10-301/601: Introduction to Machine Learning Lecture 11 – Regularization

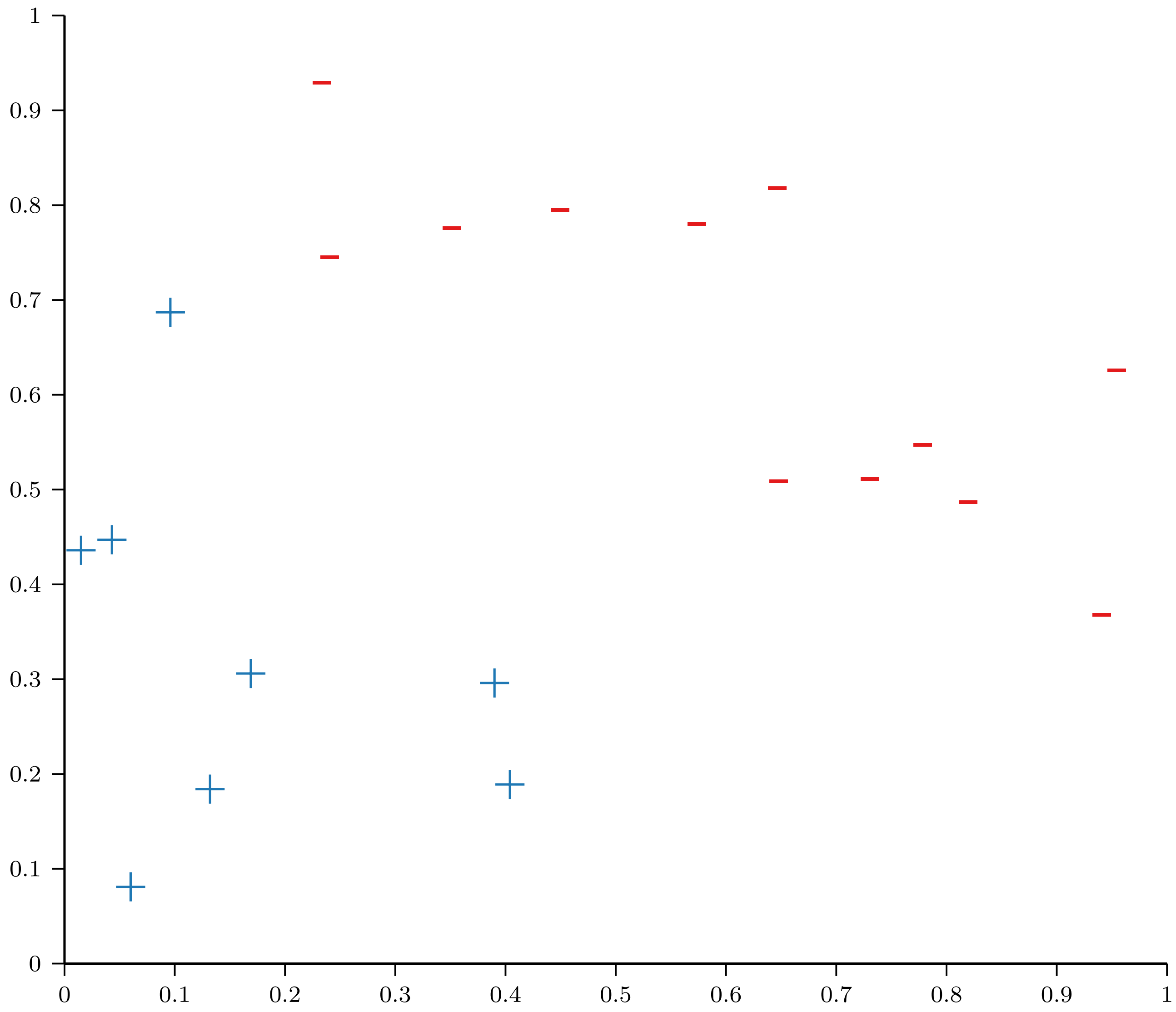
Henry Chai

6/7/23

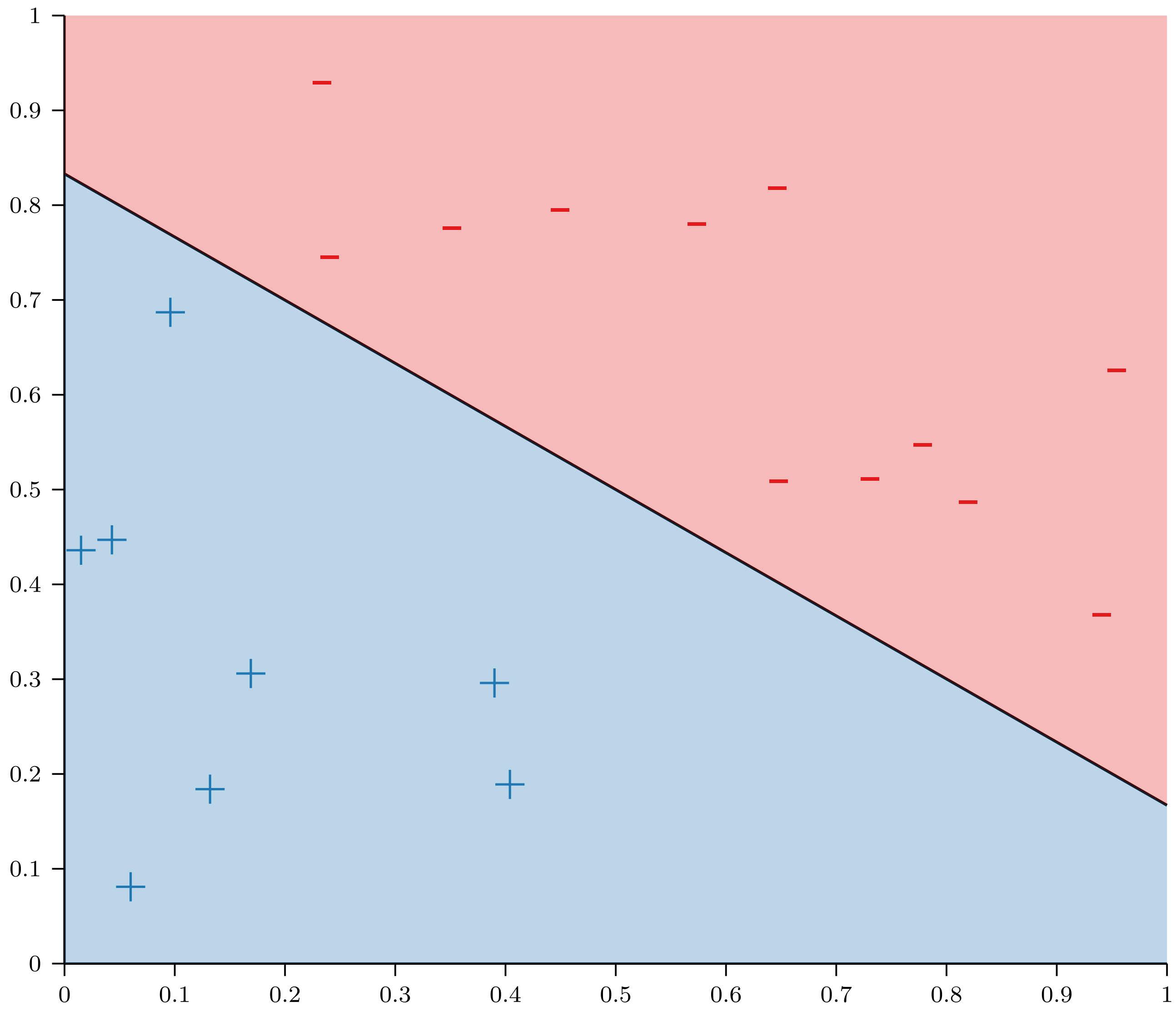
Front Matter

- Announcements:
  - PA3 released 6/8 (tomorrow), due 6/15 at 11:59 PM
  - Midterm on 6/23, two weeks from Friday
    - Practice problems for the Midterm will be posted to the course website on Friday, under Recitations
- Recommended Readings:
  - Murphy, Chapter 7.5

