CSE 417T
Introduction to Machine Learning

Lecture 11
Instructor: Chien-Ju (CJ) Ho

Logistics

- Homework 2: due on **Oct 7** (Friday)

- Exam 1: **October 27 (Thursday)**
  - Topics: LFD Chapters 1 to 5
  - Timed exam (75 min) during lecture time
  - Location TBD
  - Closed-book exam with 2 letter-size cheat sheets allowed (4 pages in total)
    - No format limitations (it can be typed, written, or a combination)

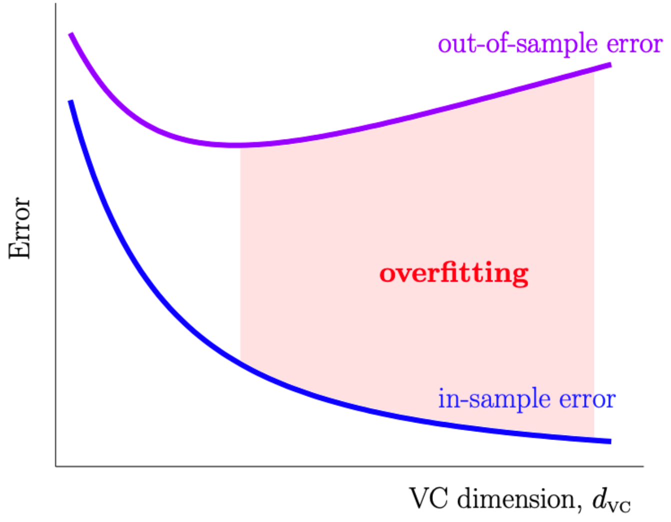
- Homework 3 will be posted later this week

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: Constrain H to make overfitting less likely to happen

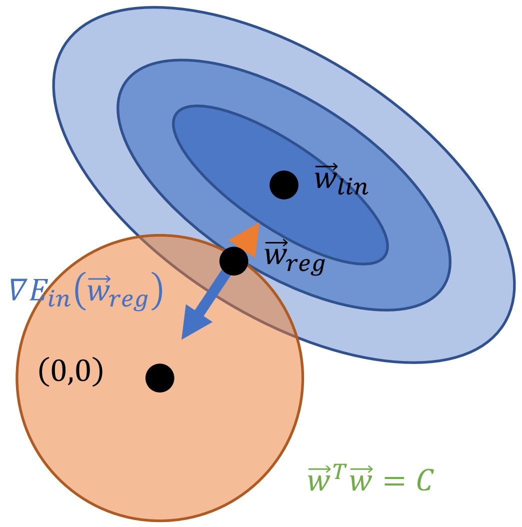
- Validation
  - Intuition: Reserve data to estimate E_{out}

Regularization (Constrain 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

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

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

- Does the form look familiar? Recall in the VC Theory (treating δ as a constant)
  - $E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{vc}\frac{\ln N}{N}}\right)$

- What the impacts of picking Ω and λ?

Summary of Regularization

- Regularization is everywhere in machine learning

- Two main ways of thinking about regularization
  - Constrain 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}
    - $E_{aug}(h, \lambda, \Omega) = E_{in}(h) + \frac{\lambda}{N}\Omega(h)$

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

Today’s Lecture

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

Let me know if you spot errors.

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 goal of learning is to minimize E_{out})

Review of Test Set (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 (i.e., not in training)

- Test set $D_{test} = \{(\vec{x}_1, y_1), \dots, (\vec{x}_K, y_K)\}$
  - Reserve K data points
  - 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

Short Discussion on HW2

- In HW2, you are asked to perform “normalization” on the training/test datasets. How should you do it?

1. Calculate the mean/variance of the combined data.
   Normalize them using the overall mean/variance.
2. Calculate the means/variances of the training and test datasets separately.
   Normalize them using their respective mean/variance.
3. Calculate the mean/variance of the training dataset.
   Normalize both datasets using the training mean/variance.

Short Discussion on HW2

- In HW2, you are asked to perform "normalization" on the training/test datasets. How should you do it?

   1. Calculate the mean/variance of the combined data.
      Normalize them using the overall mean/variance.
   2. Calculate the means/variances of the training and test datasets separately.
      Normalize them using their respective mean/variance.
   3. Calculate the mean/variance of the training dataset.
      Normalize both datasets using the training mean/variance.

