CSE 417T
Introduction to Machine Learning

Lecture 11

Instructor: Chien-Ju (CJ) Ho

Logistics: Exam 1

- Exam 1 Date: March 3, 2020 (Tuesday)
  - In-class exam (the same time/location as the lecture)
  - Exam duration: 75 minutes
  - Planned exam content: LFD Chapter 1 to 5
    - Everything in textbook/lectures are included, except for parts labeled as “safe to skip”.
  - Exam format
    - 2 sections
      - Written-response questions
      - Multiple choice questions

Logistics: Exam 1

- More about Exam 1
  - Closed-book exam
  - You can bring two cheat-sheets
    - Up to letter size, front and back (up to 4 pages)
    - No format limitations (it can be typed, written, or a combination)
  - No calculators (you don’t need them)

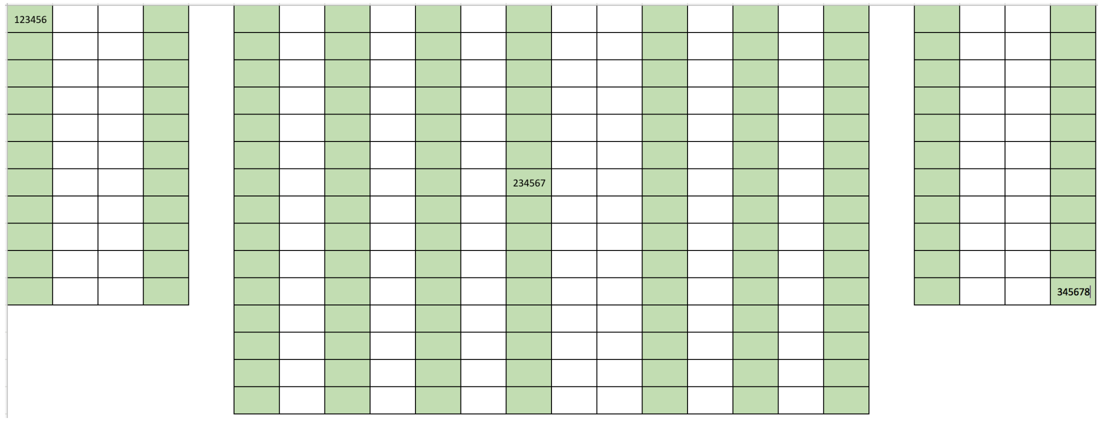
Logistics: Lectures Before Exam 1

- Feb 18 (Tue): Regularization (LFD 4.2)
- Feb 20 (Thu): Validation (LFD 4.3)
- Feb 25 (Tue): Three Learning Principles (LFD Chapter 5)
  - In the unlikely event that we can't finish Chapter 5, we will remove it from the exam.

- Feb 27 (Thu): Review
  - I'll post practice questions around Feb 25 and discuss answers in lectures

- Mar 3 (Tue): Exam 1

Logistics: Policies



- I plan to arrange random seat assignments
  - Will be announced the night before the exam

- If you have a question or if you finish before time is up:
  - **Do not get up**
  - Raise your hand and I will come to you
  - I may or may not answer your question

- When time is called:
  - **Stop writing**
  - **Do not get up**
  - Proctors will come around and collect your exam

