Variance, Bias, Generalization, etc.

Loss, Training, Cost

- The most typical Loss we've been using so far

$$J(\theta) = \frac{1}{n} \sum_i \left(y^{(i)} - h_\theta(x^{(i)}) \right)^2$$

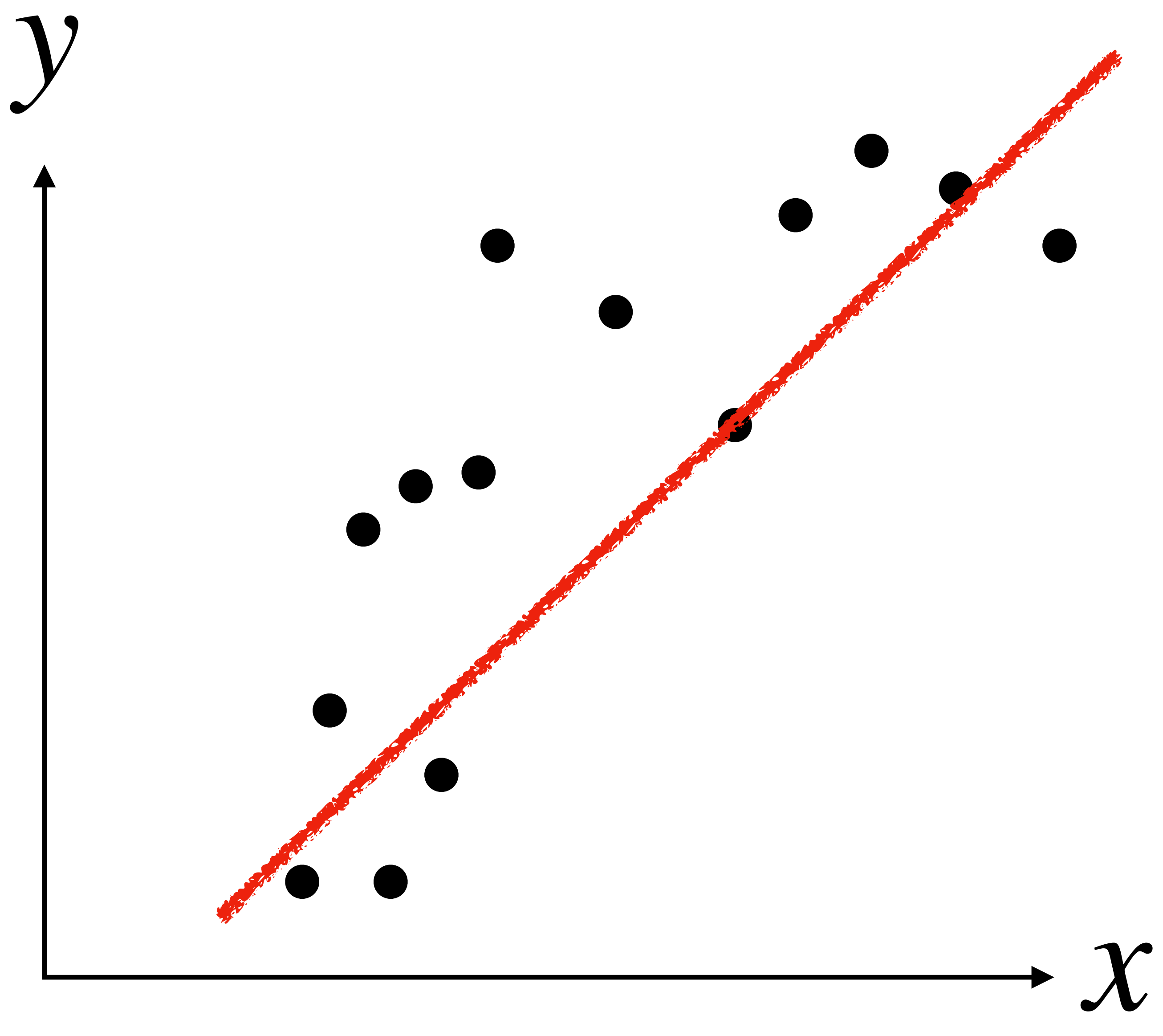
- How did we find it?
- **Maximum Likelihood Estimation**
- $\max P(y \mid x; \theta)$**:** use the negative log-likelihood as the training loss

How to choose $\phi(x)$?

