10-301/601: Introduction to Machine Learning Lecture 12 – Regularization

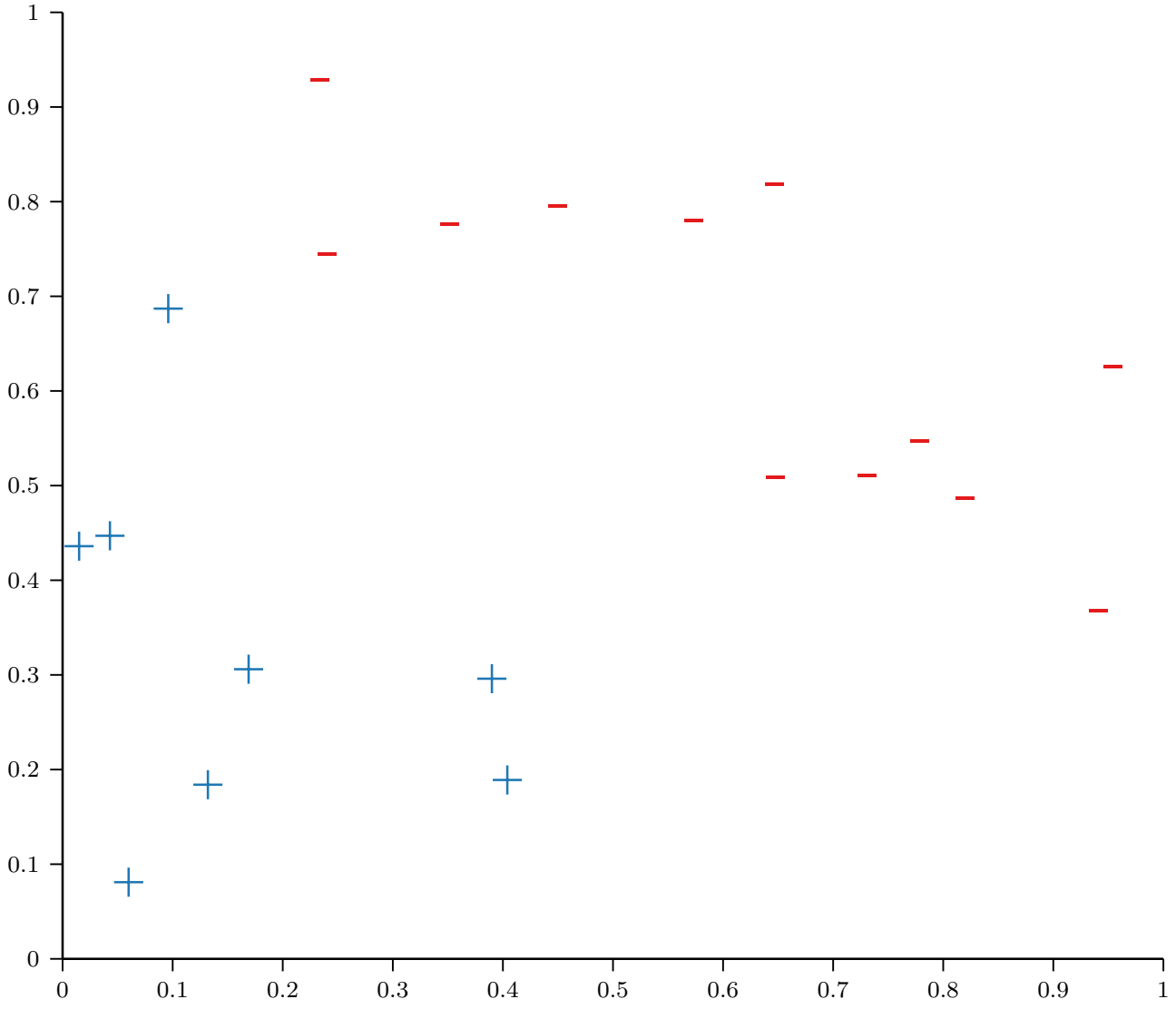
Henry Chai

5/21/25

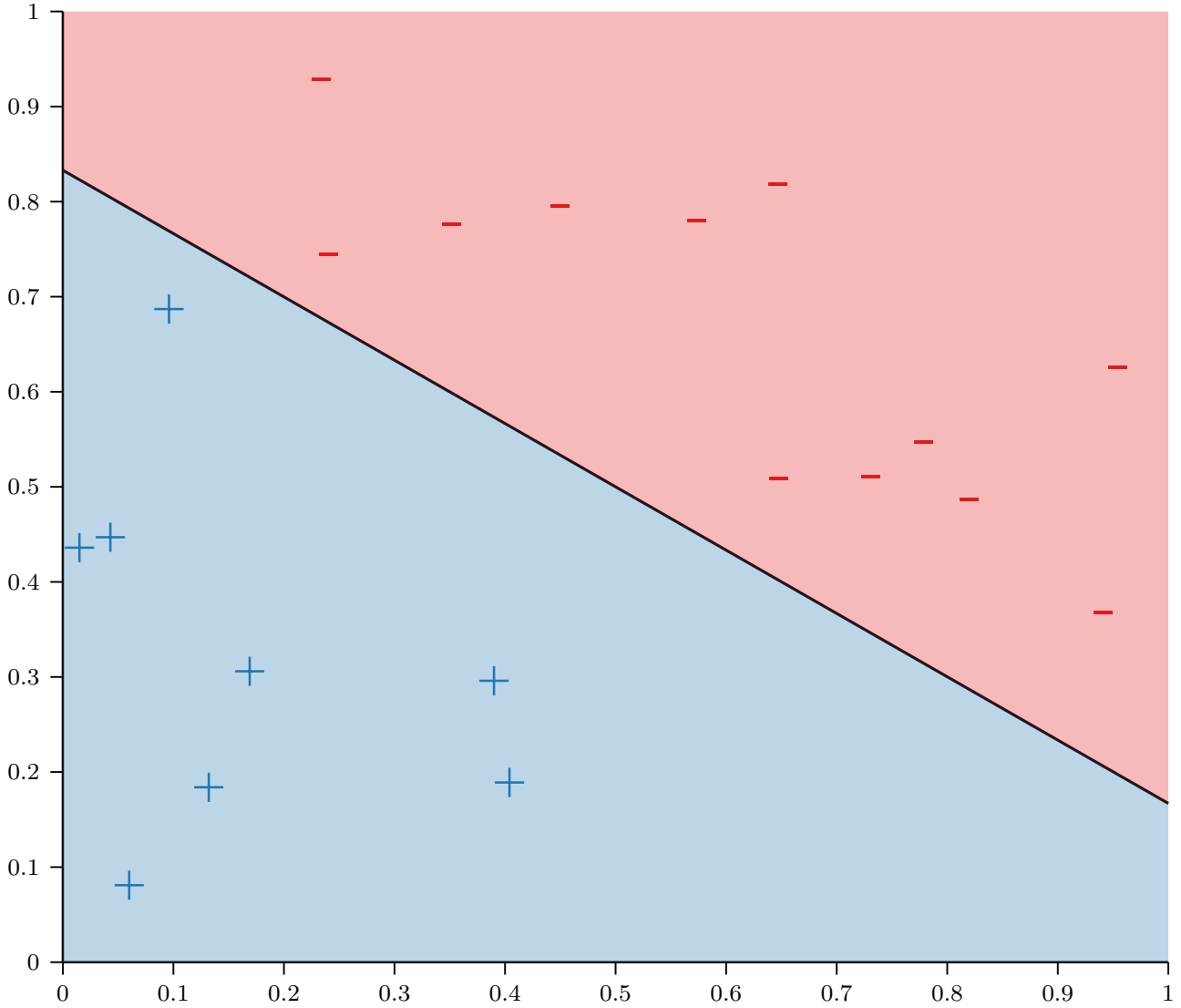
Front Matter

- 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**

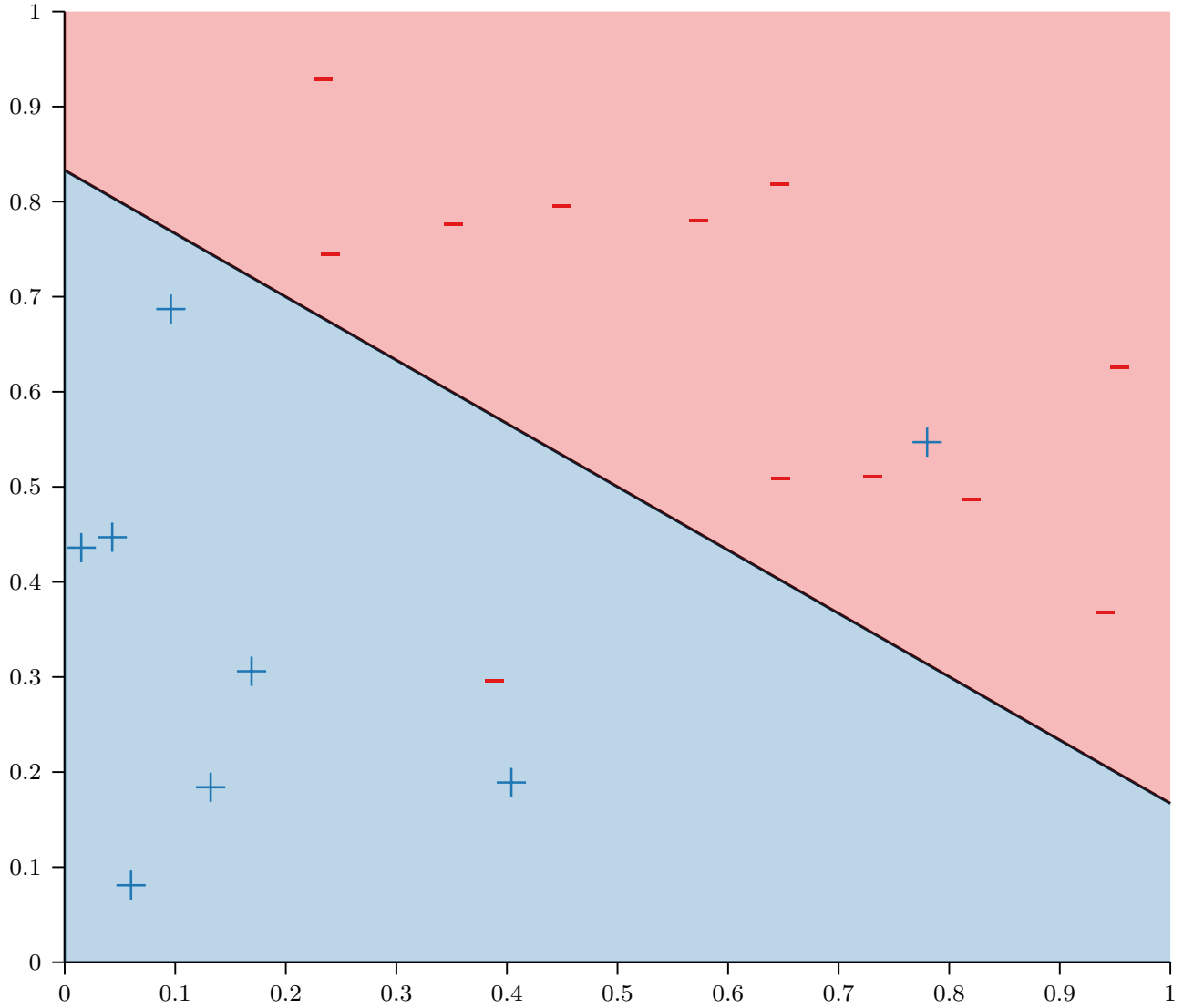