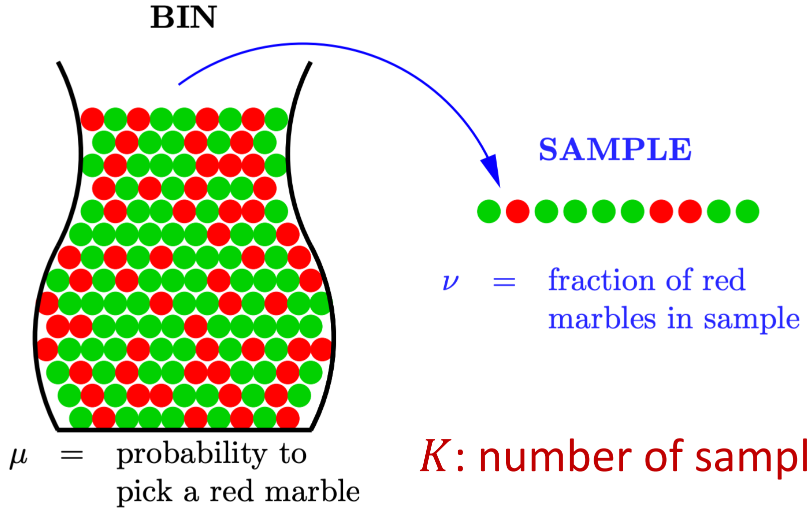
Two important properties we want to preserve

1. Training and test data are drawn from the same distribution.
2. Test data is never used in training.

Test Set

- Test set $D_{test} = \{(\vec{x}_1, y_1), \dots, (\vec{x}_K, y_K)\}$
- For a g learned using only the training dataset
  - g is a "fixed" hypothesis for D_{test}

- 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
    - $E_{out}(g) \leq E_{test}(g) + O\left(\sqrt{\frac{1}{K}}\right)$

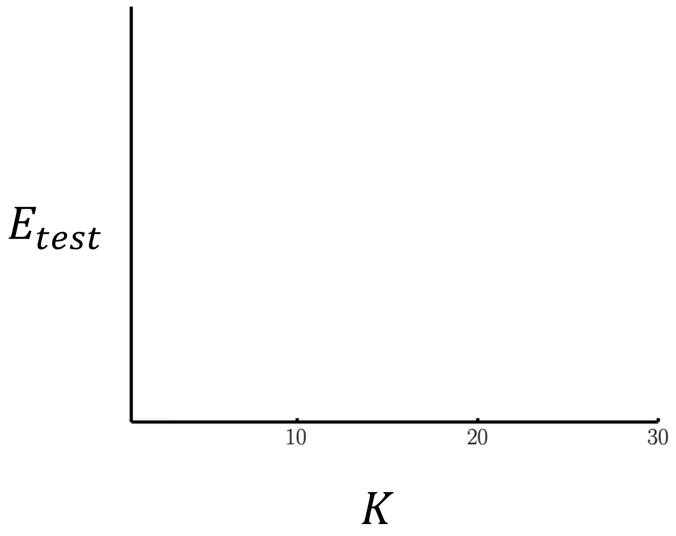


K: number of samples

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

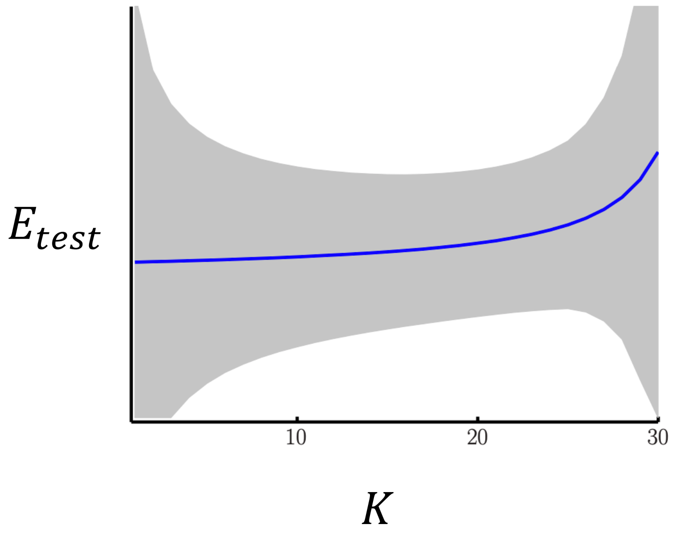
- Effect of the choice of K



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}
- Can we do better than g^- ?
  - Yes! Learn g using the entire D; return g and $E_{test}(g^-)$
- Generally (Informal, 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?

