CSE 417T
Introduction to Machine Learning

Lecture 12

Instructor: Chien-Ju (CJ) Ho

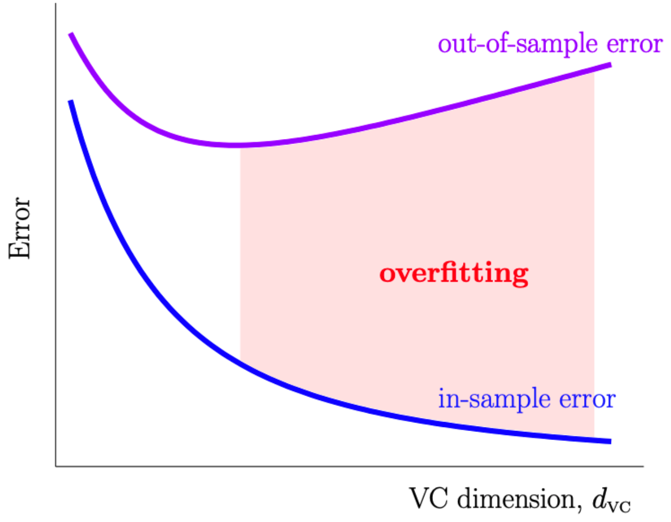
Logistics: Reminders

- HW 2: Feb 24, 2020 (Monday)
  - Reserve time for submissions
  - No extensions will be given for last-minute technical reasons
- Exam 1: 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
  - Check seat assignments on Piazza the night before the exam
  - More details in the Slides on Feb 18

Recap

Overfitting and Its Cures

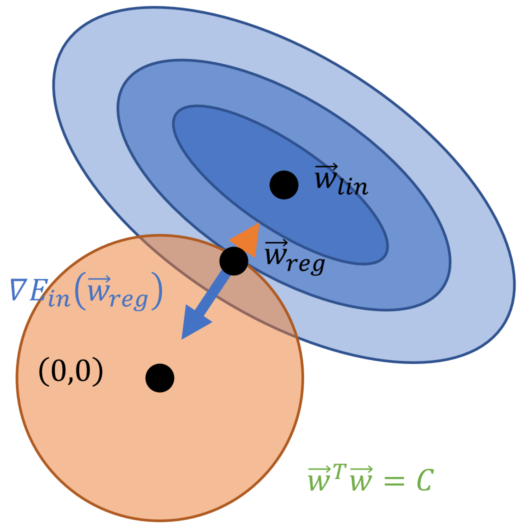
- Overfitting
  - Fitting the data more than is warranted
  - Fitting the noise instead of the pattern of the data
  - Decreasing E_{in} but getting larger E_{out}
  - When H is too strong, but N is not large enough



- Regularization
  - Intuition: Constraining H to make overfitting less likely to happen

- Validation
  - Intuition: Reserve data to estimate E_{out}

Regularization (Constraining H)



- Weight decay

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

- Algorithm: Find $g \in H(C)$ such that $g \approx f$

Constrained optimization

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

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

equivalent

Unconstrained optimization

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

Augmented error

Augmented Error

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

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

- 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.

General Form of Regularization

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

- Key components
  - Ω: Regularizer
  - λ: Amount of regularization
- Does the form look familiar? Recall in the VC Theory (treating δ as a constant)
  - $E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{vc}\frac{\ln N}{N}}\right)$
- If we pick the right Ω, E_{aug} can be a good proxy for E_{out}

How to Pick the Right Ω

- Intuition: pick Ω that leads to "smoother" hypothesis
  - Overfitting is due to noise
  - Informally, noise is generally "high frequency"

- Computation: prefer Ω that makes the optimization easier (e.g., convex/differentiable)
  - Similar to picking the error measure

- We might have some other objective in mind
  - Ex: L-1 regularizer leads to weight vectors with more 0s

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.

More Discussion on Regularization

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

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

- The update rule would be

$$\vec{w}(t+1) \leftarrow \vec{w}(t) - \eta \nabla_{\vec{\mathrm{w}}} E_{aug}(\vec{w}(t))$$

$$\Rightarrow \vec{w}(t+1) \leftarrow (1 - 2\eta\lambda_C)\vec{w}(t) - \eta \nabla_{\vec{\mathrm{w}}} E_{in}(\vec{w}(t))$$

We are decaying the weights first, then do the update

Linear Regression with Weight Decay

- $E_{aug}(\vec{w}) = E_{in}(w) + \frac{\lambda_C}{N}\vec{w}^T\vec{w} = \frac{1}{N}\|X\vec{w} - \vec{y}\|^2 + \frac{\lambda_C}{N}\vec{w}^T\vec{w}$