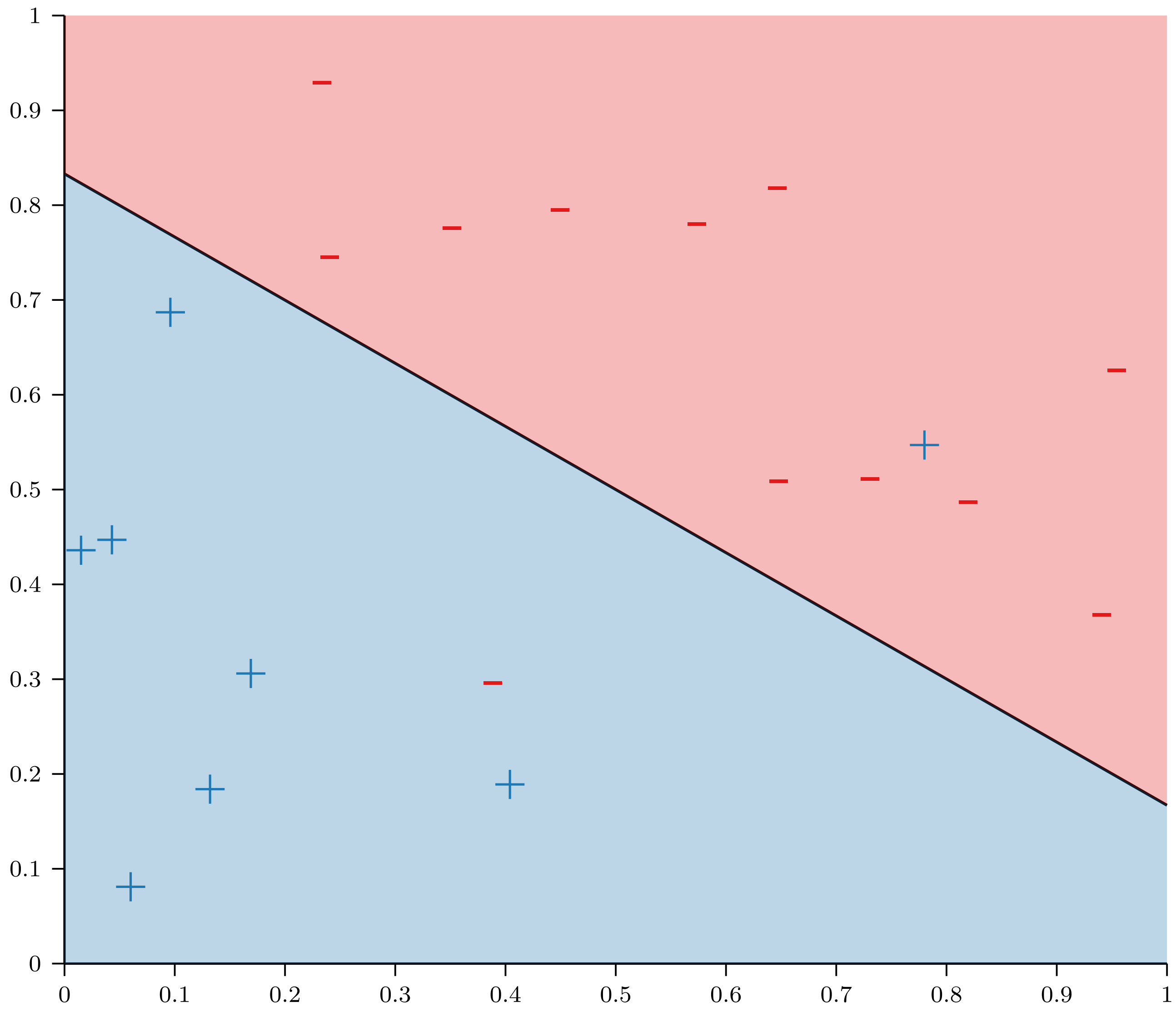
Linear Models



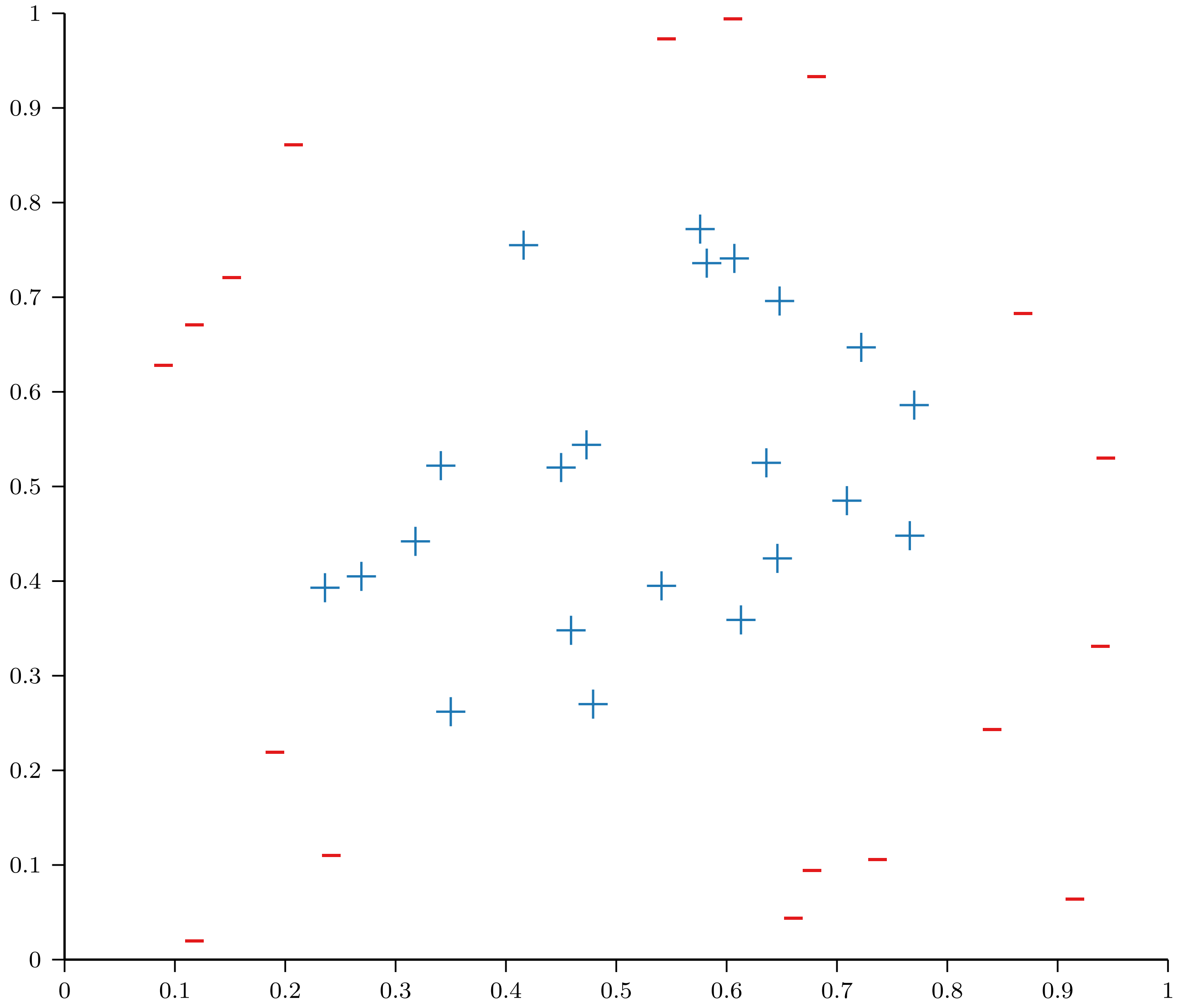
Linear Models



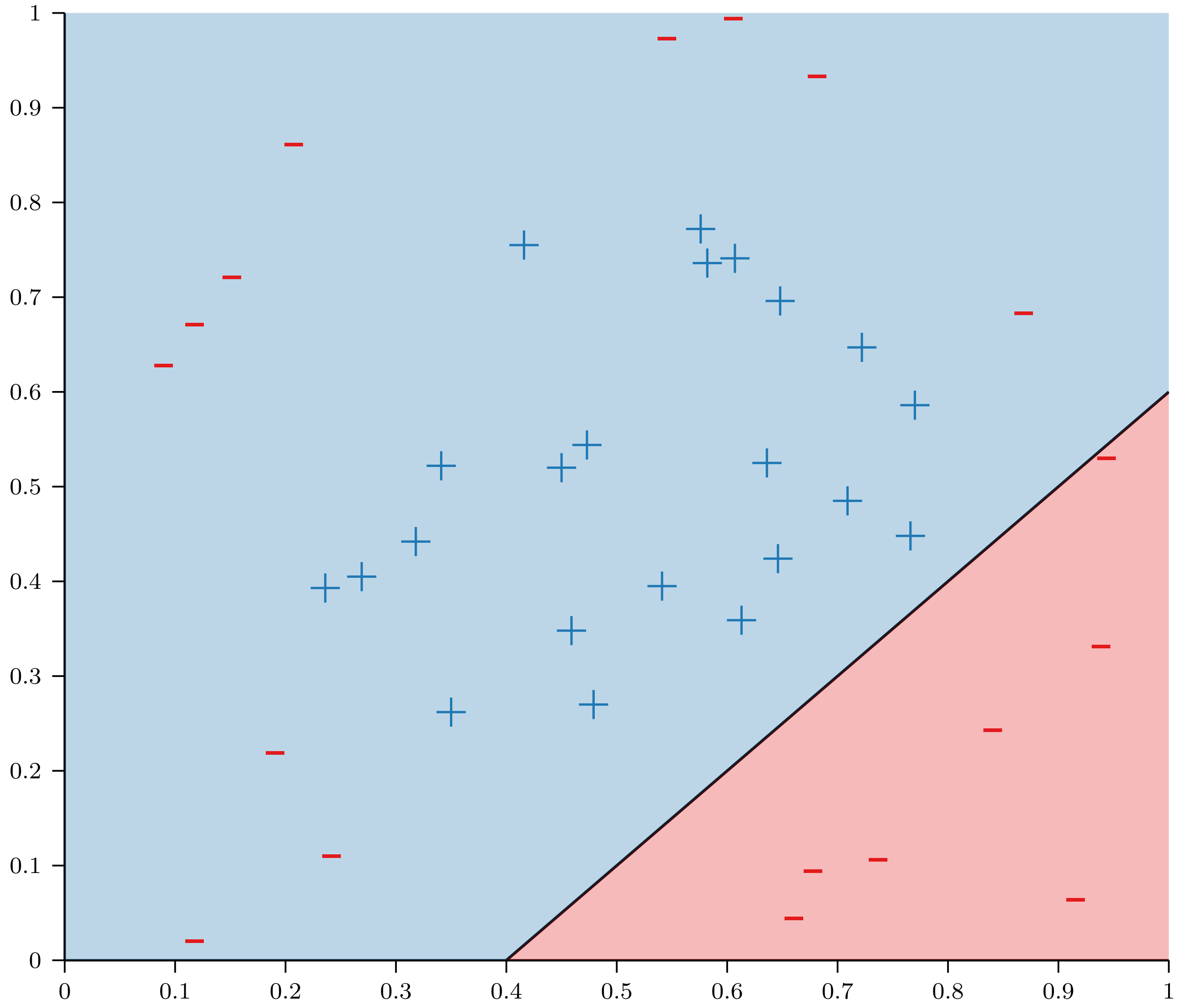
Linear Models



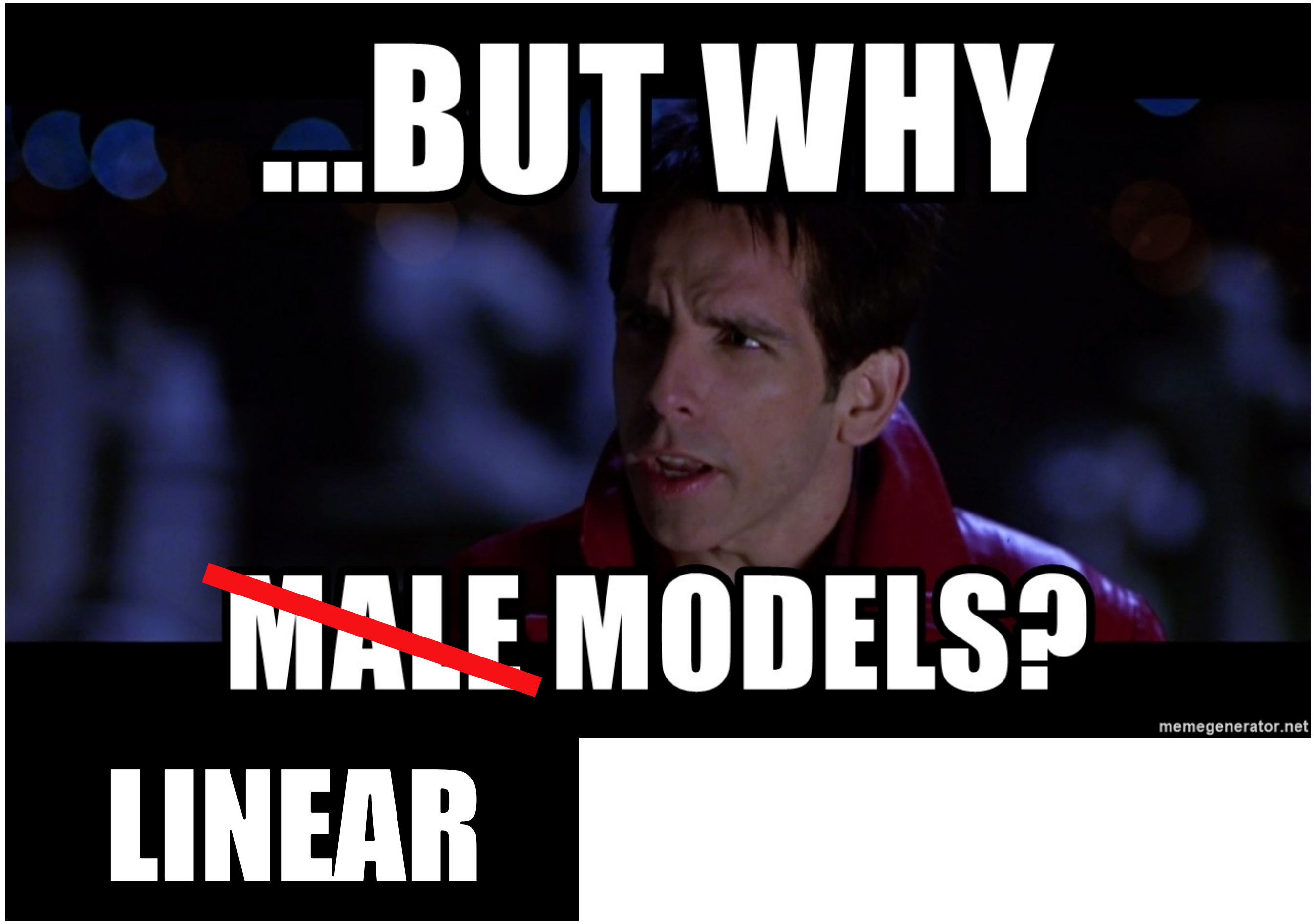
Linear Models?



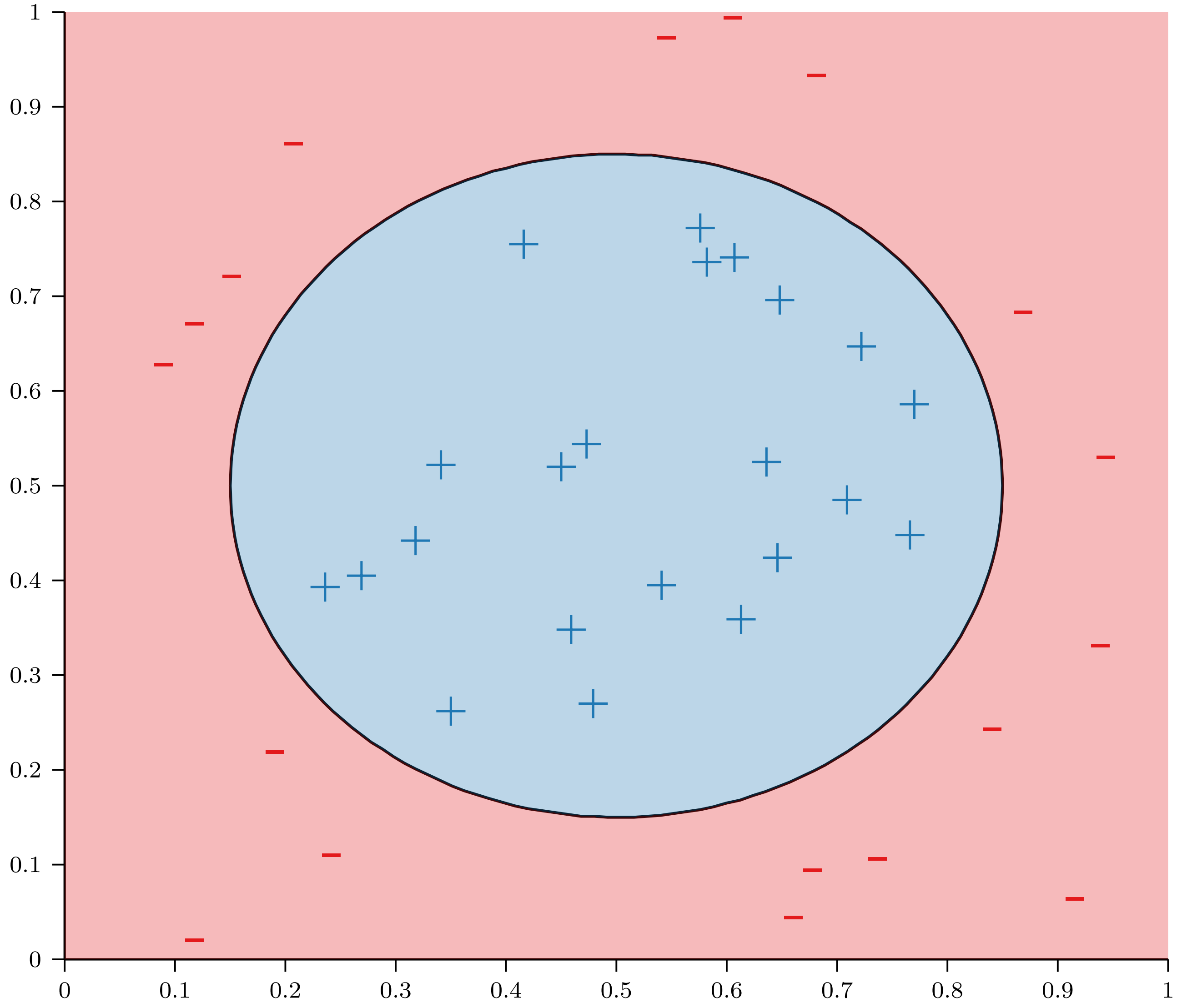
Linear Models?



Linear Models?