Validation: Beyond Test Set

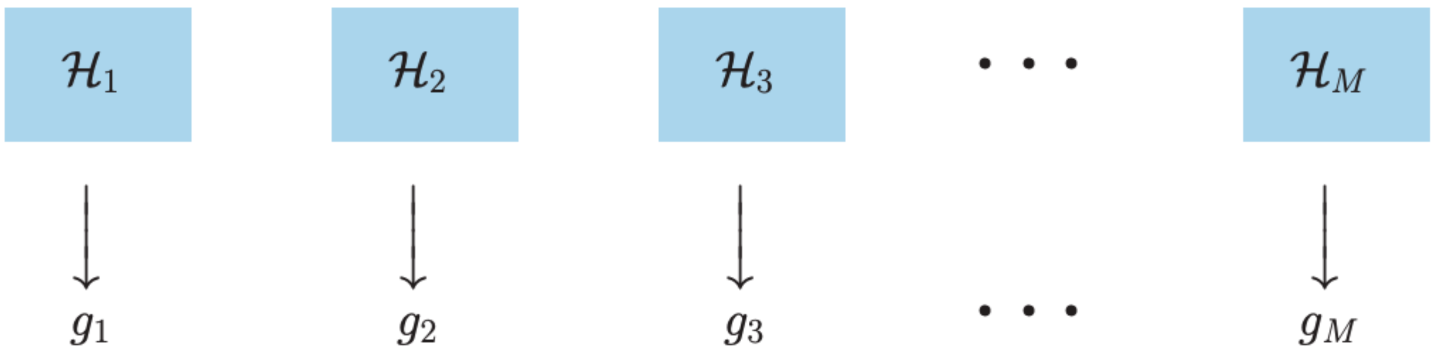
- Model selection:
  - Should I use linear models or decision trees?
  - Should I set the regularization parameter λ to 0.1, 0.01, or 0.001?
    - A model with different λ can be considered as different model

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

  - Key difference to the test set
    - D_{val} could be used multiple times for model selection
    - We need to account for the multiple usages of D_{val}

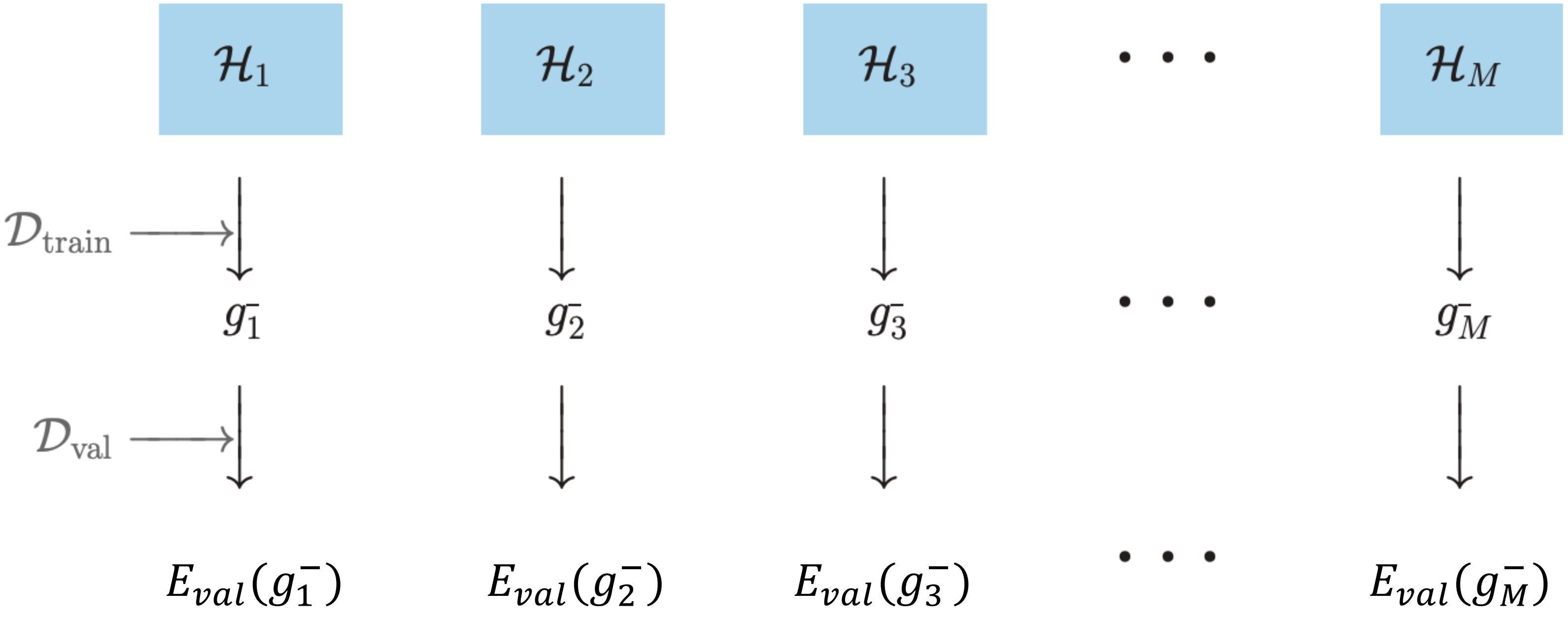
Model Selection

- Which model should we choose?