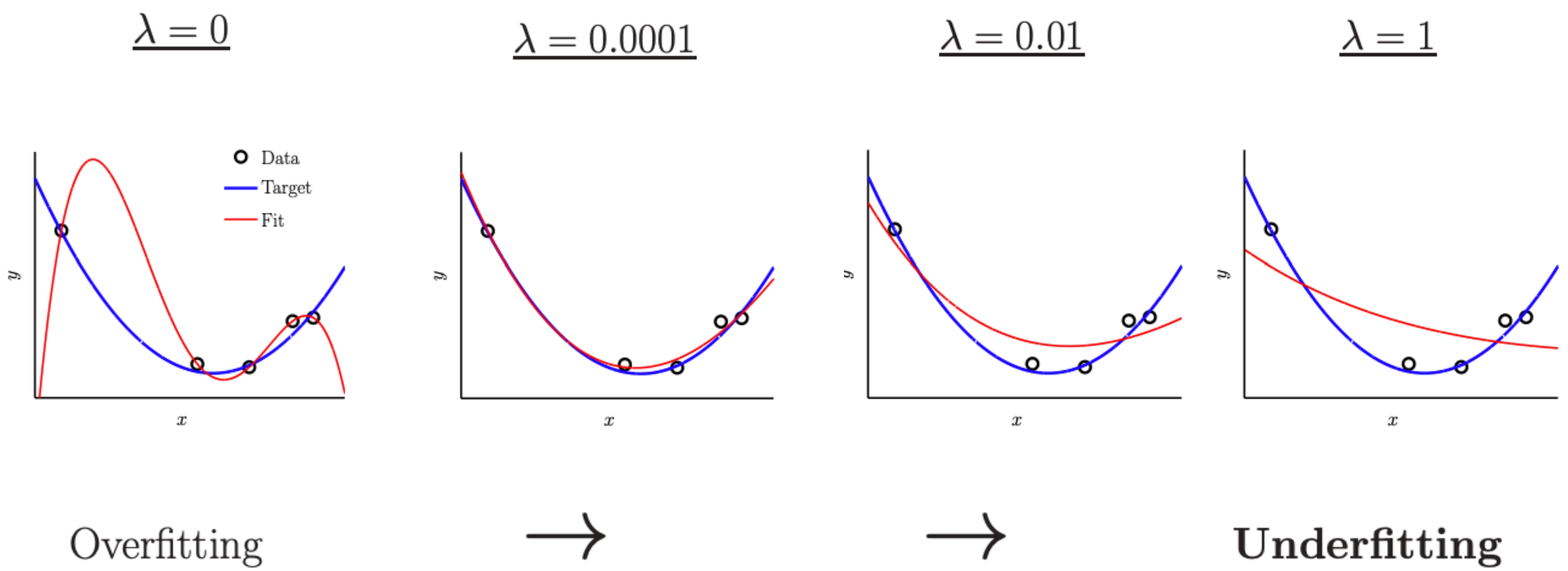
- Solve $\nabla_{\vec{w}} E_{aug}(\vec{w})|_{\vec{w}=\vec{w}_{reg}} = 0$, we get
  - $\frac{2}{N}\left(X^T X\vec{w}_{reg} - X^T\vec{y} + \lambda_C\vec{w}_{reg}\right) = 0$
  - $(X^T X + \lambda_C I)\vec{w}_{reg} = X^T\vec{y}$
  - $\vec{w}_{reg} = (X^T X + \lambda_C I)^{-1}X^T\vec{y}$

Notation: I is an identity matrix: only the elements in the diagonals are 1, and all others are 0.

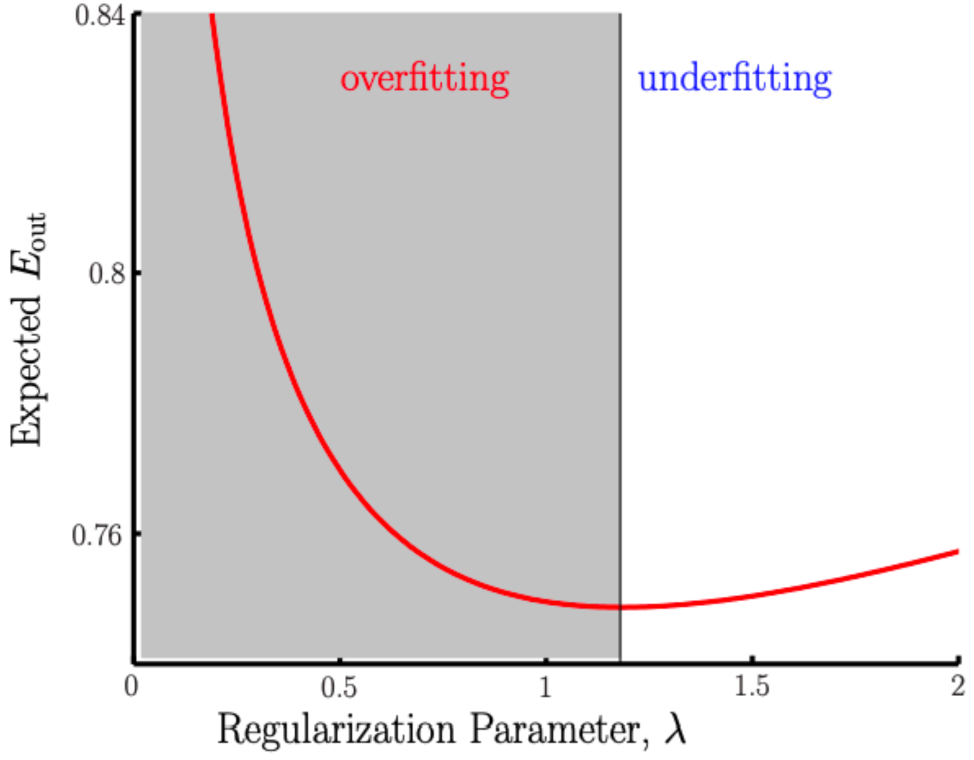
This is called “Ridge Regression” in statistics.