Nonlinear Models

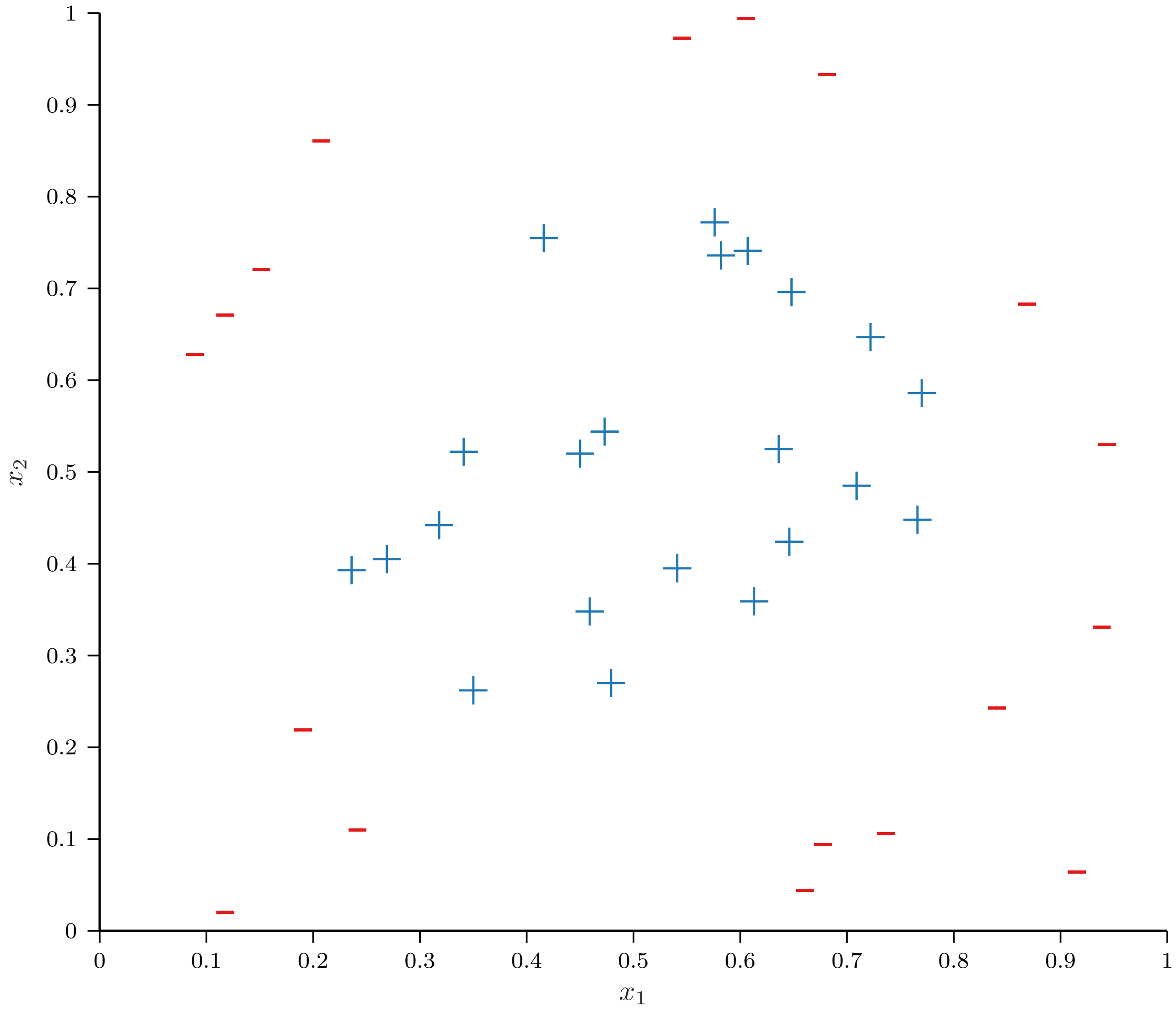


Feature Transforms

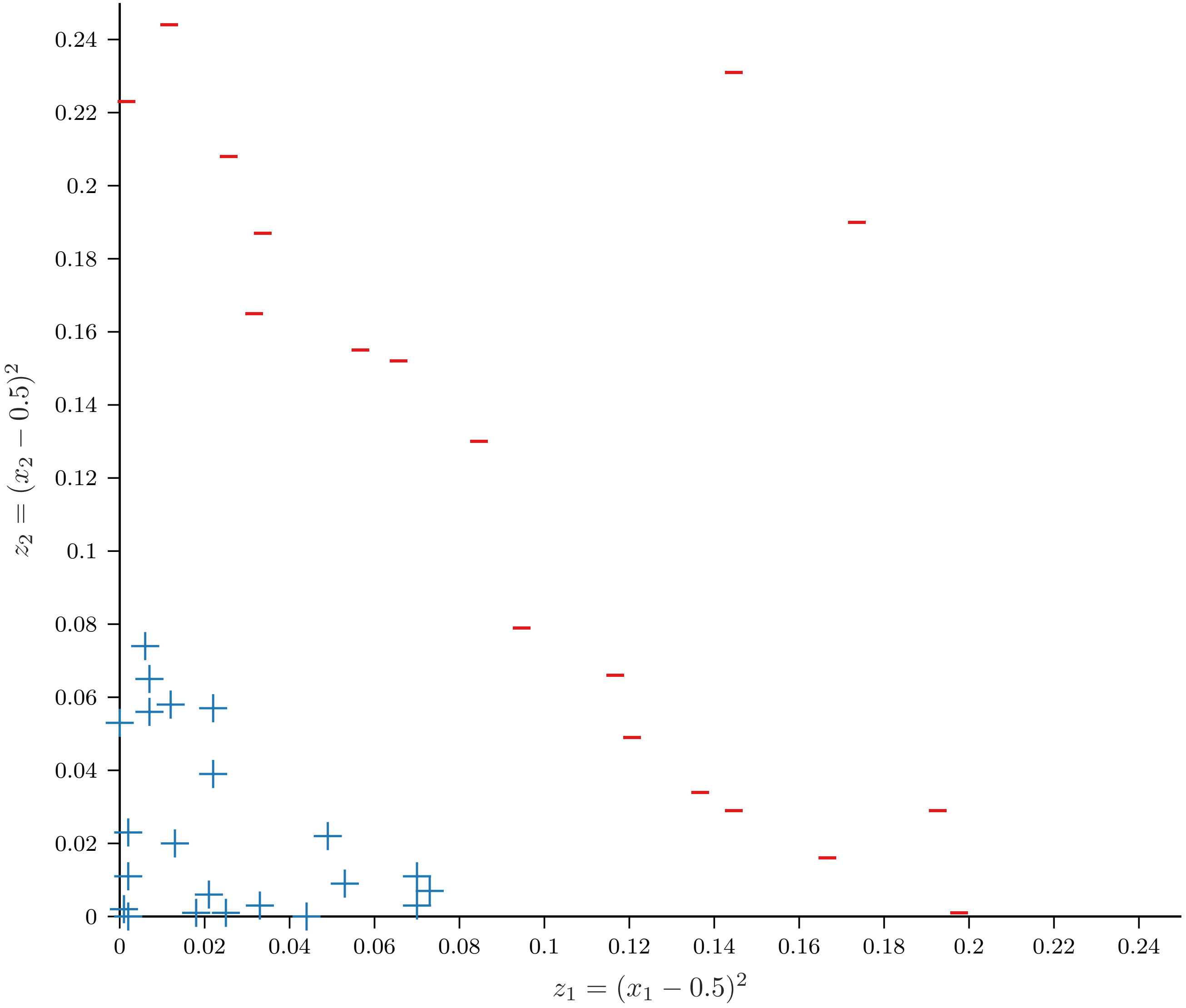
- Given D-dimensional inputs $\boldsymbol{x} = [x_1, \ldots, x_D]$, first compute some transformation of our input, e.g.,