How to optimize over $\phi(x)$

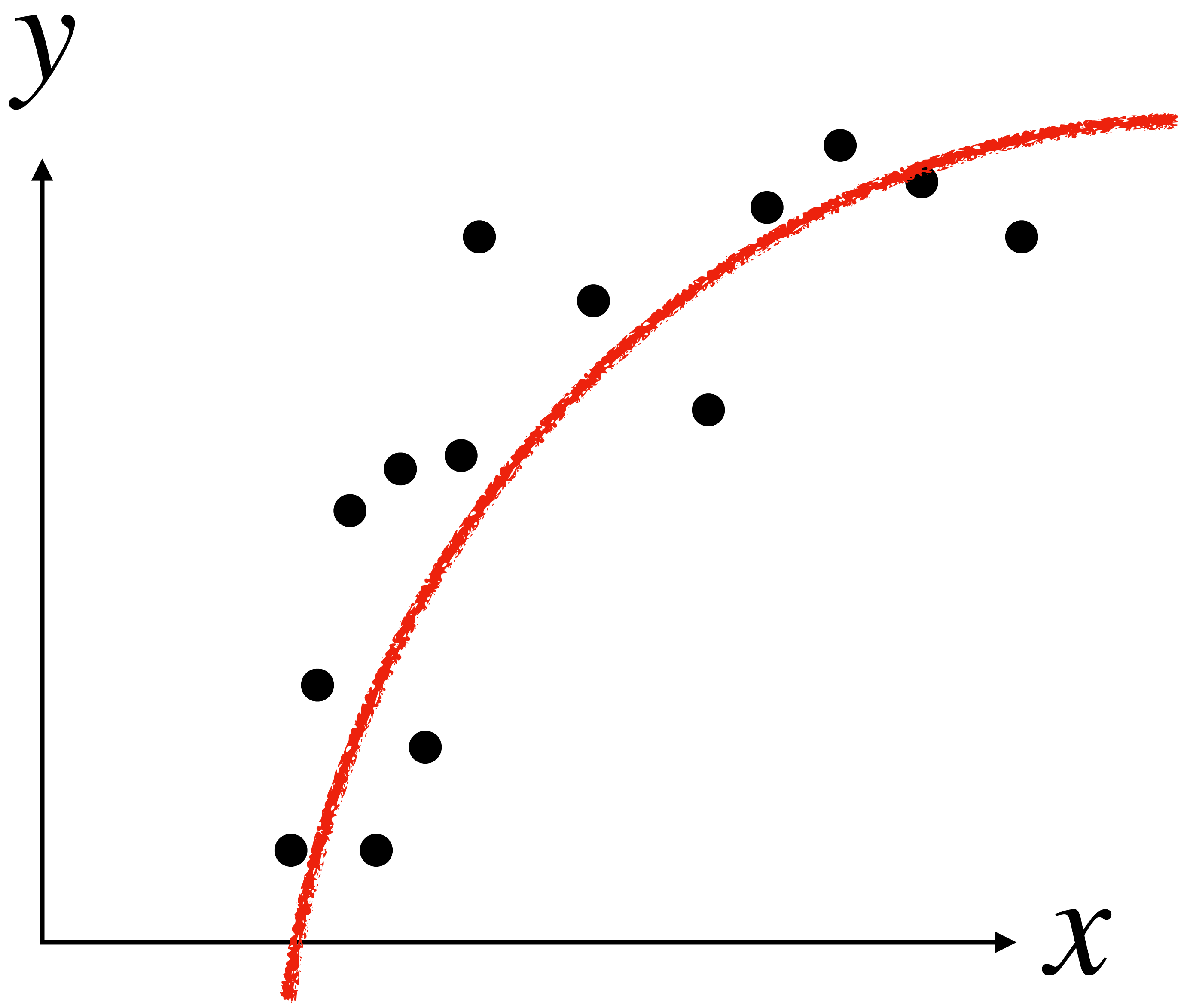
Underfitting

High Bias



$$\phi(x) = [1, x]$$

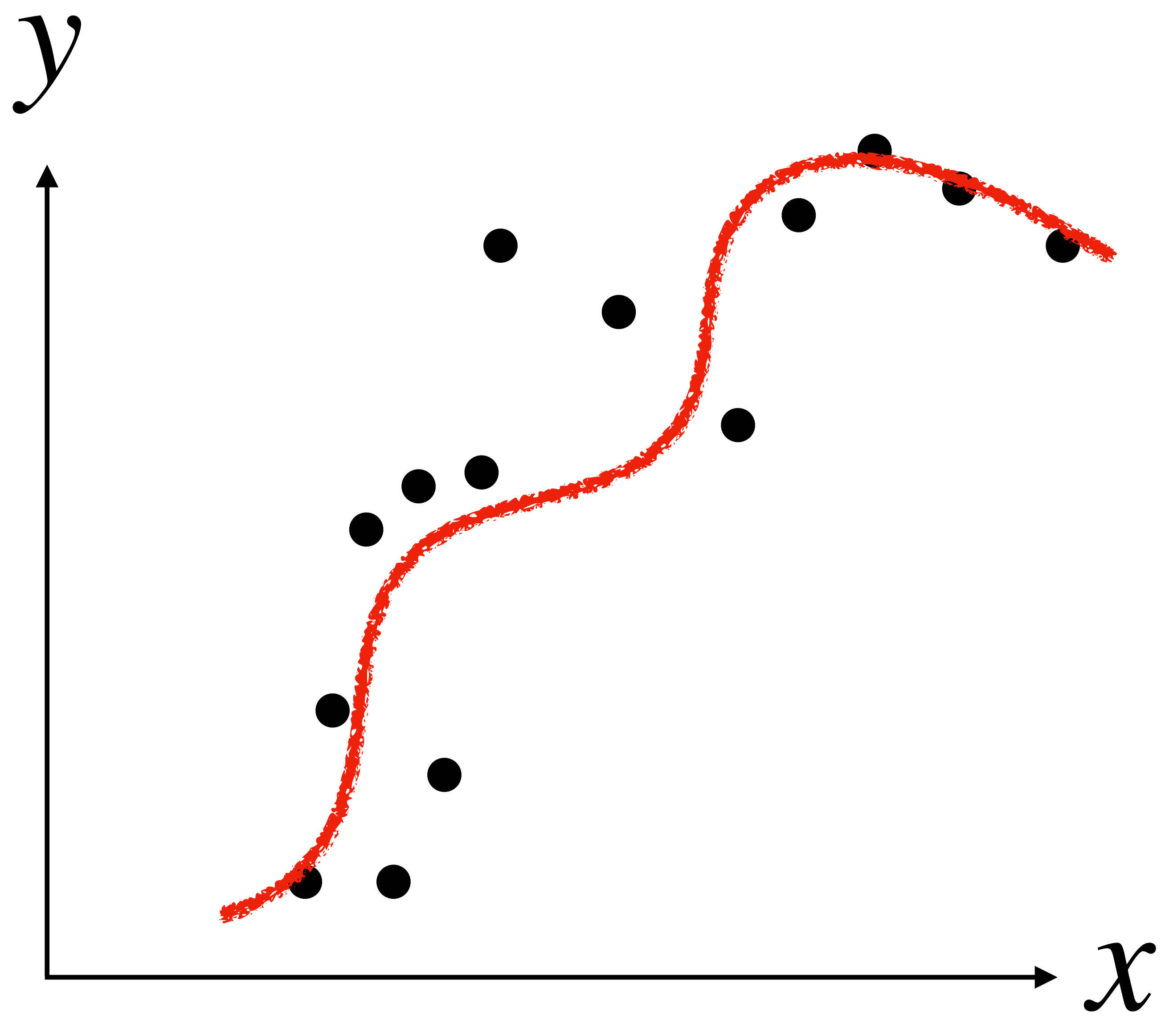
Just right



$$\phi(x) = [1, x, x^2]$$

Overfitting

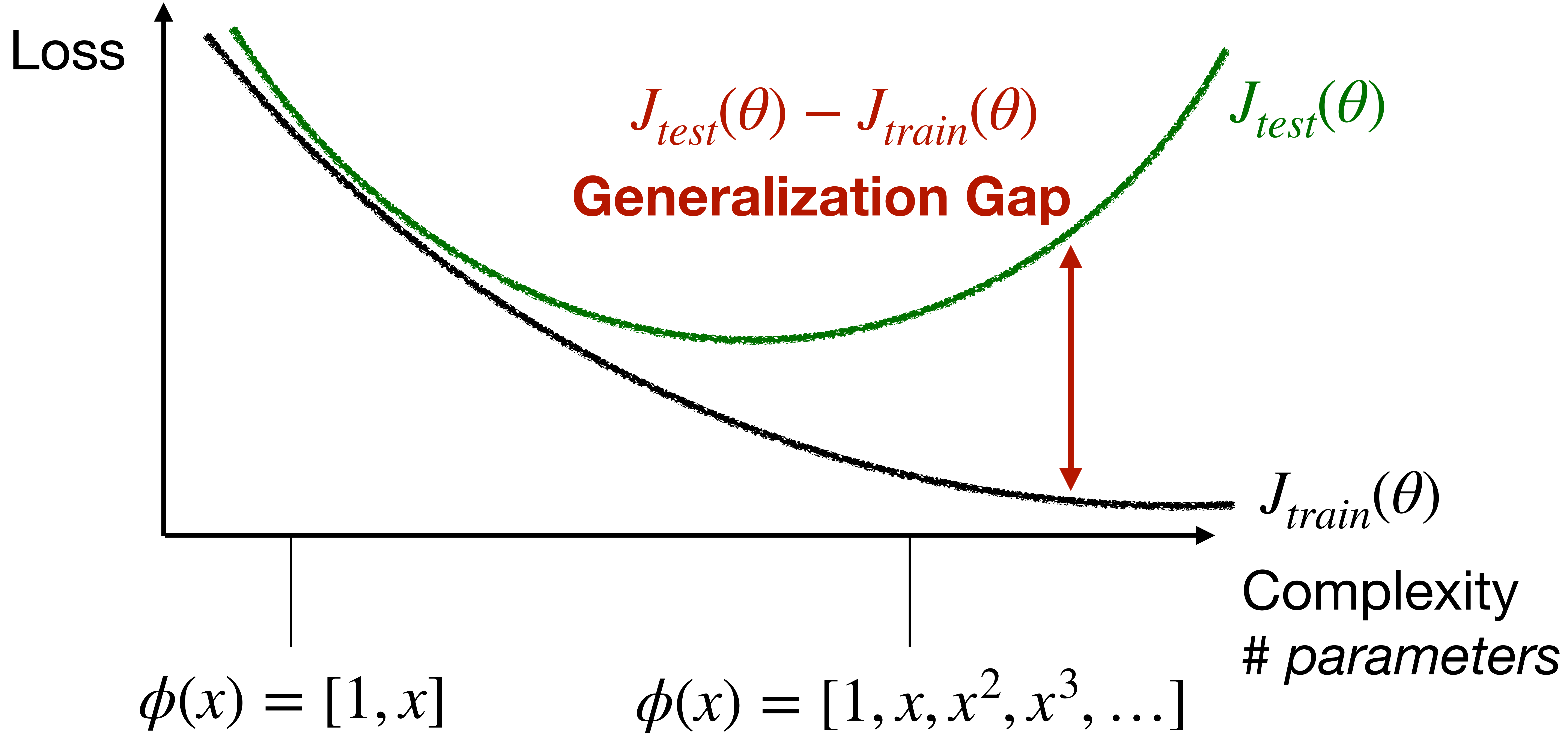
High Variance



$$\phi(x) = [1, x, x^2, x^3, \ldots]$$

Variance Bias Trade-off

Error as a function of complexity

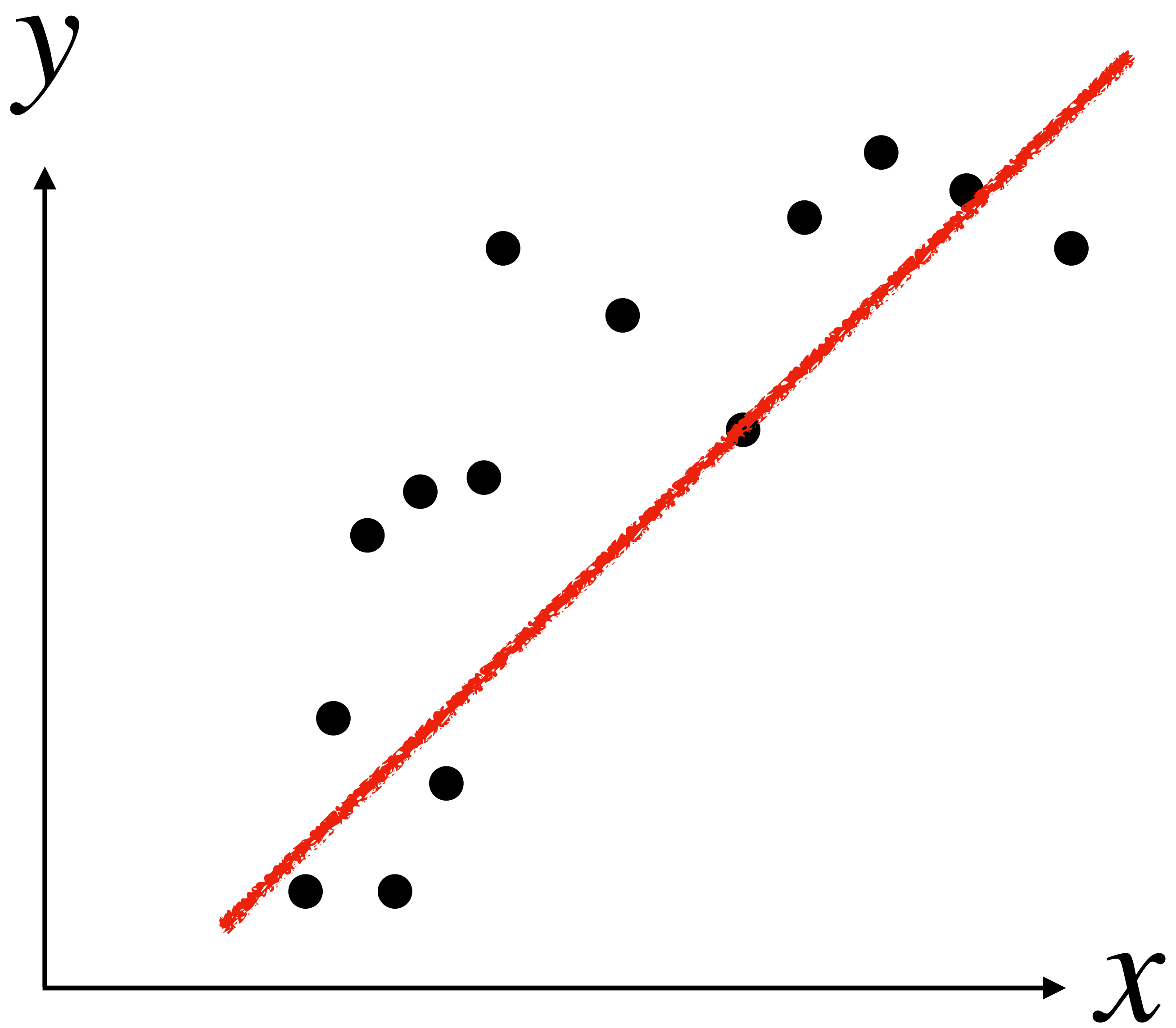


How to choose $\phi(x)$?