Model Selection using Validation

- Which model should we choose?



Key: D_{val} is used to choose from M hypothesis

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

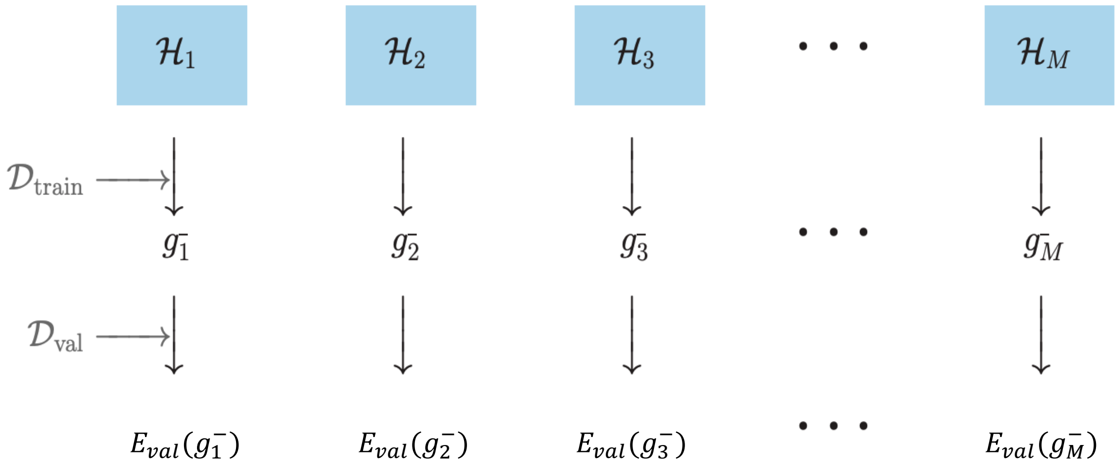
Question…

- 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

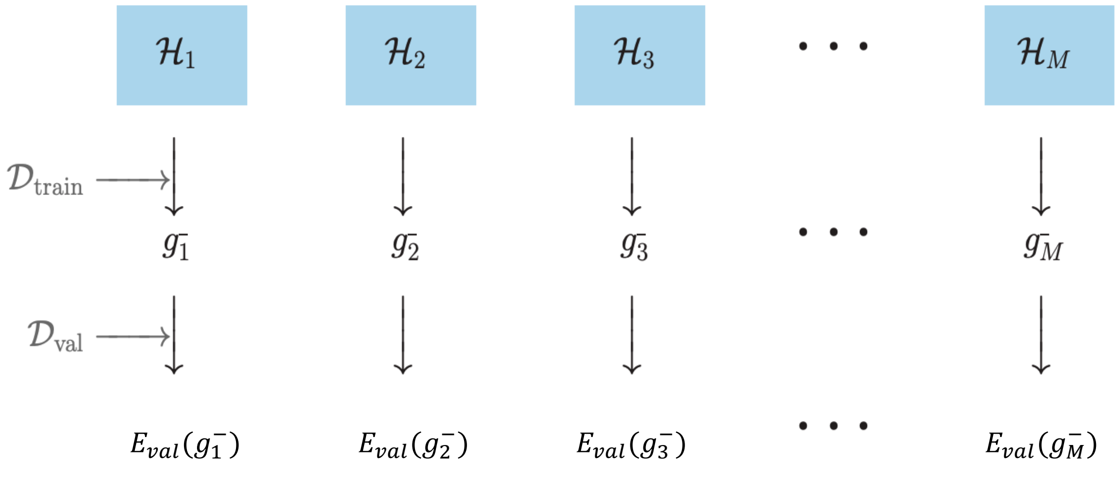
Question…

- 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 adjusted for Multiple Hypothesis

