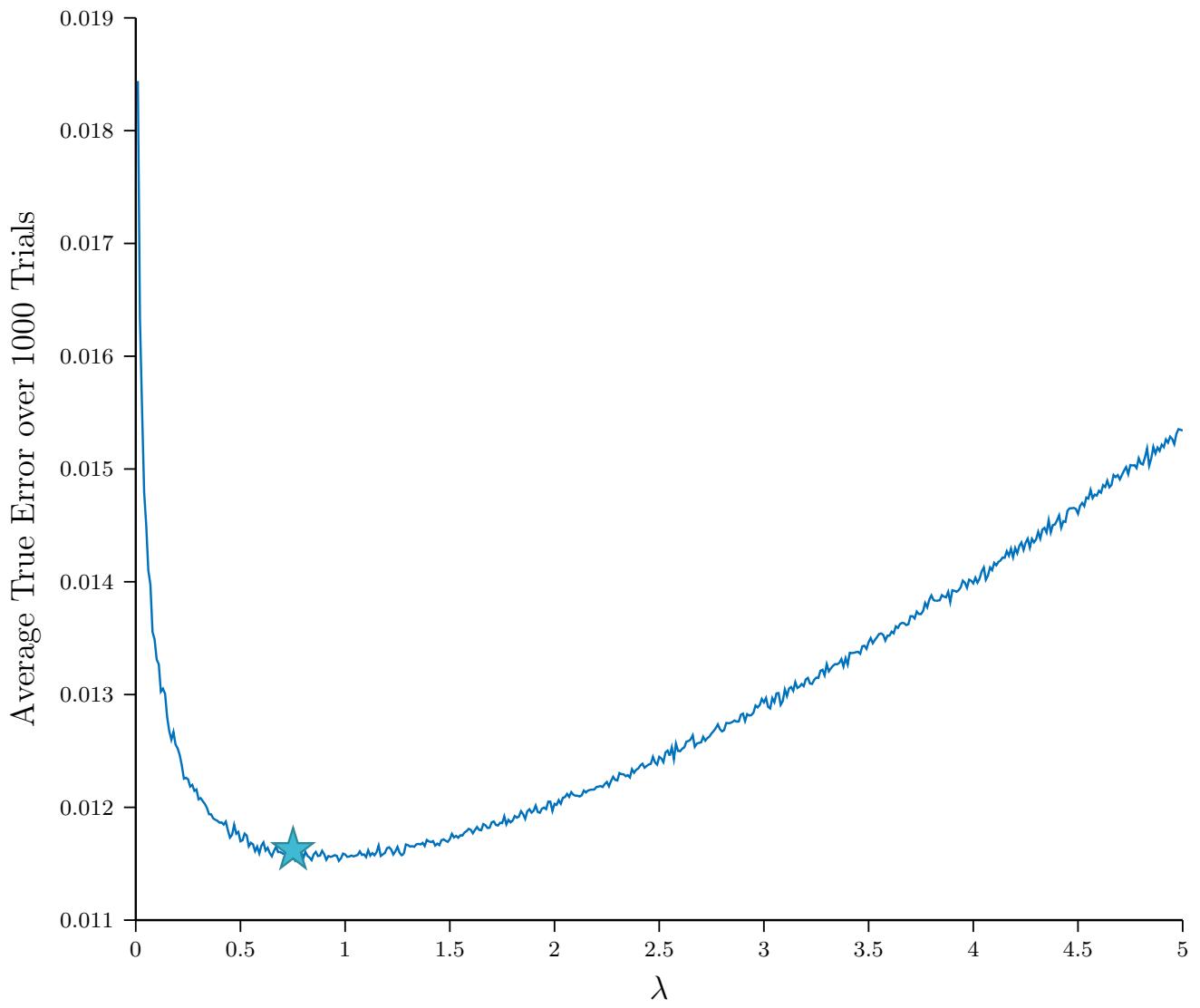
Setting λ



Setting λ

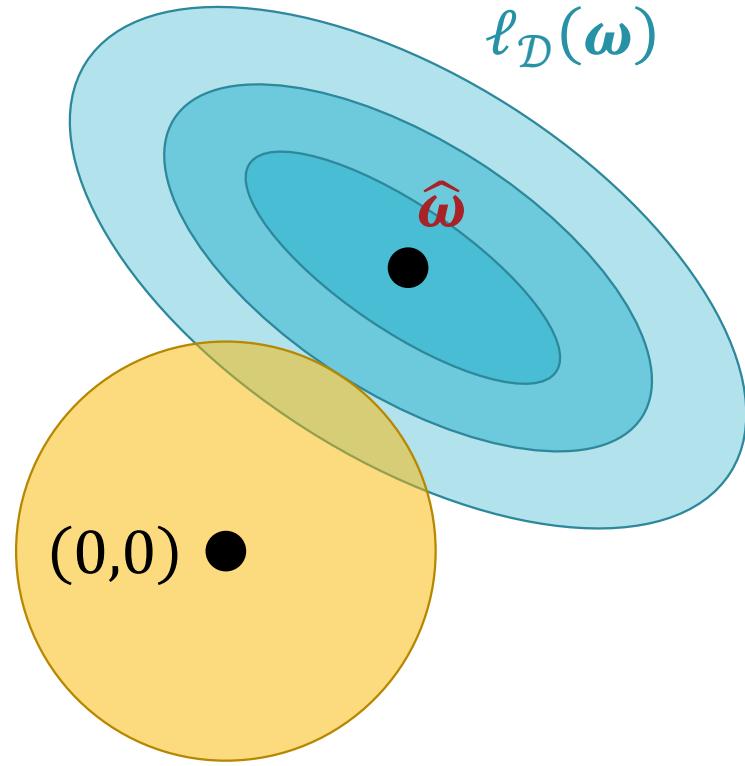


Setting λ

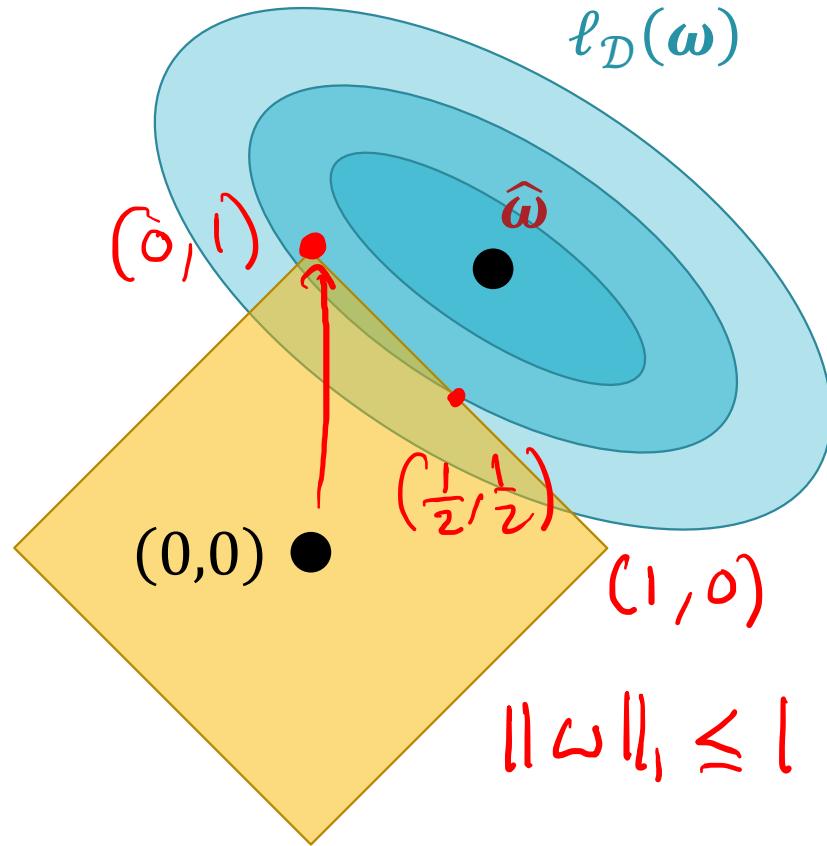


Other Regularizers

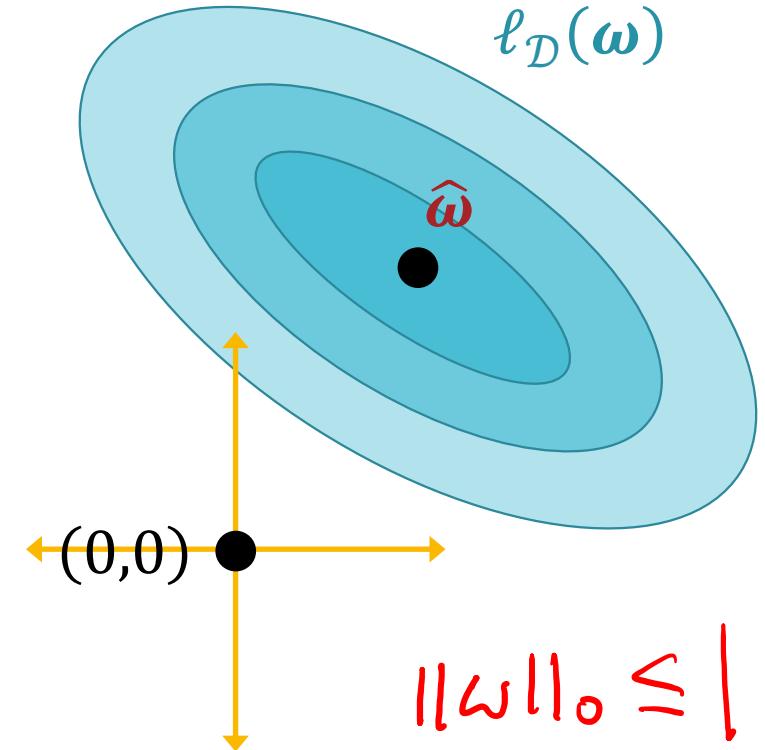
$\ell_{\mathcal{D}}(\boldsymbol{\omega}) + \lambda r(\boldsymbol{\omega})$			
Ridge or $L2$	$r(\boldsymbol{\omega}) = \ \boldsymbol{\omega}\ _2^2 = \sum_{d=0}^D \omega_d^2$		Encourages small weights
Lasso or $L1$	$r(\boldsymbol{\omega}) = \ \boldsymbol{\omega}\ _1 = \sum_{d=0}^D \omega_d $		Encourages sparsity
$L0$	$r(\boldsymbol{\omega}) = \ \boldsymbol{\omega}\ _0 = \sum_{d=0}^D \mathbb{1}(\omega_d \neq 0)$		Encourages sparsity (intractable)



Ridge or $L2$



Lasso or $L1$



$L0$

Other Regularizers

Key Takeaways

- Where do features come from?
 - Engineered (hand-crafted) vs. learned
- Polynomial/non-linear feature transformations allow for learning non-linear functions/decision boundaries
 - Can lead to overfitting...
 - Address with regularization!
 - Analogous to constrained optimization, solve via method of Lagrange multipliers
 - Regularization level is a hyperparameter