Linear Models



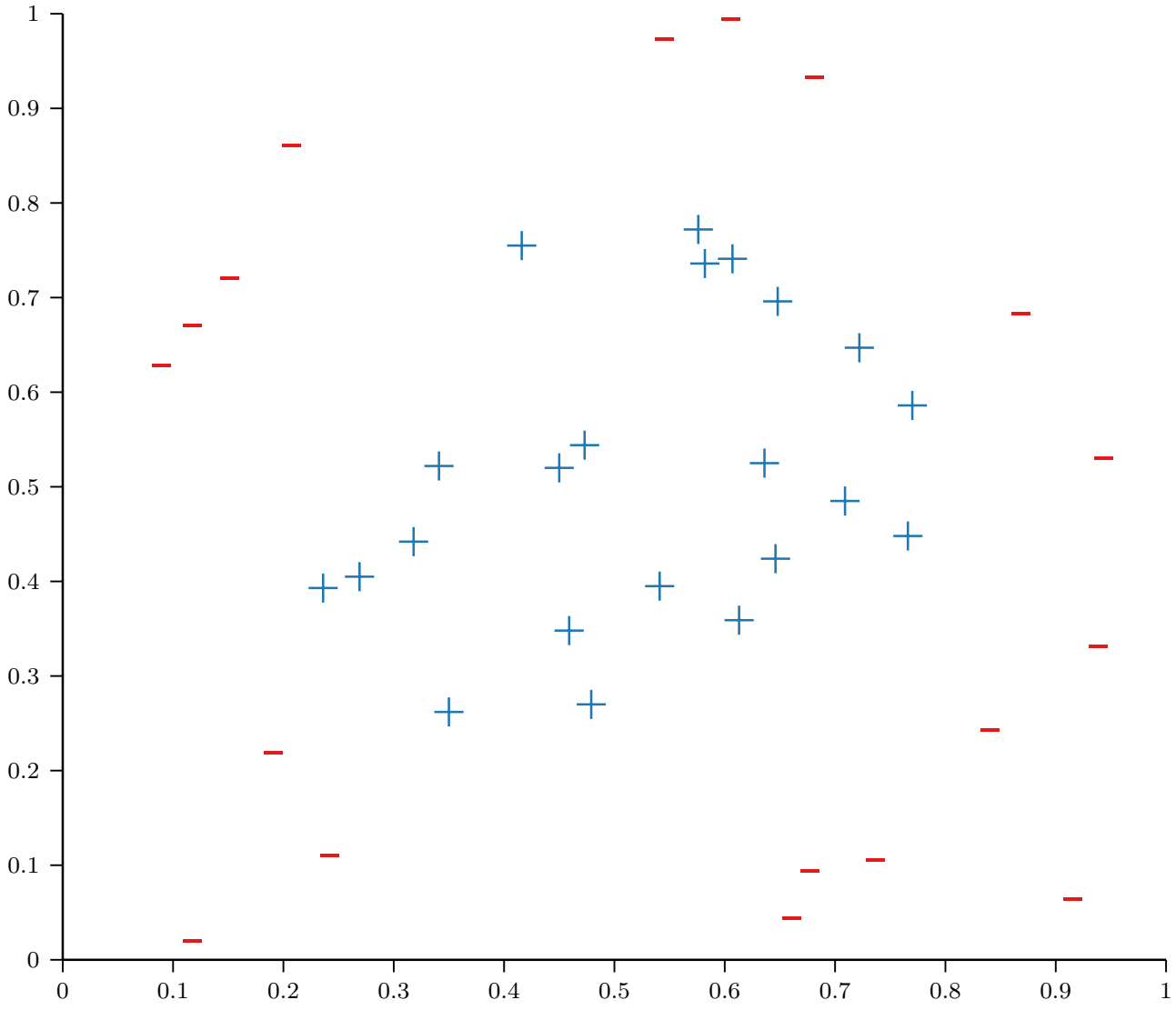
Linear Models



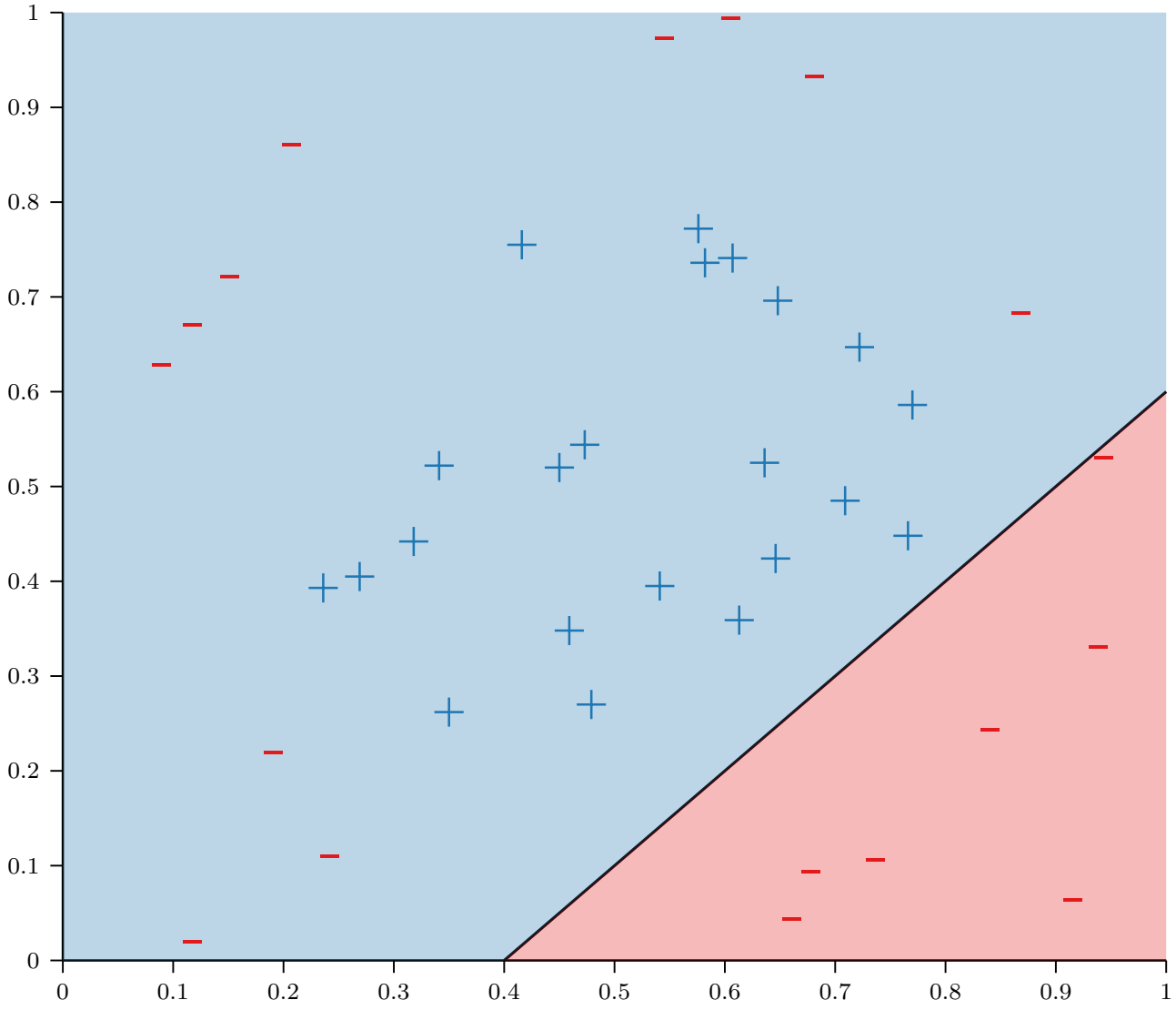
Linear Models



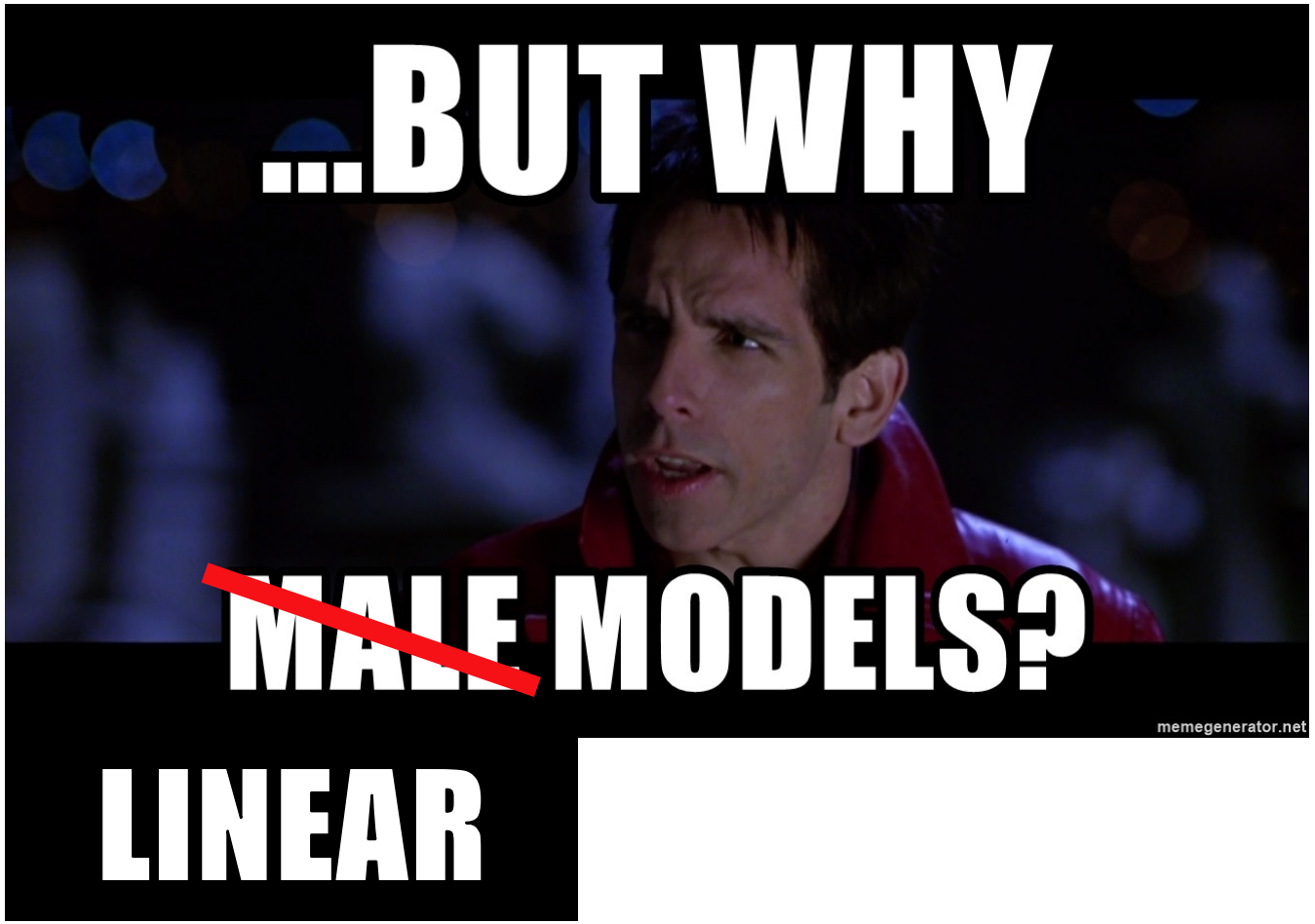
Linear Models?



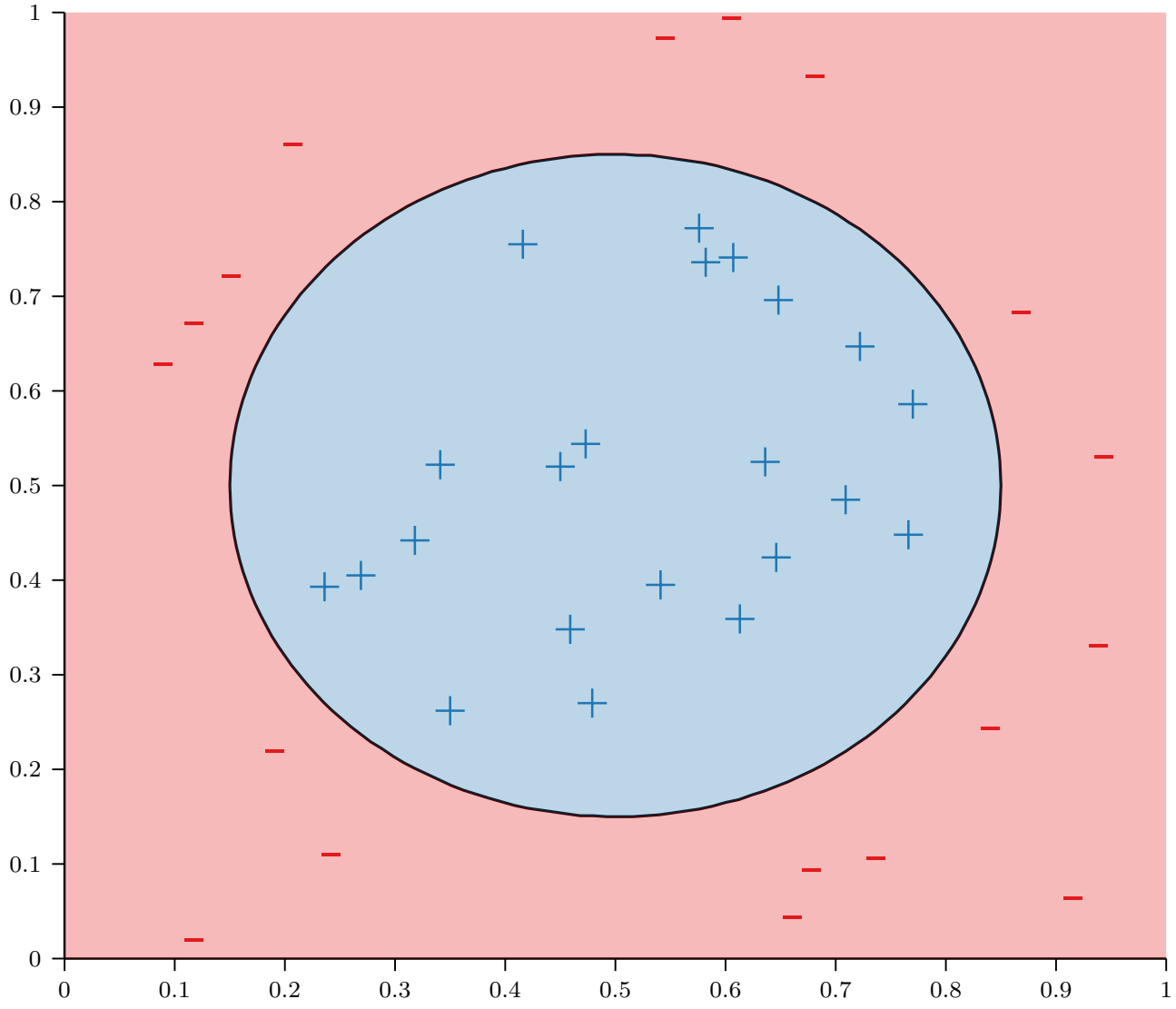
Linear Models?



Linear Models?



