Day 2: Overfitting, regularization

Introduction to Machine Learning Summer School
June 18, 2018 - June 29, 2018, Chicago

Instructor: Suriya Gunasekar, TTI Chicago

19 June 2018

TOYOTA TECHNOLOGICAL INSTITUTE AT CHICAGO
THE UNIVERSITY OF CHICAGO

Review

- Yesterday
  - Supervised learning
  - Linear regression - polynomial curve fitting
  - Empirical risk minimization, evaluation
- Today
  - Overfitting
  - Model selection
  - Regularization
  - Gradient descent
- Schedule:

9:00am-10:25am	Lecture 2.a: Overfitting, model selection
10:35am-noon	Lecture 2.b: Regularization, gradient descent
noon-1:00pm	Lunch
1:00pm-3:30pm	Programming
3:30pm-5:00pm	Invited Talk - Mathew Walter

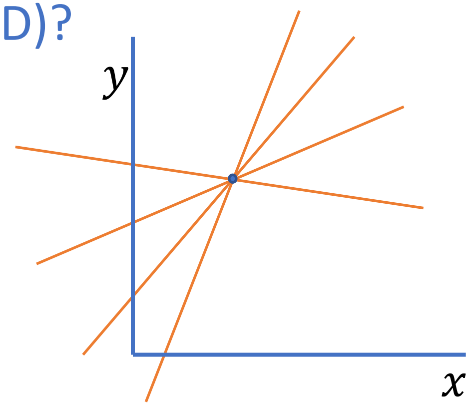
Overfitting

Dataset size and linear regression

- Recall linear regression
  - Input $\boldsymbol{x} \in \mathbb{R}^d$, output $y \in \mathbb{R}$, training data $S = \{(\boldsymbol{x}^{(i)}, y^{(i)}) : i = 1,2, \dots, N\}$
  - Estimate $\boldsymbol{w} \in \mathbb{R}^d$ and bias $w_0 \in \mathbb{R}$ by minimizing training loss

$$\widehat{\boldsymbol{w}}, \widehat{w}_0 = \underset{\boldsymbol{w}, w_0}{\operatorname{argmin}} \sum_{i=1}^{N} \left(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0 - y^{(i)}\right)^2$$

- What happens when we only have a single data point (in 1D)?
  - Ill-posed problem: an infinite number of lines perfectly fit the data

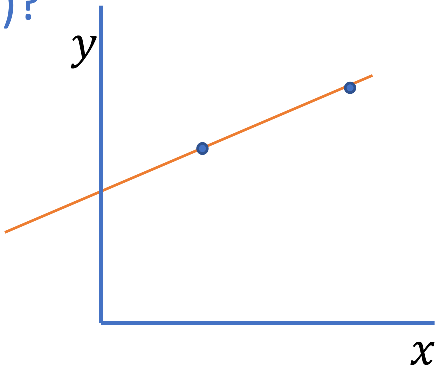


Dataset size and linear regression

- Recall linear regression
  - Input $\boldsymbol{x} \in \mathbb{R}^d$, output $y \in \mathbb{R}$, training data $S = \{(\boldsymbol{x}^{(i)}, y^{(i)}) : i = 1,2, \ldots, N\}$
  - Estimate $\boldsymbol{w} \in \mathbb{R}^d$ and bias $w_0 \in \mathbb{R}$ by minimizing training loss

$$\widehat{\boldsymbol{w}}, \widehat{w}_0 = \underset{\boldsymbol{w}, w_0}{\operatorname{argmin}} \sum_{i=1}^{N} \left(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0 - y^{(i)}\right)^2$$

- What happens when we only have a single data point (in 1D)?
  - Ill-posed problem: an infinite number of lines perfectly fit the data