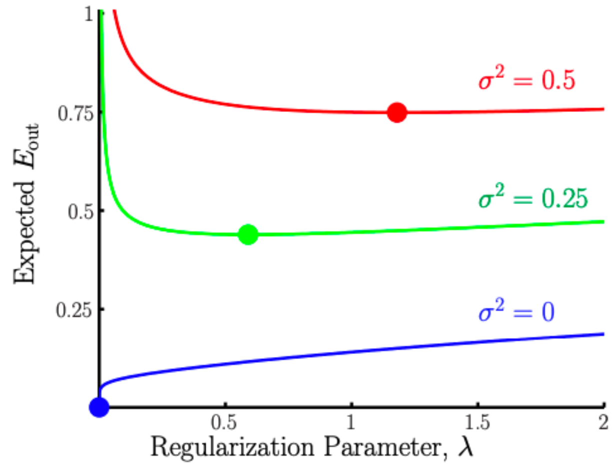
Effect of Regularization (Different λ)

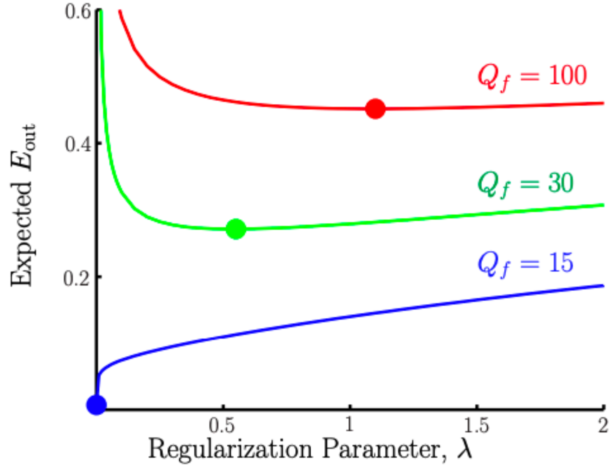
- Minimizing $E_{in}(\vec{w}) + \frac{\lambda}{N}\vec{w}^T\vec{w}$ with different λ



Overfitting and Underfitting



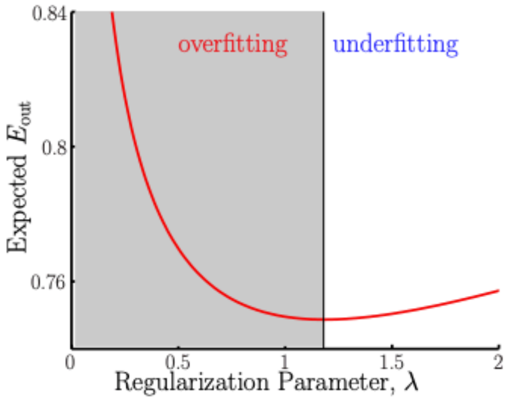




Need to pick the right λ:
Using validation: Focus of this lecture

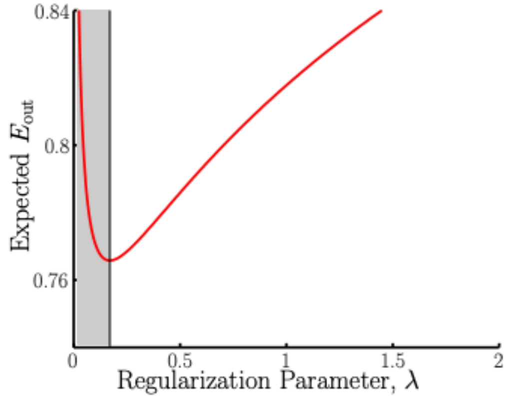
Variations on Weight Decay (Different Ω)

Uniform Weight Decay



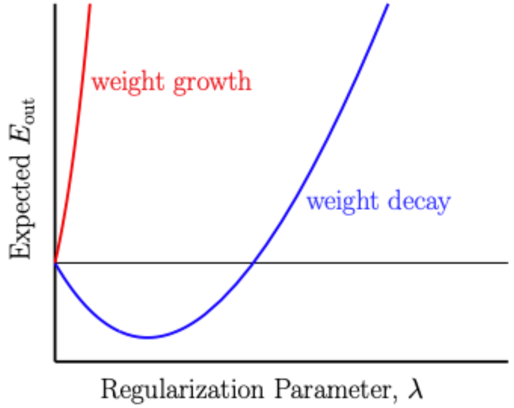
$$\sum_{q=0}^{Q} w_q^2$$

Low Order Fit



$$\sum_{q=0}^{Q} q w_q^2$$

Weight Growth!



$$\sum_{q=0}^{Q} \frac{1}{w_q^2}$$

How to Pick the Right Ω

- As discussed earlier
  - Intuition: pick Ω that leads to “smoother” hypothesis
    - Overfitting is due to noise
    - Informally, noise is generally ”high frequency”
  - Computation: prefer Ω that makes the optimization easier (e.g., convex/differentiable)
    - Similar to picking 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 (weight growth)
  - We might still fix it by picking the right λ – using validation

Summarizing Regularization

- Regularization is everywhere in machine learning

- Two main ways of thinking about regularization
  - Constraining H to make overfitting less likely to happen
    - Will discuss more regularization methods in the 2nd half of the semester
    - Pruning for decision trees, early stopping / dropout for neural networks, etc

  - Define augmented error E_{aug} to better approximate E_{out}
    - $E_{aug}(h, \lambda, \Omega) = E_{in}(\vec{w}) + \frac{\lambda}{N}\Omega(h)$

- We show the equivalence of the two for weight decay
  - The conceptual equivalence is general with Lagrangian relaxation (will cover later in the semester)

Validation

Prevent Overfitting

$$E_{out}(g) = E_{in}(g) + \text{overfit penalty}$$

- Regularization
  - Choose a regularizer Ω to approximate the penalty
- Validation
  - Directly estimate E_{out} (The real goal of learning is to minimize E_{out})

Test Set (Want to Estimate E_{out})

- Out-of-sample error $E_{out}(g) = \mathbb{E}_{\vec{x}}[e(g(\vec{x}), y)]$
  - Key: $\vec{x}$ need to be out of sample