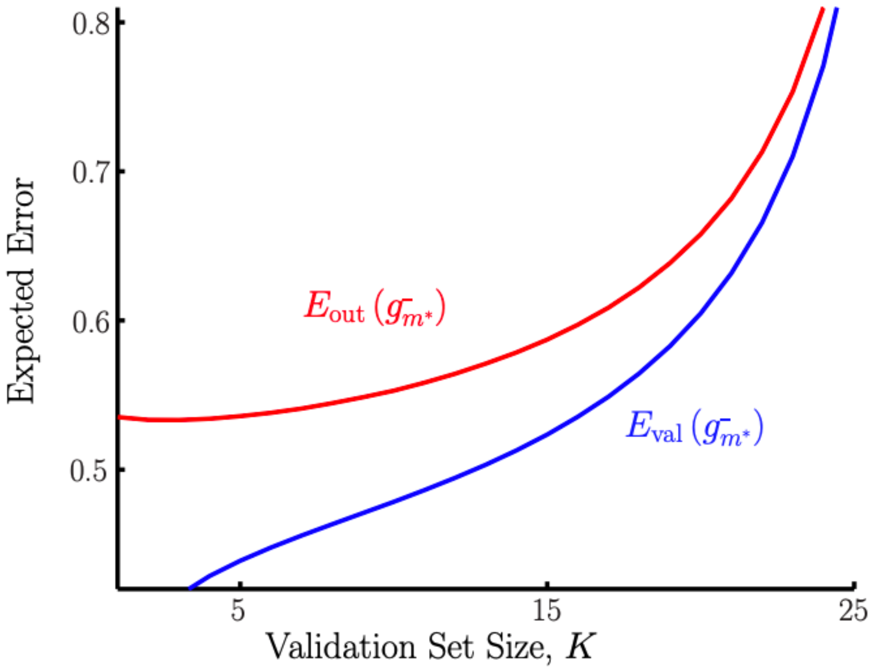
Question…

- Which of the following is true?

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

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

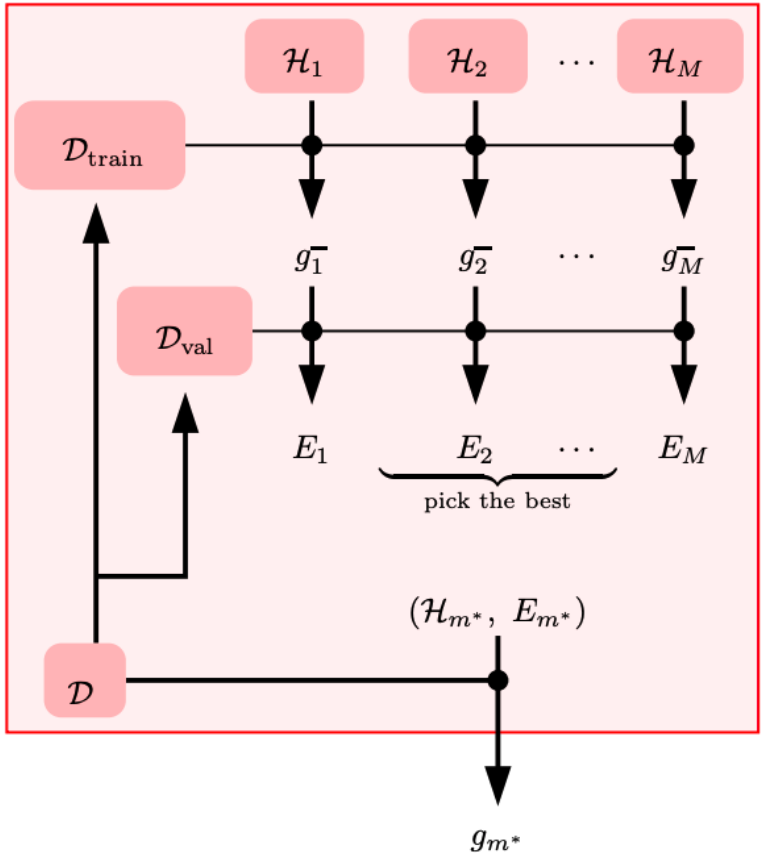
(c) $\mathbb{E}[E_{val}(g_{m^*}^-)] \geq E_{out}(g_{m^*}^-)$