How to optimize over $\phi(x)$

Underfitting

High Bias

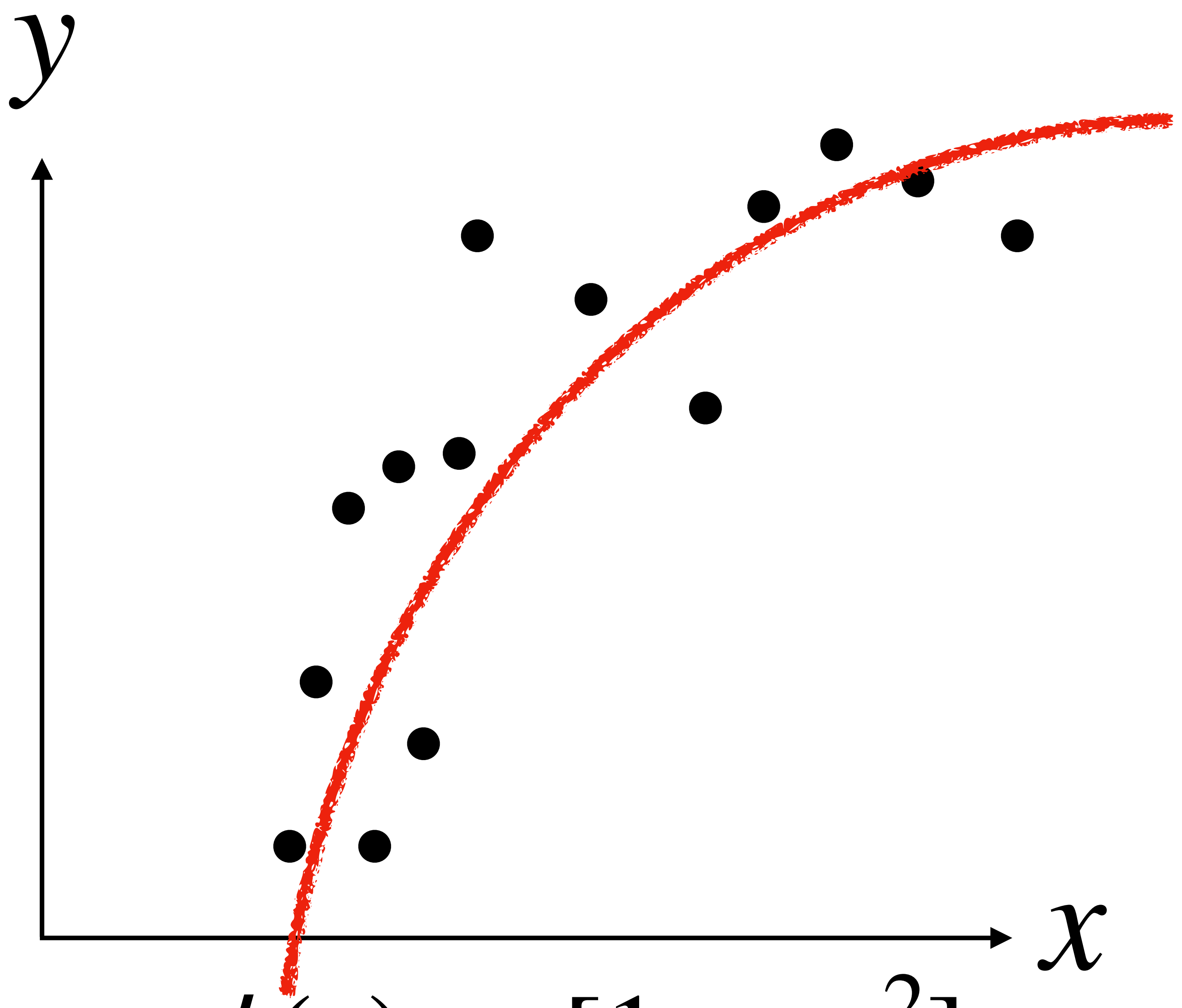


$$\phi(x) = [1, x]$$

J_{test} and J_{train} are big

Bias ≈ what you can get with infinite data

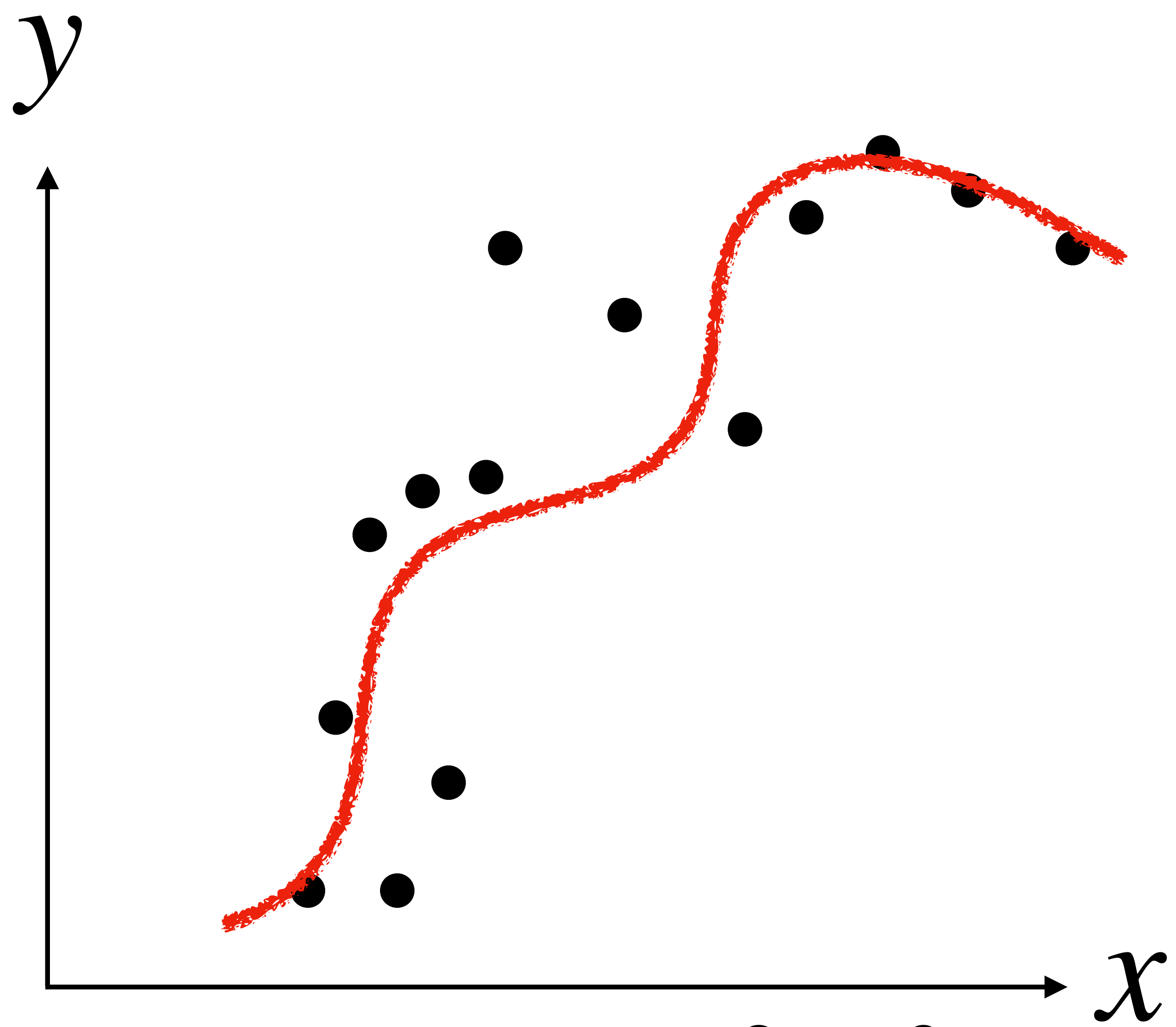
Just right



$$\phi(x) = [1, x, x^2]$$

Overfitting

High Variance



$$\phi(x) = [1, x, x^2, x^3, \ldots]$$

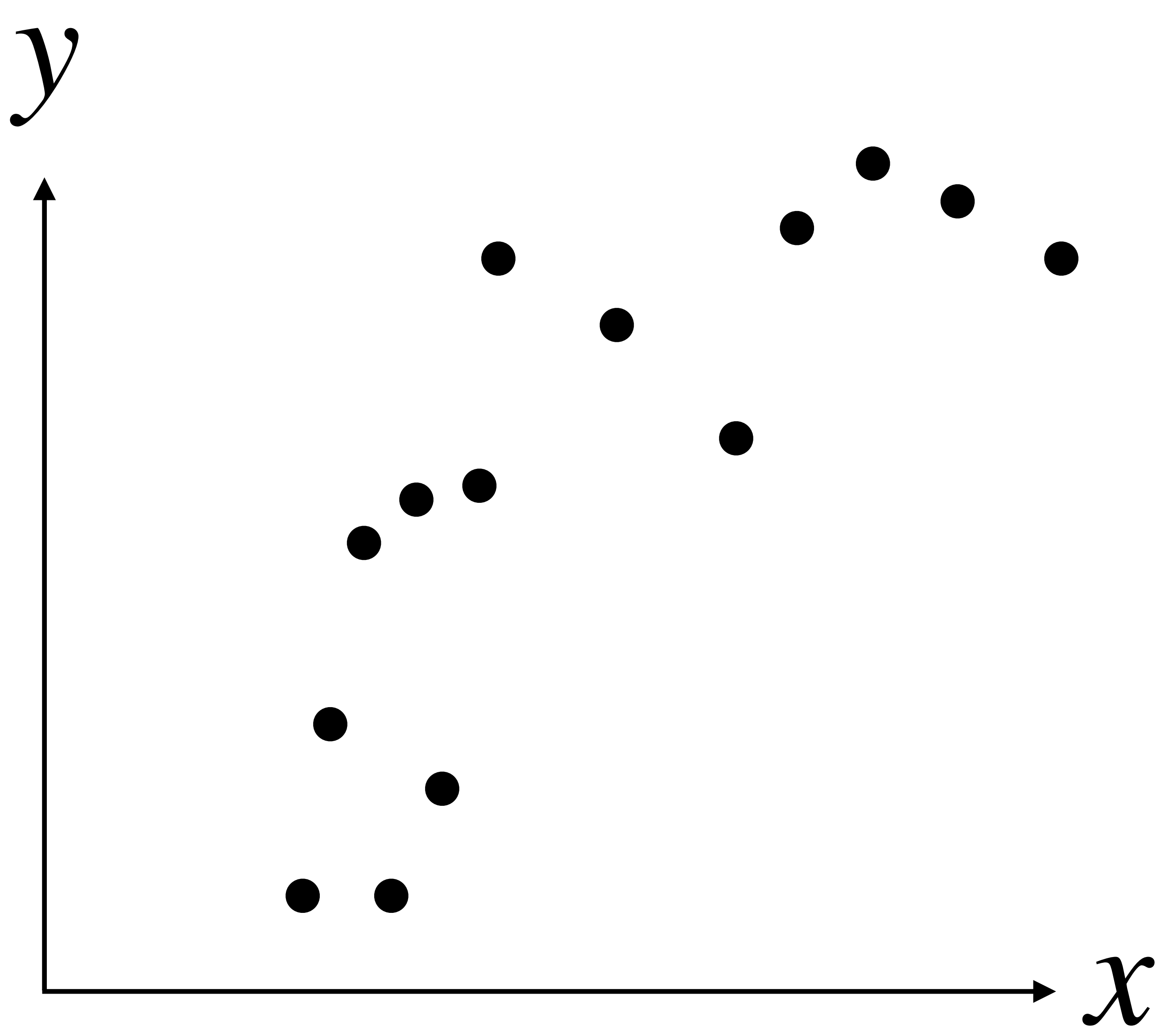
J_{test} is big, J_{train} is small

Sensitive to redrawing new samples

How to choose $\phi(x)$?