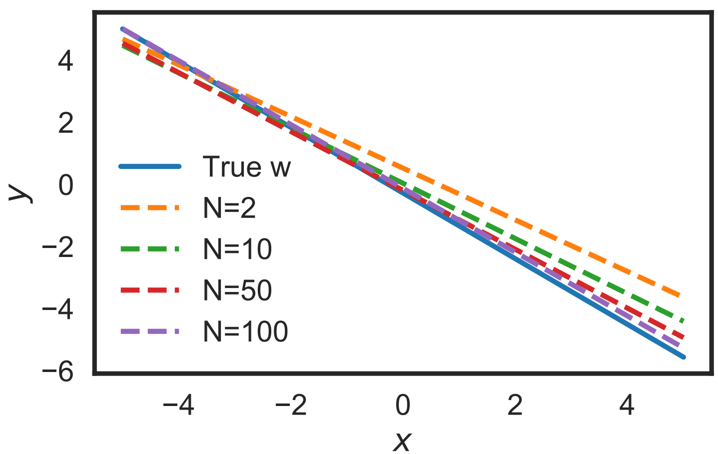
- Two points in 1D?
- Two points in 2D?
  - the amount of data needed to obtain a meaningful estimate of a model is related to the number of parameters in the model (its complexity)

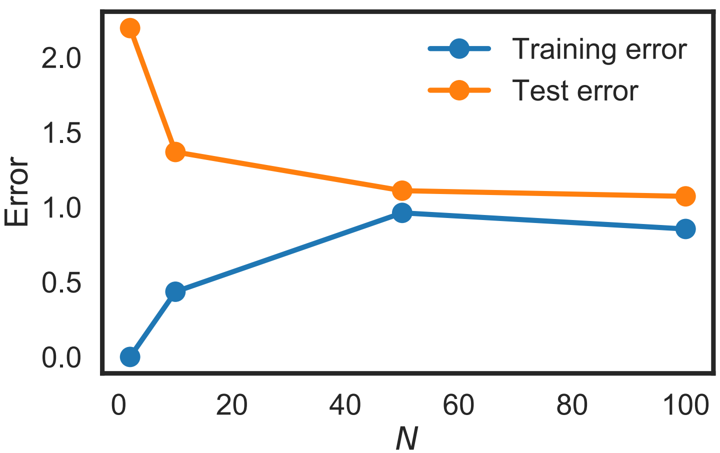
Linear regression - generalization

Consider 1D example

- $S_{train} = \{(x^{(i)}, y^{(i)}): i = 1,2,\dots,N\}$ where
  - $x^{(i)} \sim uniform(-5,5)$
  - $y^{(i)} = w^* x^{(i)} + \epsilon^{(i)}$ for true w^* and noise $\epsilon^{(i)} \sim \mathcal{N}(0,1)$
- S_{test} similarly generated

$$\hat{w} = \operatorname*{argmin}_w \sum_{i=1}^{N} \left(wx^{(i)} - y^{(i)}\right)^2$$





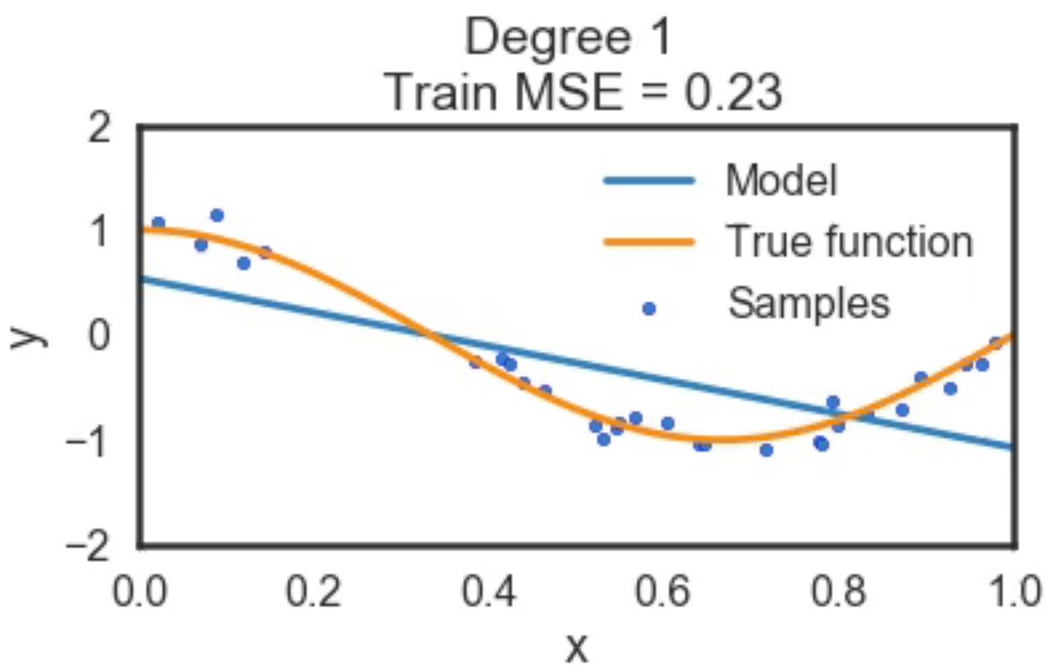
- The training error increases with the size of training data?