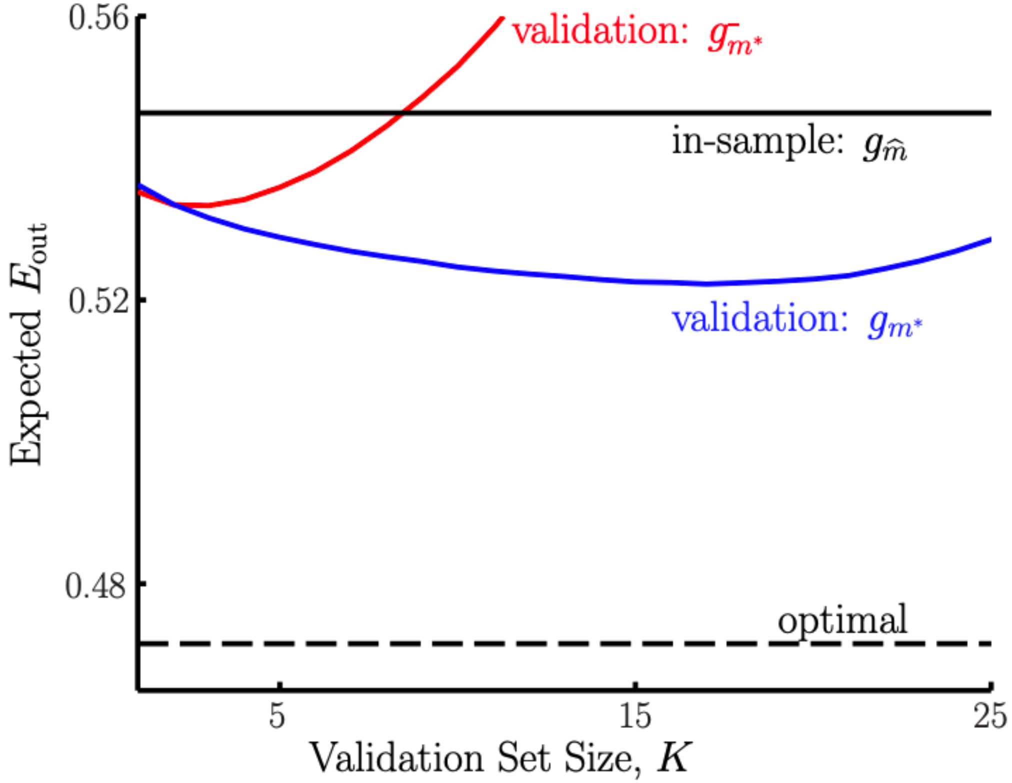


Equivalent to use D_{val} to choose from $H = \{g_1^-, \dots, g_M^-\}$

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

Utilizing the Whole D





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

	Outlook	Relationship to E_{out}
E_{in}		
E_{val} (when used for model selection)		
E_{test}		

When a validation set is not used for model selection (i.e., used only once), it is essentially a test set

	Outlook	Relationship to E_{out}
E_{in}	Incredibly optimistic	
E_{val} (when used for model selection)	Slightly optimistic	
E_{test}	Unbiased	

	Outlook	Relationship to E_{out}
E_{in}	Incredibly optimistic	VC-bound
E_{val} (when used for model selection)	Slightly optimistic	Hoeffding's bound (adjusted for 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

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

The Dilemma When Choosing K

- The main ideas behind validation

Want large K
(E_{val} estimates E_{out} well)

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

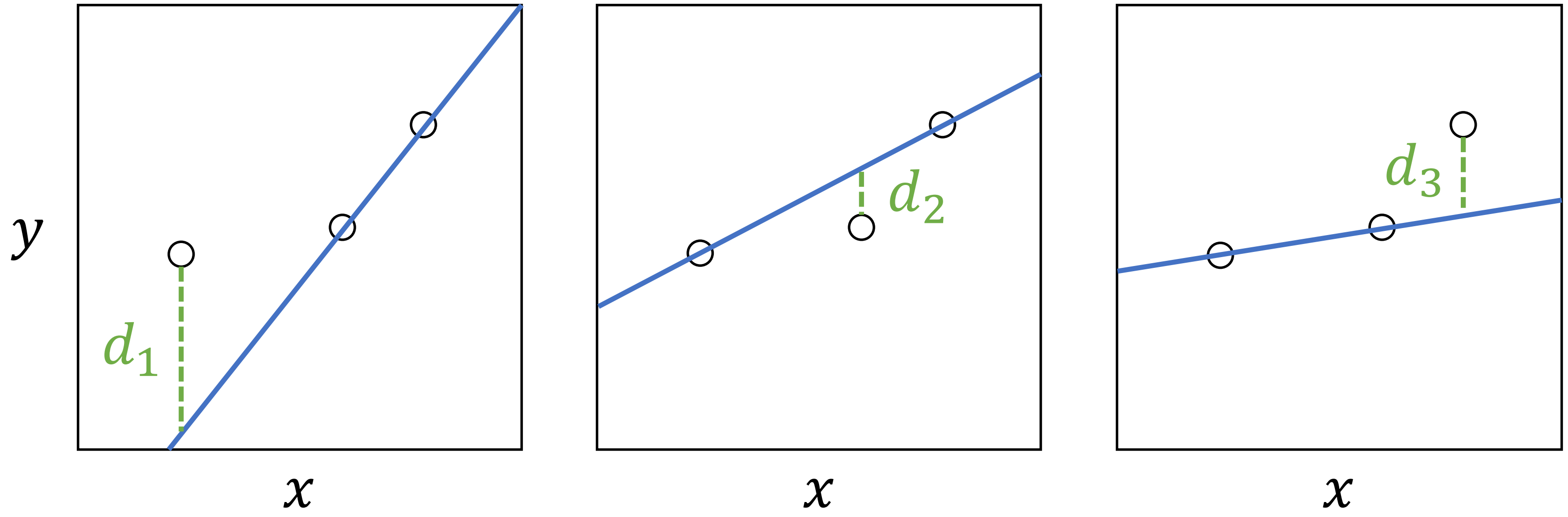
Want small K
(didn’t sacrifice too much training data)