How to optimize over $\phi(x)$

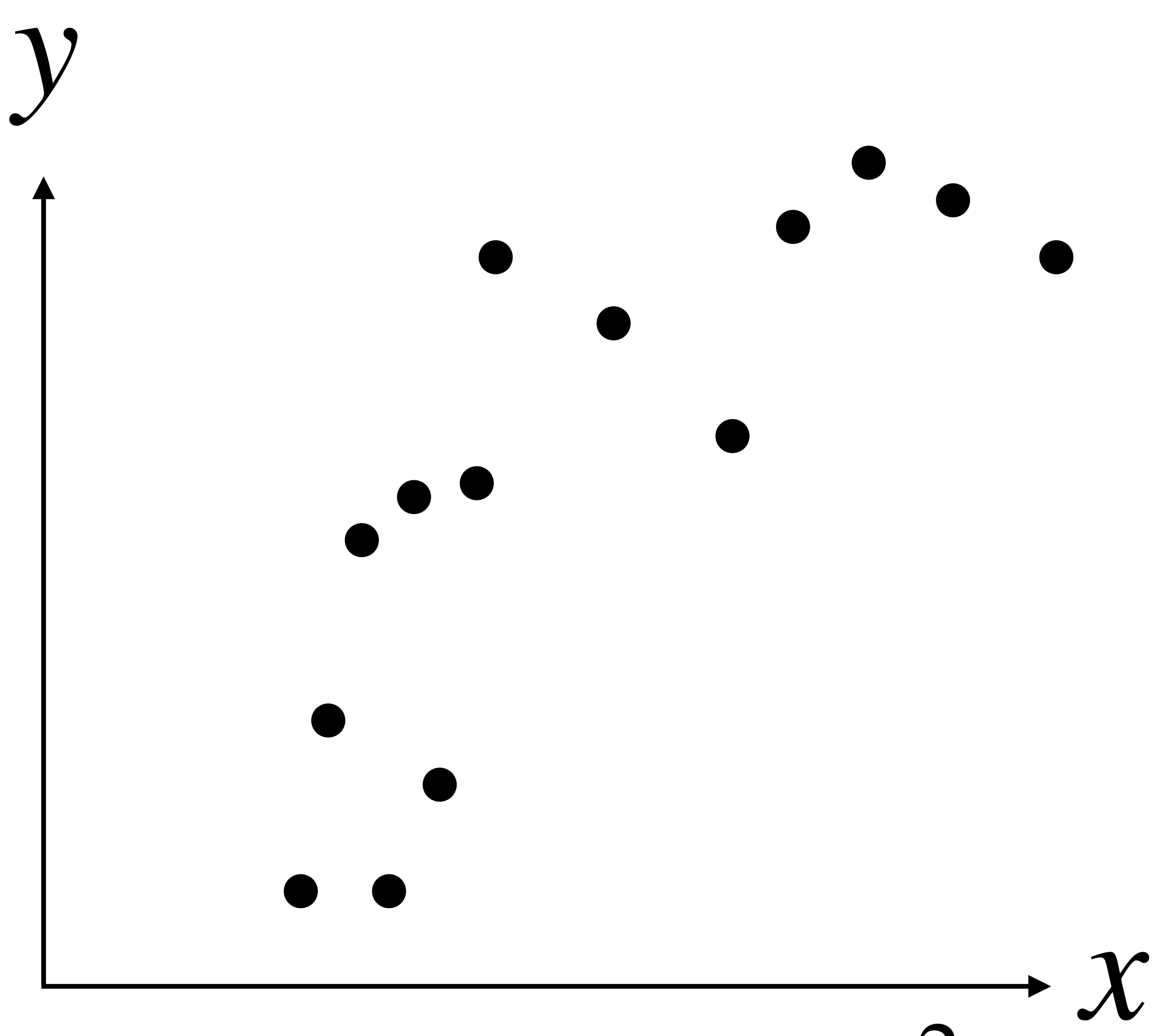
Just right



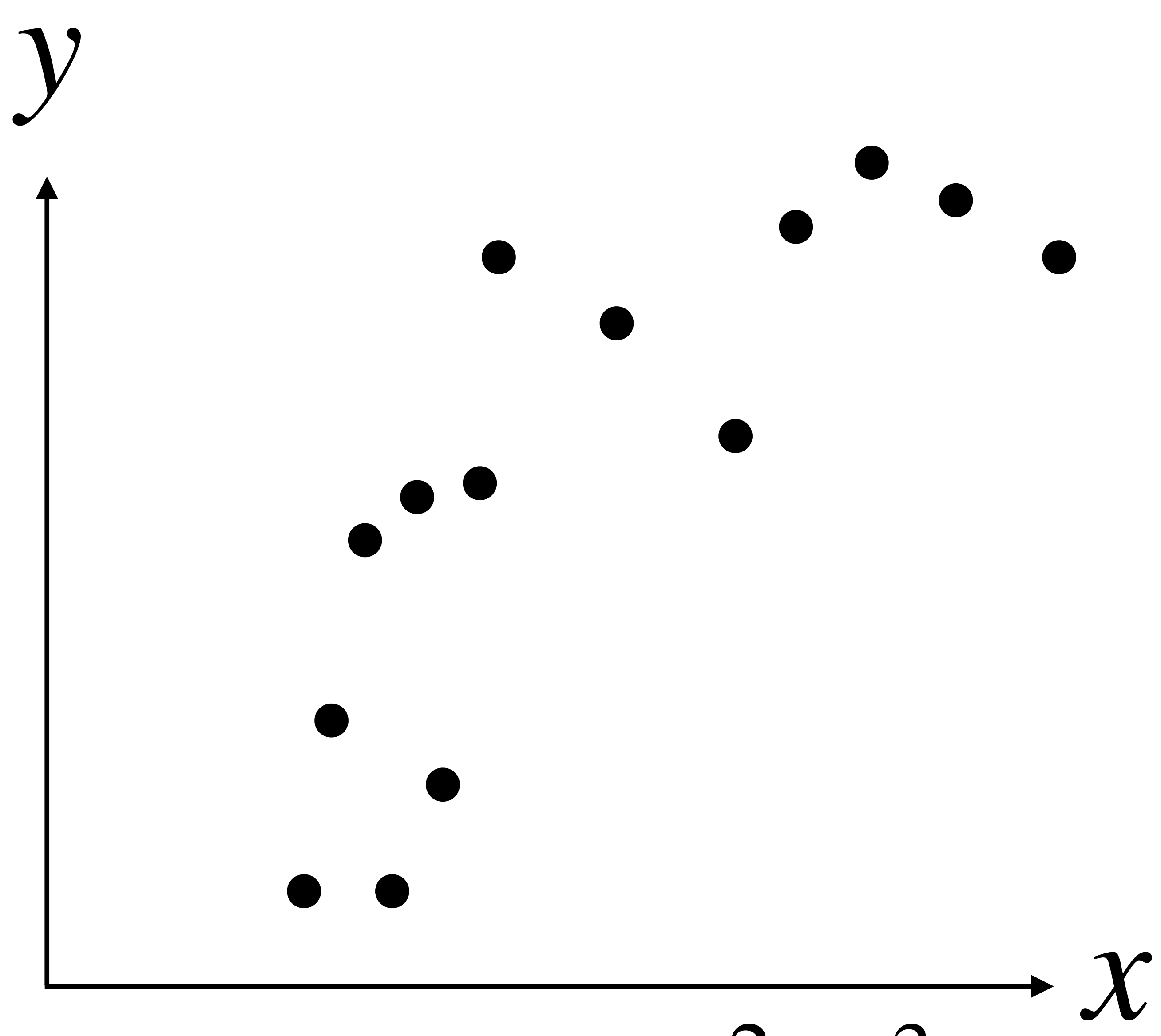
$\phi(x) = [1, x]$

J_{test} and J_{train} are big

Bias ≈ what you can get with infinite data



$\phi(x) = [1, x, x^2]$

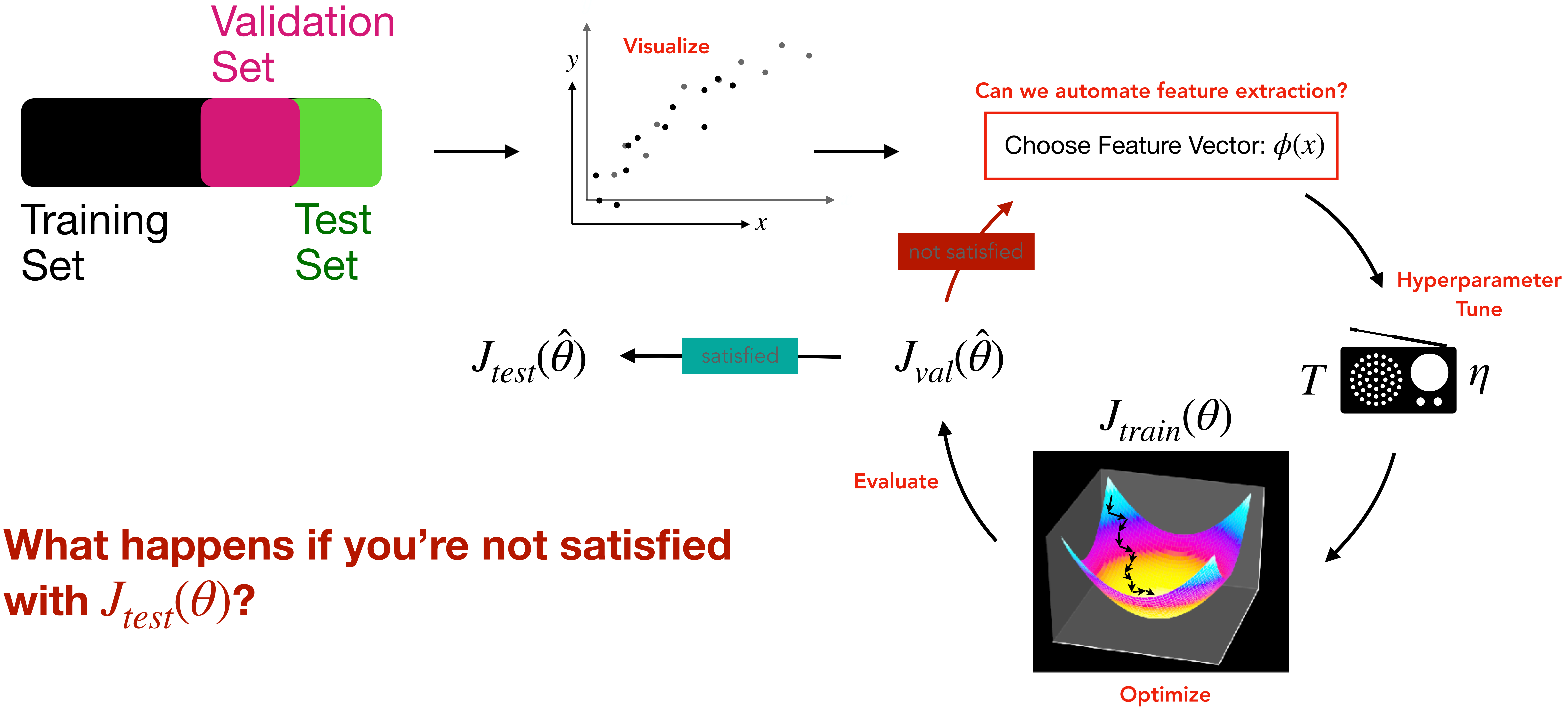


$\phi(x) = [1, x, x^2, x^3, \ldots]$

J_{test} is big, J_{train} is small

Sensitive to redrawing new samples

Machine Learning workflow - Cross Validation

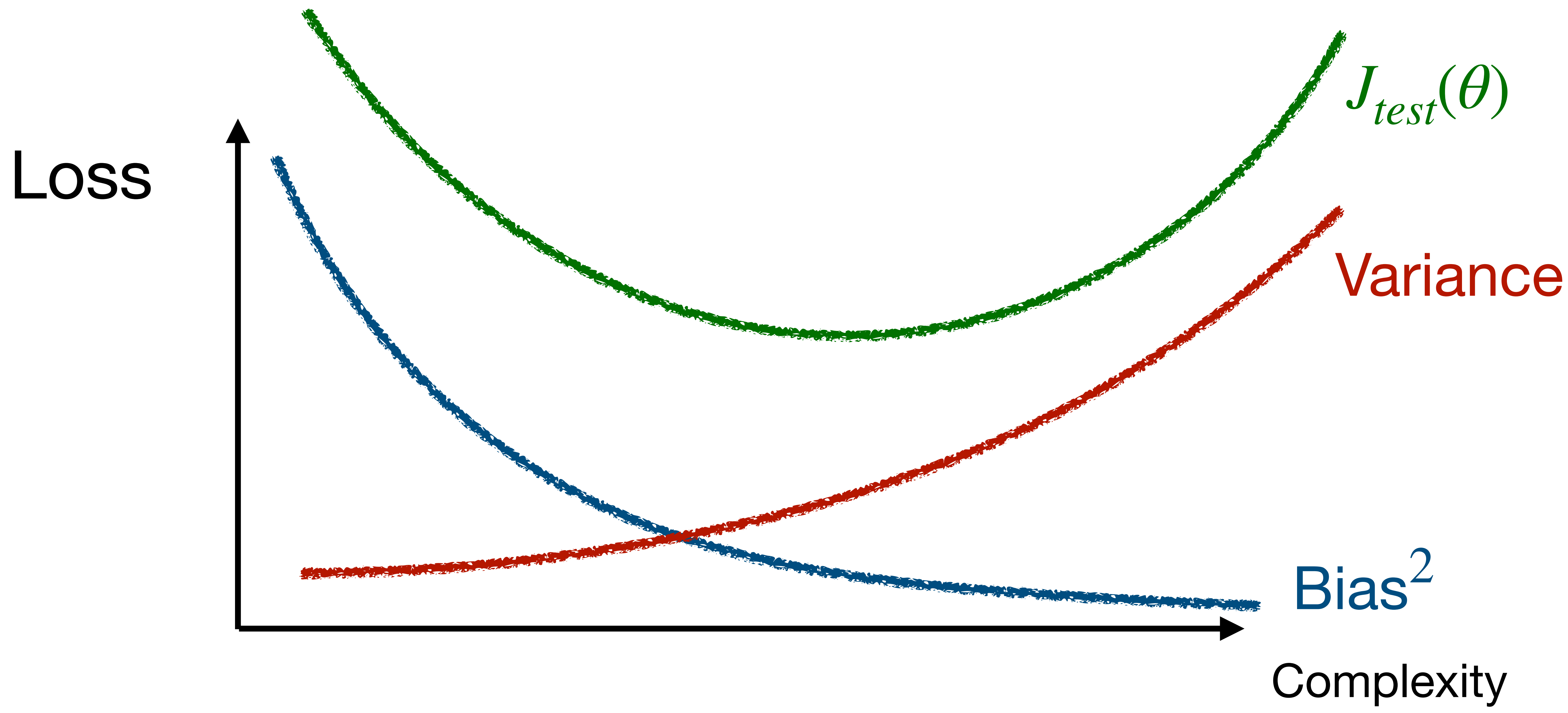


What happens if you're not satisfied with $J_{test}(\theta)$?

Decomposition of Test Error

- Test error can be written as

$$J_{test}(\theta) \sim Bias^2 + Variance$$



Decomposition of Test Error

See derivation in Section 8.1.1

- Draw a training dataset $S = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ such that $y^{(i)} = h^*(x^{(i)}) + \xi^{(i)}$ where $\xi^{(i)} \sim \mathcal{N}(0,\sigma^2)$

- Train a model on the dataset, denoted by $\hat{h}_S$

- Take a test example (x, y) such that $y = h^*(x) + \xi$ where $\xi \sim \mathcal{N}(0,\sigma^2)$ and measure the expected test error (averaged over the random draw of the training set S and the randomness of ξ)