Nonlinear Models

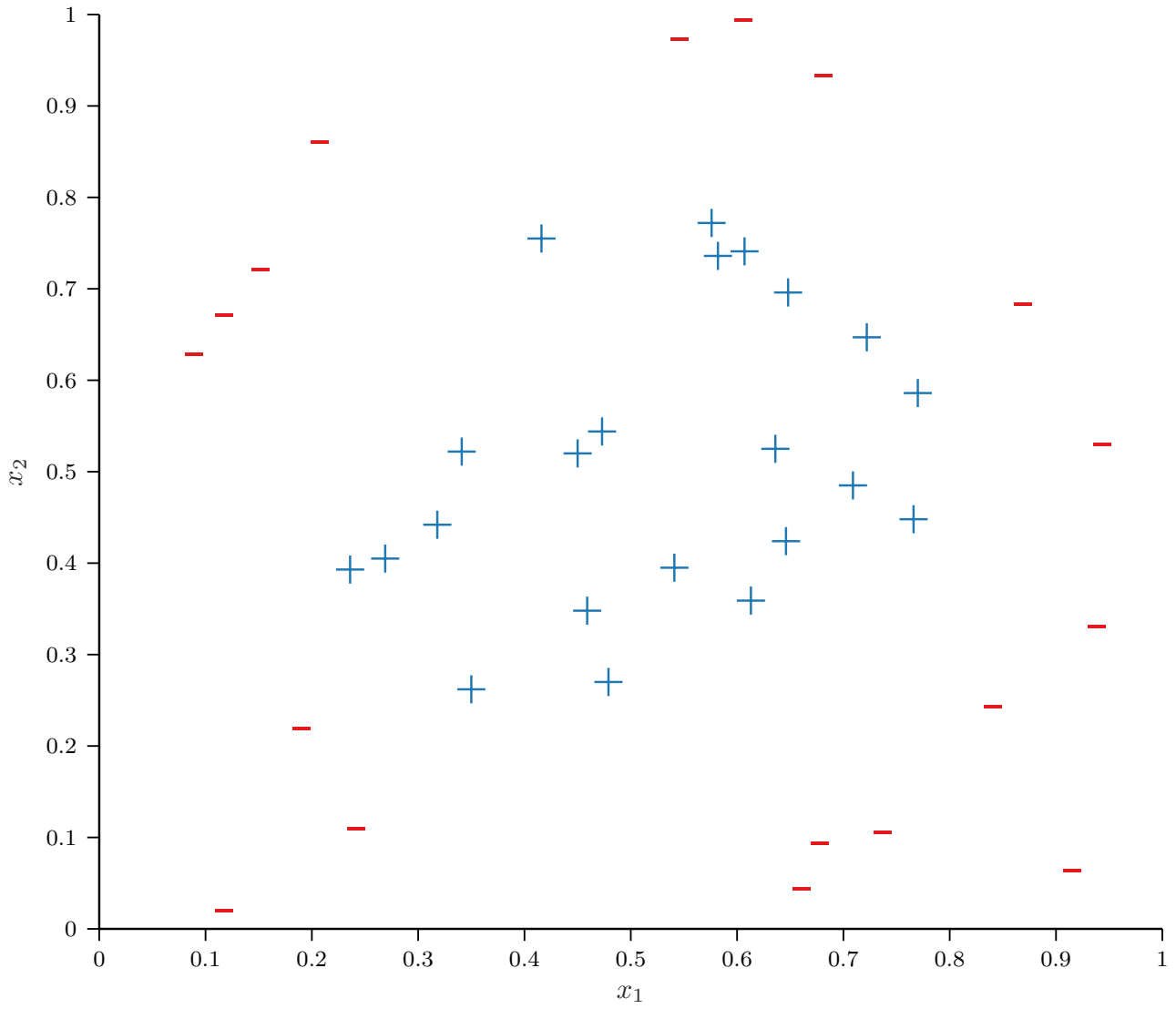


Feature Transforms

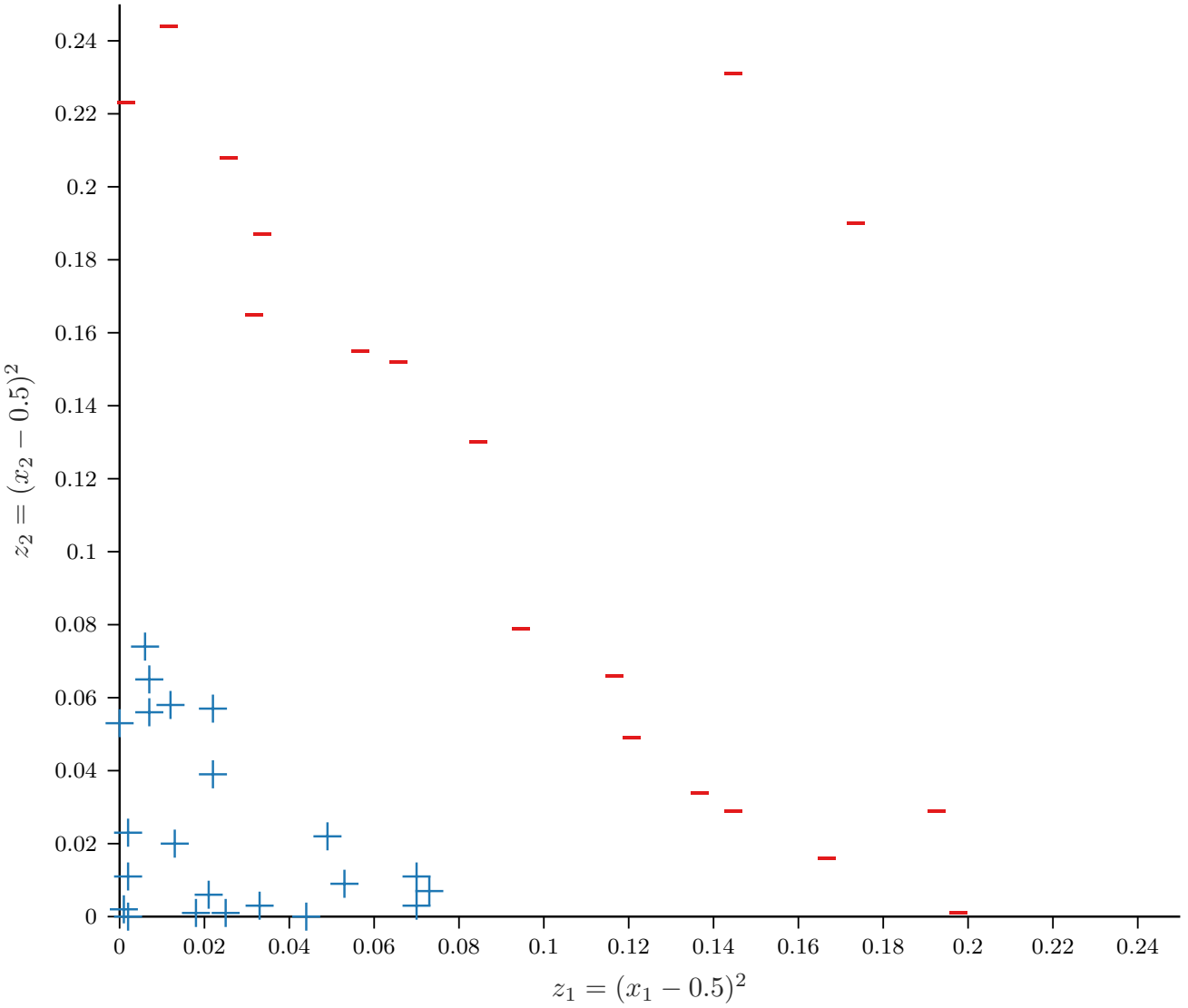
- Given D-dimensional inputs $\boldsymbol{x} = [x_1, \dots, x_D]^T$, first compute some transformation of our input, e.g.,