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

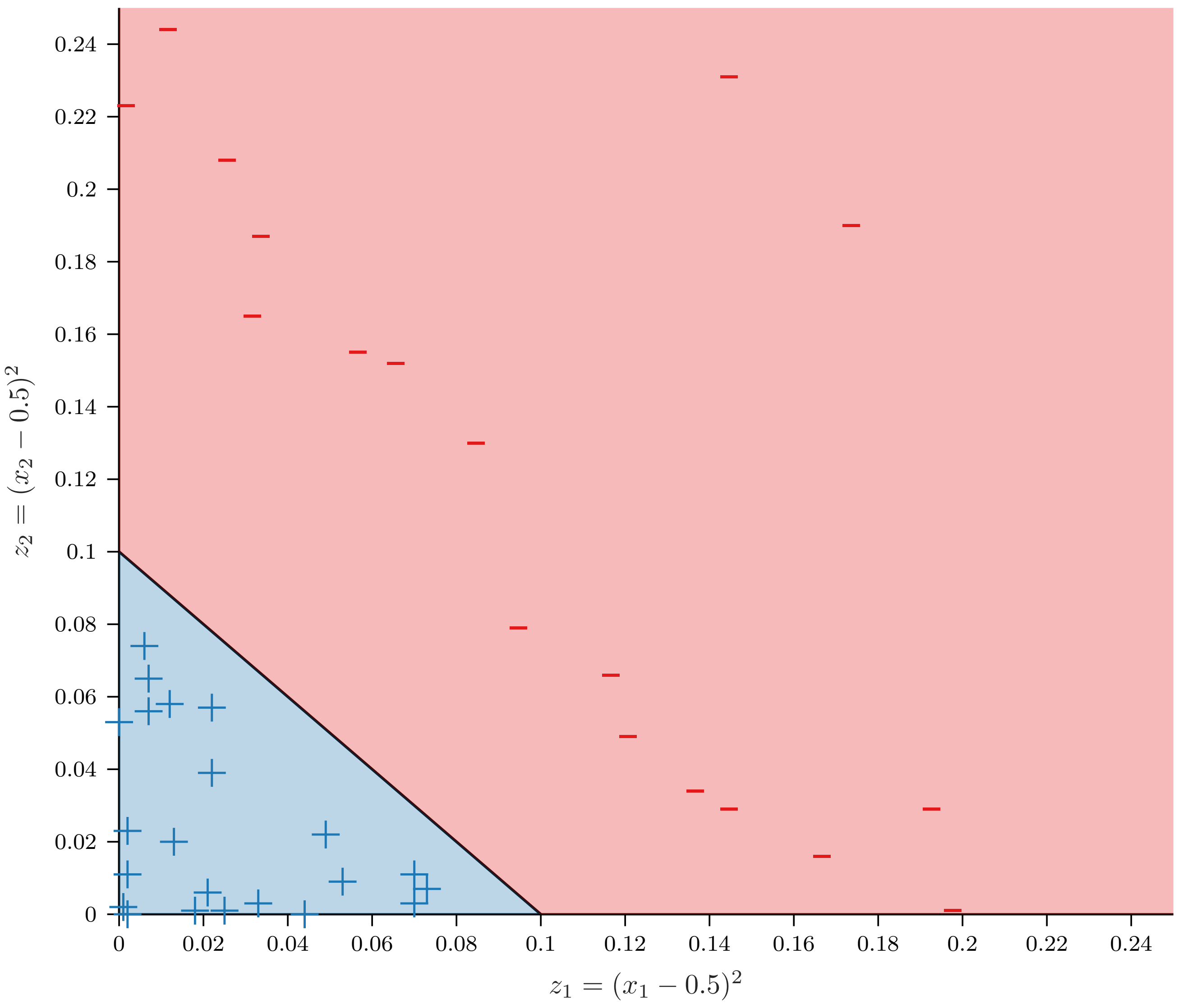
Nonlinear Models



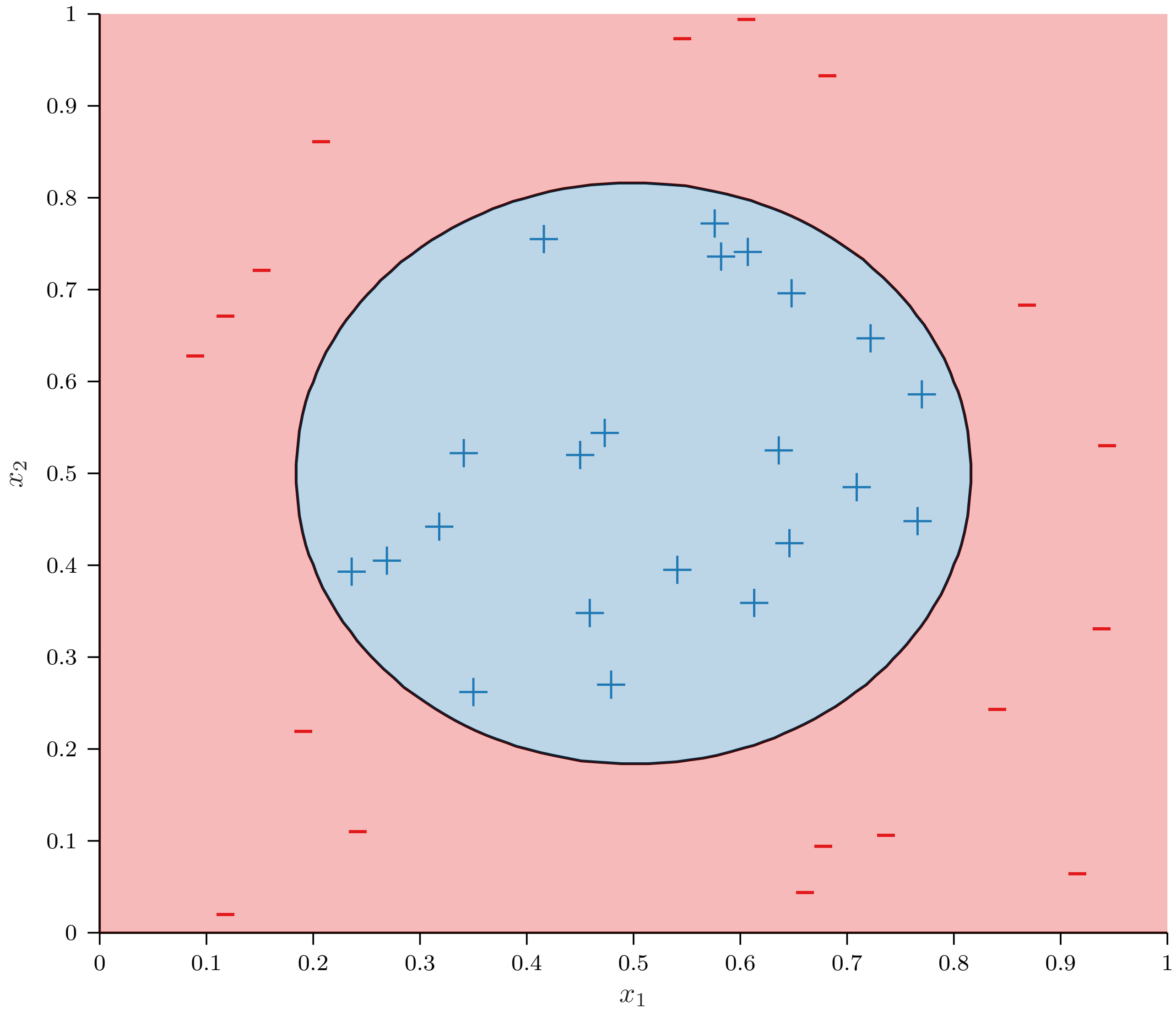
Nonlinear Models



Nonlinear Models



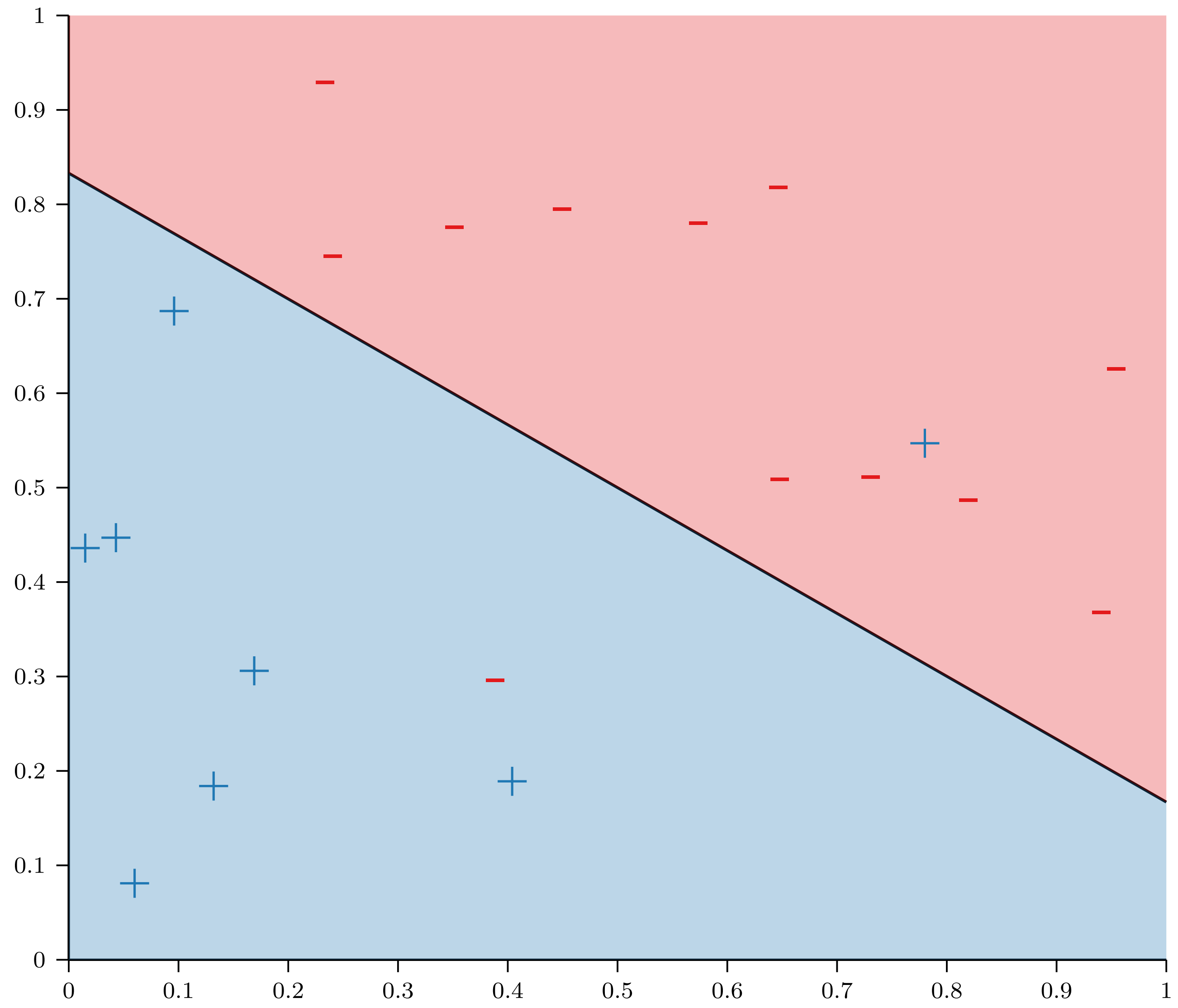
Nonlinear Models



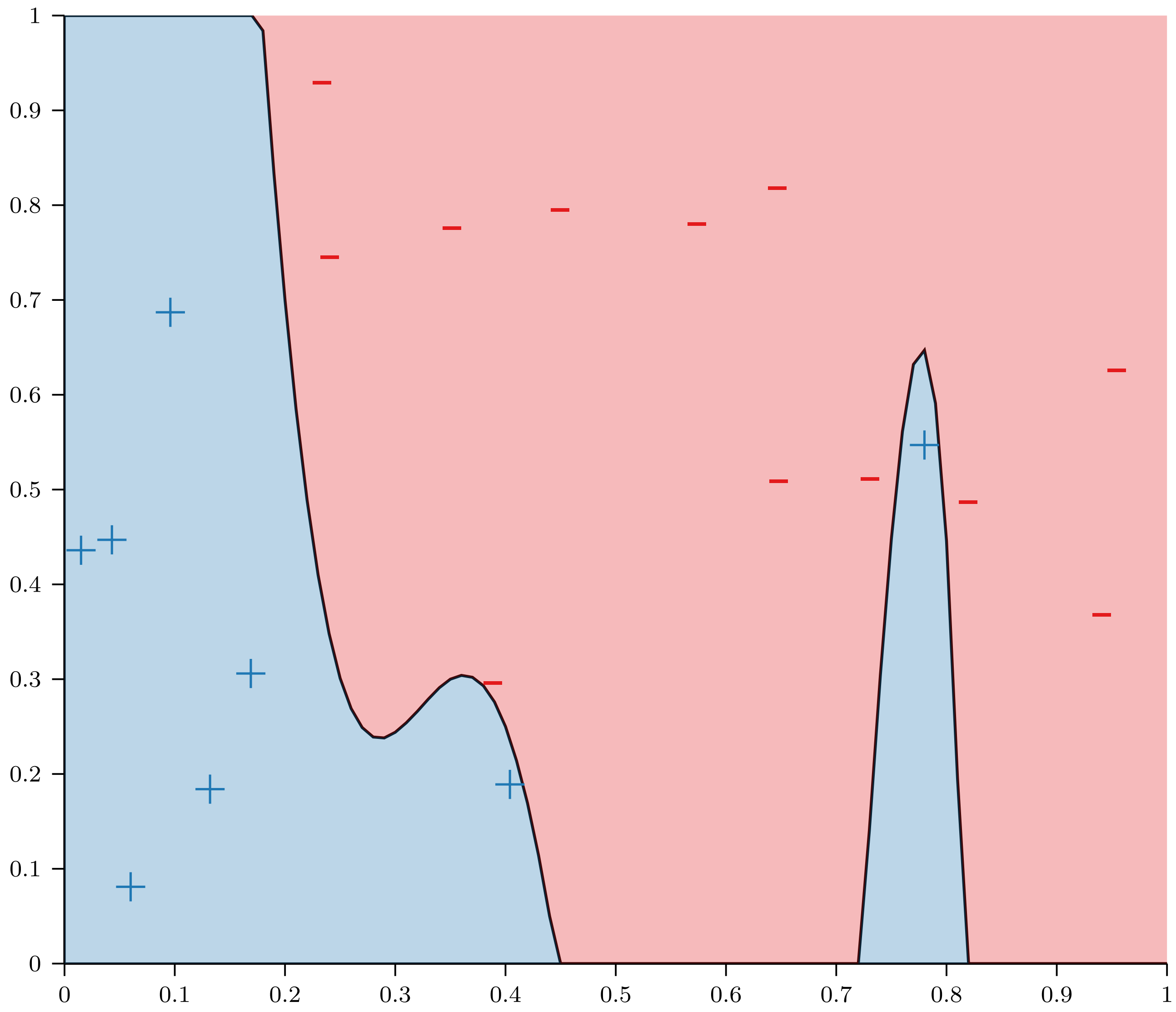
General Q^{th}-order Transforms

- $\phi_{2,2}([x_1, x_2]) = [x_1, x_2, x_1^2, x_1 x_2, x_2^2]$
- $\phi_{2,3}([x_1, x_2]) = [x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3]$
- $\phi_{2,4}([x_1, x_2]) =$
  $[x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3, x_1^4, x_1^3 x_2, x_1^2 x_2^2, x_1 x_2^3, x_2^4]$
- $\phi_{2,Q}$ 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

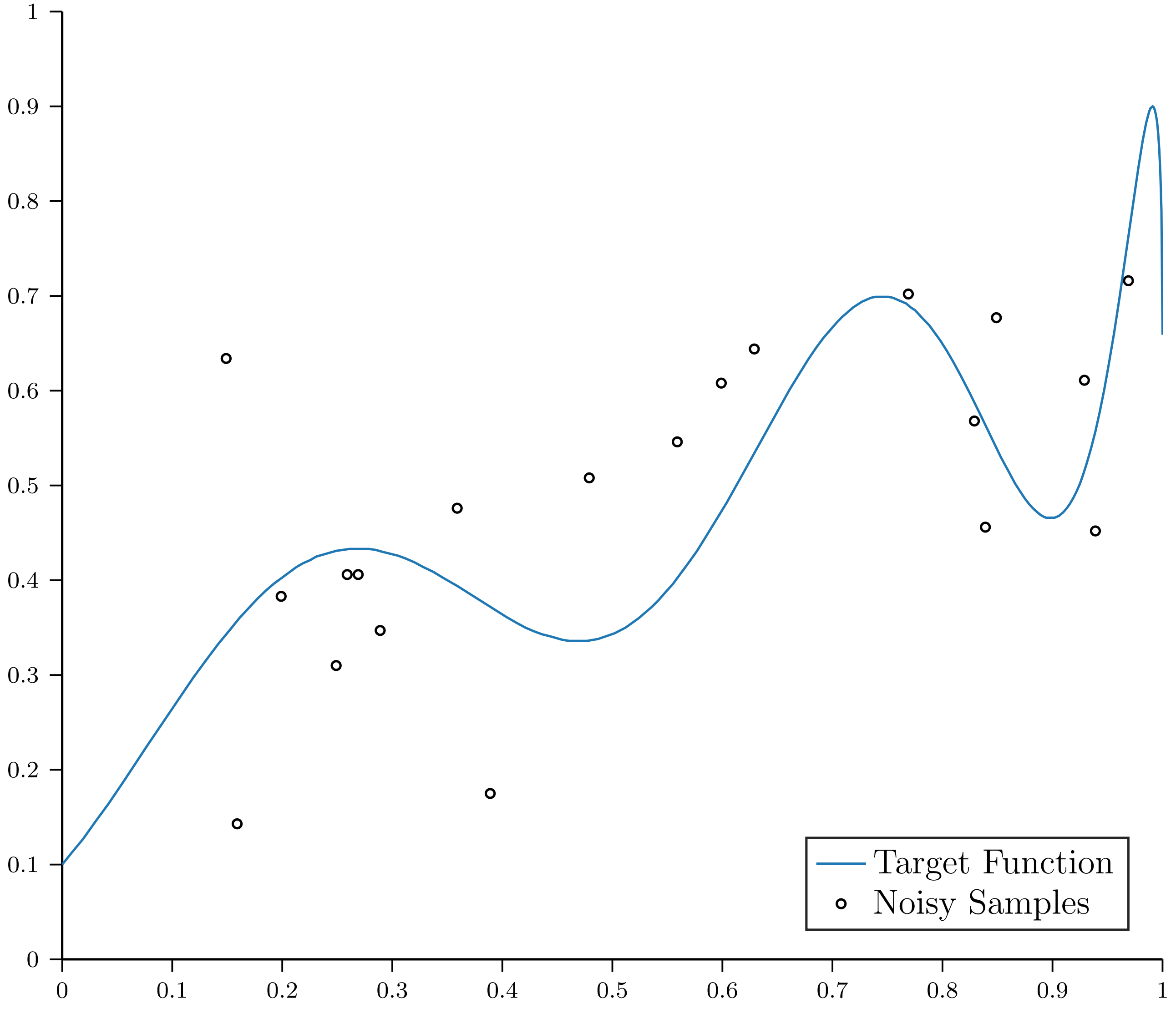
- $x \in \mathbb{R}$, $y \in \mathbb{R}$ and $N = 20$
- Targets are generated by a 10^{th}-order polynomial in x with additive Gaussian noise:

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

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

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



⚠ When survey is active, respond at **pollev.com/301601polls**

Lecture 11 Polls

0 done

0 underway

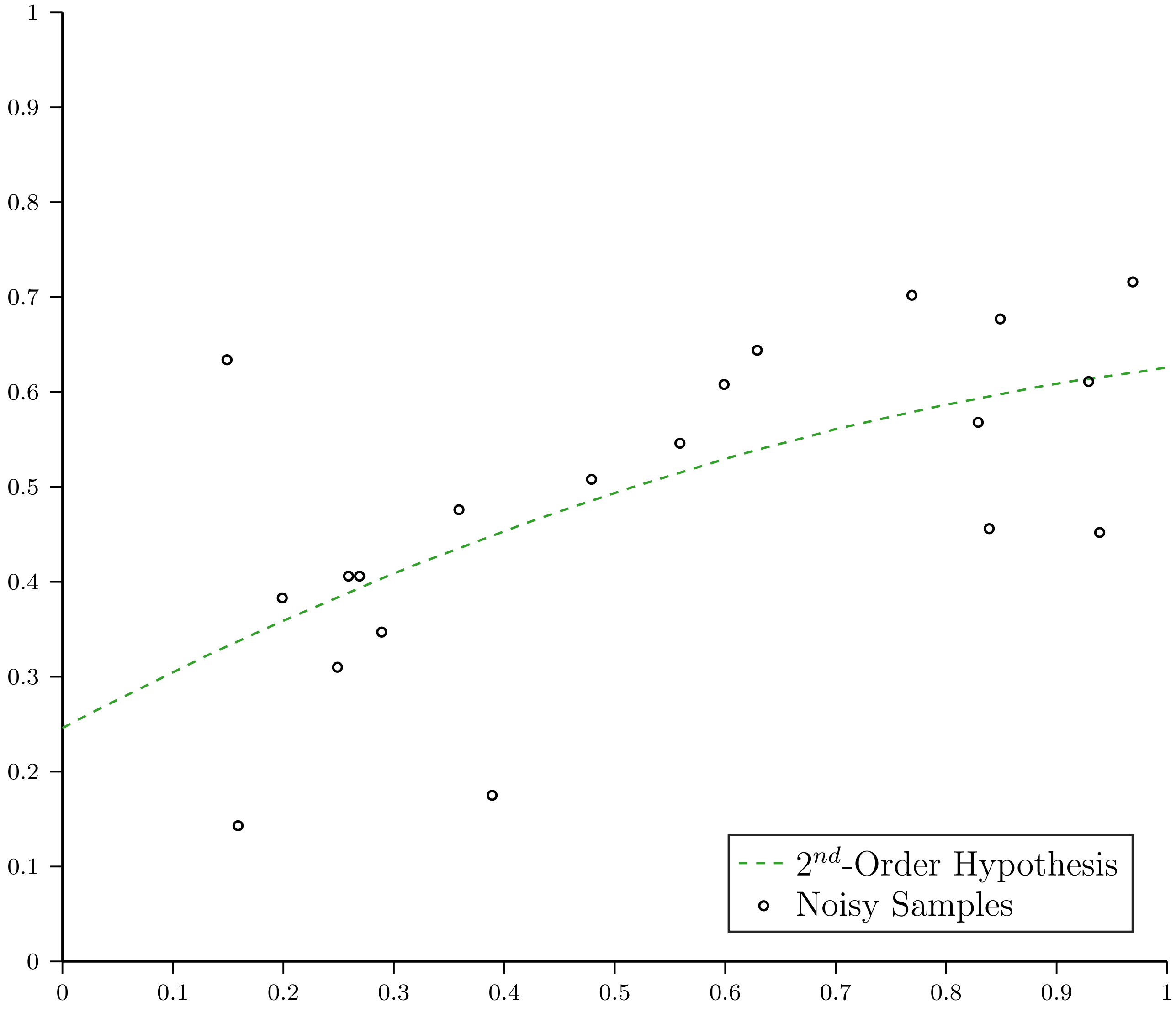
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Which model do you think will have a lower true error in this setting ([*loading eqn.*])?