$$\text{MSE}(x) = \mathbb{E}_{S,\xi}\left[(y - \hat{h}_S(x))^2\right]$$

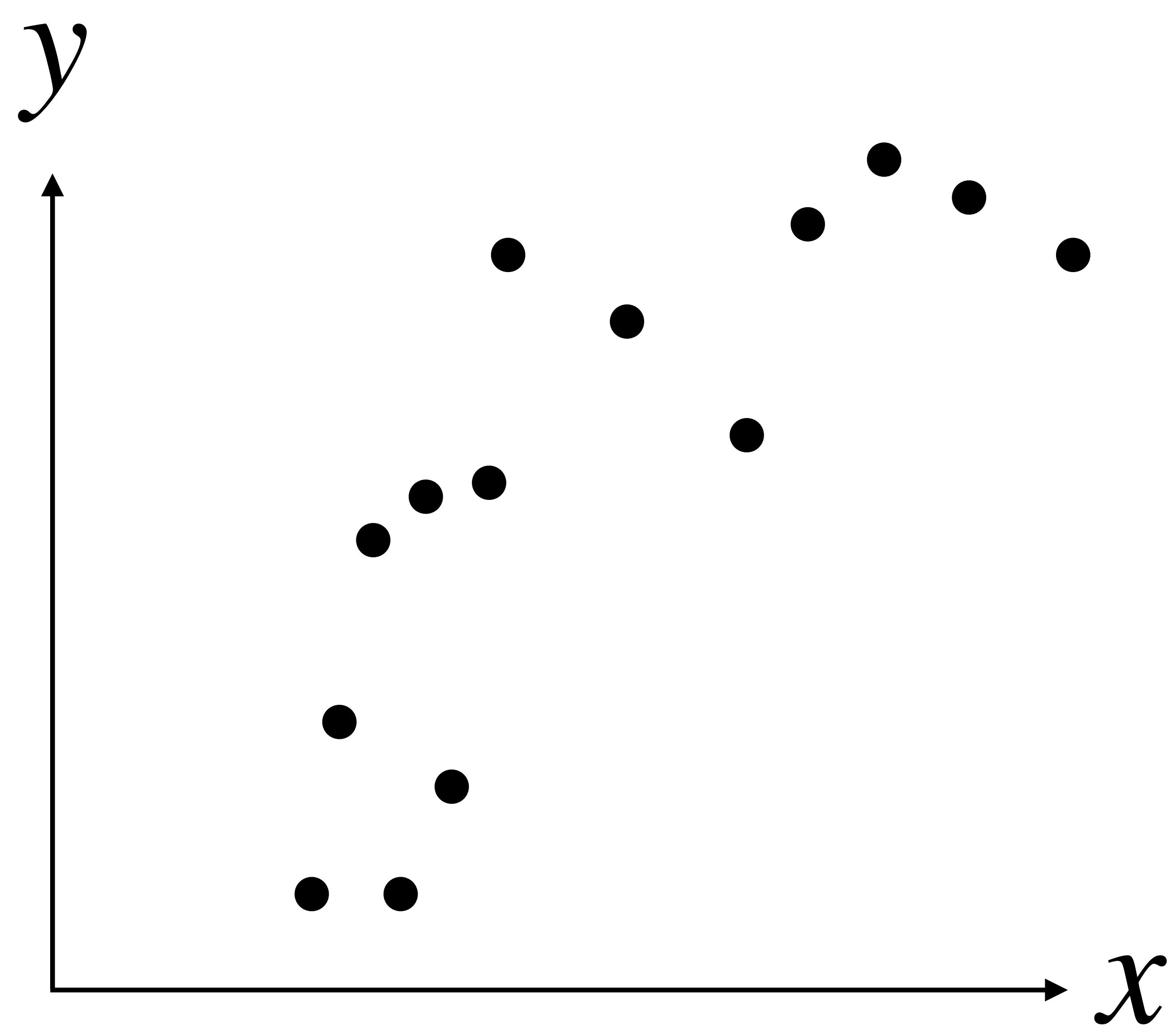
Decomposition of Test Error for square loss

See derivation in Section 8.1.1

$$
\begin{aligned}
\mathrm{MSE}(x) &= \mathbb{E}\left[(y - \hat{h}_S(x))^2\right] \\
&= \mathbb{E}\left[(\xi + (h^*(x) - \hat{h}_S(x)))^2\right] \\
&= \mathbb{E}\left[\xi^2\right] + \mathbb{E}\left[(h^*(x) - \hat{h}_S(x))^2\right] \\
&= \sigma^2 + \mathbb{E}\left[(h^*(x) - \hat{h}_S(x))^2\right] \\
&= \sigma^2 + (h^*(x) - h_{\mathrm{avg}}(x))^2 + \mathbb{E}\left[(h_{\mathrm{avg}}(x) - \hat{h}_S(x))^2\right] \\
&= \underbrace{\sigma^2}_{\text{unavoidable}} + \underbrace{(h^*(x) - h_{\mathrm{avg}}(x))^2}_{\text{bias}^2} + \underbrace{\mathrm{var}(\hat{h}_S(x))}_{\text{variance}}
\end{aligned}
$$

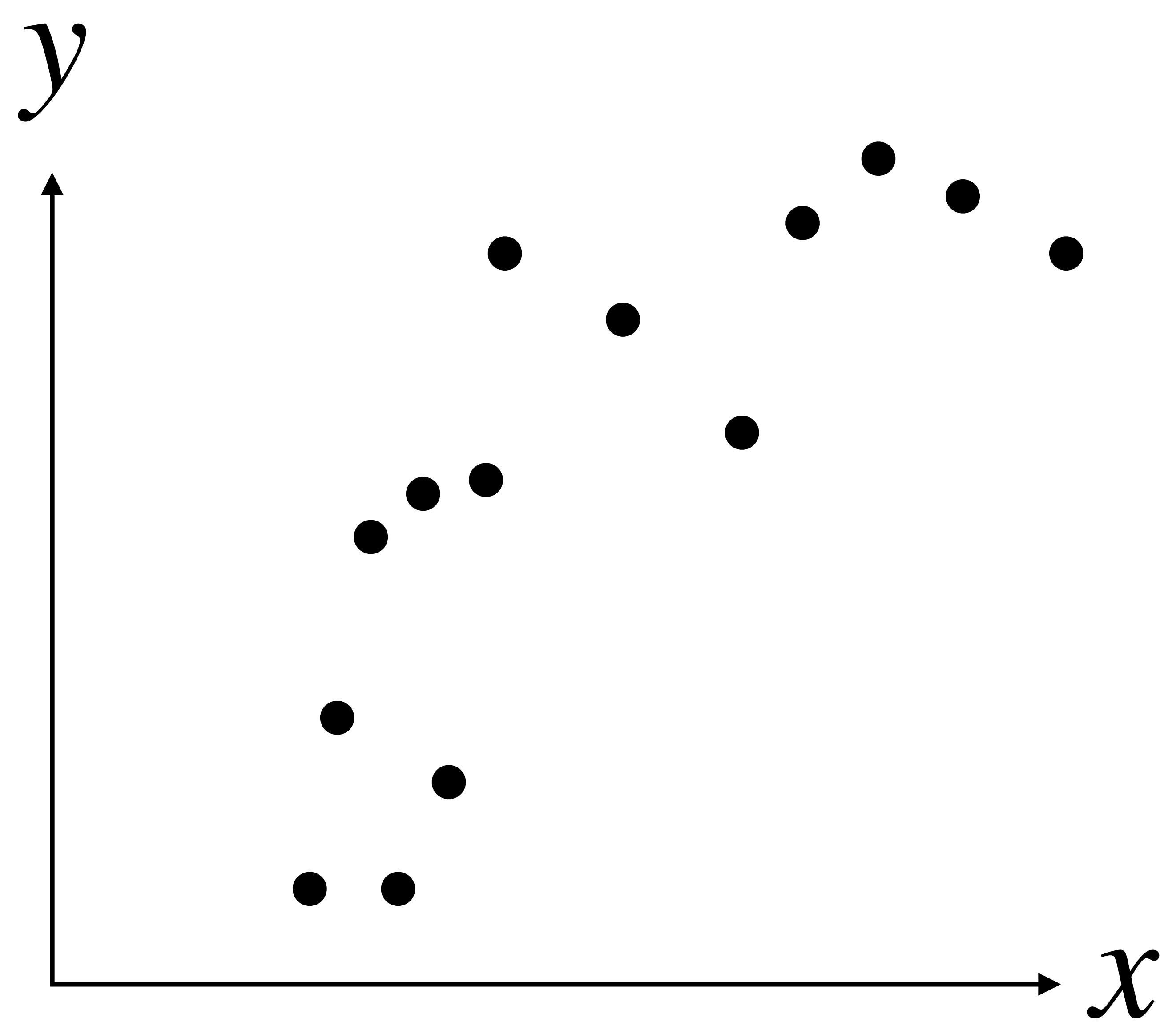
Solving Bias and Variance Problems

- **High Variance:**
    - **Problem:** Lack of data, and model is too expressive
    - **Solution:** More data, and simpler model



Solving Bias and Variance Problems

- **High Bias:**
  - **Problem:** Lack of expressivity (doesn't depend on data)
  - **Solution:** Make model more complex



Regularization

Force fitting parameters to be smaller - ‘shrink’ hypothesis class

$$h_\theta(x) = 100.2 + 50.6x + 70.4x^2 + 1345x^3 + 200.3x^4$$

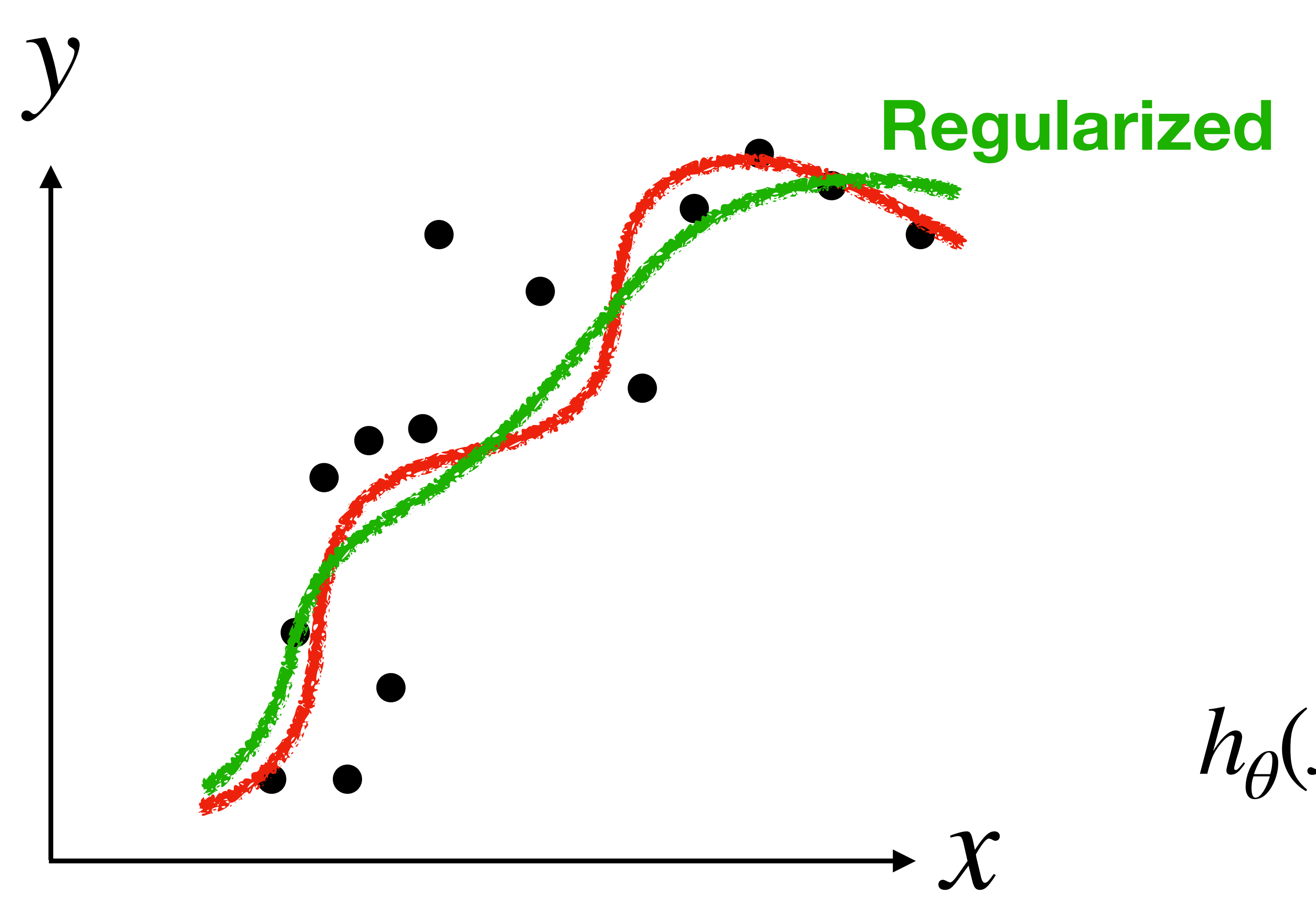
$$J_{reg}(\theta) = J(\theta) + \lambda\, R(\theta)$$