- Test set $D_{test} = \{(\vec{x}_1, y_1), \dots, (\vec{x}_K, y_K)\}$
  - Reserve K data points used to estimate E_{out}
  - None of the data points in test set can be involved in training

- Using the data in test set to estimate E_{out}
  - Since all data points in D_{test} are out of sample

Test Set

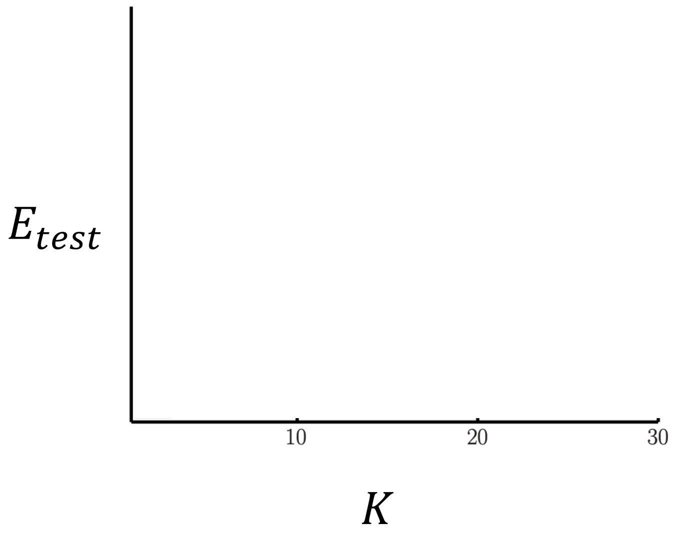
- Test set $D_{test} = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_K, y_K)\}$
- For a g learned using only training set

- Let $E_{test}(g) = \frac{1}{K}\sum_{k=1}^{K} e(g(\vec{x}_k), y_k)$
  - $E_{test}(g)$ is an unbiased estimate of $E_{out}(g)$
    - $\mathbb{E}[E_{test}(g)] = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[e(g(\vec{x}_k), y_k)] = E_{out}(g)$
  - Single hypothesis Hoeffding bound applies
    - $E_{out}(g) \leq E_{test}(g) + O\left(\sqrt{\frac{1}{K}}\right)$

Where are Test Set From?

- Given a data set D of N points
  - $D = D_{train} \cup D_{test}$
  - Reserving K points for test set means we only have $N - K$ points for training

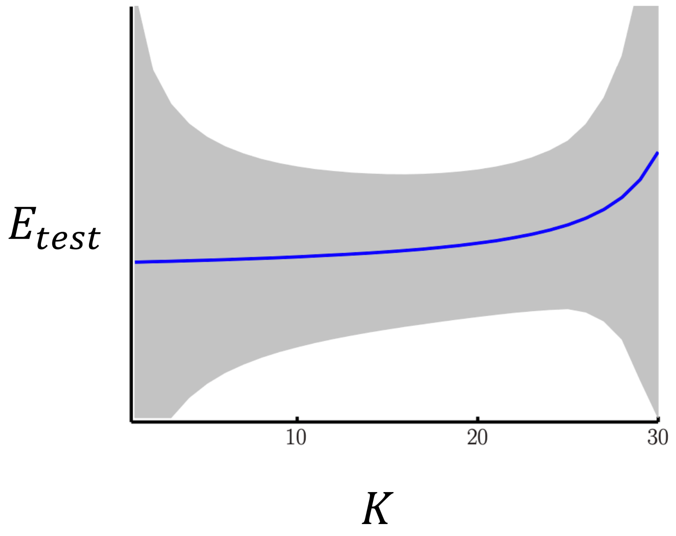
- Effect of the choice of K



Where are Test Set From?

- Given a data set D of N points
  - $D = D_{train} \cup D_{test}$
  - Reserving K points for test set means we only have $N - K$ points for training

- Effect of the choice of K



Rule of Thumb: $K^* = \frac{N}{5}$

Utilizing the Whole D

- Process:
  - $D = D_{train} \cup D_{test}$ where $|D_{test}| = K, |D_{train}| = N - K$
  - Learn some hypothesis g^- using only D_{train}
  - Estimate $E_{out}(g^-)$ using D_{test}
  - Let g be the hypothesis that would be learned using D

- Generally (informally, not theoretically proven)
  - Training on more data leads to better learned hypothesis
  - $E_{out}(g) \leq E_{out}(g^-)$

Validation: Beyond Test Set

- What if we want to estimate E_{out} multiple times?

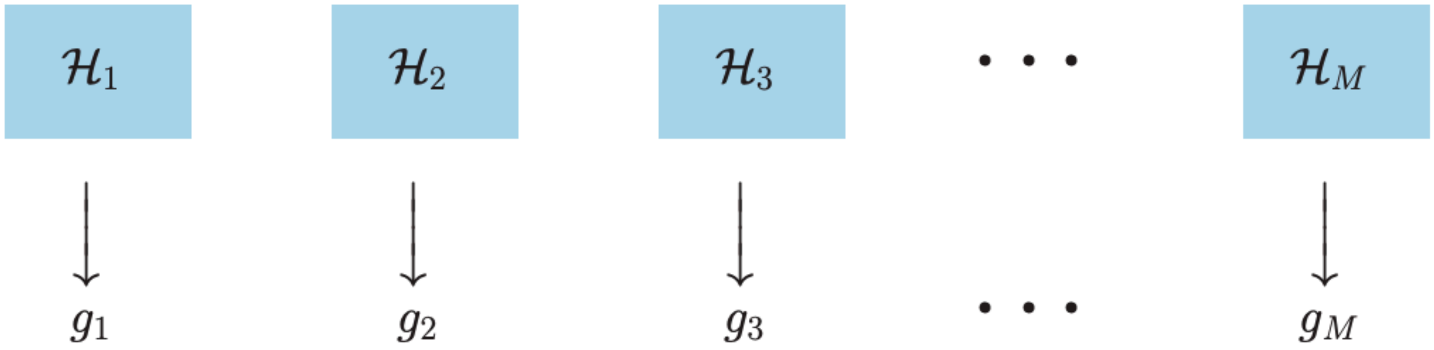
- Model selection:
  - Should I use linear models or decision trees?
  - Should I set the regularization parameter λ to 0.1, 0.01, or 0.001?
    - A model with different λ can be considered as different model