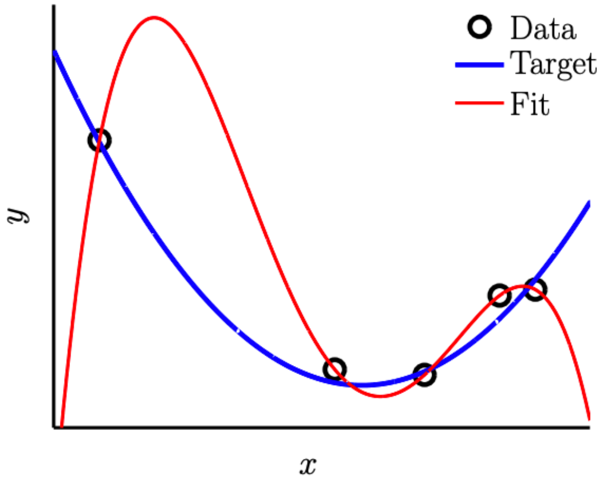
Recap

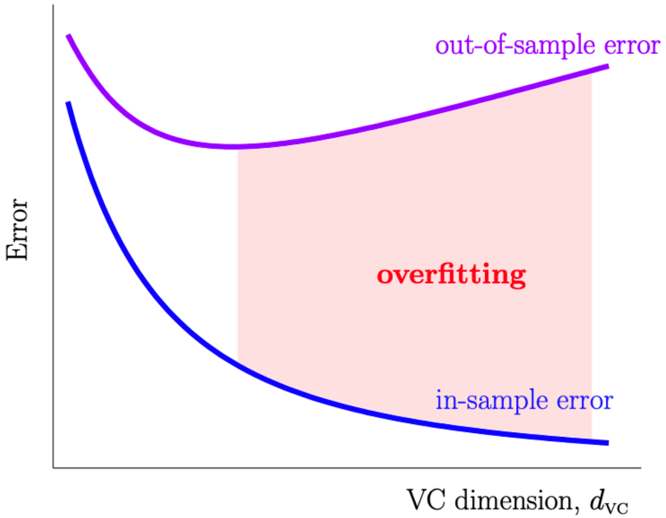
Related Note for HW2: Test Set

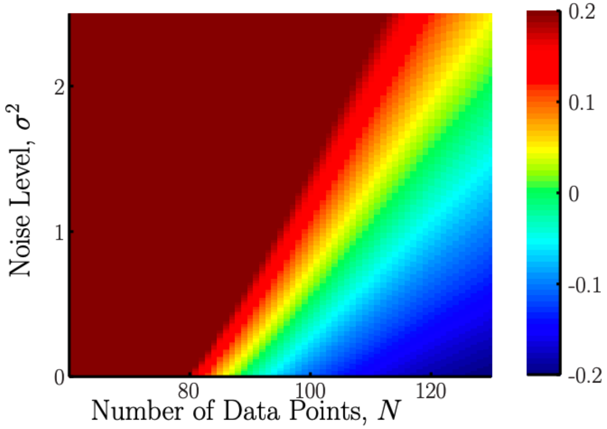
- Will discuss in detail in Validation.

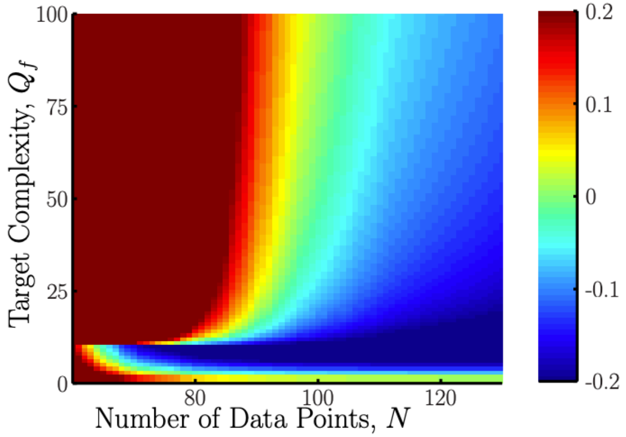
- When we are given a dataset, we often split them into training set and test set (like we did in HW2).
  - First learn a hypothesis using data in the training set.
  - Estimate E_{out} using the performance on data in the test set.
    - If you use the test set only once, the test error is an unbiased estimator for E_{out}.
    - Let E_{test} be the error on test set. We usually treat E_{test} as E_{out}.