Model complexity vs fit for fixed N

- Recall polynomial regression of degree m in 1D

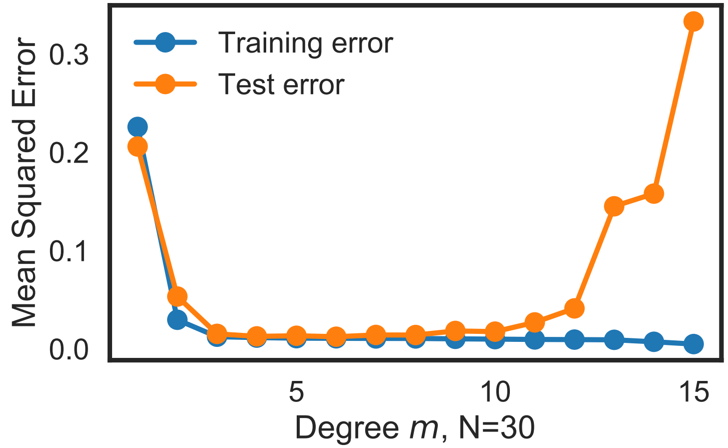
$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w} \in \mathbb{R}^{m+1}}{\operatorname{argmin}} \sum_{i=1}^{N} \left(w_0 + w_1 . x^{(i)} + w_2 . x^{(i)^2} + \cdots + w_m . x^{(i)^m} - y_t\right)^2$$

N=30



Overfitting with ERM

- For same amount of data, more complex models overfits more than simple model
  - Recall: higher degree → more number of parameters to fit
- What happens if we have more data?