L1 Regularization

$$R(\theta) = \|\theta\|_1$$

$$h_\theta(x) = 5.1x + 7.2x^2 + 3.3x^4$$

Less coefficients



L2 Regularization

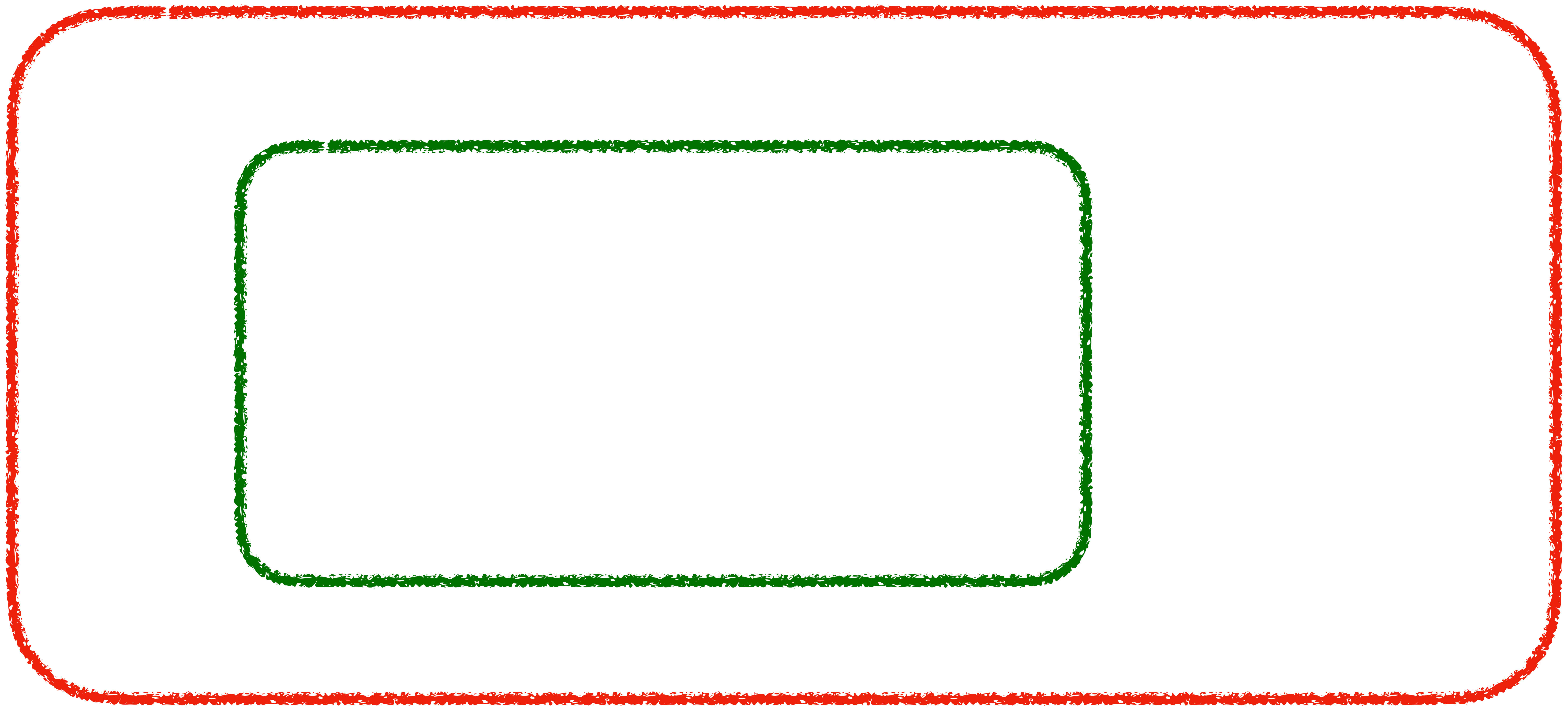
$$R(\theta) = \|\theta\|_2$$

$$h_\theta(x) = .1 + 5.2x + 7.4x^2 + .05x^3 + 2.3x^4$$

Smaller coefficients

Regularization

Force fitting parameters to be smaller - ‘shrink’ hypothesis class



Regularization

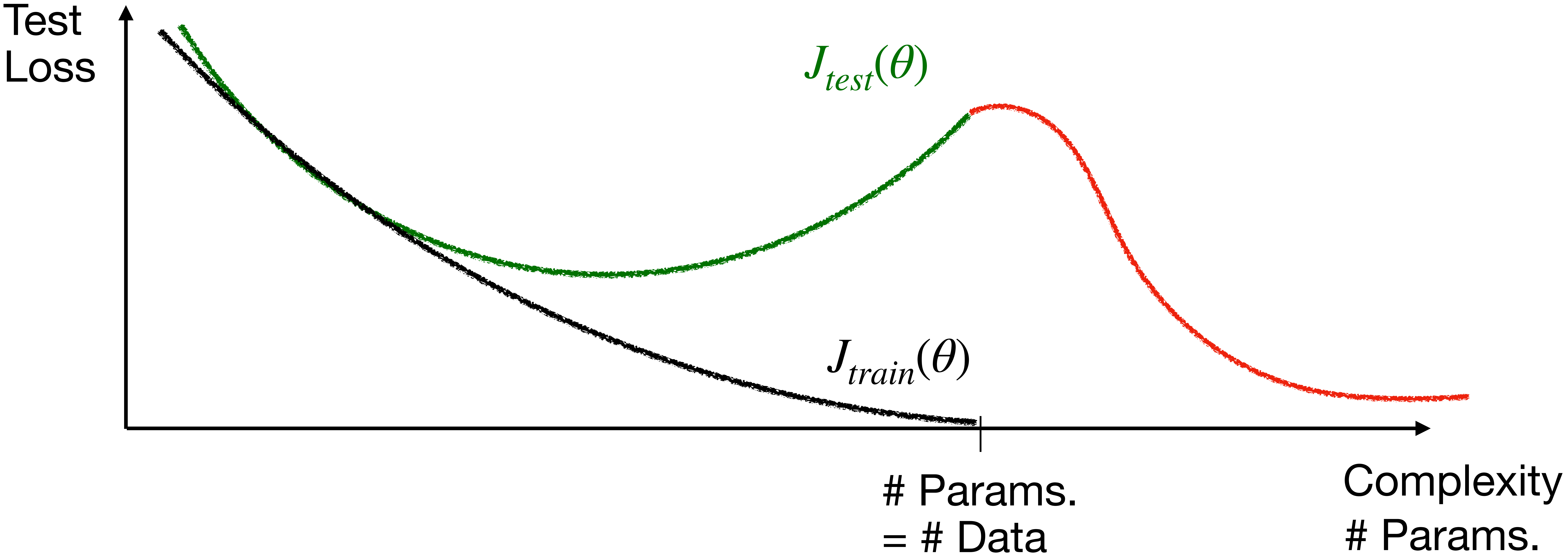
Force fitting parameters to be smaller - ‘shrink’ hypothesis class

$$J_{reg}(\theta) = J(\theta) + \lambda R(\theta)$$

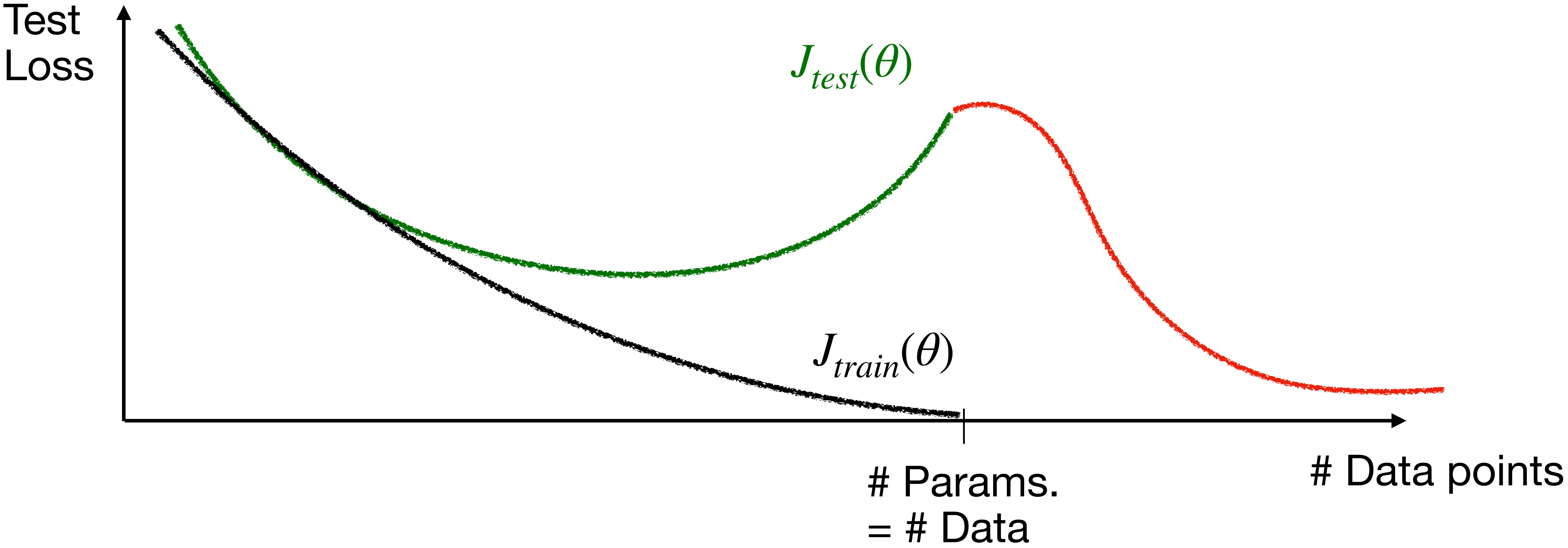
- **Other regularizations:**
  - $R(\theta) = ||\theta||_0$ **enforces sparsity**
  - **Loss related to smoothness or equation known about problem**