Overfitting









Number of data points ↑	Overfitting ↓
Noise ↑	Overfitting ↑
Target complexity ↑	Overfitting ↑

Overfitting and Its Cures

- Overfitting
  - Fitting the data more than is warranted
  - Fitting the noise instead of the pattern of the data
  - Decreasing E_{in} but getting larger E_{out}
  - When H is too strong, but N is not large enough
- Regularization
  - Intuition: Constraining H to make overfitting less likely to happen
- Validation
  - Intuition: Reserve data to estimate E_{out}

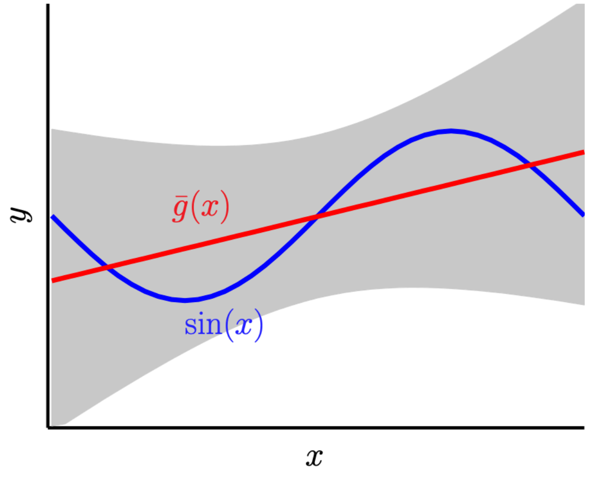
Brief Lecture Notes Today

The notes are not intended to be comprehensive. They should be accompanied by lectures and/or textbook.

Let me know if you spot errors.

Regularization (Constraining H)

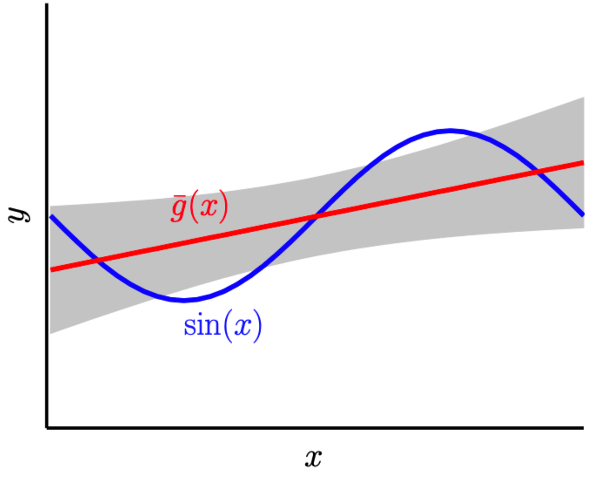
- Informal example:
  - Regression; $f = \sin(\pi x)$; $H = \{h(x) = ax + b\}$; $N = 2$



no regularization

bias $= 0.21$
var $= 1.69$

Regularization:
Constrain the hypothesis set
Avoid large a and b



regularization

bias $= 0.23$
var $= 0.33$

How to do this in a principled way?

Hard Constraints

- We have seen hard constraints already

$$H_2 = \{h(x) = w_0 + w_1 x + w_2 x^2\}$$

$$H_{10} = \{h(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{10} x^{10}\}$$

- H_2 can be written as constrained H_{10}

$$H_2 = \{h \in H_{10} \text{ and } \underbrace{w_3 = w_4 = \cdots = w_{10} = 0}_{\text{Constraints}}\}$$

Soft-Order Constraints

- Instead of setting the weights to 0

$$H(C) = \left\{ h \in H_Q \text{ and } \sum_{q=0}^{Q} w_q^2 \le C \right\}$$

$$= \left\{ h \in H_Q \text{ and } \vec{w}^T \vec{w} \le C \right\}$$

- Observations
  - When $C \to \infty$, $H(C) = H_Q$
  - When $C_1 \le C_2$, $H(C_1) \subseteq H(C_2)$ and therefore $d_{vc}\big(H(C_1)\big) \le d_{vc}(H(C_2))$
  - A smoother way to tune the complexity of hypothesis set

Soft-Order Constraints

$$H(C) = \{h \in H_Q \text{ and } \vec{w}^T\vec{w} \leq C\}$$

- Two main questions
  - How do we choose C
    - Model selection: The same question as selecting H
    - The focus of the next lecture
  - How do we perform learning, i.e., find a $g \in H(C)$ such that $g \approx f$
    - Solve the following constrained optimization problem