- Validation set
  - $D = D_{train} \cup D_{val}$

  - Key difference: We need to account for the multiple usage of D_{val}

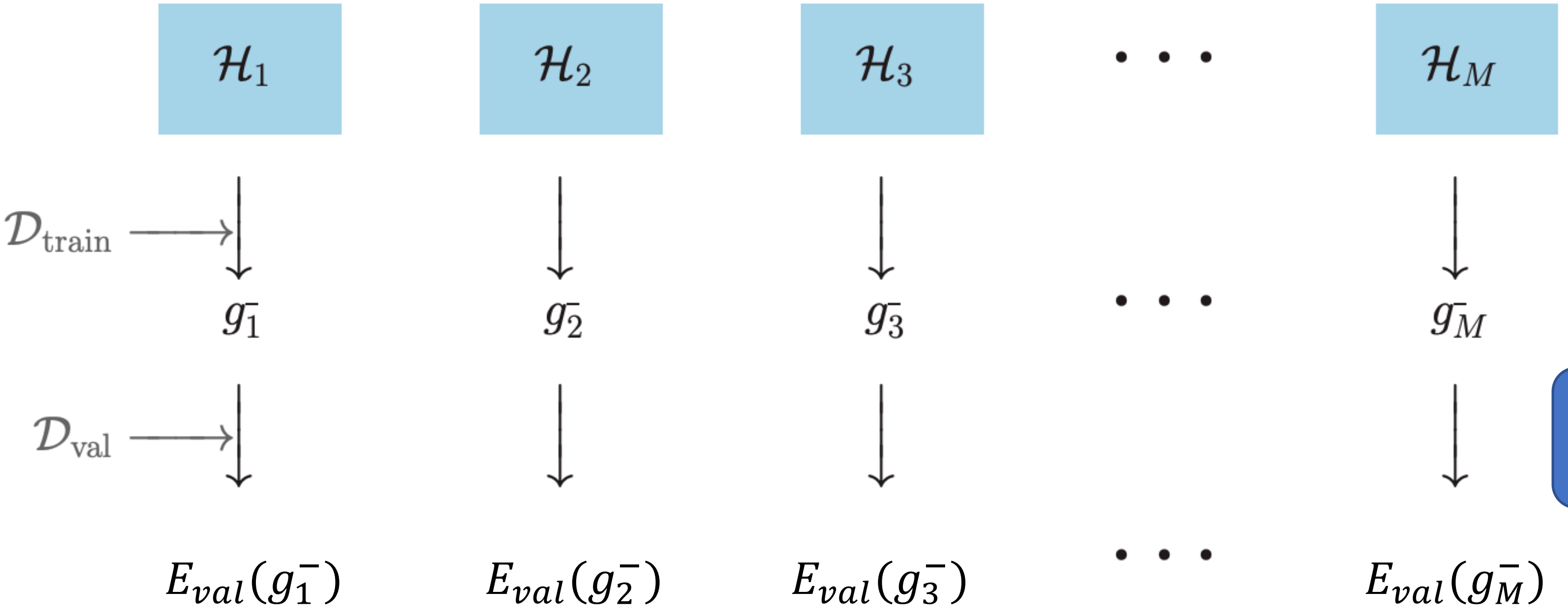
Model Selection

- Which model should we choose?



Model Selection using Validation

- Which model should we choose?



Key: D_{val} is used M times

Choose H_{m^*} such that $E_{val}(g_{m^*}^-) \leq E_{val}(g_m^-)$ for all m

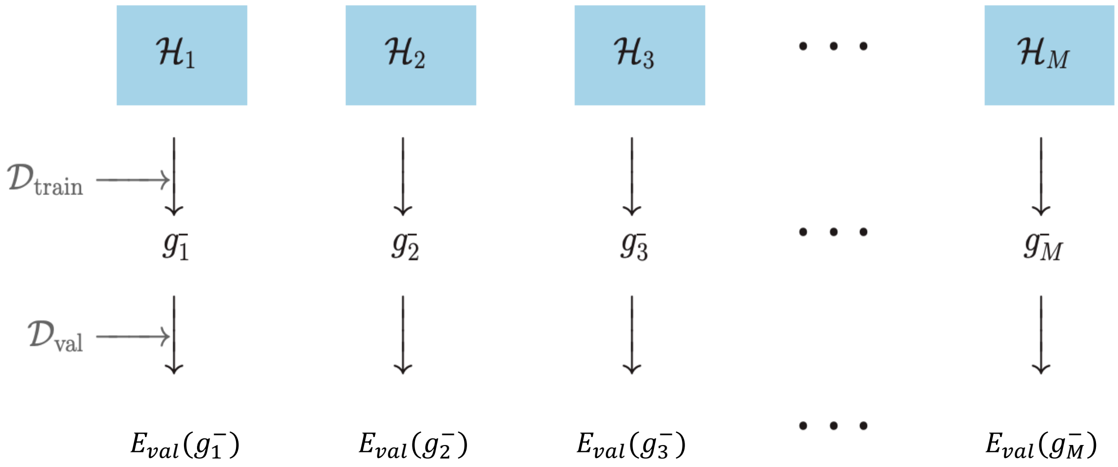
Question…

- Which of the following is true?