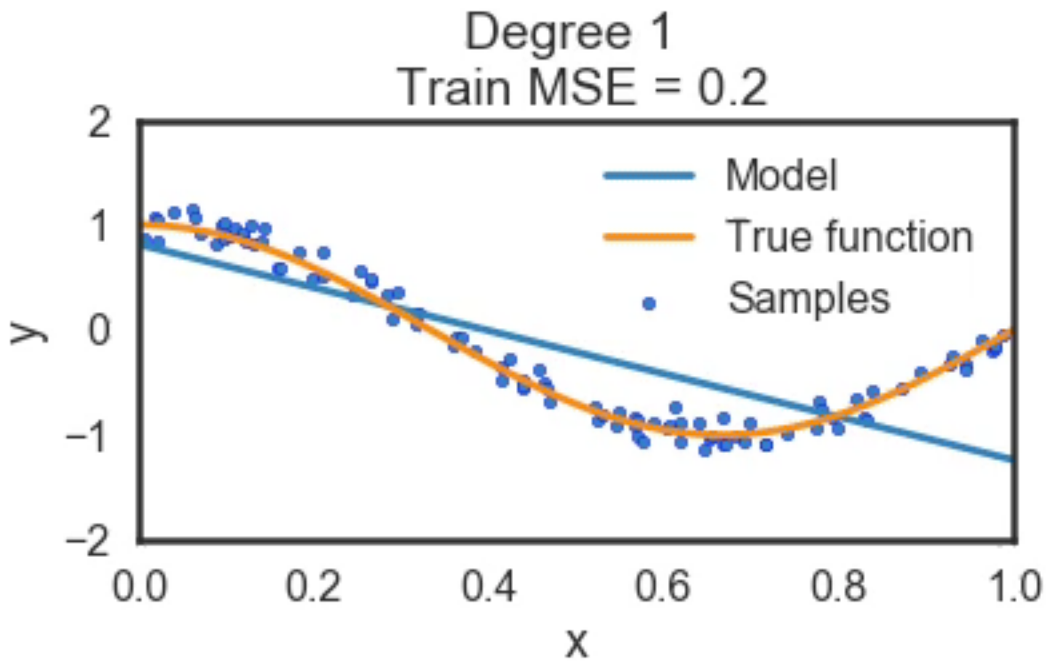


Model complexity vs fit for fixed N

- Recall polynomial regression of degree m in 1D

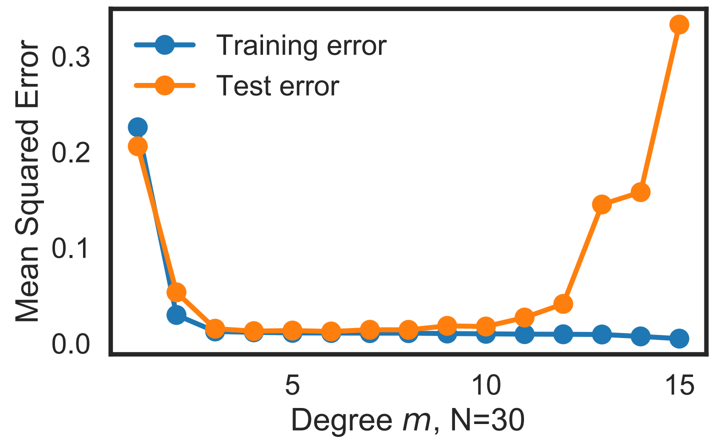
$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}\in\mathbb{R}^{m+1}}{\operatorname{argmin}} \sum_{i=1}^{N}\left(w_0 + w_1.x^{(i)} + w_2.x^{(i)^2} + \cdots + w_m.x^{(i)^m} - y_t\right)^2$$

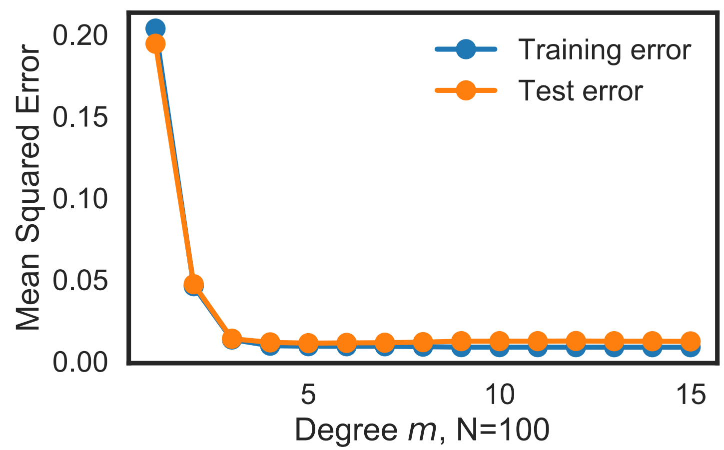
N=100



Overfitting with ERM

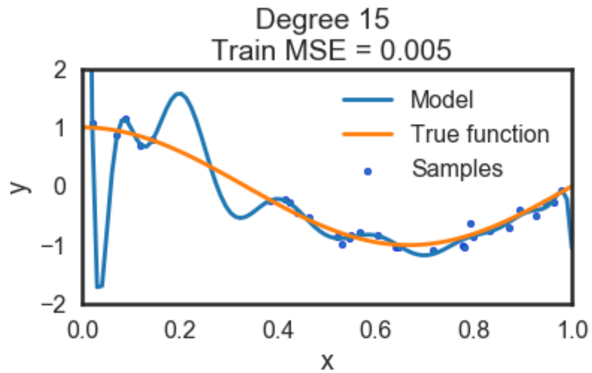
- For same amount of data, complex models overfit more than simple models
  - Recall: higher degree → more number of parameters to fit
- What happens if we have more data?
  - More complex models require more data to avoid overfitting

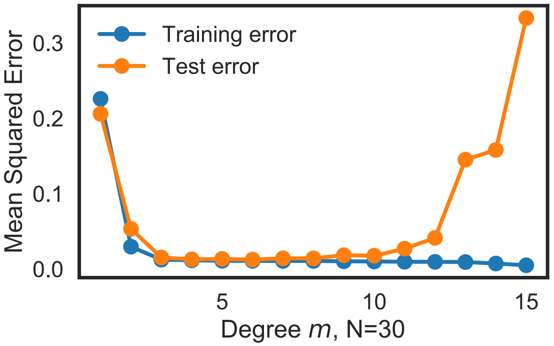




How to avoid overfitting?

- How to **detect** overfitting?