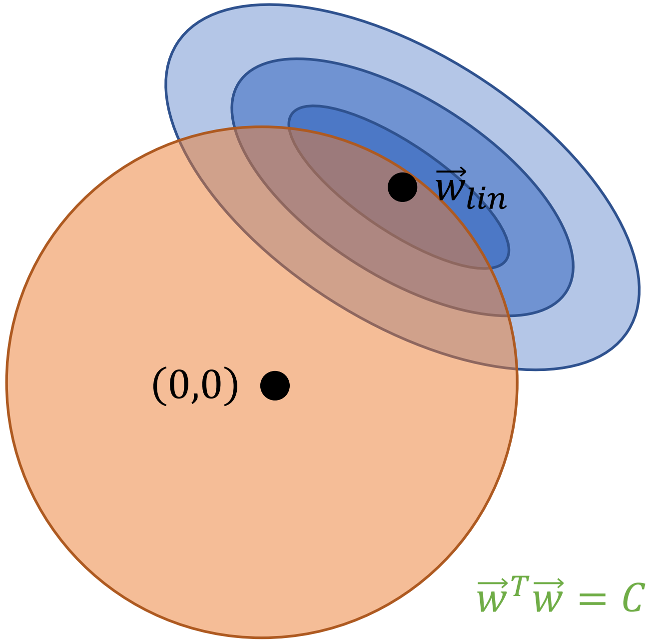
minimize $E_{in}(\vec{w})$

subject to $\vec{w}^T\vec{w} \leq C$

minimize $E_{in}(\vec{w})$ subject to $\vec{w}^T\vec{w} \leq C$

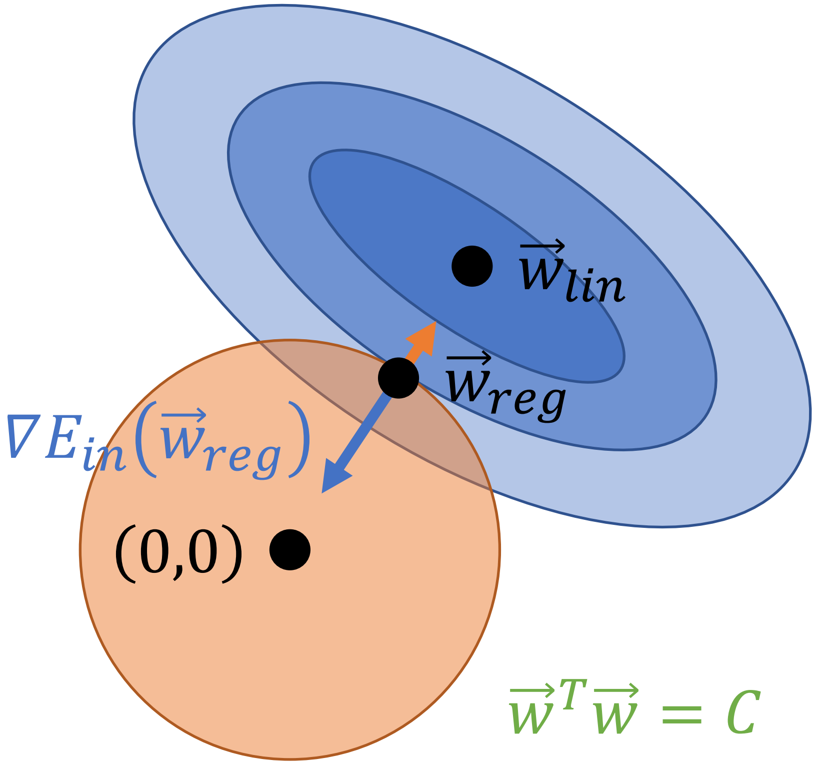
- Notations
  - $\vec{w}_{lin}$: the solution for $\min E_{in}(\vec{w})$
  - $\vec{w}_{reg}$:the solution for $\min E_{in}(\vec{w})$ subject to $\vec{w}^T\vec{w} \leq C$