Leave-One-Out Cross Validation (LOOCV)

Getting the best of both worlds

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

Illustrative Example



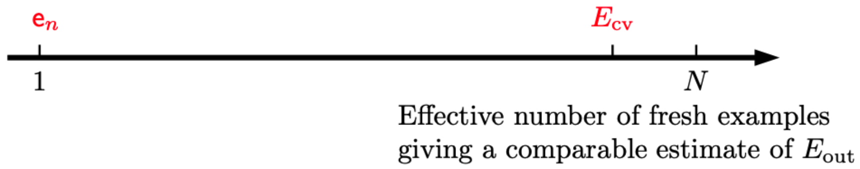
$$E_{cv} = \frac{1}{3}(d_1^2 + d_2^2 + d_3^2)$$

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)

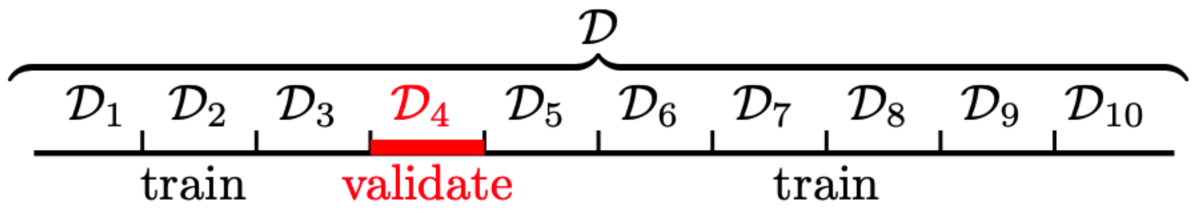
- The "effective number" of examples in E_{CV} estimation is high for LOOCV



- 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: $D_1, D_2, \dots, D_V$
  - 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: $V = 10$

VC Dimension of d-dim Perceptron

Recall the Definitions

- Shatter
  - H **shatters** $(\vec{x}_1, \ldots, \vec{x}_N)$ if $|H(\vec{x}_1, \ldots, \vec{x}_N)| = 2^N$
  - H can induce all label combinations for $(\vec{x}_1, \ldots, \vec{x}_N)$
- Break point
  - k is a **break point** for H if no data set of size k can be shattered by H
  - k is a break point for $H \leftrightarrow m_H(k) < 2^k$
- VC Dimension: $d_{vc}(H)$ or d_{vc}
  - The VC dimension of H is the largest N such that $m_H(N) = 2^N$
  - Equivalently, if k^* is the smallest break point for H, $d_{vc}(H) = k^* - 1$

VC Dimension of d-dimension Perceptron

- Claim:
  - The VC Dimension of d-dim perceptron is $d + 1$

- How to prove it?
  1. Show that the VC dimension of d-dim perceptron $\geq d + 1$
  2. Show that the VC dimension of d-dim perceptron $\leq d + 1$

- To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?
    A. There is a set of $d + 1$ points that can be shattered by H
    B. There is a set of $d + 1$ points that cannot be shattered by H
    C. Every set of $d + 1$ points can be shattered by H
    D. Every set of $d + 1$ points cannot be shattered by H

- To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?

    A. There is a set of $d + 1$ points that can be shattered by H
    B. There is a set of $d + 1$ points that cannot be shattered by H
    C. Every set of $d + 1$ points can be shattered by H
    D. Every set of $d + 1$ points cannot be shattered by H

- To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?

  A. There is a set of $d + 1$ points that can be shattered by H
  B. There is a set of $d + 1$ points that cannot be shattered by H
  C. Every set of $d + 1$ points can be shattered by H
  D. Every set of $d + 1$ points cannot be shattered by H

- To prove $d_{vc}(H) \leq d + 1$, what do we need to prove?

  A. There is a set of $d + 1$ points that can be shattered by H
  B. There is a set of $d + 2$ points that cannot be shattered by H
  C. Every set of $d + 2$ points can be shattered by H
  D. Every set of $d + 1$ points cannot be shattered by H
  E. Every set of $d + 2$ points cannot be shattered by H

- To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?

  A. There is a set of $d + 1$ points that can be shattered by H
  B. There is a set of $d + 1$ points that cannot be shattered by H
  C. Every set of $d + 1$ points can be shattered by H
  D. Every set of $d + 1$ points cannot be shattered by H

- To prove $d_{vc}(H) \leq d + 1$, what do we need to prove?