$$\phi([x_1, x_2]^T) = [z_1 = (x_1 - 0.5)^2, z_2 = (x_2 - 0.5)^2]$$

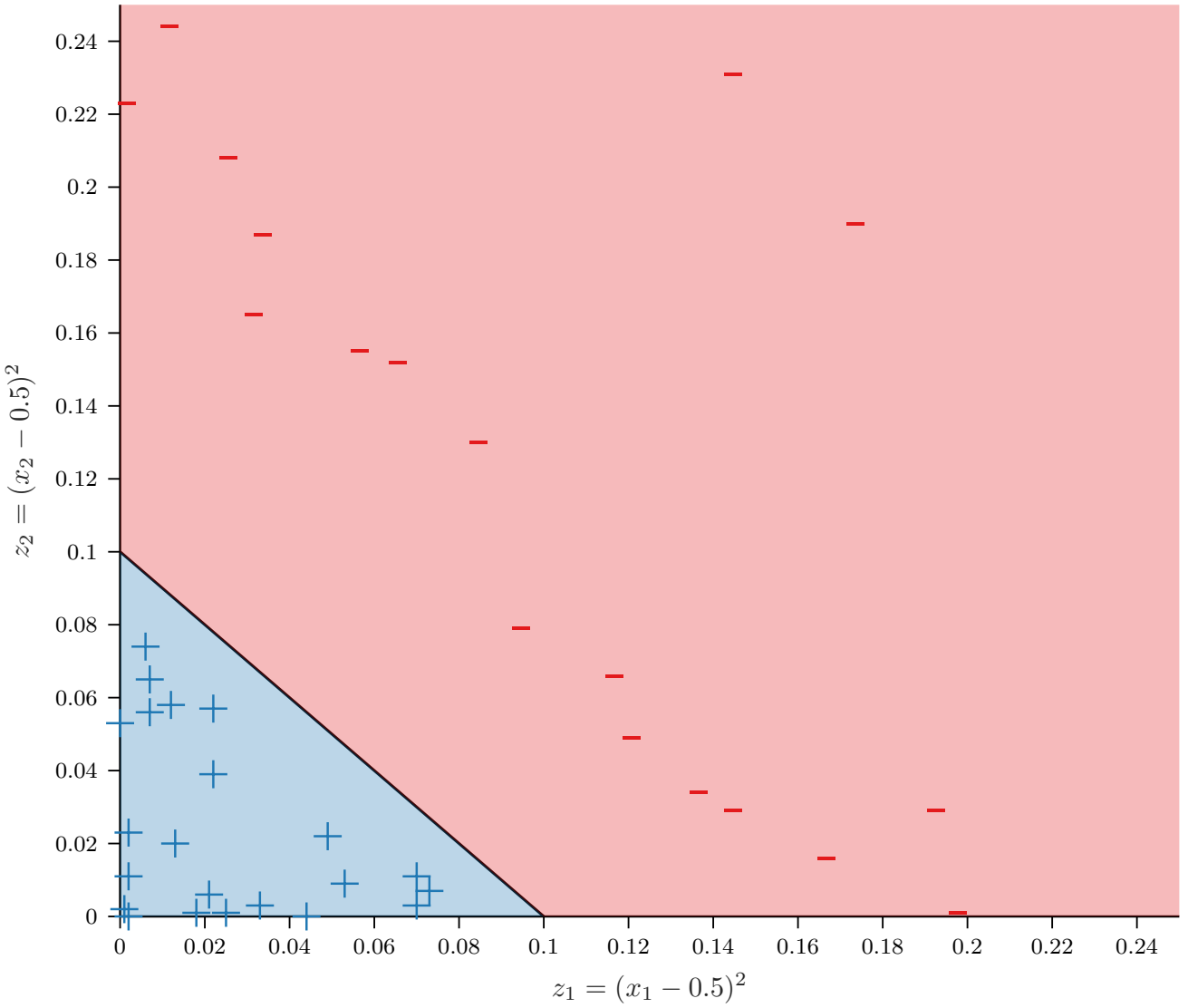
Nonlinear Models



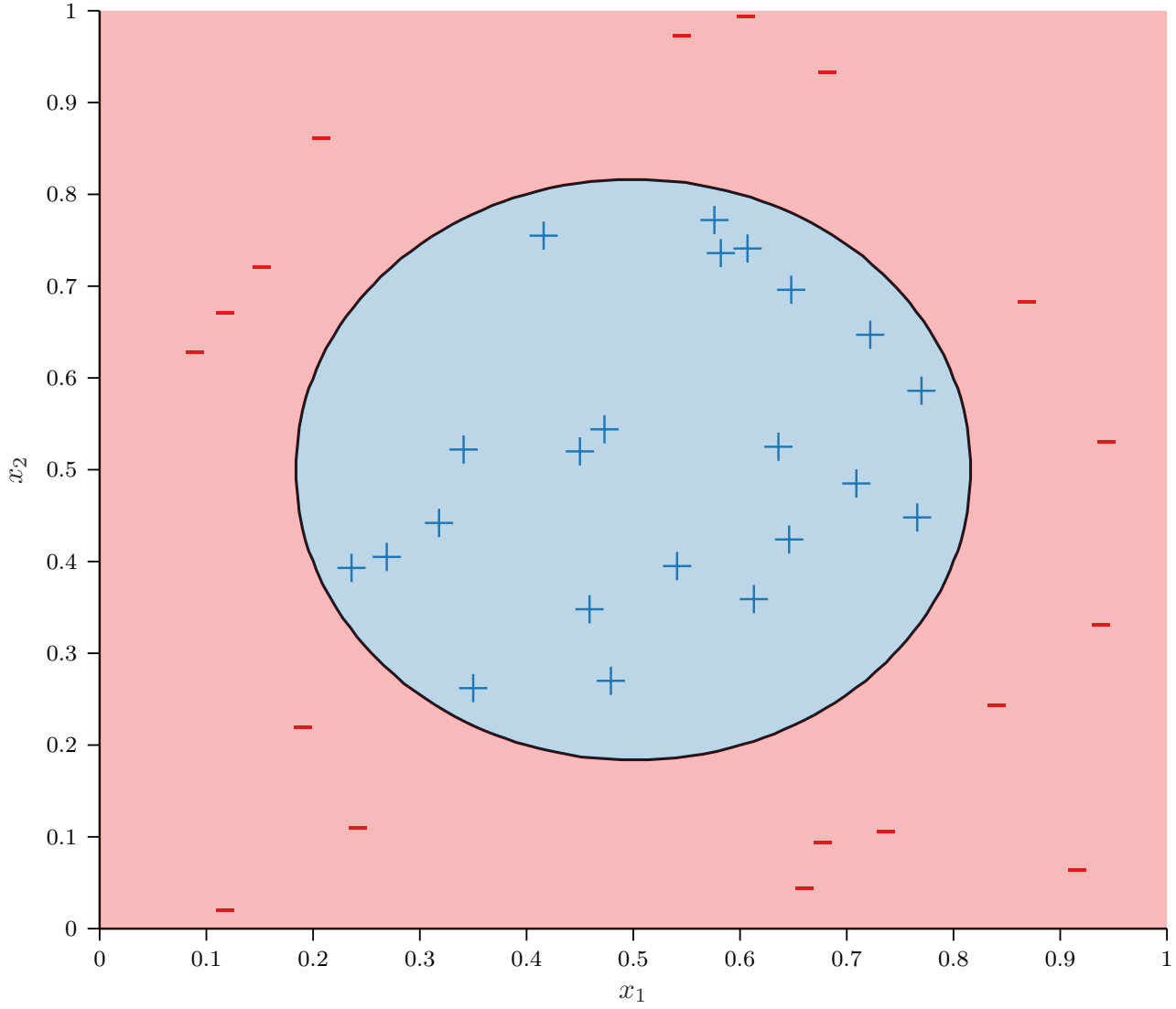
Nonlinear Models



Nonlinear Models



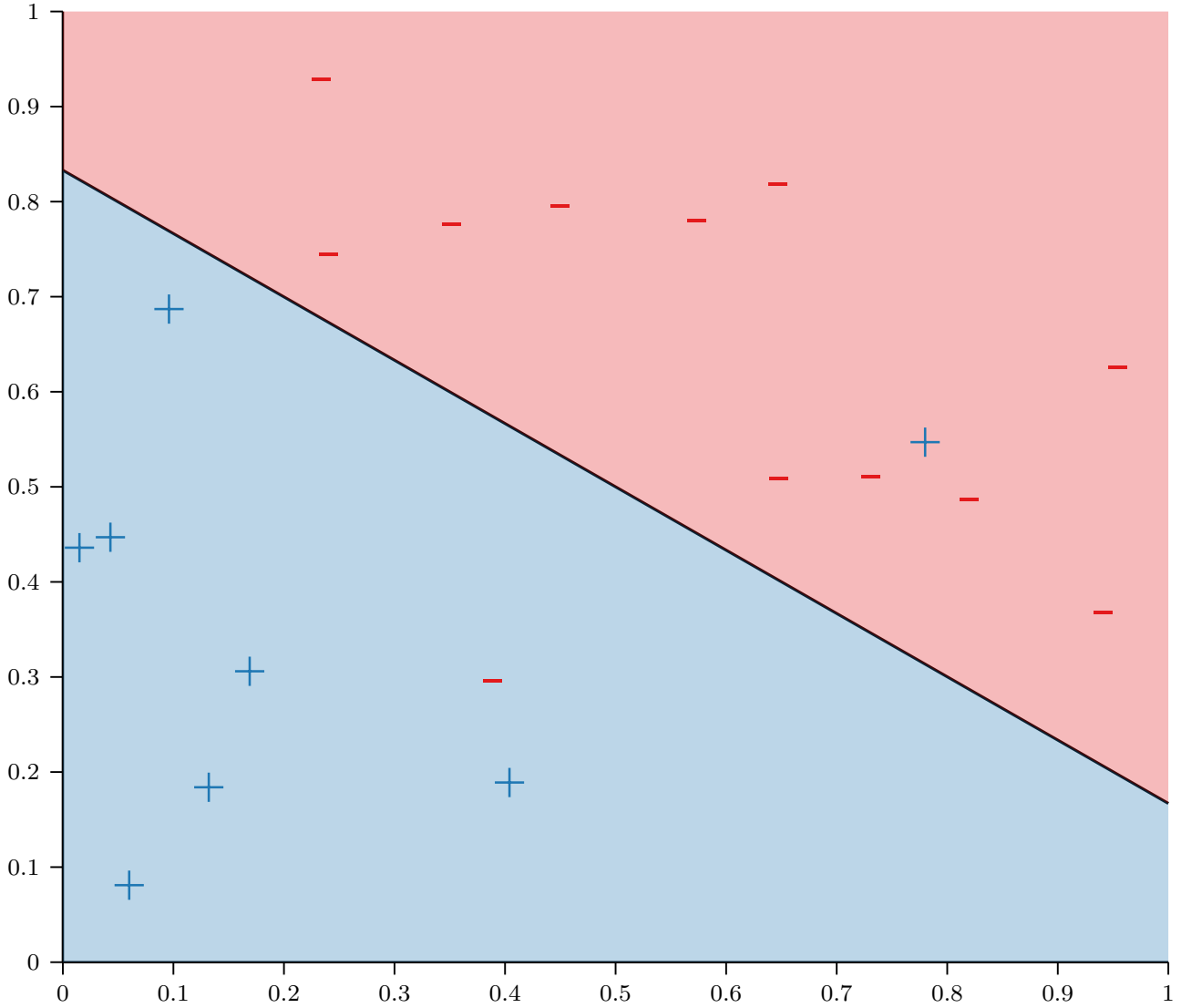
Nonlinear Models



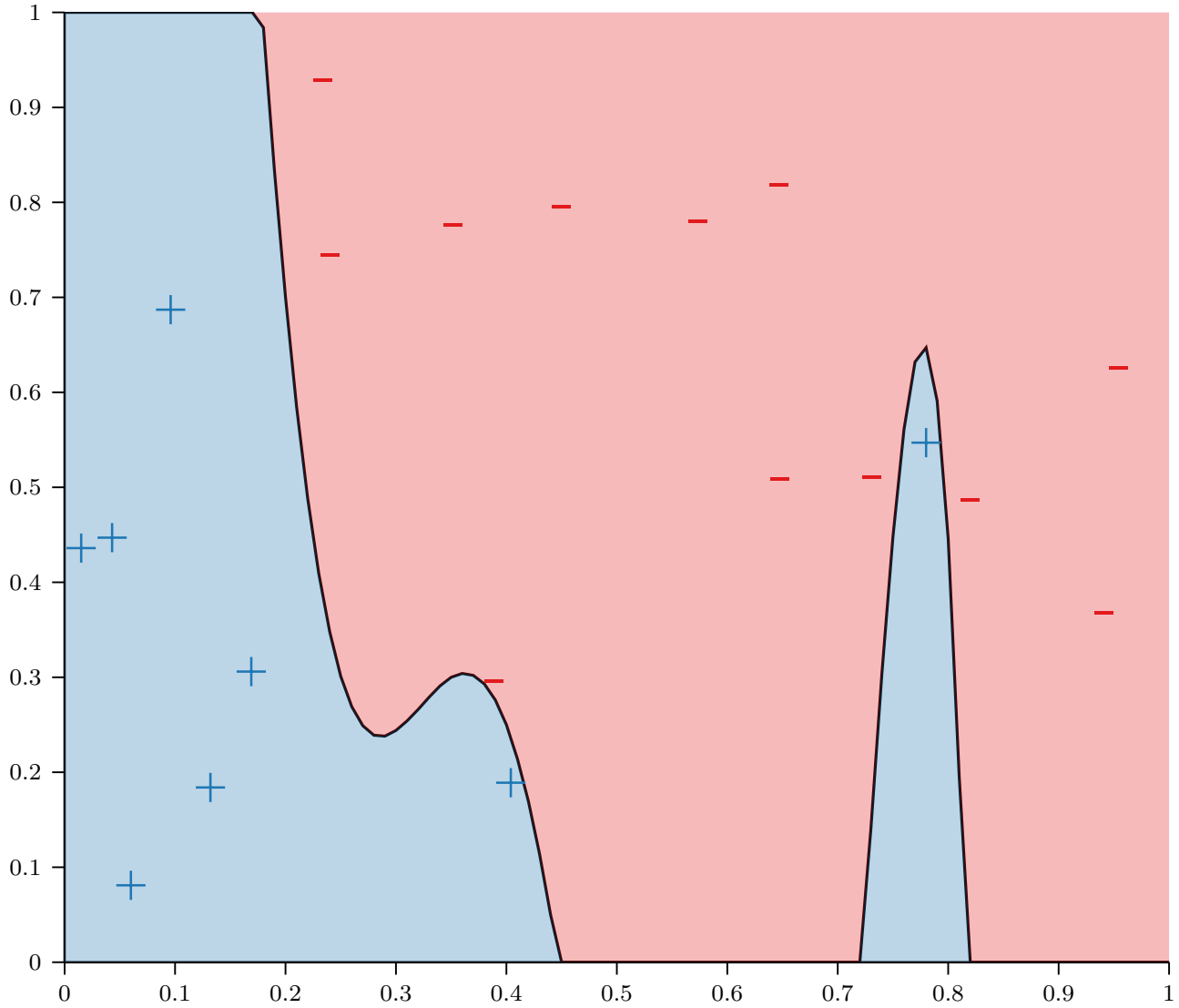
General Q^{th}-order Transforms

- $\phi_{2,2}([x_1, x_2]^T) = [x_1, x_2, x_1^2, x_1x_2, x_2^2]^T$
- $\phi_{2,3}([x_1, x_2]^T) = [x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3]^T$
- $\phi_{2,4}([x_1, x_2]^T) =$
  $[x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_1^4, x_1^3x_2, x_1^2x_2^2, x_1x_2^3, x_2^4]^T$
- $\phi_{2,Q}$ maps a 2-dimensional input to a $\frac{Q(Q+3)}{2}$-dimensional output
- Scales even worse for higher-dimensional inputs…

Linear Models

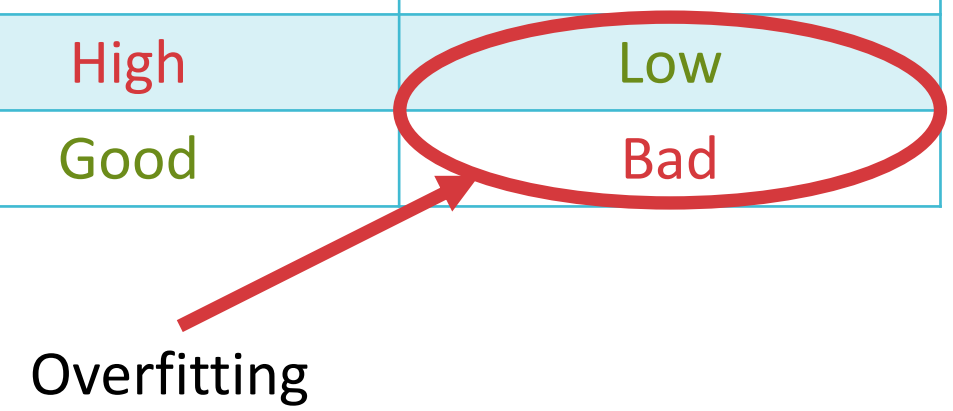


Nonlinear Models?



Feature Transforms: Tradeoffs

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



Feature Transforms: Experiment

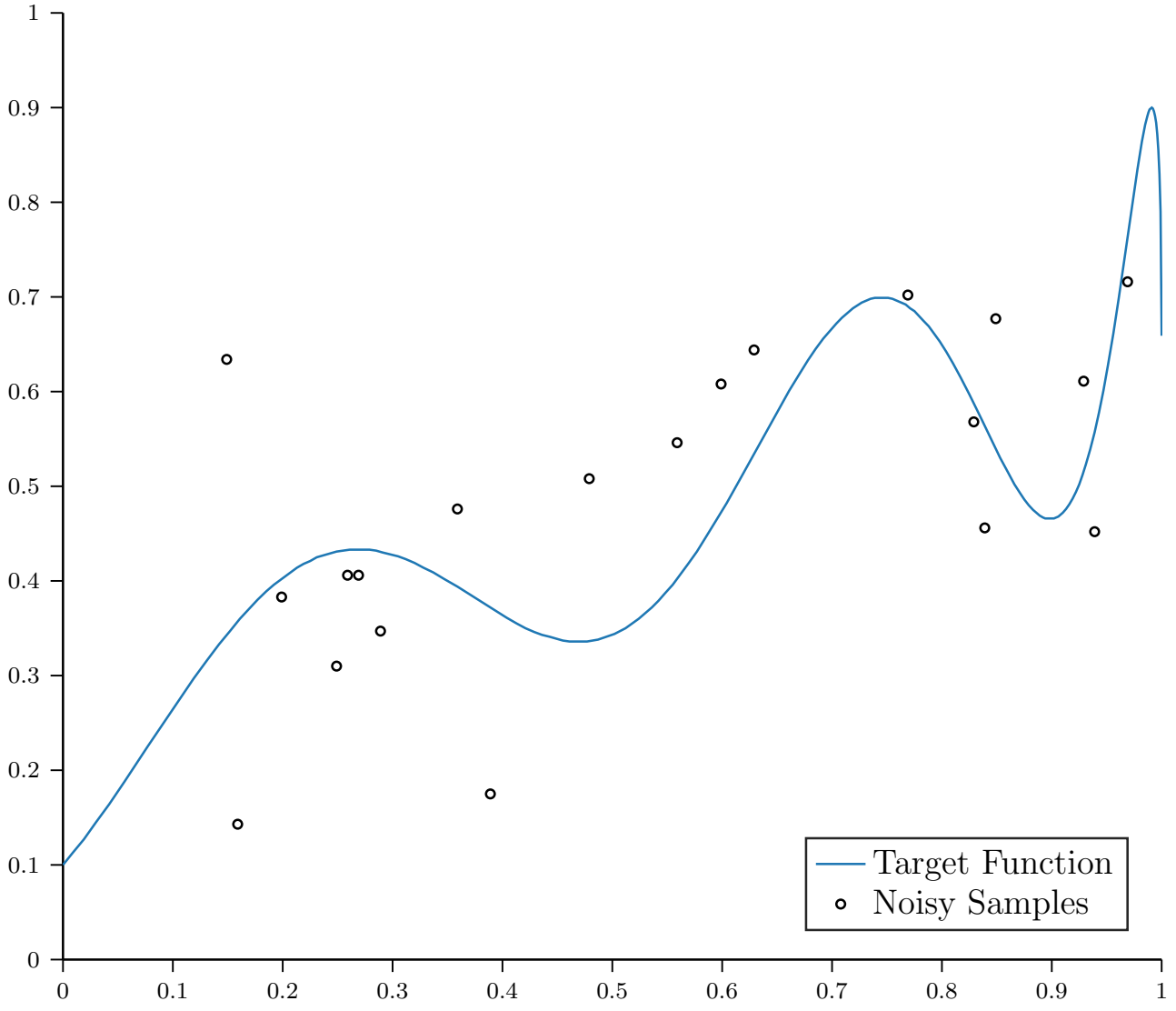
- $x \in \mathbb{R}$, $y \in \mathbb{R}$ and $N = 20$
- Targets are generated by a 10^{th}-order polynomial in x with additive Gaussian noise:

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

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

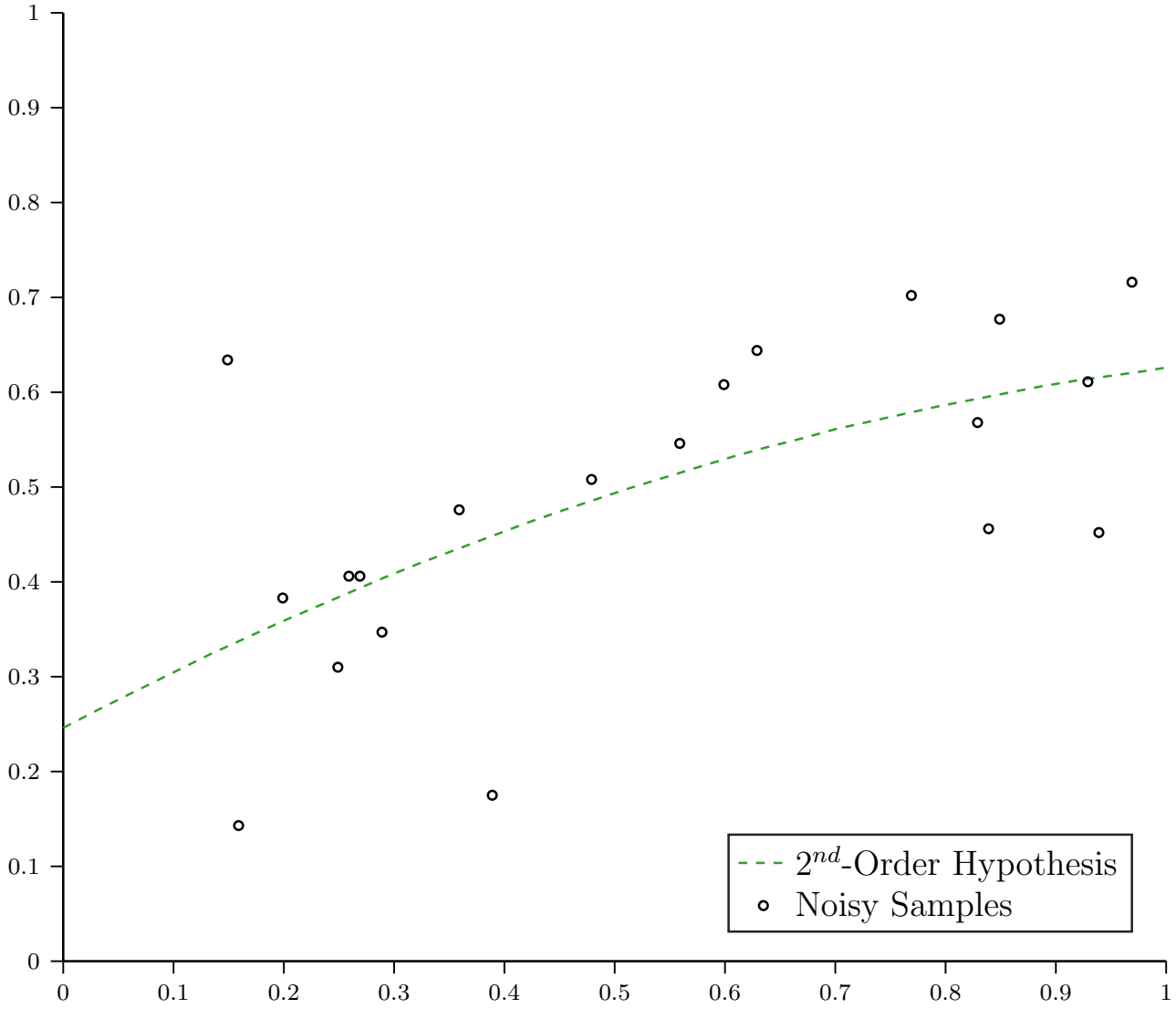
Noisy Targets

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



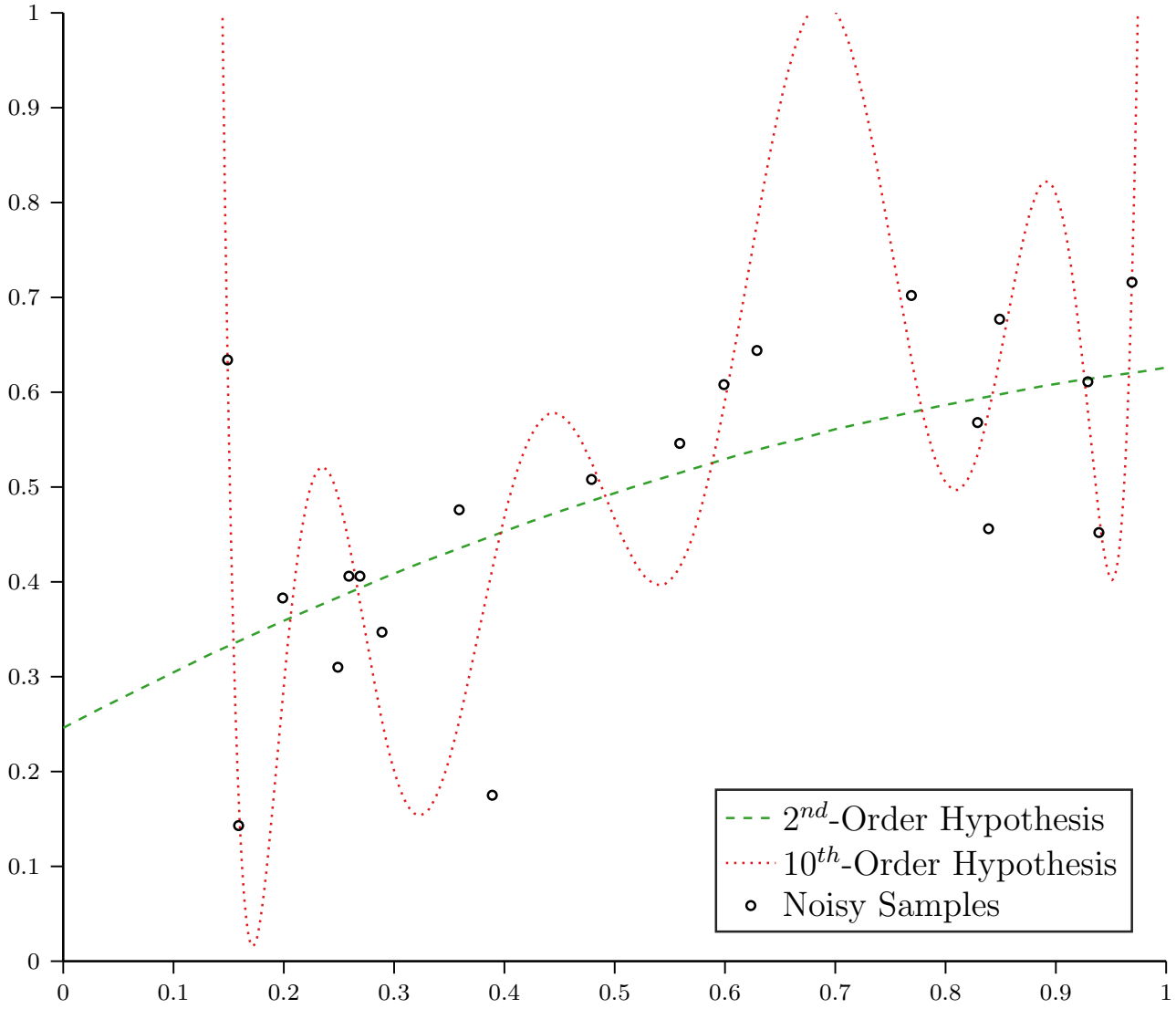
Noisy Targets

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



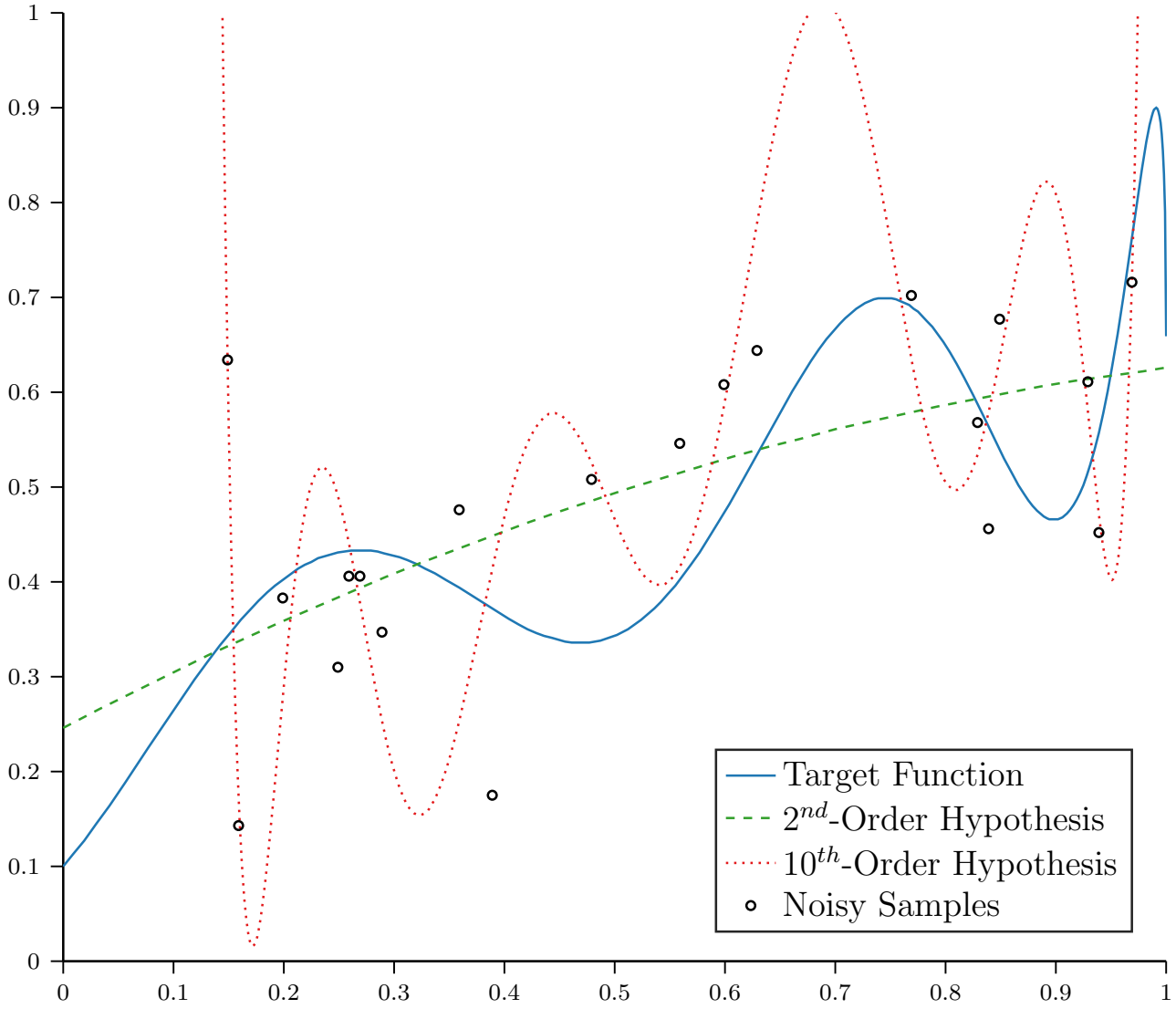
Noisy Targets

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



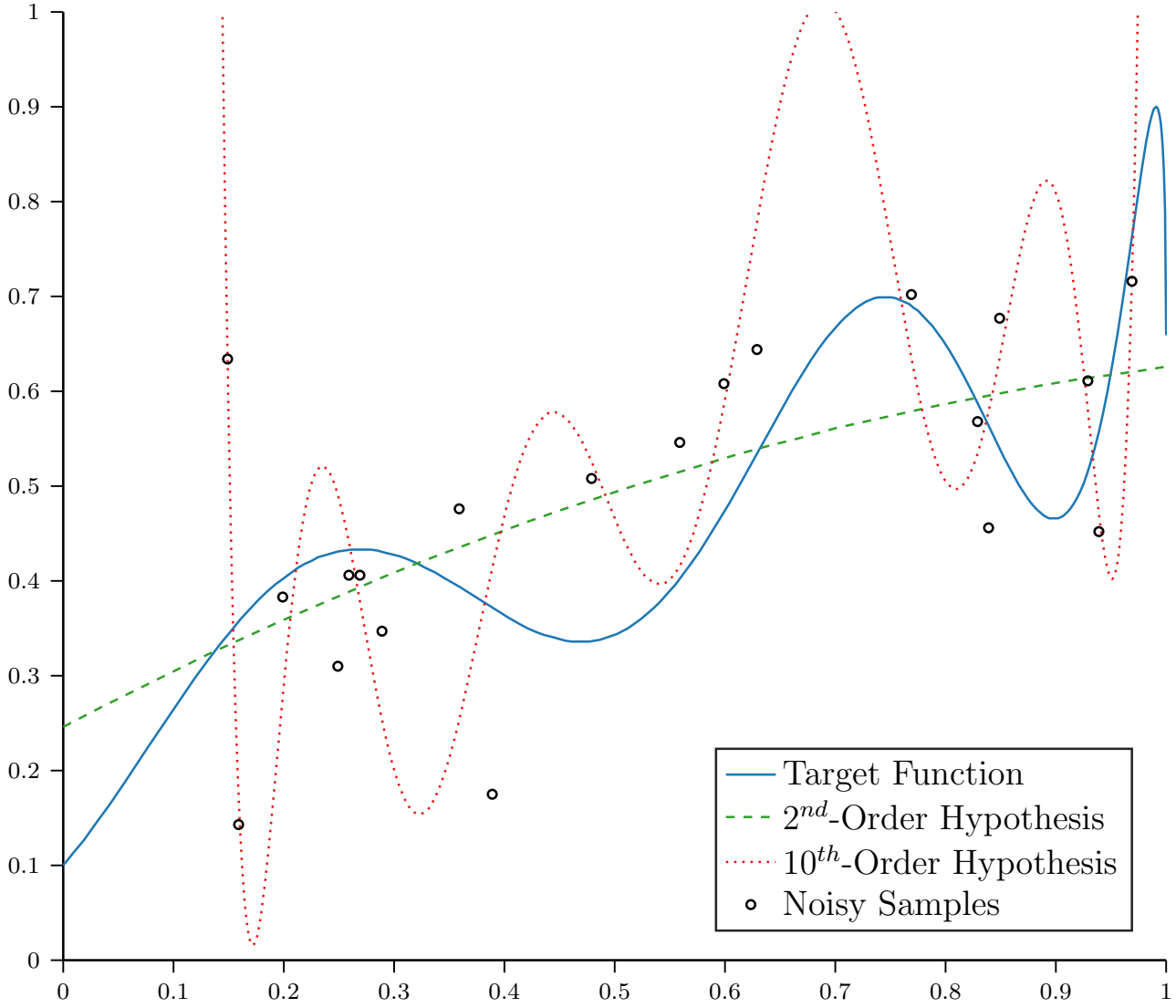
Noisy Targets

- 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

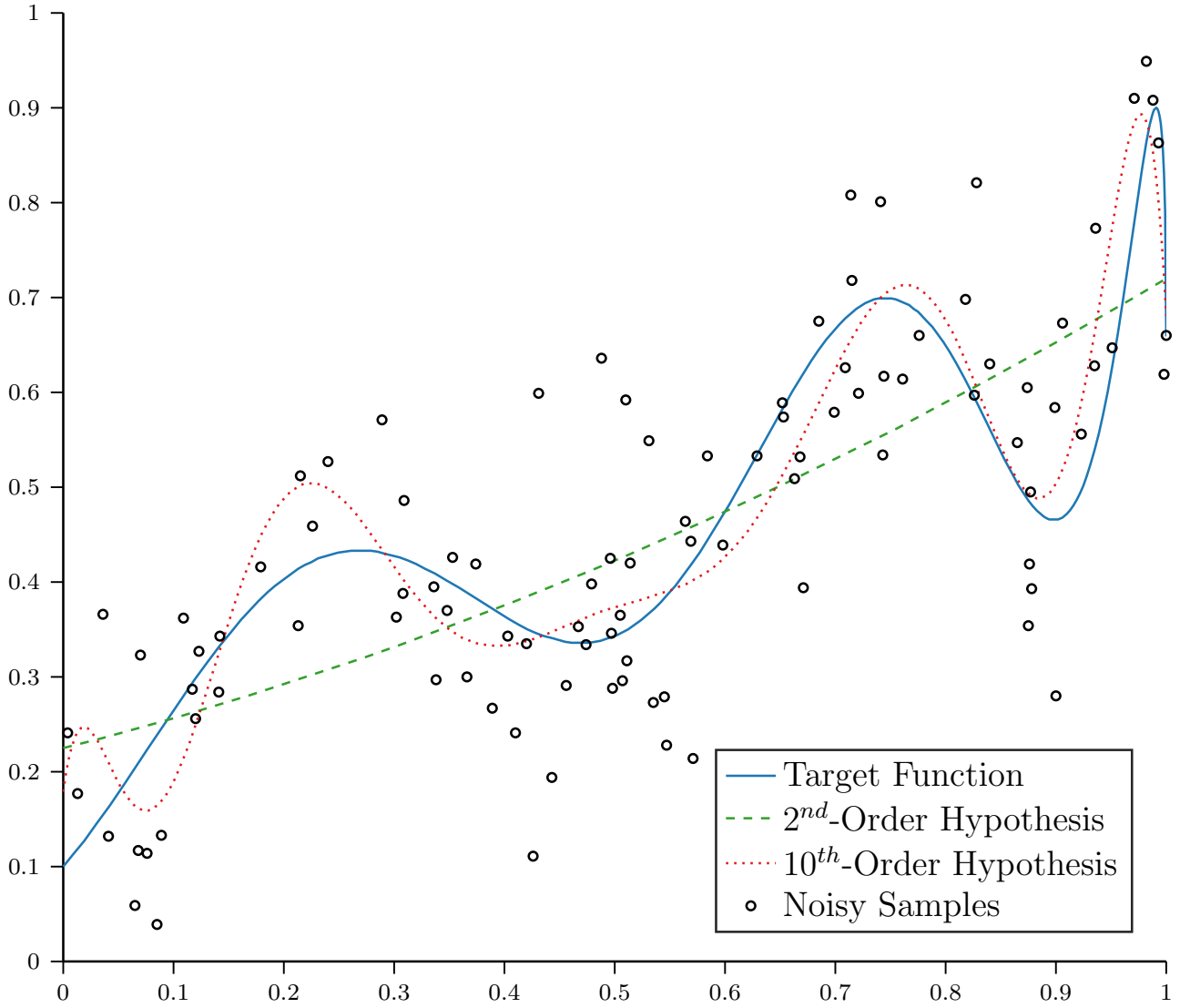
- $x \in \mathbb{R}$, $y \in \mathbb{R}$ and $N = 100$
- Targets are generated by a 10^{th}-order polynomial in x with additive Gaussian noise:

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

- $\mathcal{H}_2 = 2^{\text{nd}}$-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

	$\mathcal{H}_2$	$\mathcal{H}_{10}$
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 $X = \begin{bmatrix} 1 & \phi_{1,10}\left(x^{(1)}\right)^T \\ 1 & \phi_{1,10}\left(x^{(2)}\right)^T \\ \vdots & \vdots \\ 1 & \phi_{1,10}\left(x^{(N)}\right)^T \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ 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

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

- 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 $X = \begin{bmatrix} 1 & \phi_{1,10}\left(x^{(1)}\right)^T \\ 1 & \phi_{1,10}\left(x^{(2)}\right)^T \\ \vdots & \vdots \\ 1 & \phi_{1,10}\left(x^{(N)}\right)^T \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ 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 $X = \begin{bmatrix} 1 & \phi_{1,10}\left(x^{(1)}\right)^T \\ 1 & \phi_{1,10}\left(x^{(2)}\right)^T \\ \vdots & \vdots \\ 1 & \phi_{1,10}\left(x^{(N)}\right)^T \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ 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