- When C is large enough, i.e., $\vec{w}_{lin}^T\vec{w}_{lin} \leq C$
  - $\vec{w}_{reg} = \vec{w}_{lin}$



minimize $E_{in}(\vec{w})$ subject to $\vec{w}^T\vec{w} \leq C$

- When C is not large enough
  - Using graphical arguments
    - $\vec{w}_{reg} \propto -\nabla_{\vec{w}} E_{in}(\vec{w}_{reg})$
  - That is, we can find some constant λ_C such that
    - $\nabla_{\vec{w}} E_{in}(\vec{w}_{reg}) = -\frac{2\lambda_C}{N}\vec{w}_{reg}$
  - Therefore,
    - $\nabla_{\vec{w}}\left(E_{in}(\vec{w}_{reg}) + \frac{\lambda_C}{N}\vec{w}_{reg}^T\vec{w}_{reg}\right) = 0$
  - This implies, $\vec{w}_{reg}$ is the solution for
    - minimize $E_{in}(\vec{w}) + \frac{\lambda_C}{N}\vec{w}^T\vec{w}$



Constrained to Unconstrained Optimization

- Original constrained optimization problem

$$\text{minimize } E_{in}(\vec{w})$$

$$\text{subject to } \vec{w}^T\vec{w} \leq C$$

- Equivalent unconstrained optimization problem

$$\text{minimize } E_{in}(\vec{w}) + \frac{\lambda_C}{N}\vec{w}^T\vec{w}$$

Augmented Error

$\vec{w}^T\vec{w}$: weight decay

- Define augmented error
  - $E_{aug}(\vec{w}) = E_{in}(\vec{w}) + \frac{\lambda_C}{N}\vec{w}^T\vec{w}$
  - Algorithm: Find $\vec{w}^* = argmin\ E_{aug}(\vec{w})$

- A bit more discussion
  - When $C \to \infty$, $\lambda_C = 0$
  - Smaller C (stronger constraints)
    - => larger λ_C
    - => smaller H
    - => stronger regularization
  - Use λ_C to tune the level of regularization

Side notes:
You will see people/us interchangeably use λ_C and $\frac{\lambda_C}{N}$ to be the constant, depending on whether the dependency on N is emphasized.

Why $\vec{w}^T\vec{w}$ is Called Weight Decay

- Run gradient descent on $E_{aug}(\vec{w}) = E_{in}(\vec{w}) + \frac{\lambda_C}{N}\vec{w}^T\vec{w}$

- The update rule would be
  - $\vec{w}(t+1) \leftarrow \vec{w}(t) - \eta\nabla_{\vec{w}}E_{aug}(\vec{w}(t))$
    $\Rightarrow \vec{w}(t+1) \leftarrow (1 - 2\eta\lambda_C)\vec{w}(t) - \eta\nabla_{\vec{w}}E_{in}(\vec{w}(t))$

  - We are decaying the weights first, then do the update

General Form of Regularization

$$E_{aug}(h, \lambda, \Omega) = E_{in}(\vec{w}) + \frac{\lambda}{N}\Omega(h)$$

- Key parameters
  - Ω: Regularizer
  - λ: Amount of regularization

- Does the form look familiar: VC Theory
  - $E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{vc}\frac{\ln N}{N}}\right)$

- If we pick the right Ω, E_{aug} can be a better proxy for E_{out}

How to Pick the Right Ω

- No definite answer, but generally
  - We like to pick Ω that leads to "smoother" hypothesis
    - Overfitting is due to noise
    - Informally, noise is generally "high frequency"
  - We prefer Ω that makes the optimization easier (e.g., convex/differentiable)
    - Similar to pick the error measure
  - We might have some other objective in mind
    - Ex: L-1 regularizer leads to weight vectors with more 0s

- What if we pick the wrong Ω (Think about weight growth)
  - We might still fix it by picking the right λ – using validation in the next lecture