  (a) $\mathbb{E}[E_{val}(g_{m^*}^{-})] = E_{out}(g_{m^*}^{-})$

  (b) $\mathbb{E}[E_{val}(g_{m^*}^{-})] \leq E_{out}(g_{m^*}^{-})$

  (c) $\mathbb{E}[E_{val}(g_{m^*}^{-})] \geq E_{out}(g_{m^*}^{-})$



Choose H_{m^*} such that $E_{val}(g_{m^*}^{-}) \leq E_{val}(g_m^{-})$ for all m

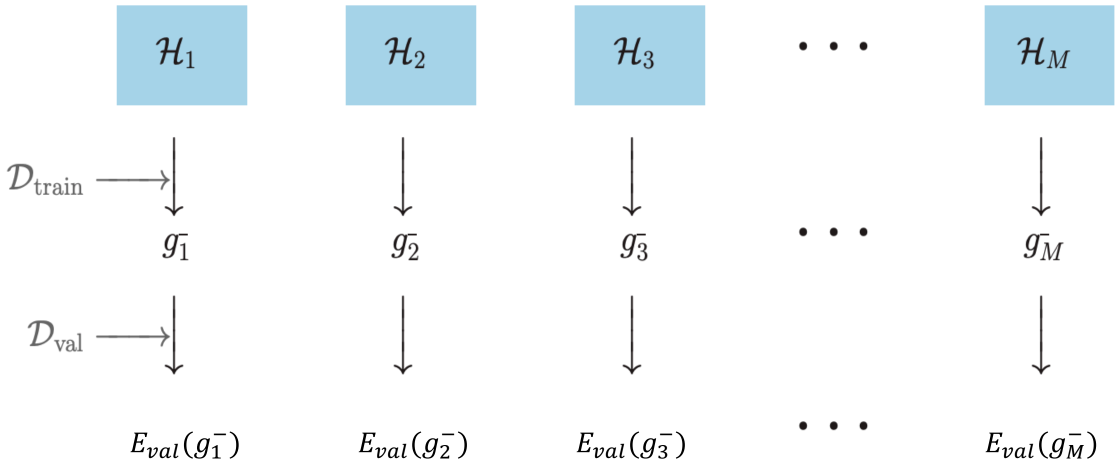
Question…

- Which of the following is true?

(a) $\mathbb{E}[E_{val}(g_{m^*}^-)] = E_{out}(g_{m^*}^-)$

(b) $\mathbb{E}[E_{val}(g_{m^*}^-)] \leq E_{out}(g_{m^*}^-)$

(c) $\mathbb{E}[E_{val}(g_{m^*}^-)] \geq E_{out}(g_{m^*}^-)$



Choose H_{m^*} such that $E_{val}(g_{m^*}^-) \leq E_{val}(g_m^-)$for all m

Equivalent to use D_{val} to choose from $H = \{g_1^-, \ldots, g_M^-\}$

$$E_{out}(g_{m^*}^-) \leq E_{val}(g_{m^*}^-) + O\left(\sqrt{\frac{\ln M}{K}}\right)$$ => Hoeffding Bound for Multiple Hypothesis

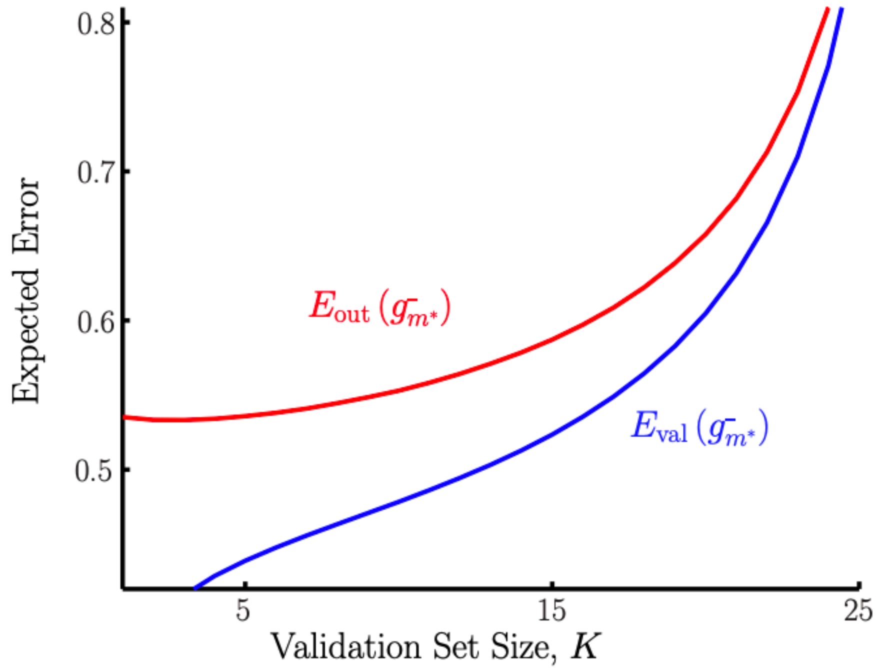
Question…

- Which of the following is true?