- $\mathcal{H}_2 = 2^{\text{nd}}$-order polynomials
  - $\phi_{1,2}(x) = [x, x^2\ ]^T$
- Given $X = \begin{bmatrix} 1 & \phi_{1,2}\left(x^{(1)}\right)^T \\ 1 & \phi_{1,2}\left(x^{(2)}\right)^T \\ \vdots & \vdots \\ 1 & \phi_{1,2}\left(x^{(N)}\right)^T \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ find

  $\boldsymbol{\omega} = [\omega_0, \omega_1, \omega_2]^T$
  that minimizes

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

- Subject to nothing!

Soft Constraints

- More generally, ϕ can be any nonlinear transformation, e.g., exp, log, sin, sqrt, etc...
- Given $X = \begin{bmatrix} 1 & \phi_1(\boldsymbol{x}^{(1)}) & \cdots & \phi_m(\boldsymbol{x}^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_1(\boldsymbol{x}^{(N)}) & \cdots & \phi_m(\boldsymbol{x}^{(N)}) \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$, find $\boldsymbol{\omega}$ that minimizes

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

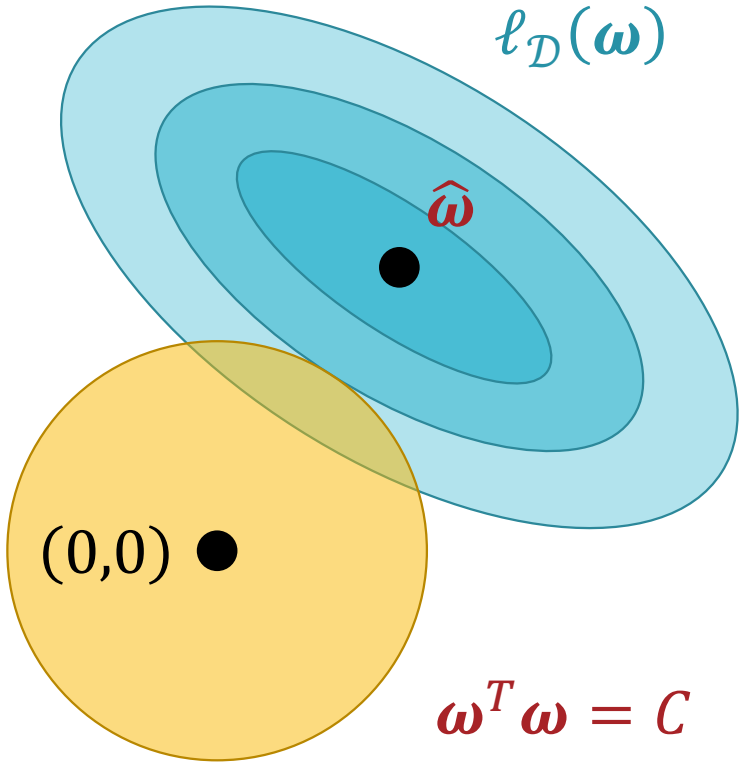
- Subject to:

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

Soft Constraints

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

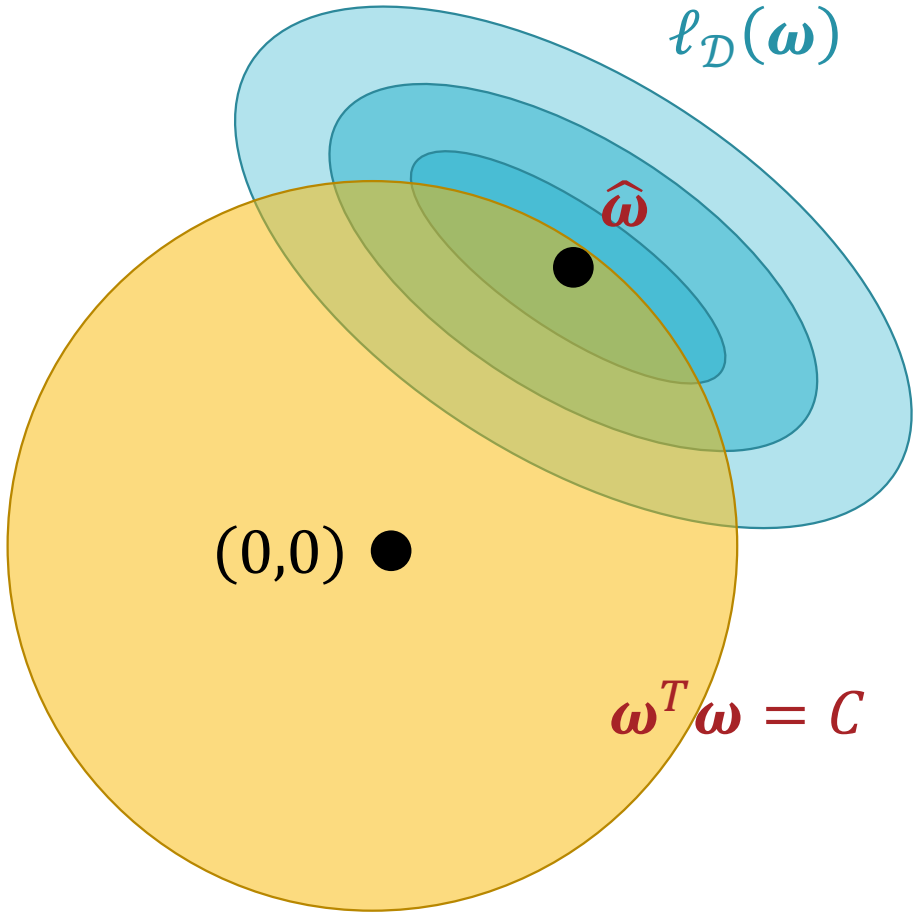
subject to $\boldsymbol{\omega}^T\boldsymbol{\omega} \leq C$



Soft Constraints

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

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



Soft Constraints

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

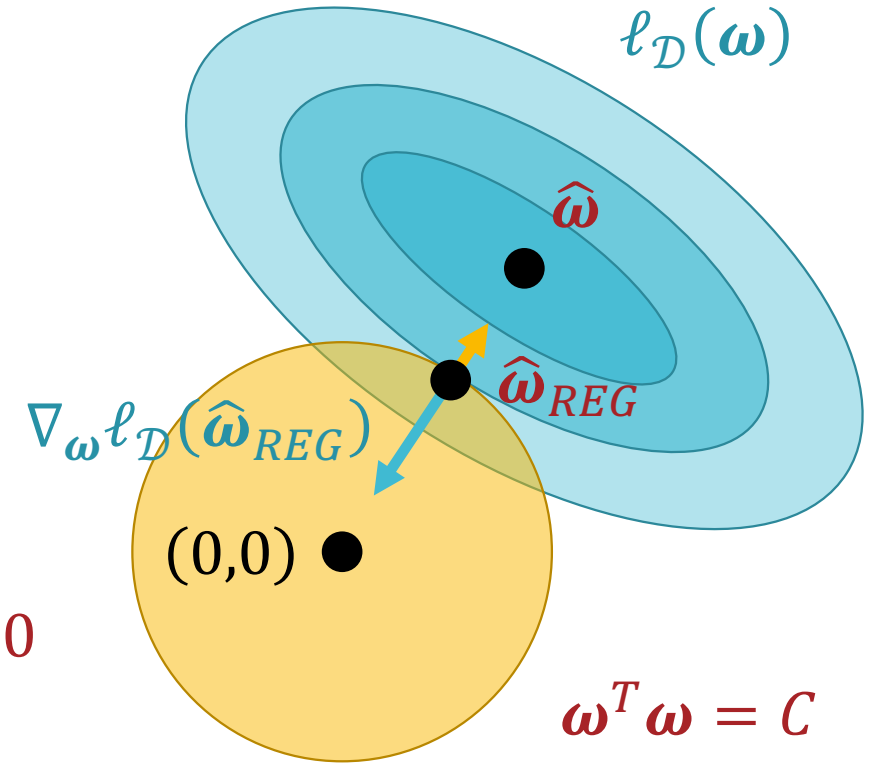
subject to $\boldsymbol{\omega}^T\boldsymbol{\omega} \leq C$