- How to **avoid** overfitting?
  - Look at test error and pick m=5?
- Split $S = S_{train} \cup S_{val} \cup S_{test}$
  - Use performance on S_{val} as proxy for test error

Model selection

- $S = S_{train} \cup S_{val} \cup S_{test}$
- m model classes $\{\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_m\}$
  - Recall each $\mathcal{H}_r$ is a set of candidate functions mapping $x \to y$
  - e.g., $\mathcal{H}_r = \{x \to w_0 + w_1 . x + w_2 . x^2 + \cdots w_r . x^r\}$
- Minimize training loss $L_{S_{train}}$ on S_{train} to pick best $\hat{f}_r \in \mathcal{H}_r$
  - e.g., $\hat{f}_r(x) = \widehat{w}_0 + \widehat{w}_1 . x + \widehat{w}_2 . x^2 + \cdots \widehat{w}_r . x^r$ *where* $\widehat{w}_0, \widehat{w}_1, \widehat{w}_2, \dots, \widehat{w}_r$

$$= \underset{w_0,\dots,w_r}{\operatorname{argmin}} \sum_{(x^{(i)}, y^{(i)}) \in S_{train}} \left(w_0 + w_1 x^{(i)} + w_2 x^{(i)^2} + \cdots + w_r x^{(i)^r} - y^{(i)}\right)^2$$

- Compute validation loss $L_{S_{val}}(\hat{f}_r)$ on S_{val} for each $\{\hat{f}_1, \hat{f}_2, \dots, \hat{f}_m\}$
- Pick $\hat{f}^* = \min\{L_{S_{val}}(\hat{f}_1), L_{S_{val}}(\hat{f}_2), \dots, L_{S_{val}}(\hat{f}_m)\} = \min_r L_{S_{val}}(\hat{f}_r)$
- Evaluate test loss $L_{S_{test}}(\hat{f}^*)$

Model selection

- Can we overfit to validation data?
  - How much data to keep aside for validation?
- What if we don't have enough data?

Cross validation

Split $S =$
$S_1 \cup S_2 \cup \cdots \cup S_K$
$\cup\, S_{test}$

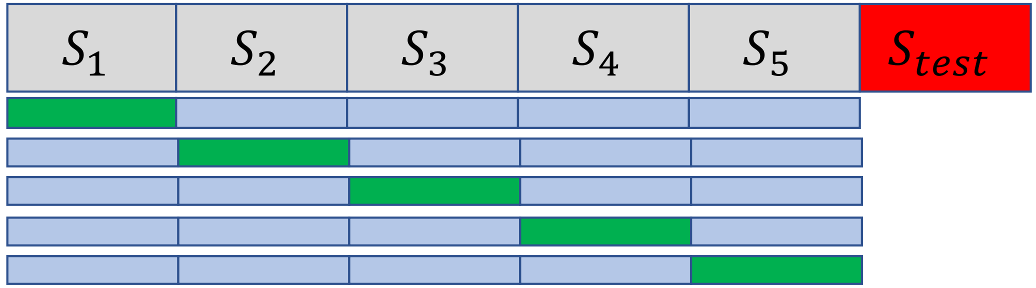


Illustration credit: Nati Srebro

Extreme case $K = N$ (leave one out cross validation)

- m model classes $\{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_m\}$
- For each k:
  - **Training loss** $L_{S^k_{train}}$ is loss on $S^k_{train} = S_1 \cup S_2 \ldots \cup S_{k-1} \cup S_{k+1} \ldots S_K$
  - Let best $\hat{f}^{(k)}_r \in \mathcal{H}_r$ by $\hat{f}^{(k)}_r = \underset{f \in \mathcal{H}_r}{\operatorname{argmin}}\, L_{S^k_{train}}(f)$
  - Compute **validation loss** $L_{S_k}\left(\hat{f}^{(k)}_r\right)$ on S_k for each r
- **Pick model based on average validation loss** $\hat{r}^* = \underset{r}{\operatorname{argmin}} \sum_{k=1}^{K} L_{S_k}\left(\hat{f}^{(k)}_r\right)$
- $\mathcal{H}_{\hat{r}^*}$ is the correct model class to use.
- $\hat{f}^* = \underset{f \in \mathcal{H}_{\hat{r}^*}}{\operatorname{argmin}}\, L_{S^k_{train} \cup S_k}(f)$ or $\hat{f}^* = \sum_k \hat{f}^{(k)}_{\hat{r}^*}$ (if it makes sense)
- Evaluate $L_{S_{test}}(\hat{f}^*)$