(a) $\mathbb{E}[E_{val}(g_{m^*}^-)] = E_{out}(g_{m^*}^-)$

(b) $\mathbb{E}[E_{val}(g_{m^*}^-)] \leq E_{out}(g_{m^*}^-)$

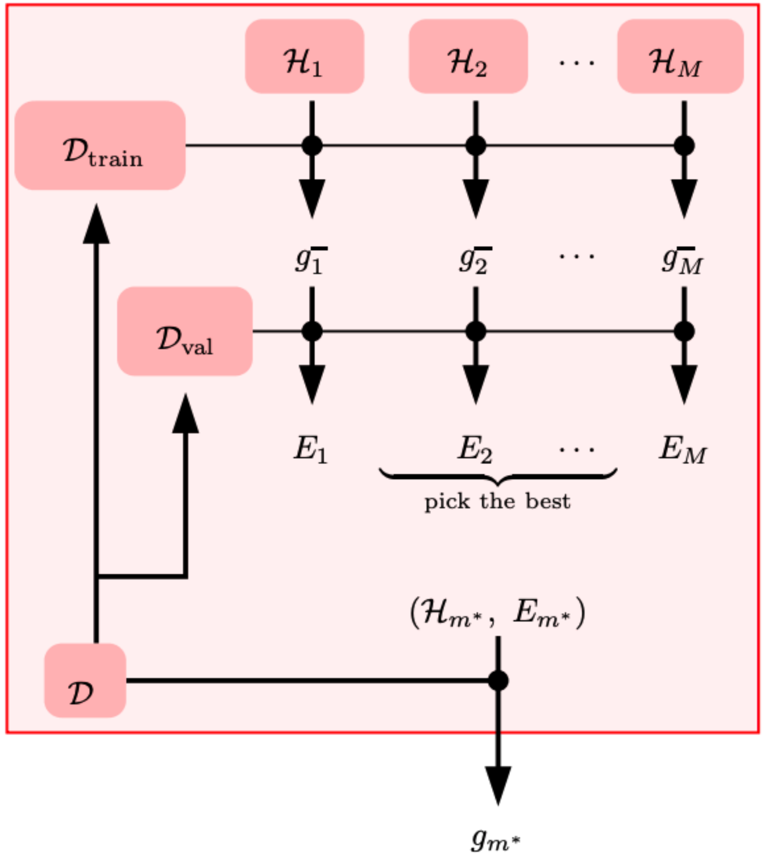
(c) $\mathbb{E}[E_{val}(g_{m^*}^-)] \geq E_{out}(g_{m^*}^-)$

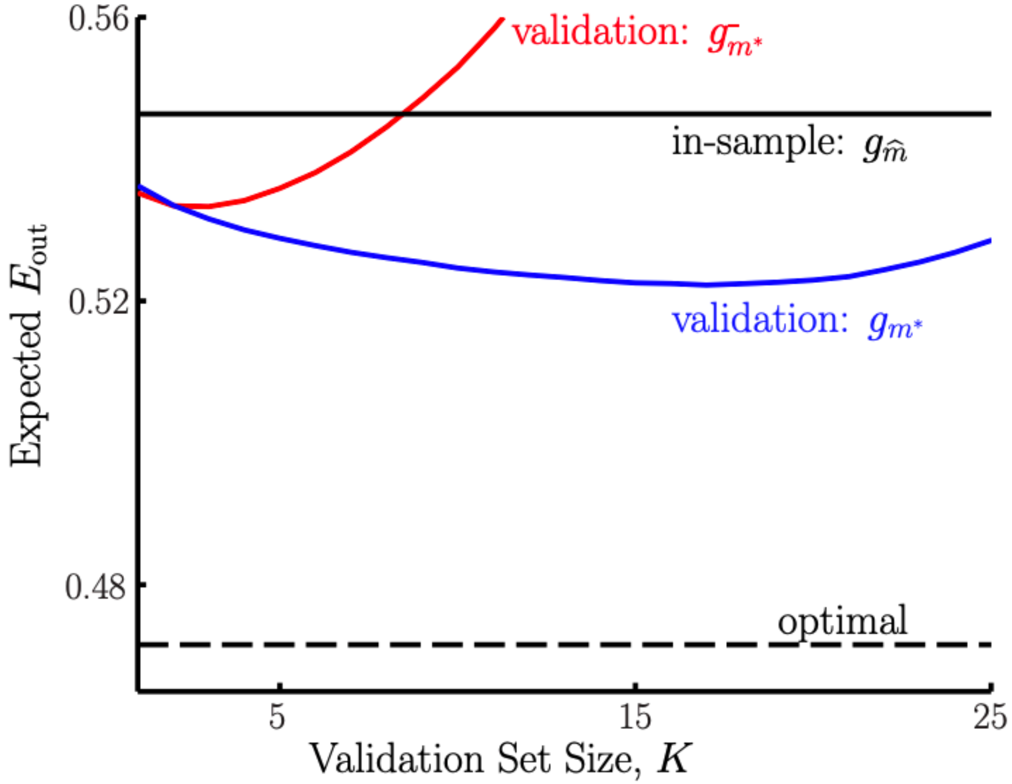


Equivalent to use D_{val} to choose from $H = \{g_1^-, \ldots, g_M^-\}$

$$E_{out}(g_{m^*}^-) \leq E_{val}(g_{m^*}^-) + O\left(\sqrt{\frac{\ln M}{N}}\right)$$ => Hoeffding Bound for Multiple Hypothesis

Utilizing the Whole D





$g_{\hat{m}}$: the hypothesis minimizes in-sample error over $\{H_1, \ldots, H_M\}$

	Outlook	Relationship to E_{out}
E_{in}		
E_{val}		
E_{test}		

	Outlook	Relationship to E_{out}
E_{in}	Incredibly optimistic	
E_{val}	Slightly optimistic	
E_{test}	Unbiased	

	Outlook	Relationship to E_{out}
E_{in}	Incredibly optimistic	VC-bound
E_{val}	Slightly optimistic	Hoeffding's bound (multiple hypotheses)
E_{test}	Unbiased	Hoeffding's bound (single hypothesis)