- $\mathcal{H}_2$
- $\mathcal{H}_{10}$

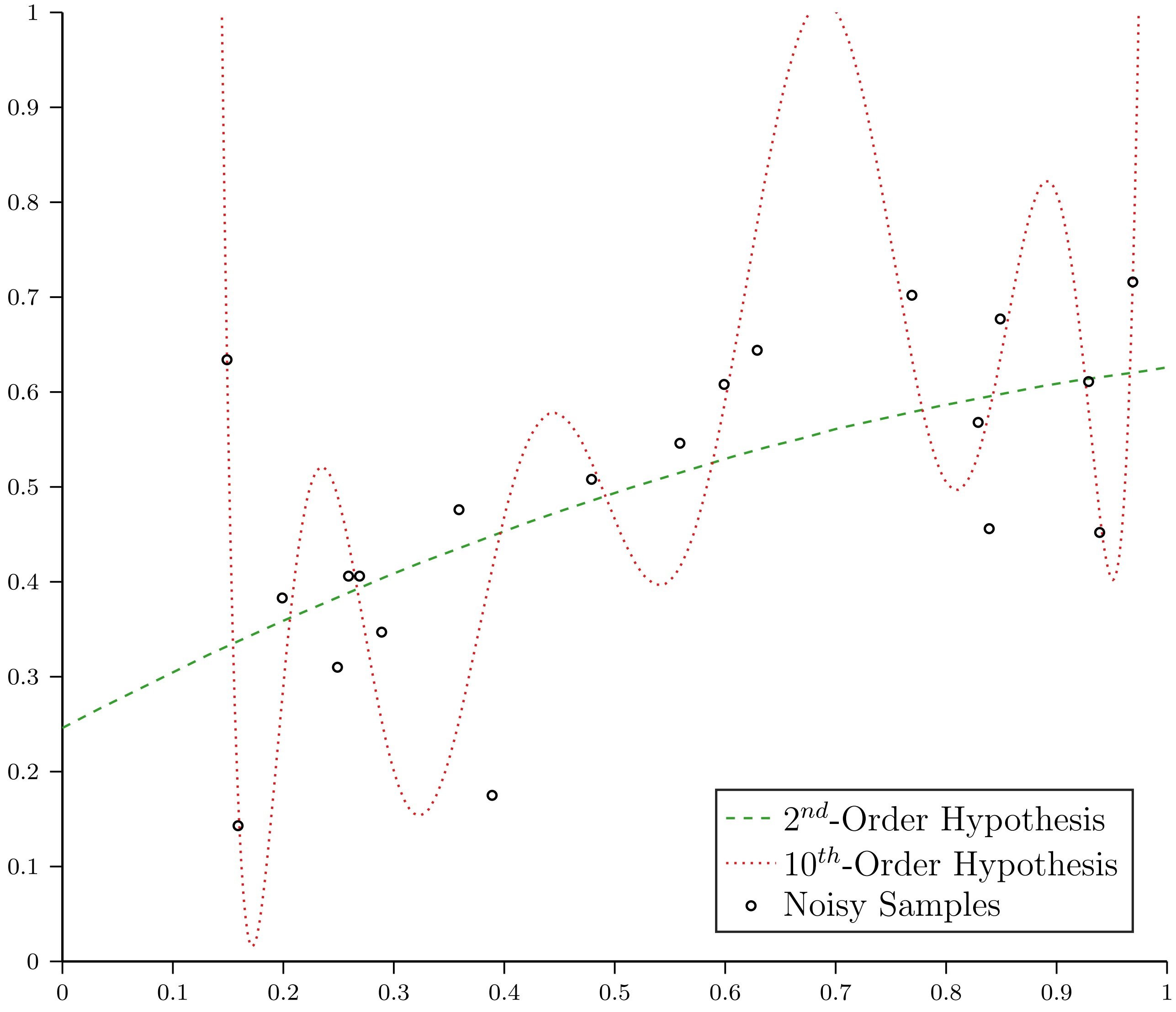
Noisy Targets

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



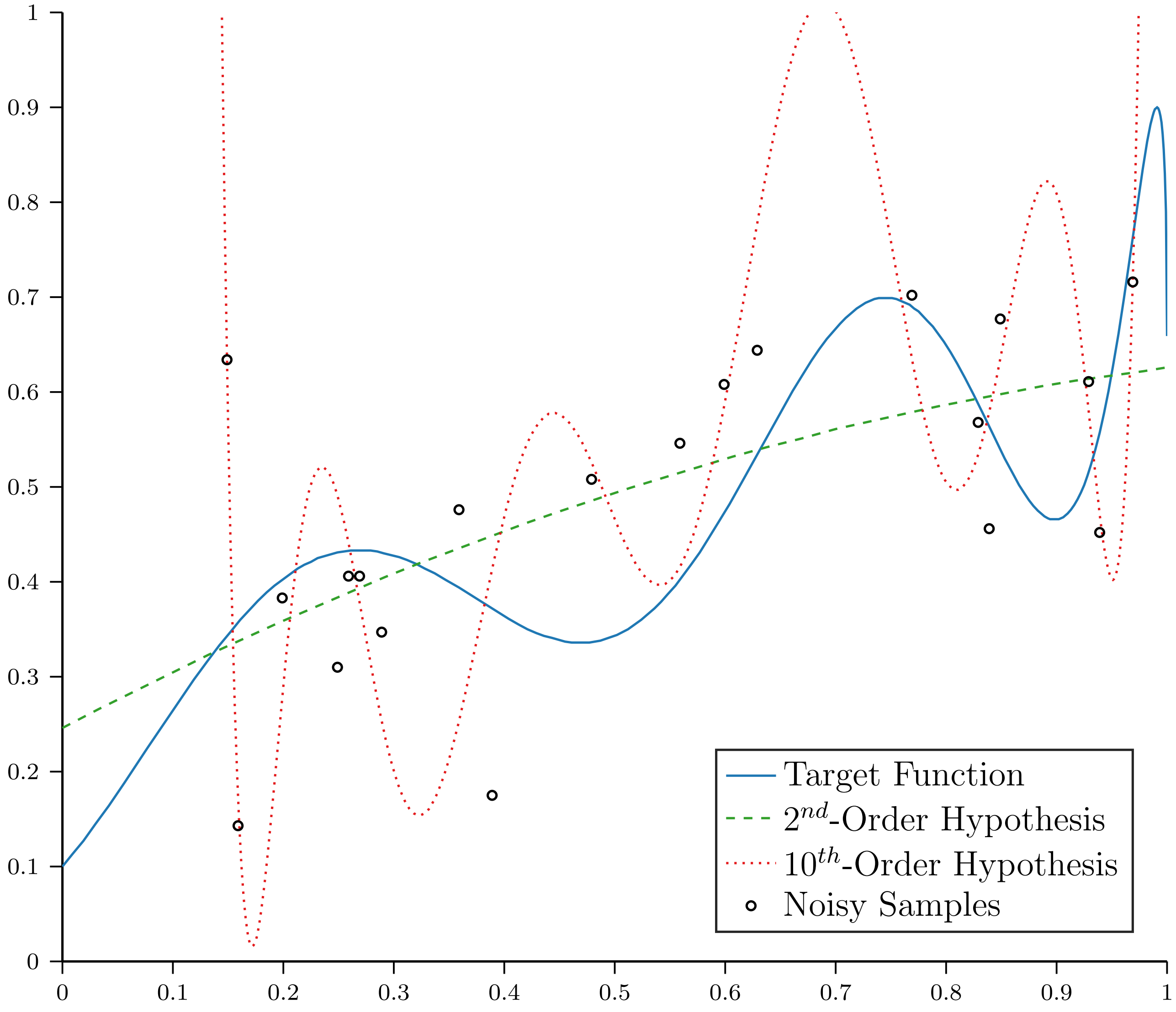
Noisy Targets

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



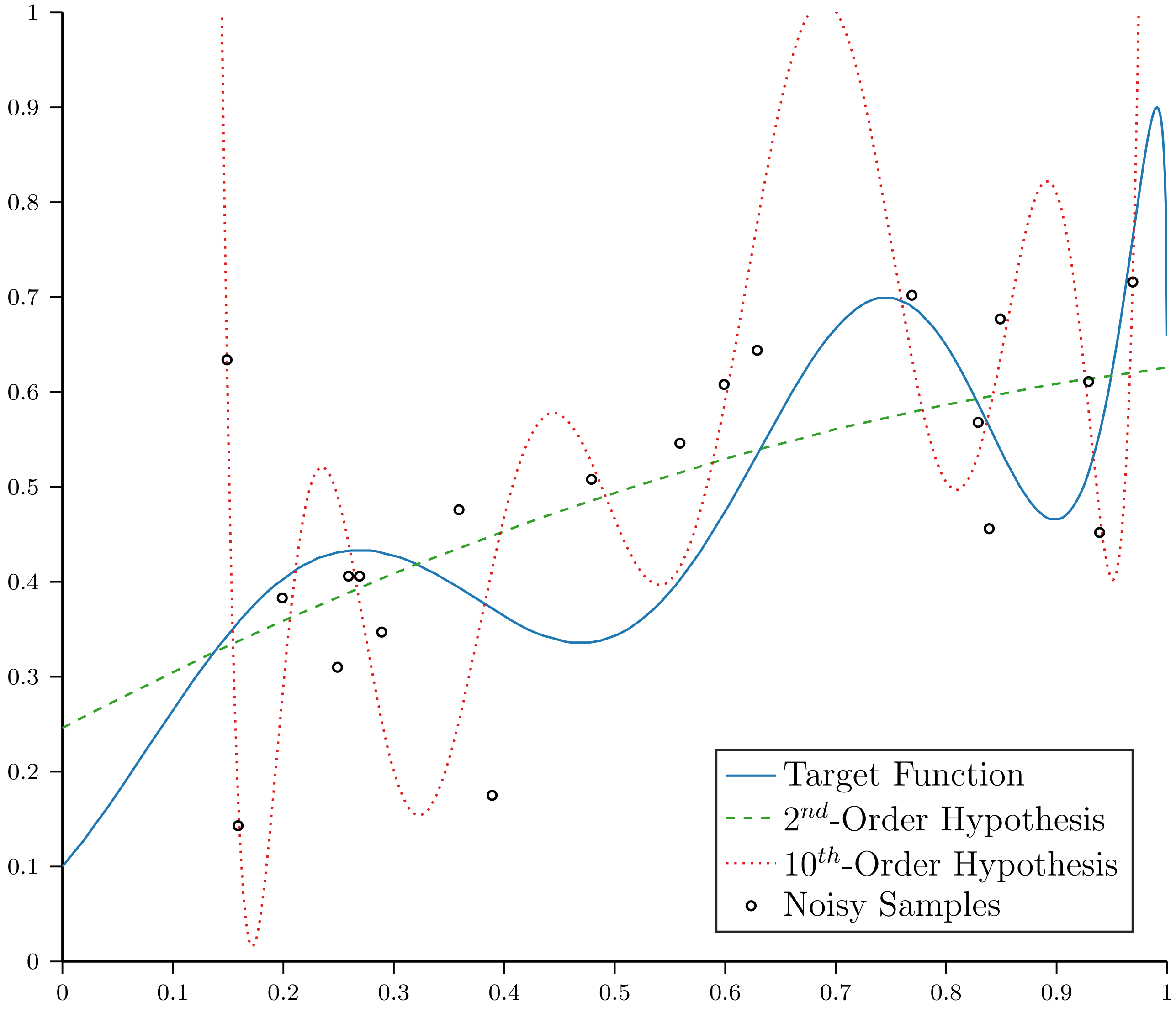
Noisy Targets

- 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 10^{th}-order polynomial in x with additive Gaussian noise:

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

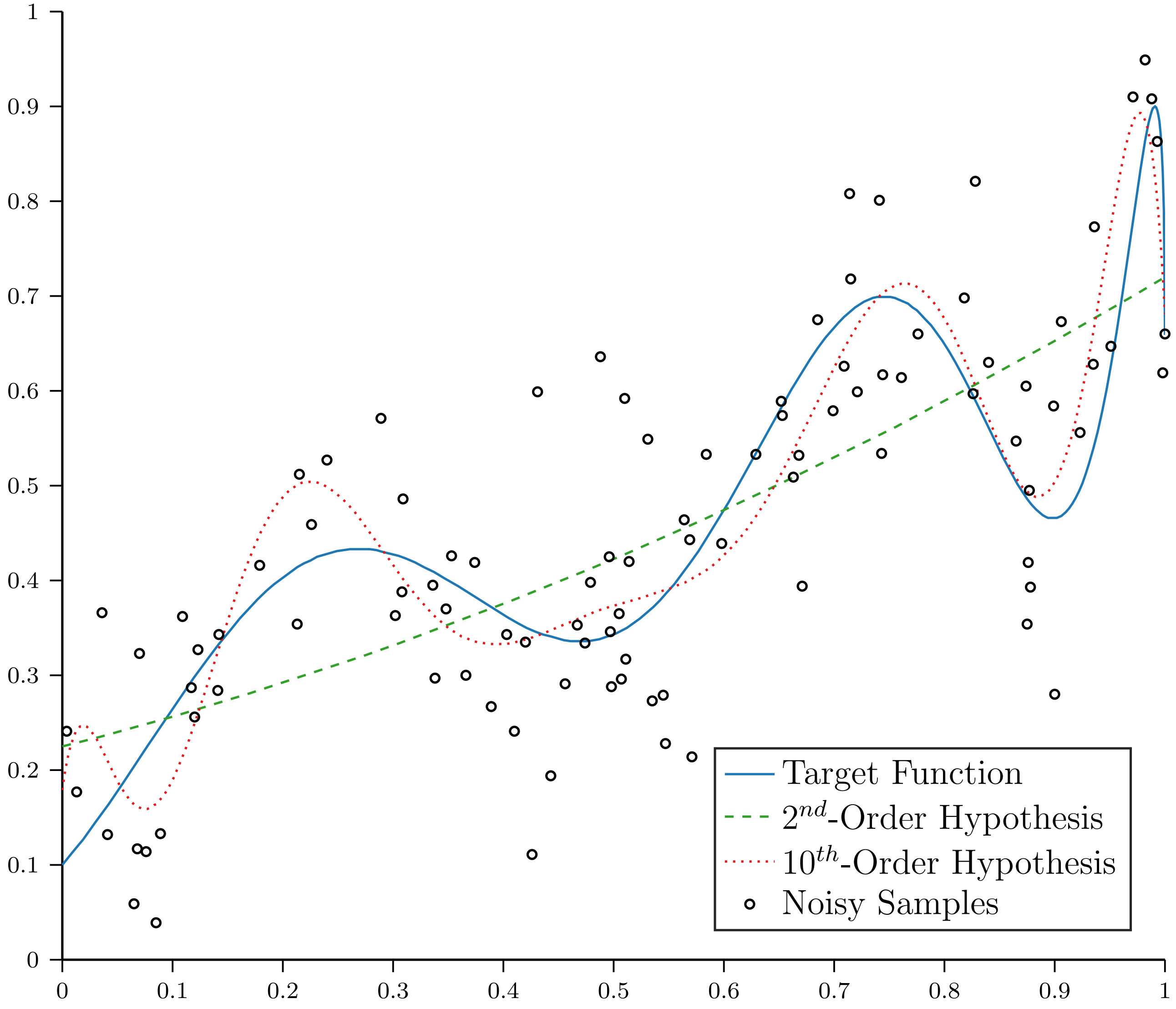
- $\mathcal{H}_2 = 2^{\text{nd}}$-order polynomials
  - $\phi_{1,2}(x) = [x, x^2]$
- $\mathcal{H}_{10} = 10^{\text{th}}$-order polynomials
  - $\phi_{1,10}(x) = [x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}]$

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^{th}-order polynomials
  - $\phi_{1,10}(x) = [x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}]$
- Given $\mathrm{X} = \begin{bmatrix} 1 & \phi_{1,10}\left(x^{(1)}\right) \\ 1 & \phi_{1,10}\left(x^{(2)}\right) \\ \vdots & \vdots \\ 1 & \phi_{1,10}\left(x^{(N)}\right) \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ find

  $\boldsymbol{\omega} = [\omega_0, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}]$ that minimizes

$$(\mathrm{X}\boldsymbol{\omega} - \boldsymbol{y})^T(\mathrm{X}\boldsymbol{\omega} - \boldsymbol{y})$$

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

$$\sum_{n=1}^{N}\left(\left(\sum_{d=0}^{10} x_d^{(n)}\omega_d\right) - y^{(n)}\right)^2$$

- 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
  - $\phi_{1,10}(x) = [x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}]$
- Given $\mathrm{X} = \begin{bmatrix} 1 & \phi_{1,10}\left(x^{(1)}\right) \\ 1 & \phi_{1,10}\left(x^{(2)}\right) \\ \vdots & \vdots \\ 1 & \phi_{1,10}\left(x^{(N)}\right) \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ find

  $\boldsymbol{\omega} = [\omega_0, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}]$ that minimizes

$$\sum_{n=1}^{N} \left(\left(\sum_{d=0}^{2} x_d^{(n)} \omega_d \right) - y^{(n)} \right)^2$$

- Subject to nothing!

Hard Constraints

- $\mathcal{H}_2$ = 2^{nd}-order polynomials
  - $\phi_{1,2}(x) = [x, x^2]$
- Given $\mathrm{X} = \begin{bmatrix} 1 & \phi_{1,2}\left(x^{(1)}\right) \\ 1 & \phi_{1,2}\left(x^{(2)}\right) \\ \vdots & \vdots \\ 1 & \phi_{1,2}\left(x^{(N)}\right) \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ find $\boldsymbol{\omega} = [\omega_0, \omega_1, \omega_2]$ that minimizes

$$(\mathrm{X}\boldsymbol{\omega} - \boldsymbol{y})^T(\mathrm{X}\boldsymbol{\omega} - \boldsymbol{y})$$

- Subject to nothing!