$\nabla_{\boldsymbol{\omega}}\ell_{\mathcal{D}}(\hat{\boldsymbol{\omega}}_{REG}) \propto -2\hat{\boldsymbol{\omega}}_{REG}$

$\nabla_{\boldsymbol{\omega}}\ell_{\mathcal{D}}(\hat{\boldsymbol{\omega}}_{REG}) = -2\lambda_C\hat{\boldsymbol{\omega}}_{REG}$

$\nabla_{\boldsymbol{\omega}}\ell_{\mathcal{D}}(\hat{\boldsymbol{\omega}}_{REG}) + 2\lambda_C\hat{\boldsymbol{\omega}}_{REG} = 0$

$\nabla_{\boldsymbol{\omega}}(\ell_{\mathcal{D}}(\hat{\boldsymbol{\omega}}_{REG}) + \lambda_C(\hat{\boldsymbol{\omega}}_{REG})^T\hat{\boldsymbol{\omega}}_{REG}) = 0$



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

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

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

$\Updownarrow$

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

Ridge Regression

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

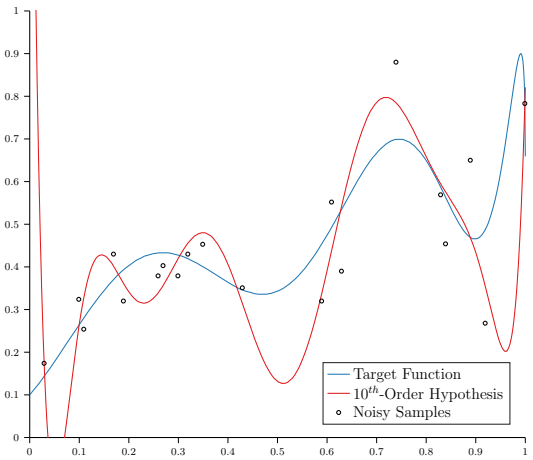
$$\nabla_{\boldsymbol{\omega}} \ell_{\mathcal{D}}^{AUG}(\boldsymbol{\omega}) = 2(X^T X \boldsymbol{\omega} - X^T \boldsymbol{y} + \lambda_C \boldsymbol{\omega})$$

$$2(X^T X \hat{\boldsymbol{\omega}}_{REG} - X^T \boldsymbol{y} + \lambda_C \hat{\boldsymbol{\omega}}_{REG}) = 0$$

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

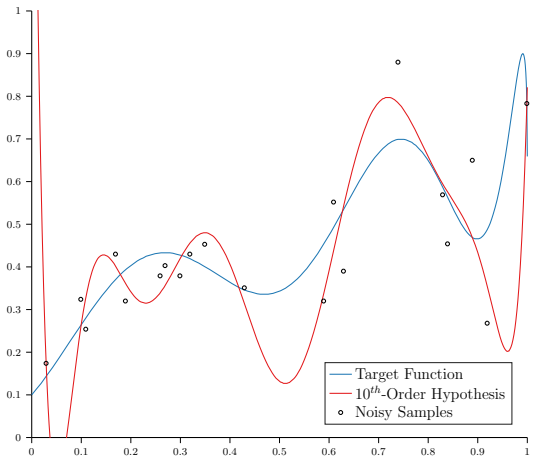
$$\hat{\boldsymbol{\omega}}_{REG} = (X^T X + \underbrace{\lambda_C I_{D+1}})^{-1} X^T \boldsymbol{y}$$

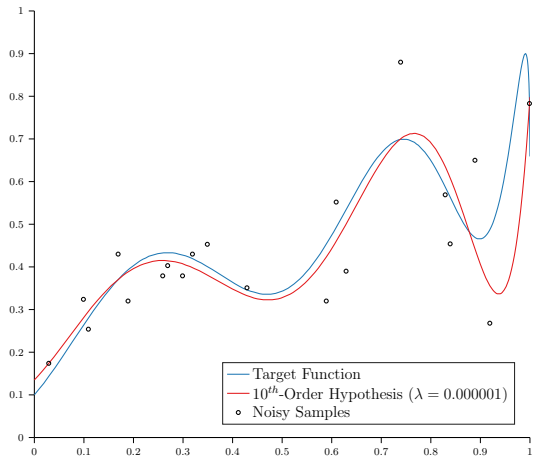
Adding this positive ($\lambda_C \geq 0$) diagonal matrix can help if $X^T X$ is not invertible!

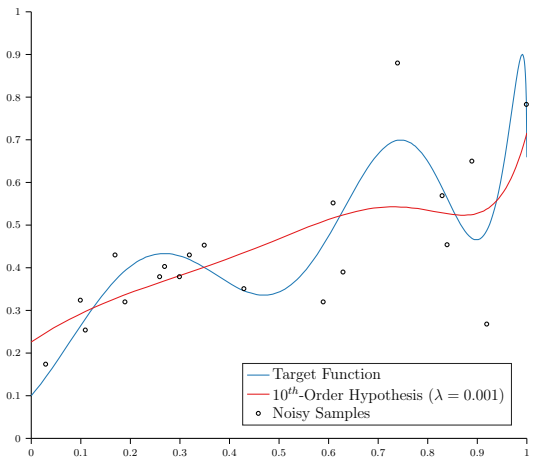


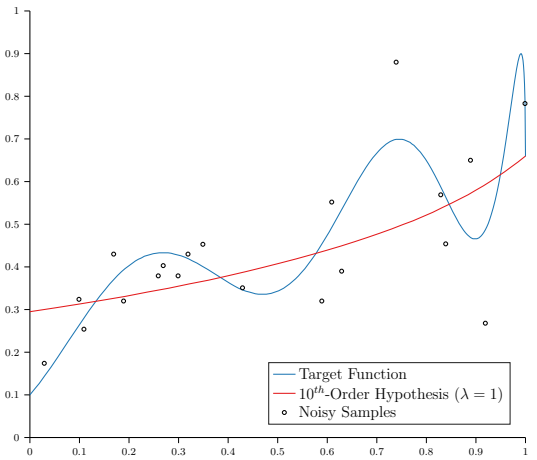
Ridge Regression

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





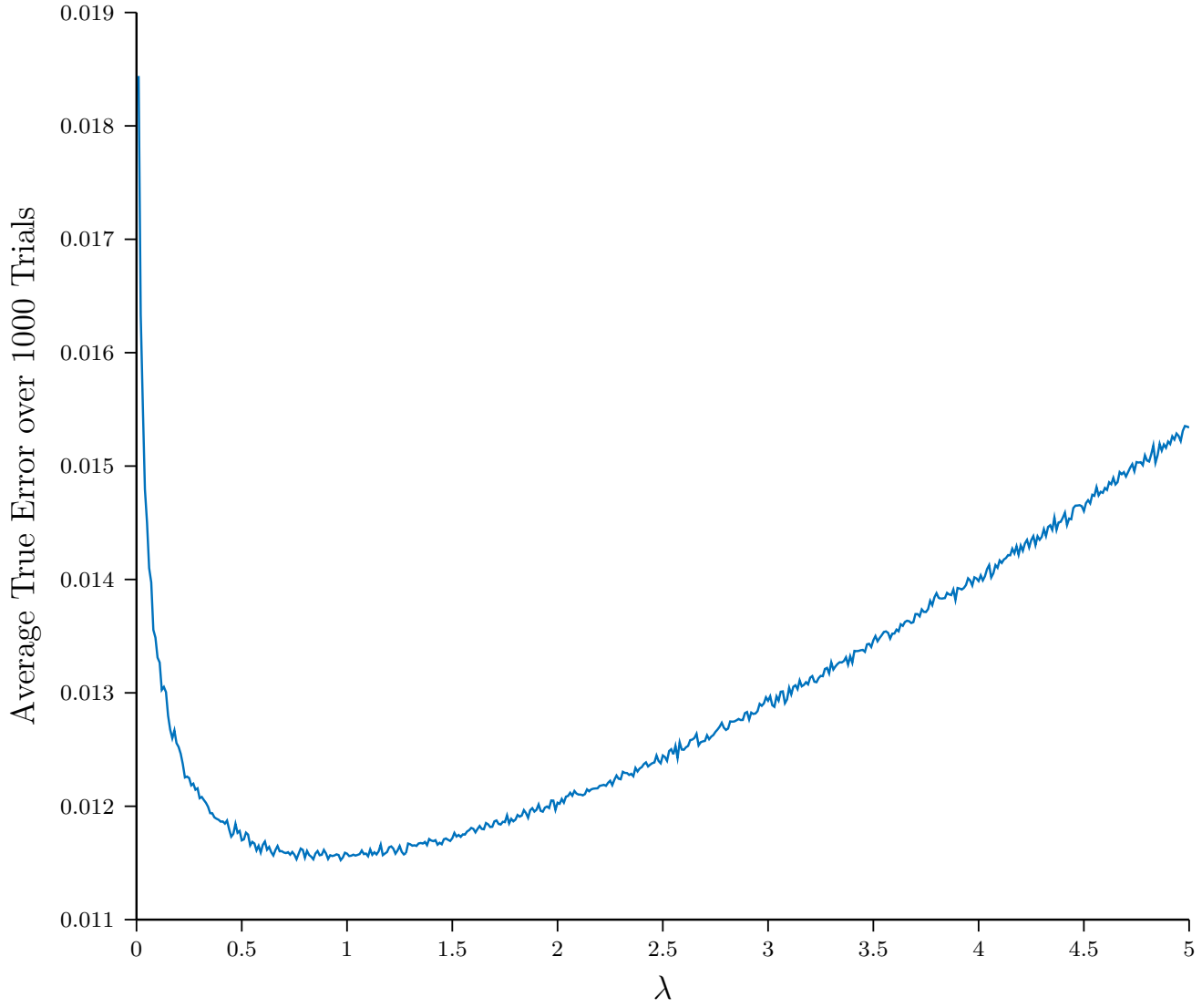




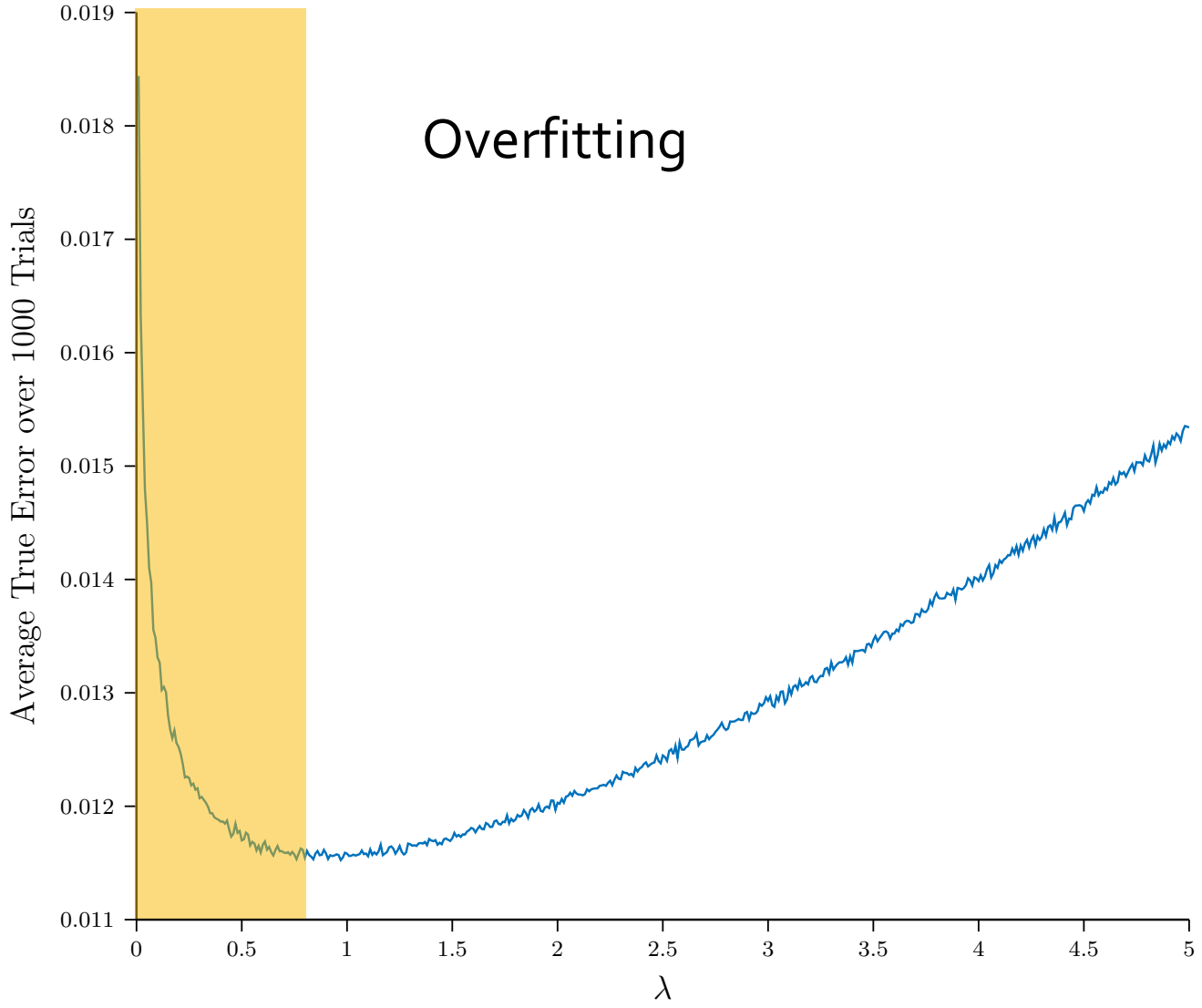
Ridge Regression

	$\lambda_C = 0$	$\lambda_C = 10^{-6}$	$\lambda_C = 10^{-3}$	$\lambda_C = 1$
True Error	0.059	0.006	0.008	0.011
	Overfit	Nice!	Wait...	Underfit