Double Descent

Model-wise



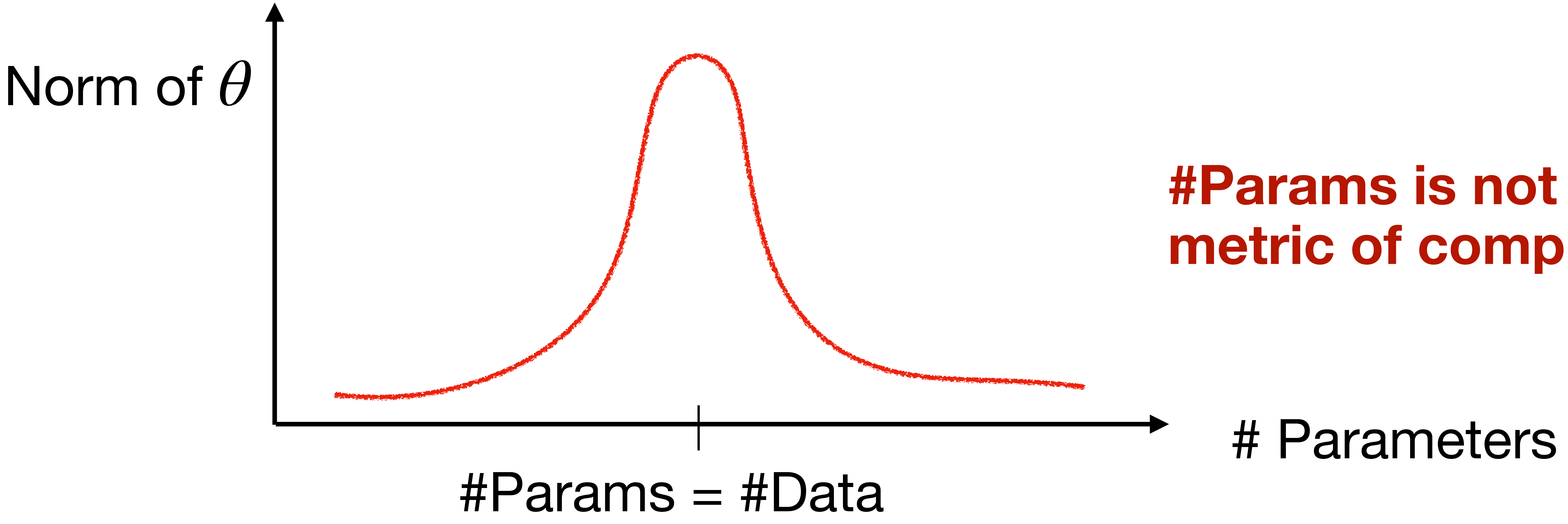
Double Descent



Double Descent

Why?

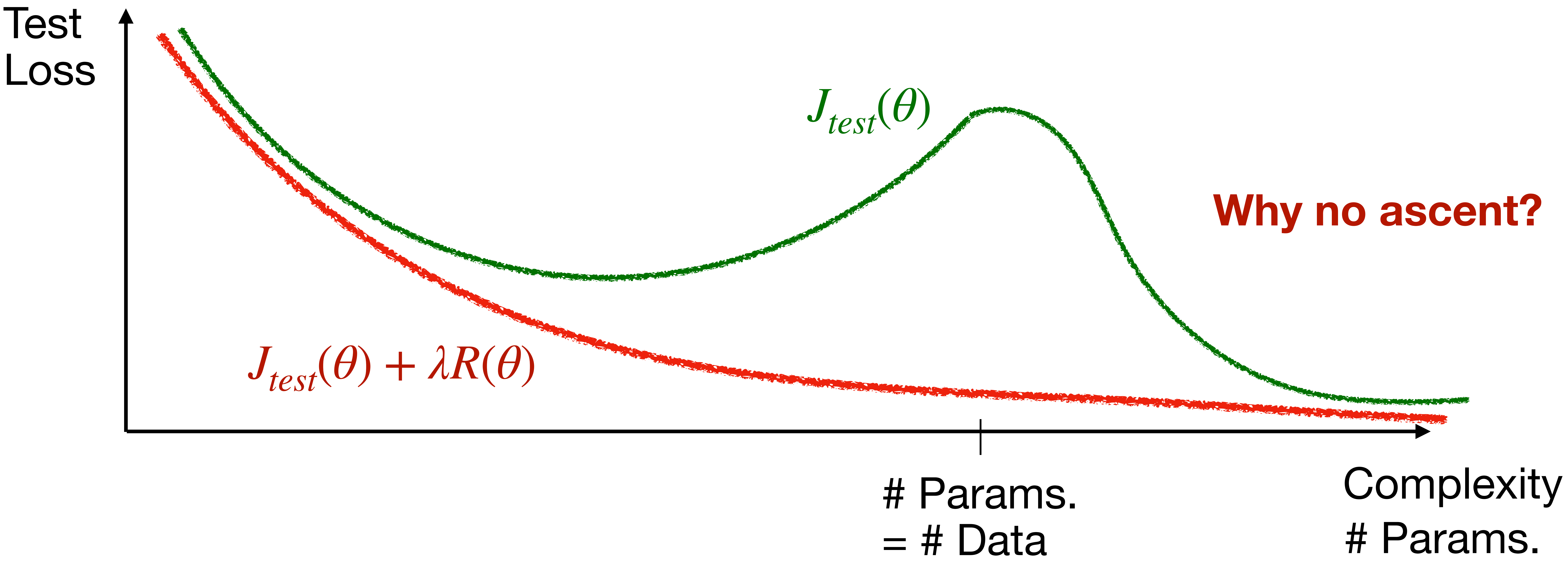
- Existing algorithms (e.g. linear models) underperform dramatically when #params = #data
- Norm of θ is big when #params = #data



#Params is not a good metric of complexity?

Double Descent

Regularization solves the problem with large parameters

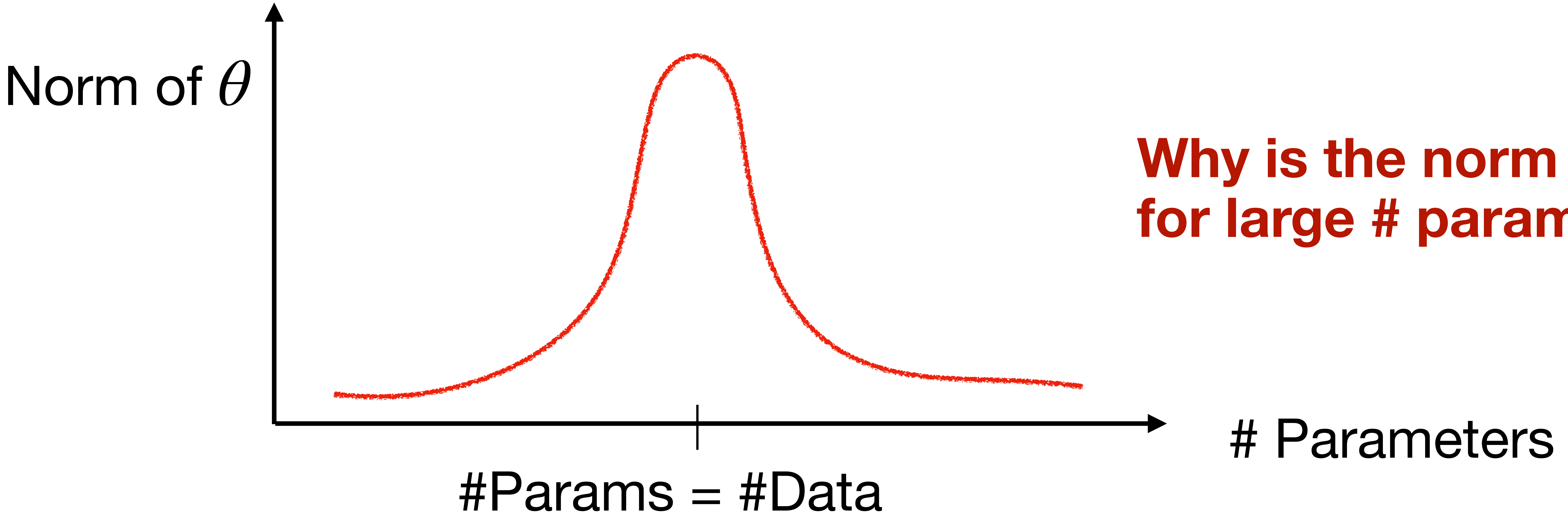


Double Descent

Why?

- Existing algorithms (e.g. linear models) underperform dramatically when #params = #data

- Norm of θ is big when #params = #data



Why is the norm small for large # parameters?