Soft Constraints

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

$$(\mathrm{X}\boldsymbol{\omega} - \boldsymbol{y})^T(\mathrm{X}\boldsymbol{\omega} - \boldsymbol{y})$$

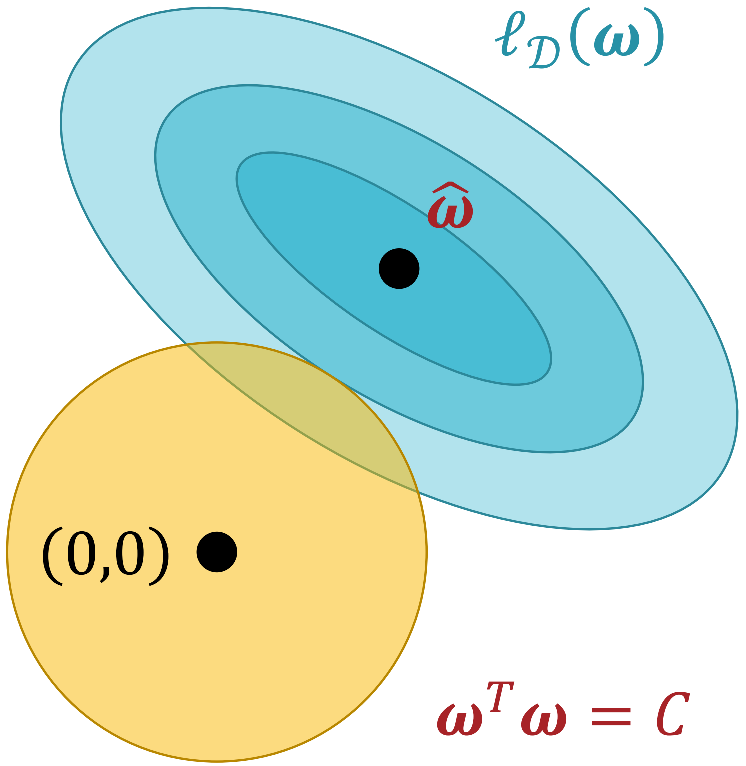
- Subject to:

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

Soft Constraints

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

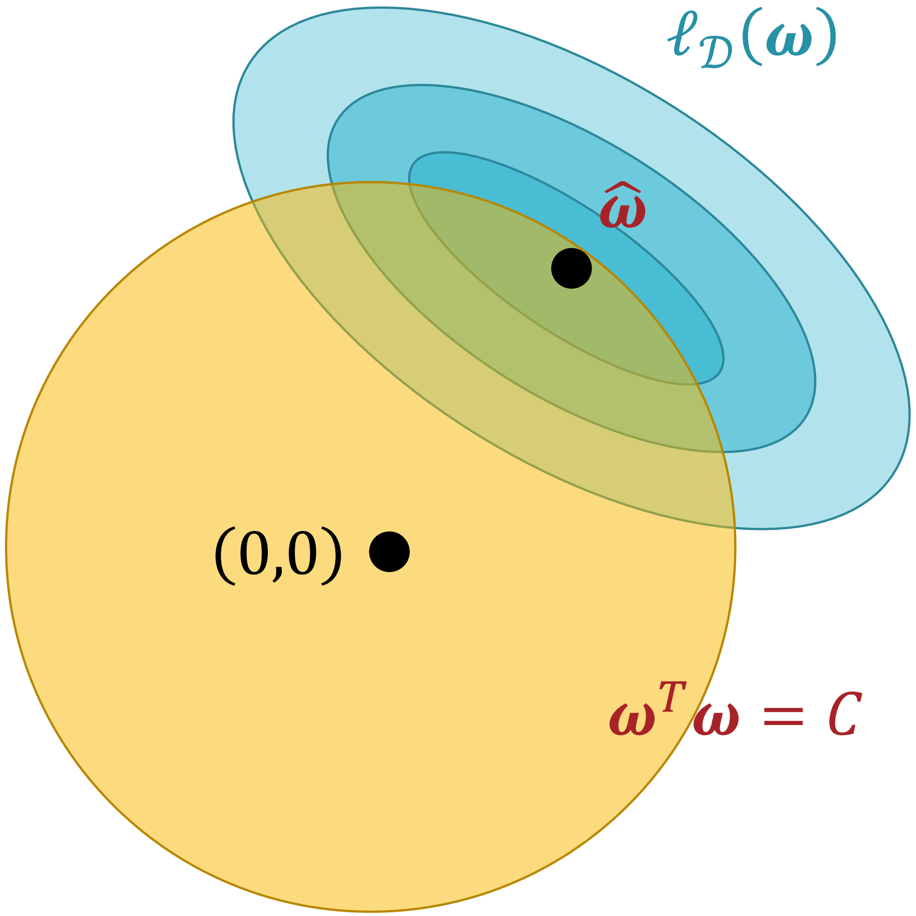
subject to $\boldsymbol{\omega}^T\boldsymbol{\omega} \leq C$



Soft Constraints

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

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



Soft Constraints

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

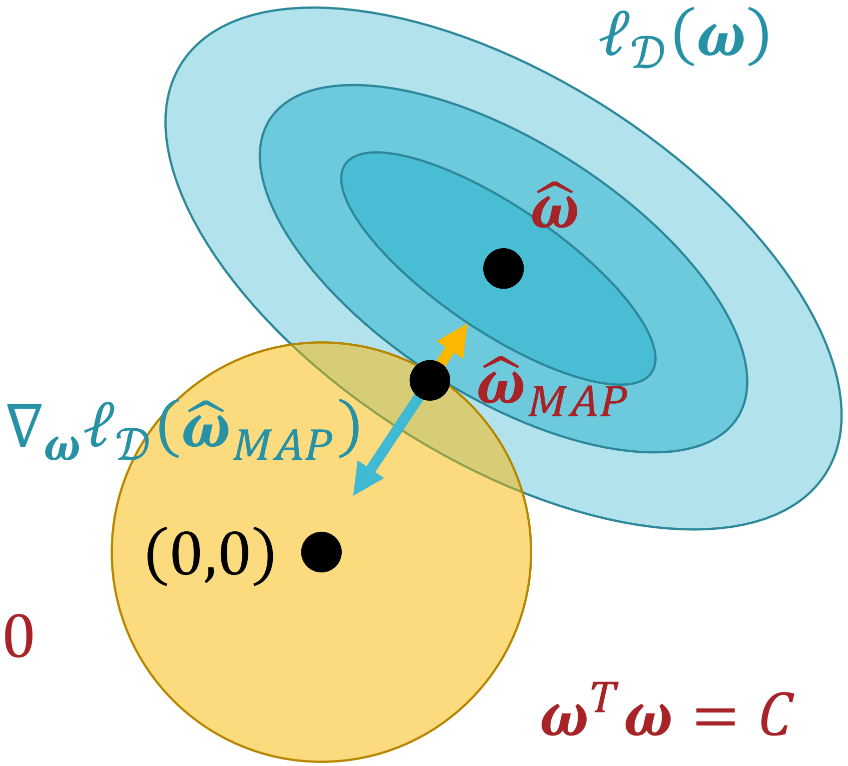
subject to $\boldsymbol{\omega}^T\boldsymbol{\omega} \leq C$

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

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

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

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



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

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

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

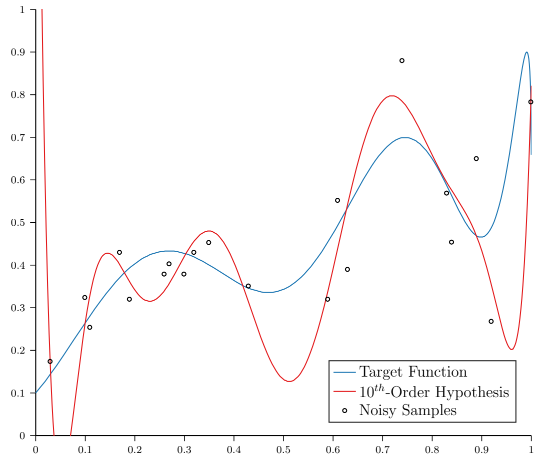
$$\nabla_{\boldsymbol{\omega}} \ell_{\mathcal{D}}^{AUG}(\boldsymbol{\omega}) = 2(\mathrm{X}^T\mathrm{X}\boldsymbol{\omega} - \mathrm{X}^T\boldsymbol{y} + \lambda_C \boldsymbol{\omega})$$

$$2(\mathrm{X}^T\mathrm{X}\widehat{\boldsymbol{\omega}}_{MAP} - \mathrm{X}^T\boldsymbol{y} + \lambda_C \widehat{\boldsymbol{\omega}}_{MAP}) = 0$$

$$(\mathrm{X}^T\mathrm{X} + \lambda_C I_{D+1})\widehat{\boldsymbol{\omega}}_{MAP} = \mathrm{X}^T\boldsymbol{y}$$

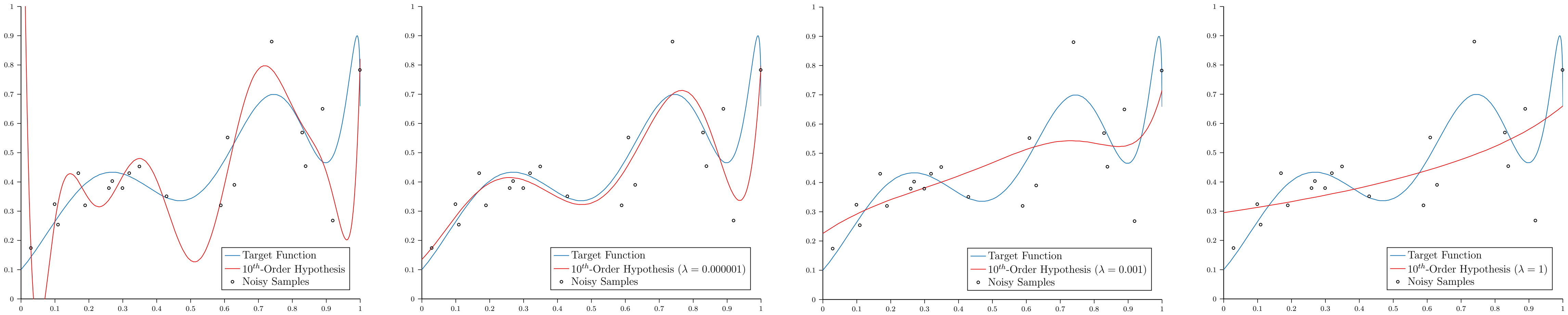
$$\widehat{\boldsymbol{\omega}}_{MAP} = (\mathrm{X}^T\mathrm{X} + \underbrace{\lambda_C I_{D+1}})^{-1}\mathrm{X}^T\boldsymbol{y}$$

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



Ridge Regression

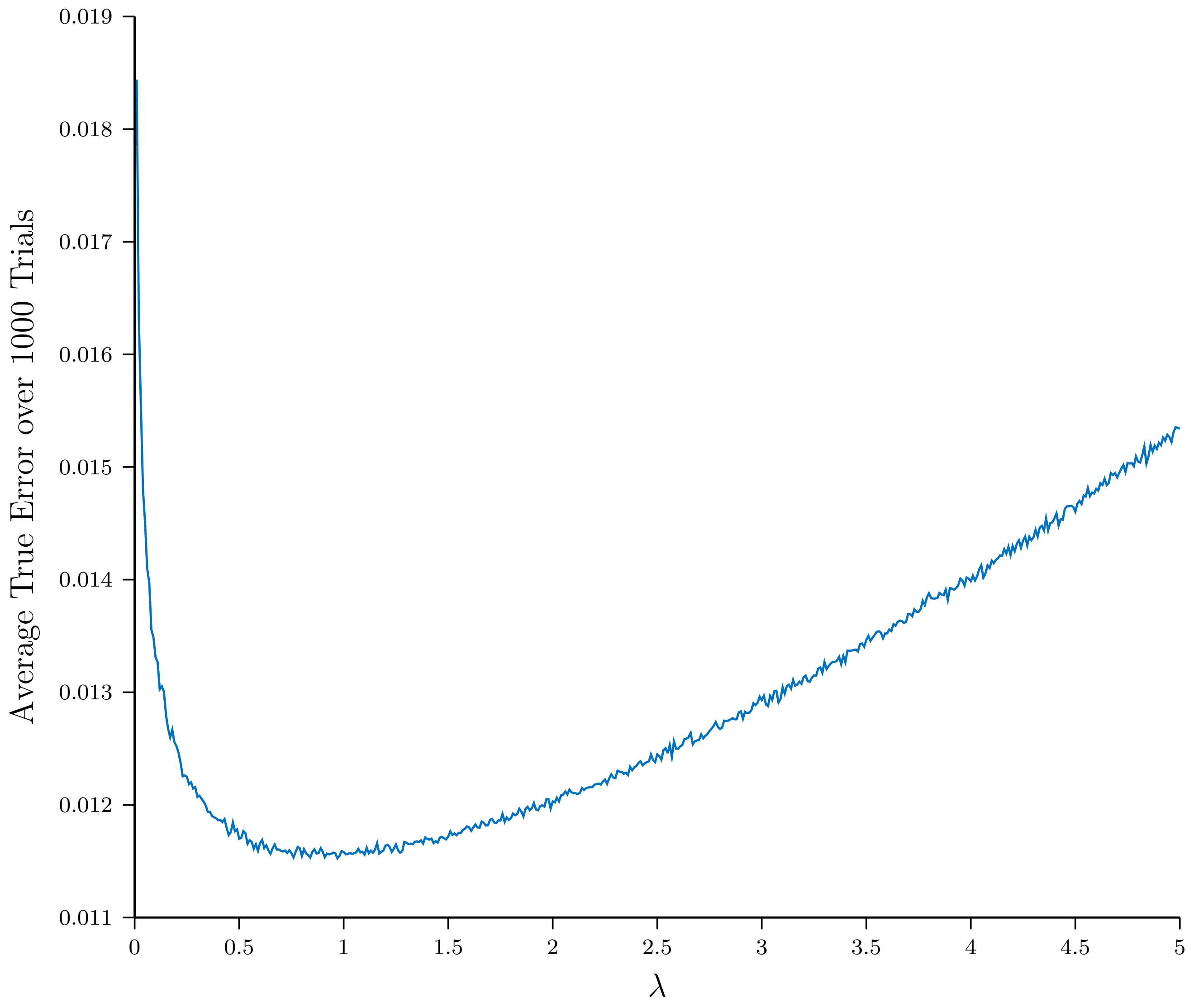
- 10-dimensional target function with additive Gaussian noise
- $\mathcal{H}_{10} = 10^{\text{th}}$-order polynomial