Note that the outlook comparisons are "in expectation"
If you only get one "draw" of $D_{train}, D_{val}, D_{test}$, you cannot say anything "for certain"

Remember that ML results are under the condition "with high probability"

The Dilemma When Choosing K

- The main ideas behind validation

Want large K

(E_{val} estimates E_{out} well)

$$E_{out}(g) \approx E_{out}(g^-) \approx E_{val}(g^-)$$

Want small K

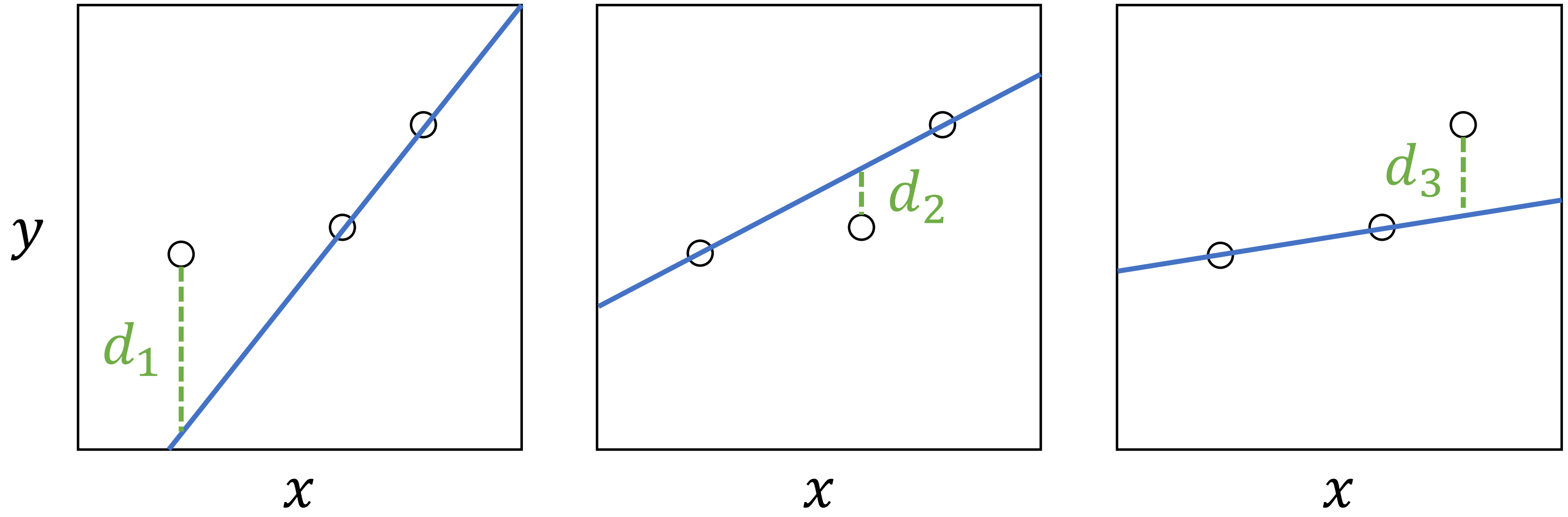
(didn’t sacrifice too much training data)

Leave-One-Out Cross Validation (LOOCV)

Getting the best of the both world

Intuition: Setting $K = 1$ but do it many times…

Illustrative Example



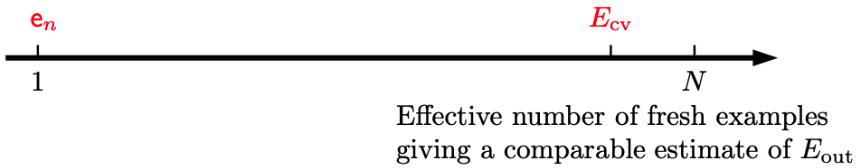
$$E_{cv} = \frac{1}{3}(d_1^2 + d_2^2 + d_3^2)$$

Properties of LOOCV

- LOOCV is unbiased (If not used for model selection)
  - E_{CV} is an unbiased estimator of $\bar{E}_{out}(N-1)$

(expected E_{out} when learning on $N-1$ points)

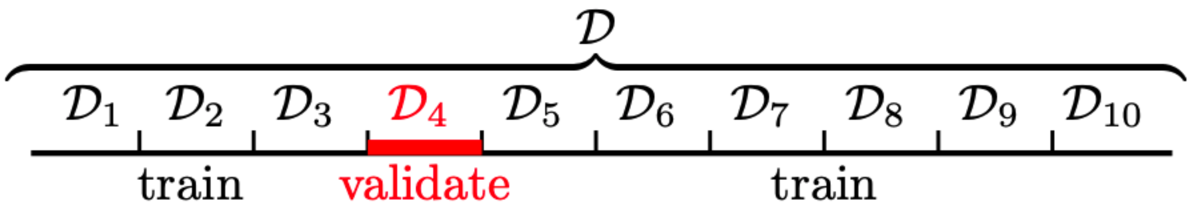
- The “effective number” of examples in E_{CV} estimation is high for LOOCV



- However, LOOCV is computationally expensive
  - Need to train N models, each on $N-1$ points

V-Fold Cross Validation

- Split D into V equally sized data sets: $D_1, D_2, \ldots, D_V$
  - Let g_i^- be the hypothesis learned using all data sets except D_i
  - Let $e_i = E_{val}(g_i^-)$ where the validation uses data set D_i
- The V-fold cross validation error is $\frac{1}{V}\sum_{i=1}^{V} e_i$



- Practical rule of thumb: $V = 10$