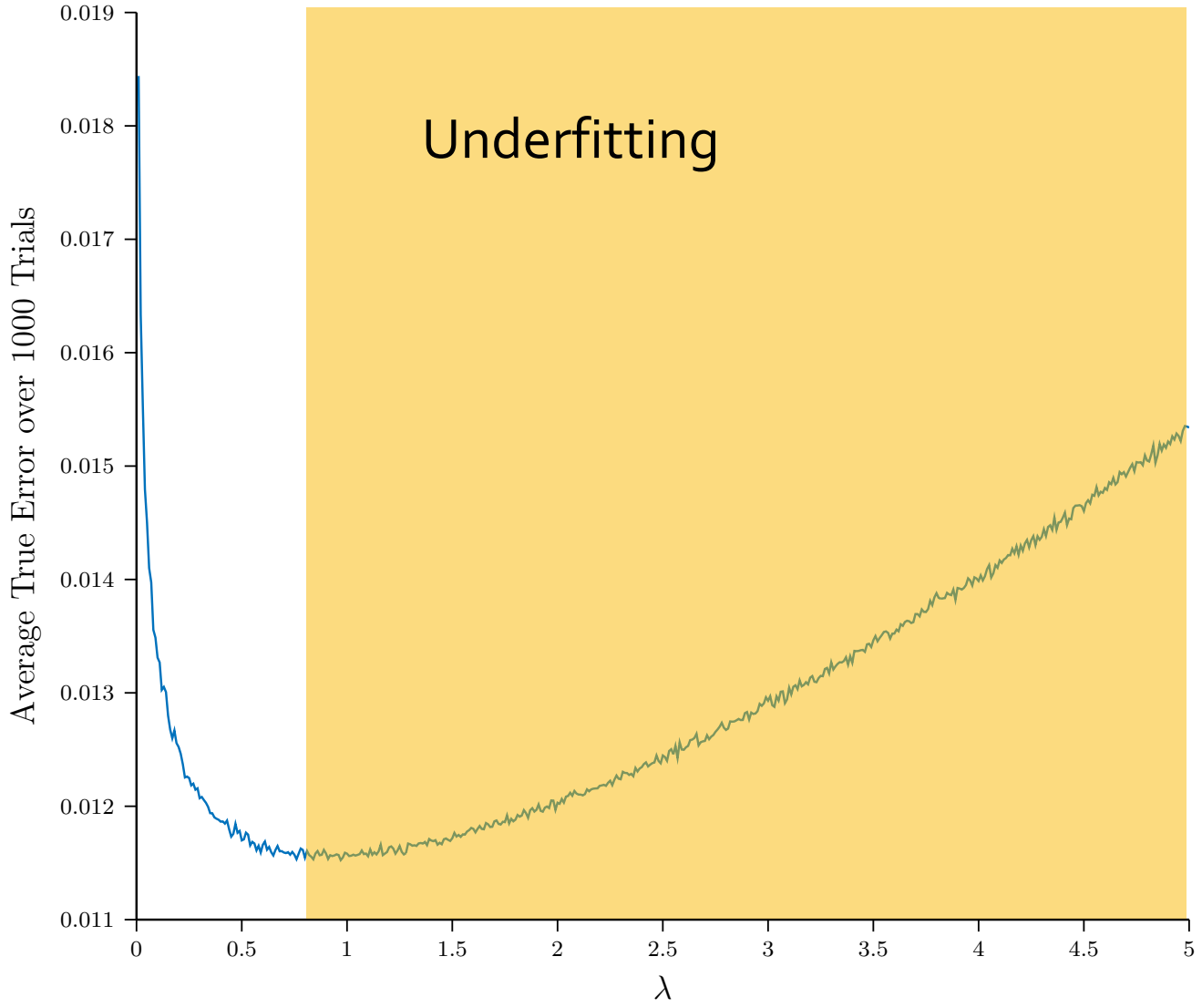
Setting λ



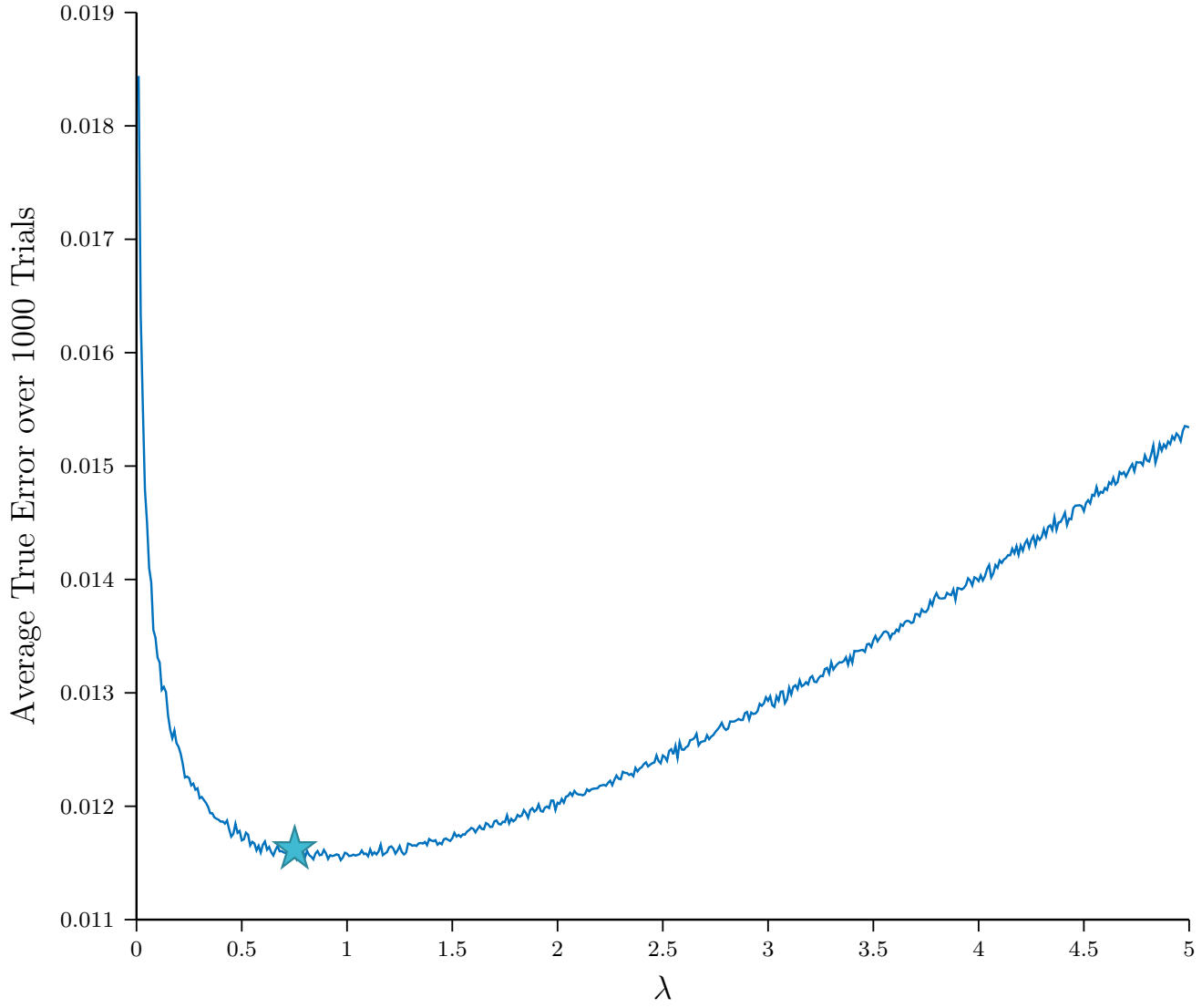
Setting λ



Setting λ

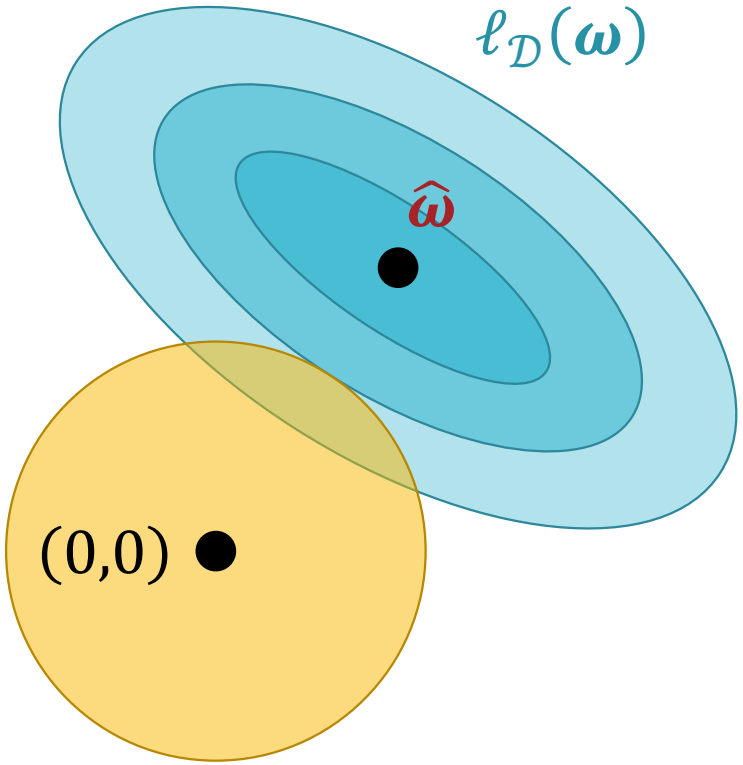


Setting λ

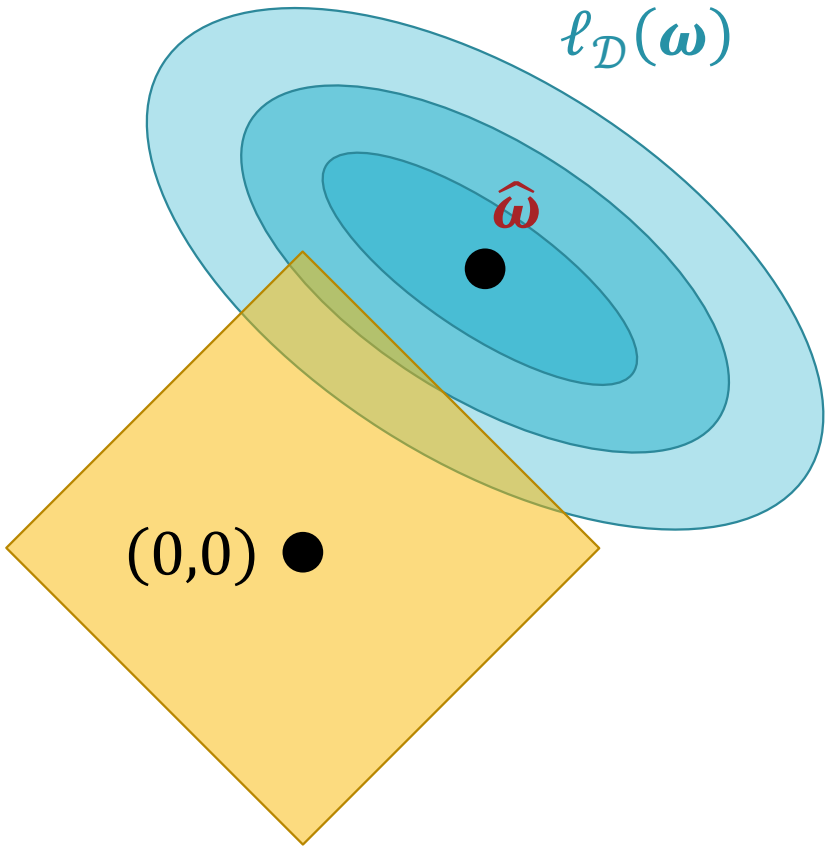


Other Regularizers

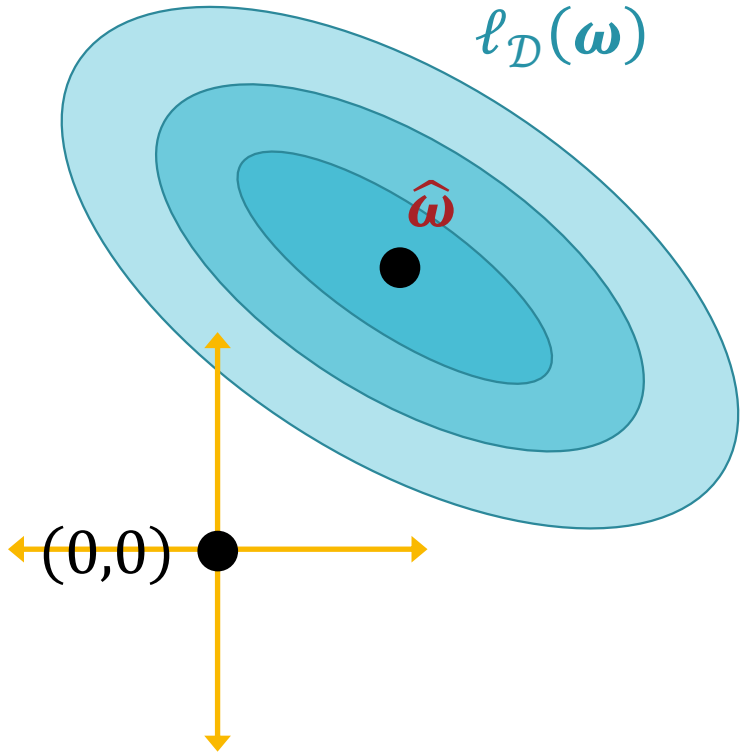
$\ell_{\mathcal{D}}(\boldsymbol{\omega}) + \lambda r(\boldsymbol{\omega})$		
Ridge or $L2$	$r(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|_2^2 = \sum_{d=0}^{D} \omega_d^2$	Encourages small weights
Lasso or $L1$	$r(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|_1 = \sum_{d=0}^{D} \|\omega_d\|$	Encourages 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