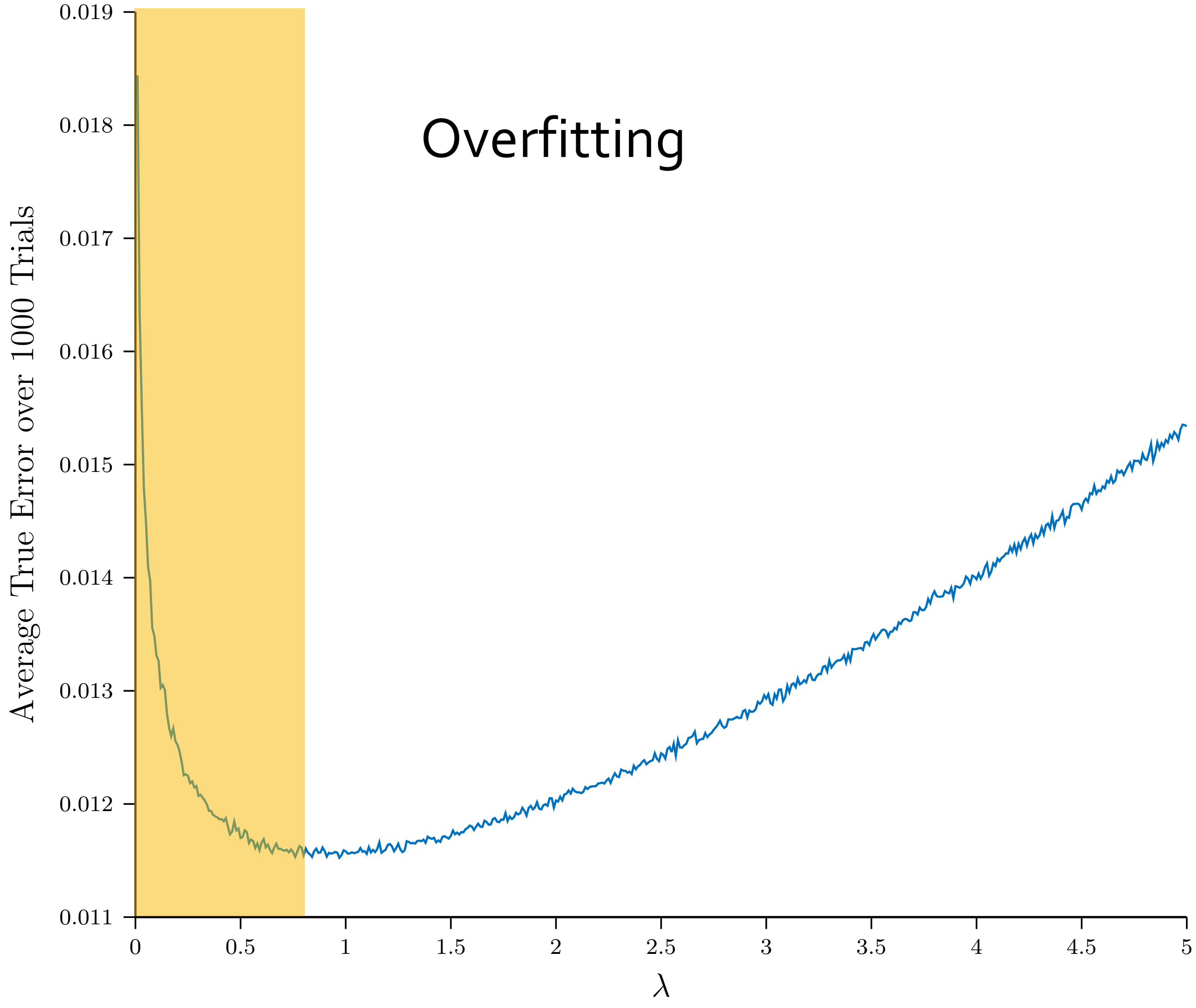
Ridge Regression

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

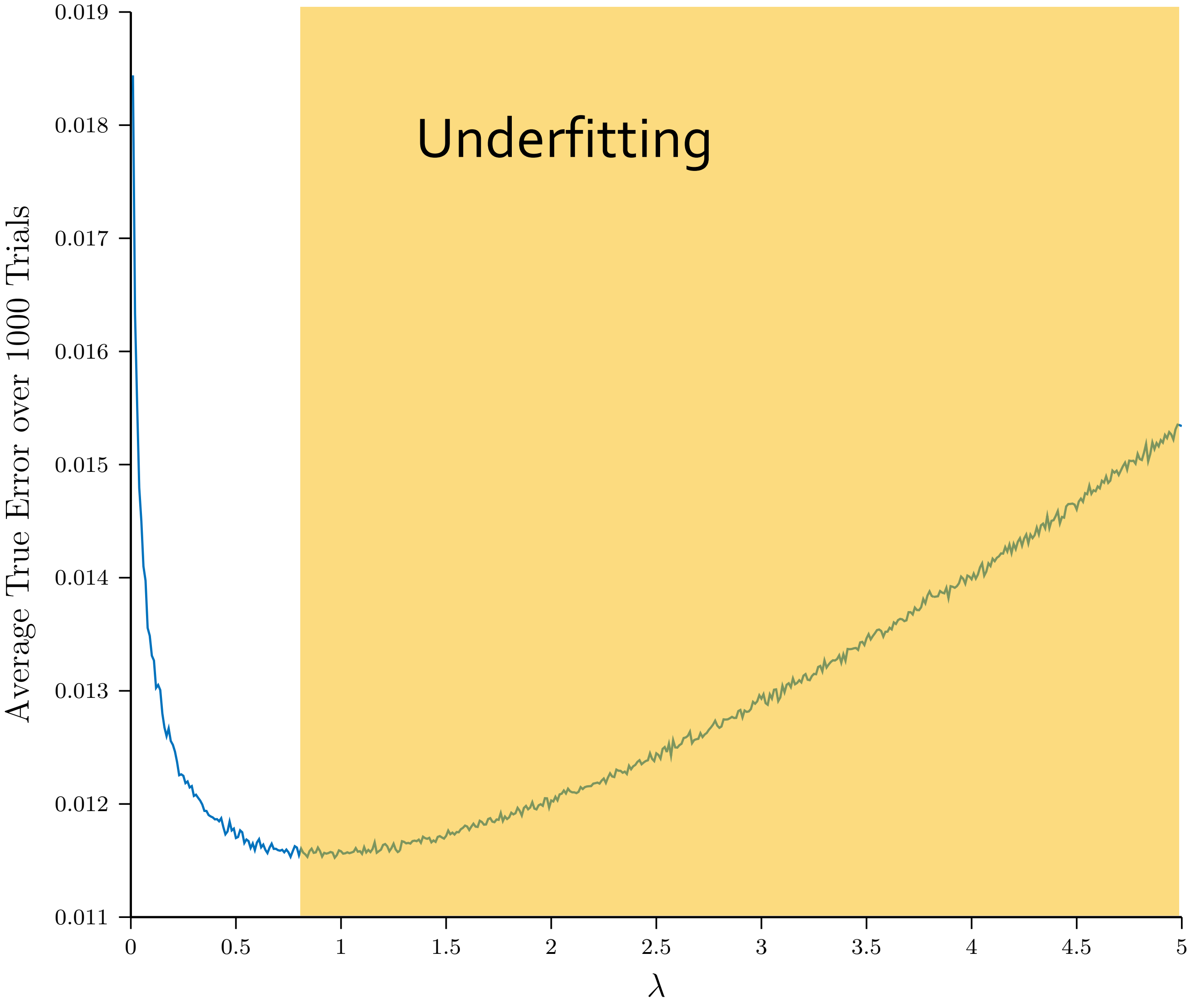
Setting λ



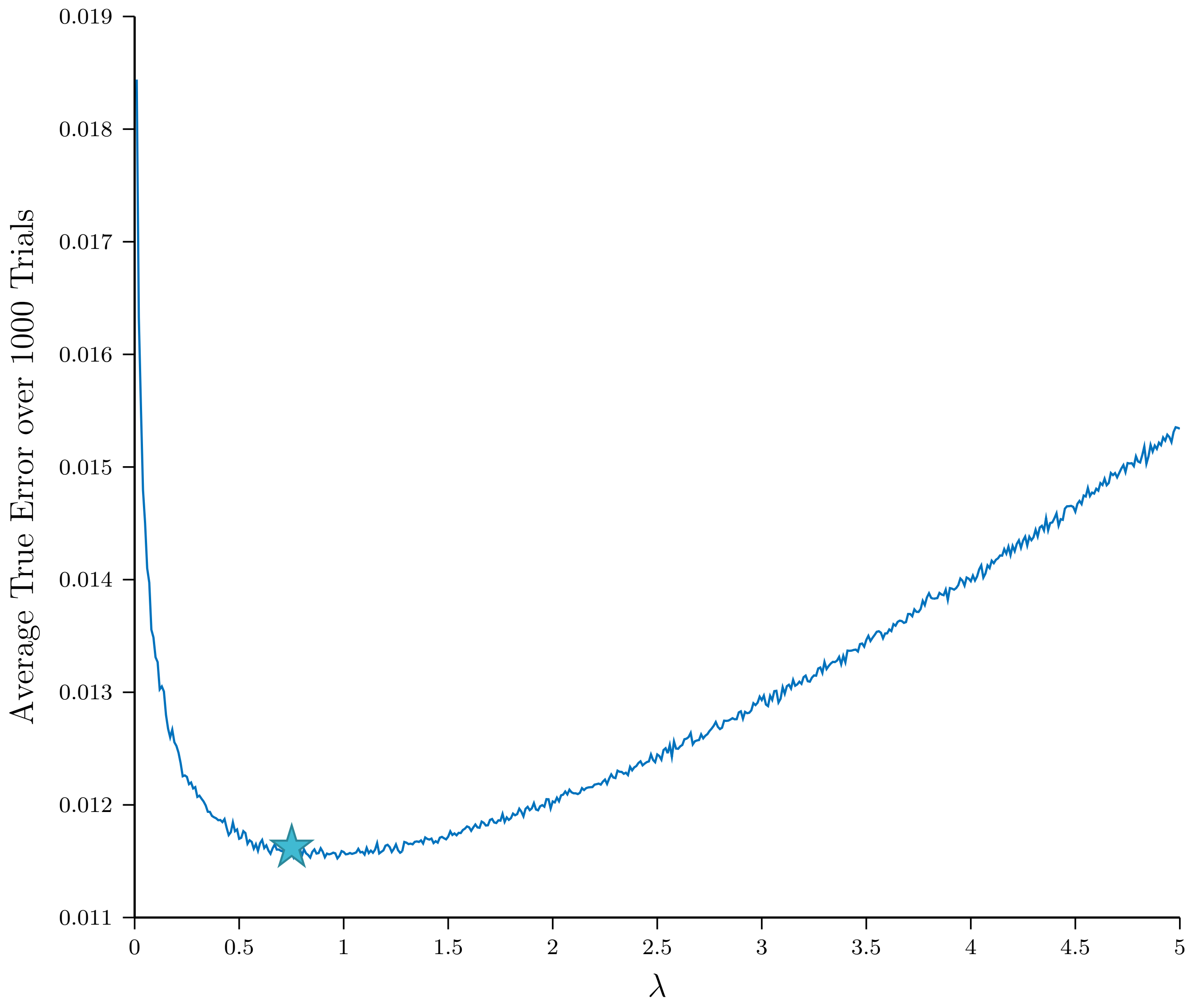
Setting λ



Setting λ

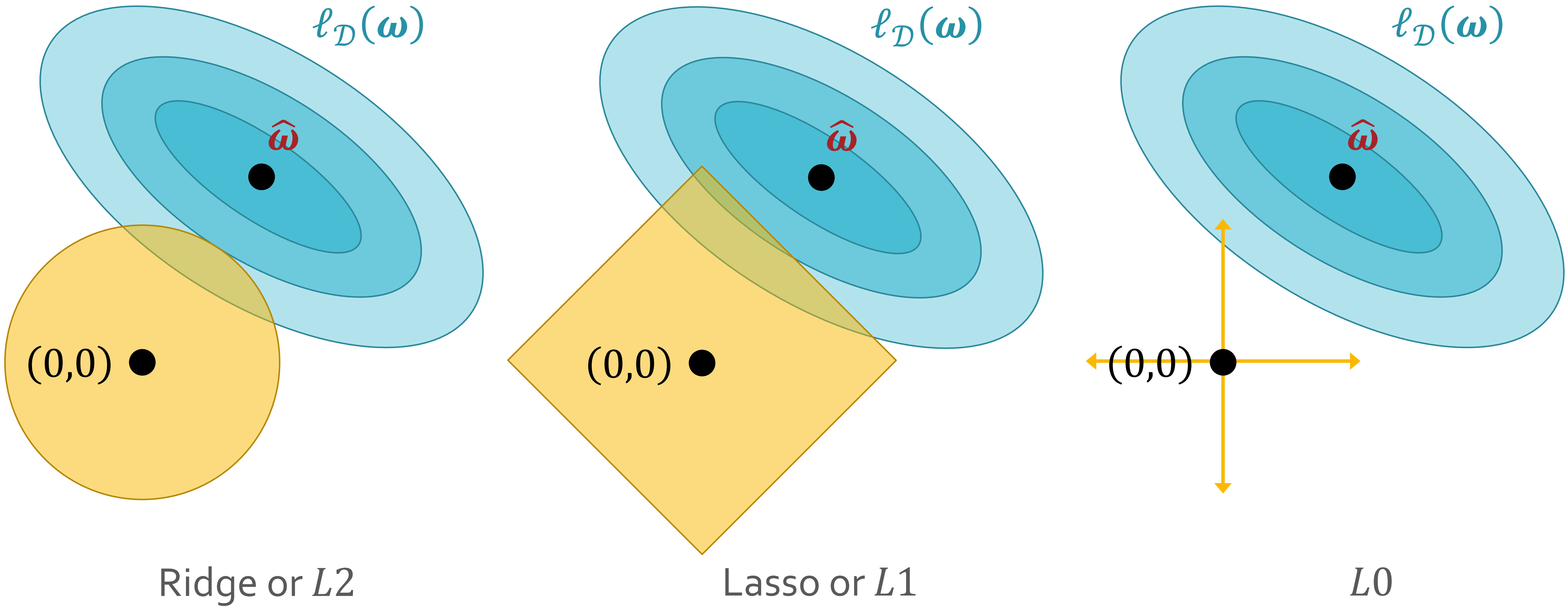


Setting λ



Other Regularizers

$\ell_{\mathcal{D}}(\boldsymbol{\omega}) + \lambda r(\boldsymbol{\omega})$		
Ridge or $L2$	$r(\boldsymbol{\omega}) = \lVert\boldsymbol{\omega}\rVert_2^2 = \sum_{d=0}^{D} \omega_d^2$	Encourages small weights
Lasso or $L1$	$r(\boldsymbol{\omega}) = \lVert\boldsymbol{\omega}\rVert_1 = \sum_{d=0}^{D} \lvert\omega_d\rvert$	Encourages sparsity
$L0$	$r(\boldsymbol{\omega}) = \lVert\boldsymbol{\omega}\rVert_0 = \sum_{d=0}^{D} \mathbb{1}(\omega_d \neq 0)$	Encourages sparsity (intractable)



Other Regularizers

M(C)LE for Linear Regression

- If we assume a linear model with additive Gaussian noise

$$y = \boldsymbol{\omega}^T \boldsymbol{x} + \epsilon \text{ where } \epsilon \sim N(0, \sigma^2) \rightarrow y \sim N(\boldsymbol{\omega}^T \boldsymbol{x}, \sigma^2)$$

- Then given $X = \begin{bmatrix} 1 & \boldsymbol{x}^{(1)} \\ 1 & \boldsymbol{x}^{(2)} \\ \vdots & \vdots \\ 1 & \boldsymbol{x}^{(N)} \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$ the MLE of $\boldsymbol{\omega}$ is

$$\widehat{\boldsymbol{\omega}} = \underset{\boldsymbol{\omega}}{\operatorname{argmax}} \log P(\boldsymbol{y} | X, \boldsymbol{\omega})$$

$$= \underset{\boldsymbol{\omega}}{\operatorname{argmax}} \log \exp\left(-\frac{1}{2\sigma^2}(X\boldsymbol{\omega} - \boldsymbol{y})^T (X\boldsymbol{\omega} - \boldsymbol{y})\right)$$

$$= \underset{\boldsymbol{\omega}}{\operatorname{argmin}} (X\boldsymbol{\omega} - \boldsymbol{y})^T (X\boldsymbol{\omega} - \boldsymbol{y}) = (X^T X)^{-1} X^T \boldsymbol{y}$$

MAP for Linear Regression

- If we assume a linear model with additive Gaussian noise

$$y = \boldsymbol{\omega}^T \boldsymbol{x} + \epsilon \text{ where } \epsilon \sim N(0, \sigma^2) \rightarrow y \sim N(\boldsymbol{\omega}^T \boldsymbol{x}, \sigma^2)$$

and independent Gaussian priors on all the weights…

$$\omega_d \sim N\left(0, \frac{\sigma^2}{\lambda}\right) \rightarrow p(\boldsymbol{\omega}) \propto \exp\left(-\frac{1}{2\sigma^2}(\lambda \boldsymbol{\omega}^T \boldsymbol{\omega})\right)$$

- … then, the MAP of $\boldsymbol{\omega}$ is the ridge regression solution!

$$\widehat{\boldsymbol{\omega}}_{MAP} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} (X\boldsymbol{\omega} - \boldsymbol{y})^T (X\boldsymbol{\omega} - \boldsymbol{y}) + \lambda \boldsymbol{\omega}^T \boldsymbol{\omega}$$

$$= \left(X^T X + \lambda I_{D+1}\right)^{-1} X^T \boldsymbol{y}$$

MAP for Linear Regression

- If we assume a linear model with additive Gaussian noise

$$y = \boldsymbol{\omega}^T \boldsymbol{x} + \epsilon \text{ where } \epsilon \sim N(0, \sigma^2) \rightarrow y \sim N(\boldsymbol{\omega}^T \boldsymbol{x}, \sigma^2)$$

and independent Laplace priors on all the weights…

$$\omega_d \sim \text{Laplace}\left(0, \frac{2\sigma^2}{\lambda}\right) \rightarrow p(\boldsymbol{\omega}) \propto \exp\left(-\frac{1}{2\sigma^2}(\lambda \|\boldsymbol{\omega}\|_1)\right)$$

- … then, the MAP of $\boldsymbol{\omega}$ is the Lasso regression solution!

$$\widehat{\boldsymbol{\omega}}_{MAP} = \underset{\boldsymbol{\omega}}{\text{argmin}}\, (X\boldsymbol{\omega} - \boldsymbol{y})^T (X\boldsymbol{\omega} - \boldsymbol{y}) + \lambda \|\boldsymbol{\omega}\|_1$$

- No closed form solution but can solve via (sub−)gradient descent

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
    - Can be interpreted as MAP for linear regression

Where do features come from?

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

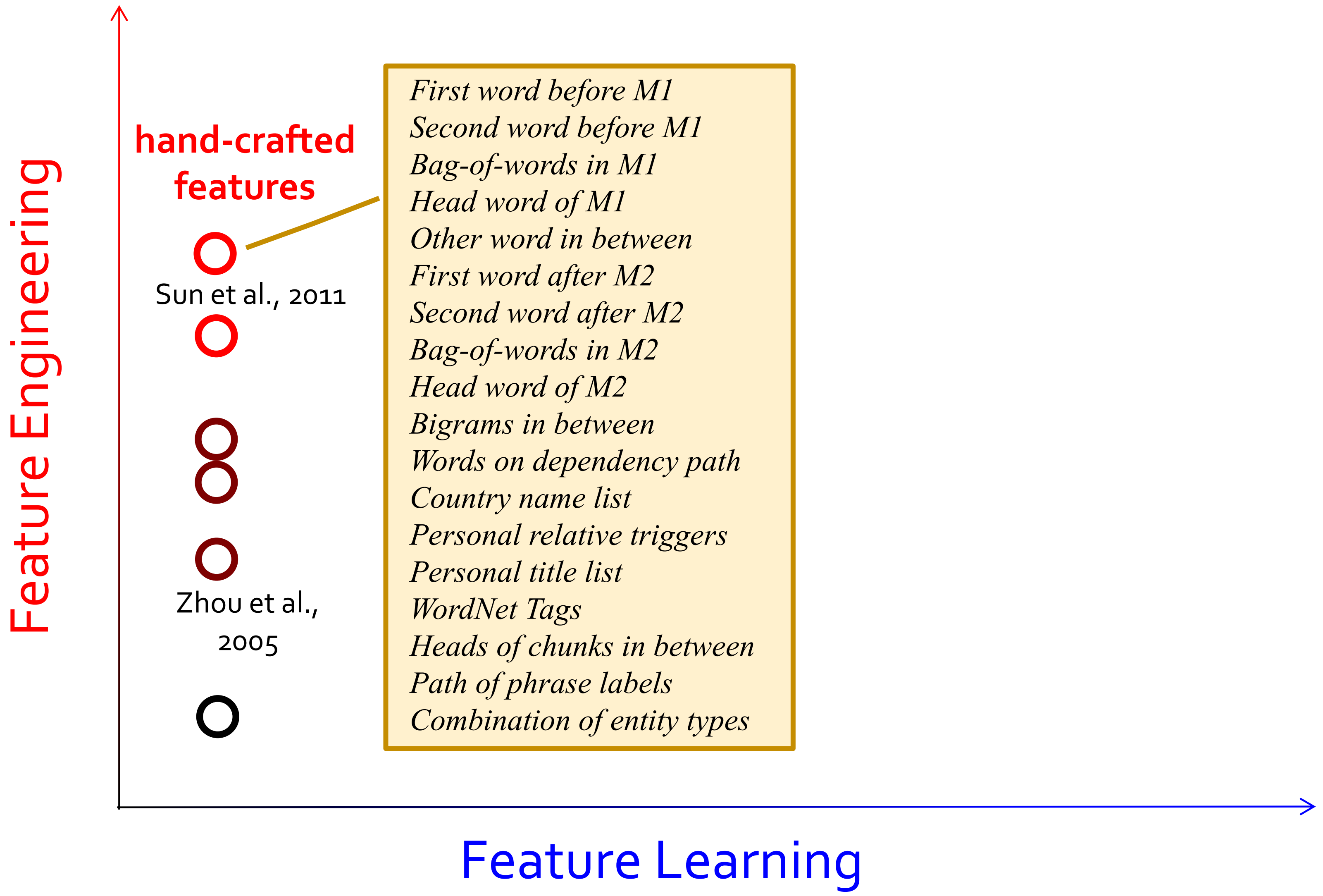
  find $\boldsymbol{\omega}$ that minimizes

$$(\mathrm{X}\boldsymbol{\omega} - \boldsymbol{y})^T(\mathrm{X}\boldsymbol{\omega} - \boldsymbol{y})$$

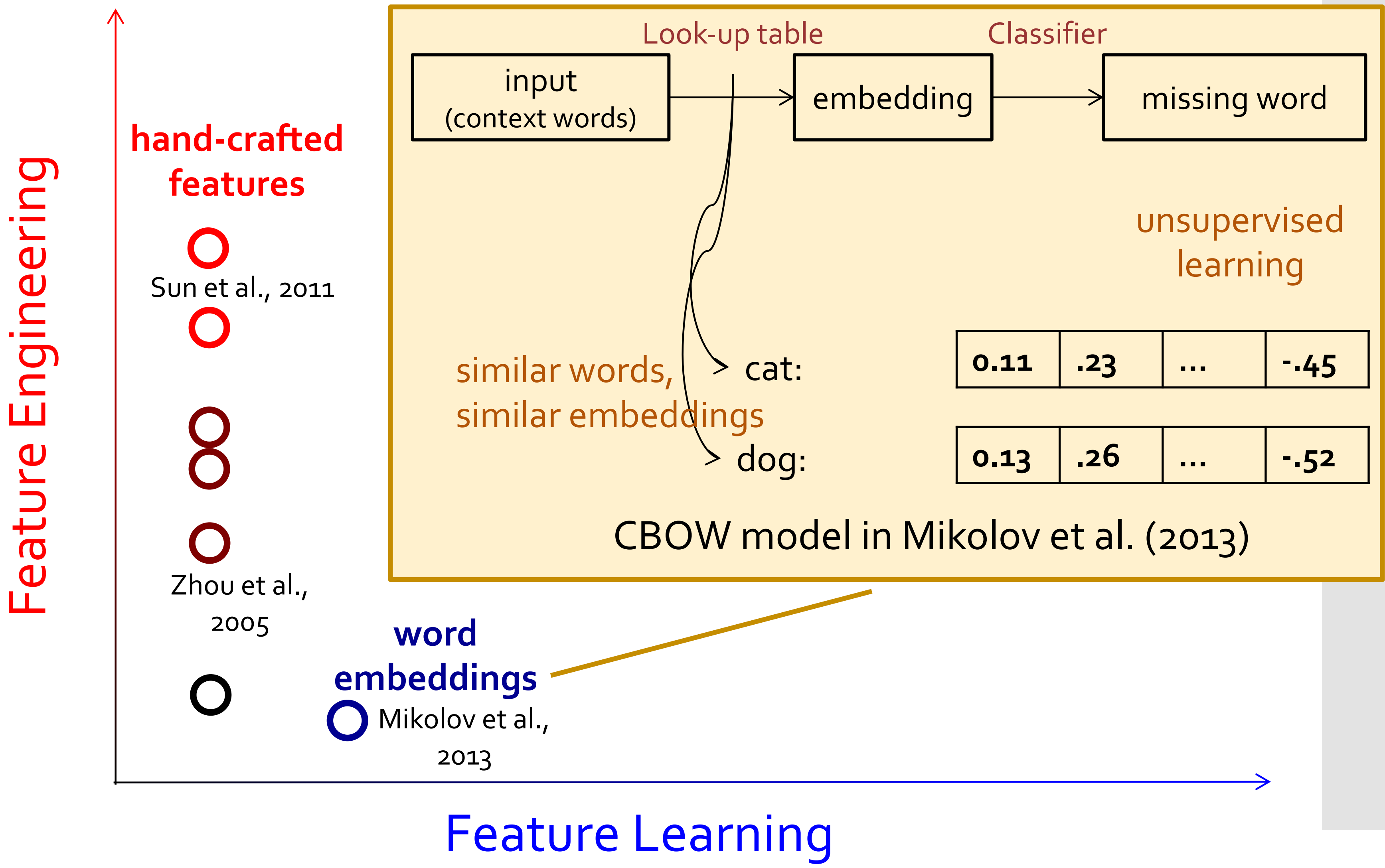
- Subject to:

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

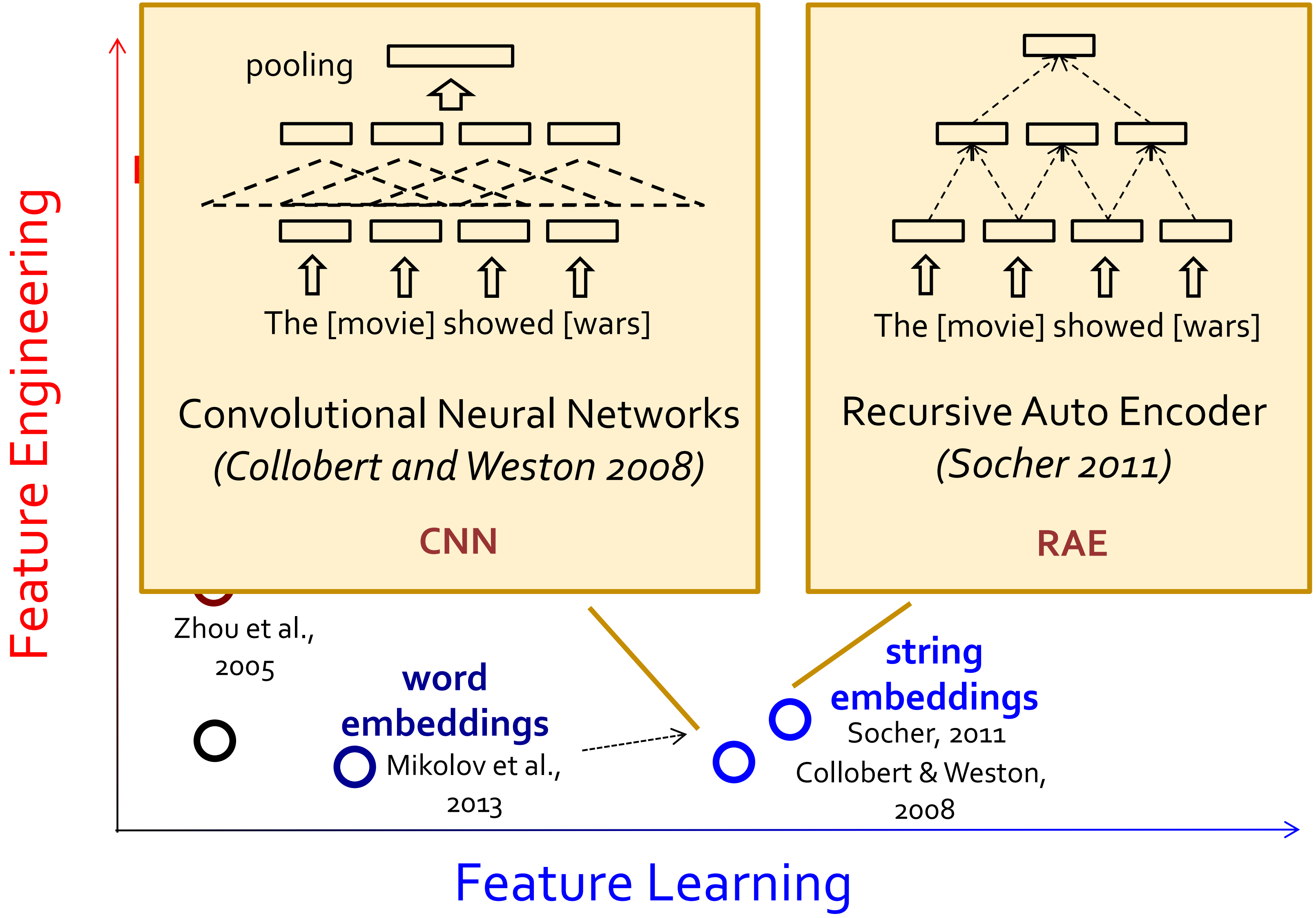
Where do features for text data come from?



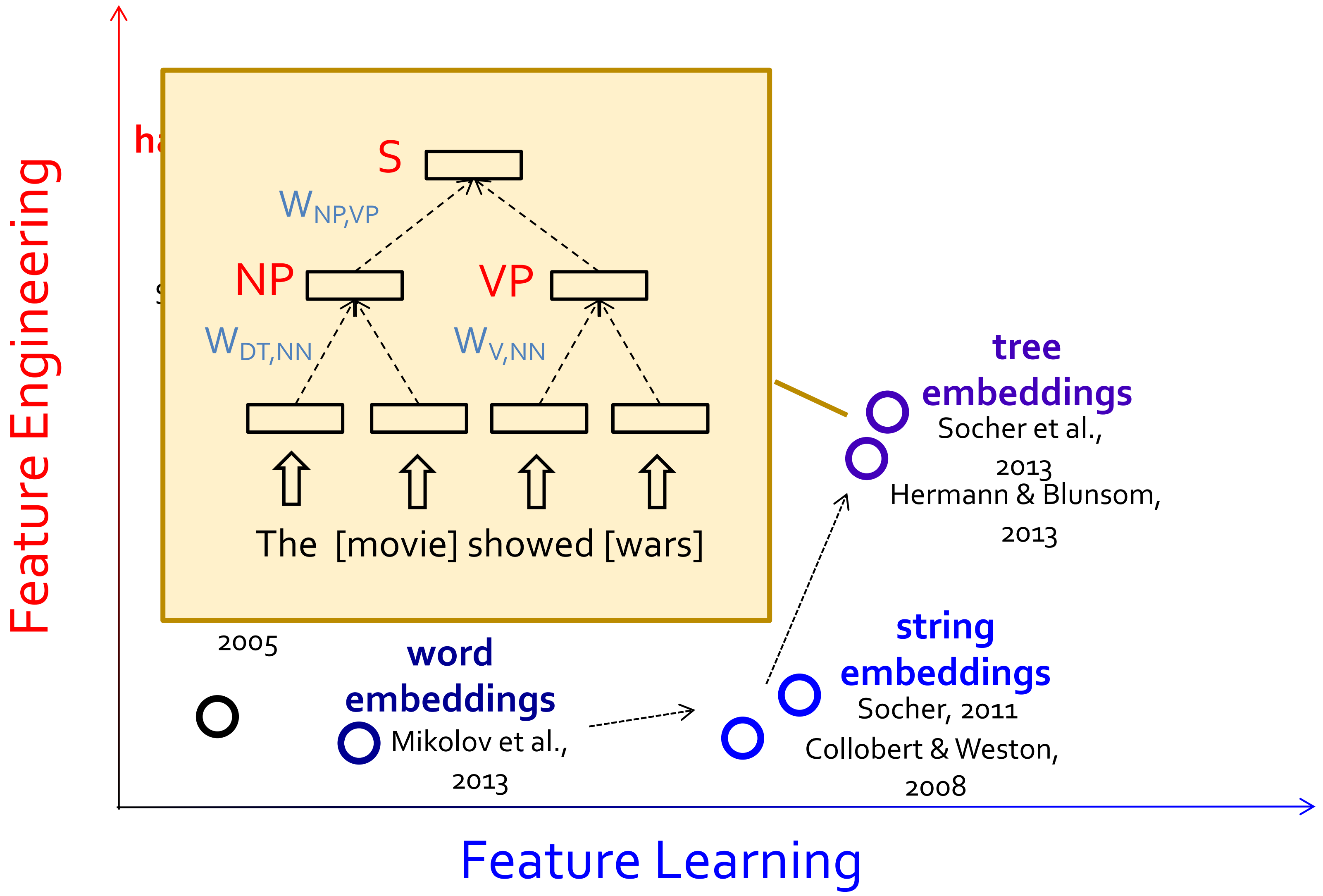
Where do features for text data come from?



Where do features for text data come from?



Where do features for text data come from?



Where do features for text data come from?