  A. There is a set of $d + 1$ points that can be shattered by H
  B. There is a set of $d + 2$ points that cannot be shattered by H
  C. Every set of $d + 2$ points can be shattered by H
  D. Every set of $d + 1$ points cannot be shattered by H
  E. Every set of $d + 2$ points cannot be shattered by H

- To prove $d_{vc}(H) \geq d+1$, what do we need to prove?
  There is a set of $d+1$ points that can be shattered by H

- To prove $d_{vc}(H) \leq d+1$, what do we need to prove?
  Every set of $d+2$ points cannot be shattered by H

- To prove $d_{vc}(H) \geq d+1$, what do we need to prove?
  There is a set of $d+1$ points that can be shattered by H

Proof Sketch:

1. Let's construct a dataset of $d+1$ points: $X = \begin{bmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_{d+1}^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 0 & \dots & 0 & 0 \end{bmatrix}$; It's easy to check that X^{-1} exist

2. For any possible dichotomy $\vec{y}$, there exists a $\vec{w}$ such that $X\vec{w} = \vec{y}$, i.e., $\vec{w} = X^{-1}\vec{y}$
3. Therefore, d-dim perceptron can shatter X

- To prove $d_{vc}(H) \leq d+1$, what do we need to prove?
  Every set of $d+2$ points cannot be shattered by H

Proof Sketch:

1. For every set of $d+2$ points (in d+1 dimensions), there exists a point that can be written as linear combinations of the others.
2. Denote the point $\vec{x}_{d+2}$, we have $\vec{x}_{d+2} = \sum_{i=1}^{d+1} a_i \vec{x}_i$
3. Consider the dichotomy $(y_1, \dots, y_{d+2}) = (\mathrm{sign}(\mathrm{a}_1), \dots, \mathrm{sign}(\mathrm{a}_{d+1}), -1)$, we can show that no linear separator can generate this dichotomy (think about why).
4. Therefore, for every set of $d+2$ points, there exist at least one dichotomy that H cannot induce.

VC "Dimension"

- Degrees of freedom for your hypothesis in H
- (effective) # of parameters that control the hypothesis

- Examples:
  - d-dim perceptron: h is represented by $(w_0, \ldots, w_d)$; $d_{vc} = d + 1$
  - Positive rays: h is represented by a threshold; $d_{vc} = 1$
  - Positive or negative rays: h is represented by a threshold and a direction; $d_{vc} = 2$
  - Positive intervals: h is represented by two thresholds; $d_{vc} = 2$
  - Positive or negative intervals: h is represented by two thresholds and a direction; $d_{vc} = 3$

- Effective # parameters: An "approximation" for VC dimension

Three Learning Principles

Occam’s Razor

Sampling Bias

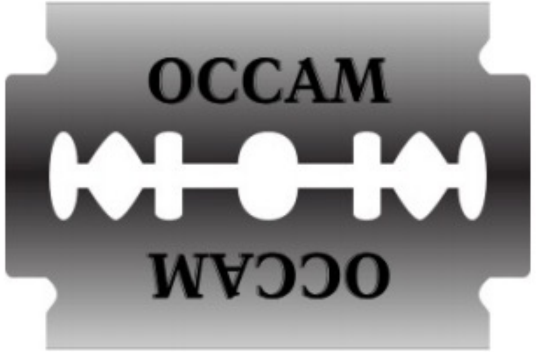
Data Snooping

Occam's Razor

"An explanation of the data should be made as simple as possible, but no simpler." -- Einstein?

"entia non sunt multiplicanda praeter necessitatem"
(entities must not be multiplied beyond necessity)
-- William of Occam

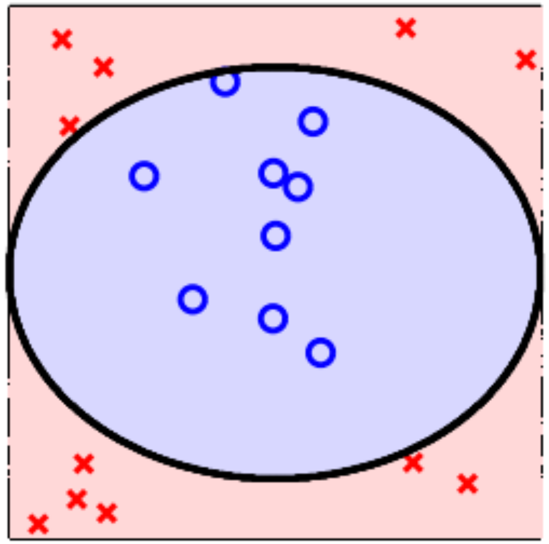
"trimming down"
unnecessary explanation

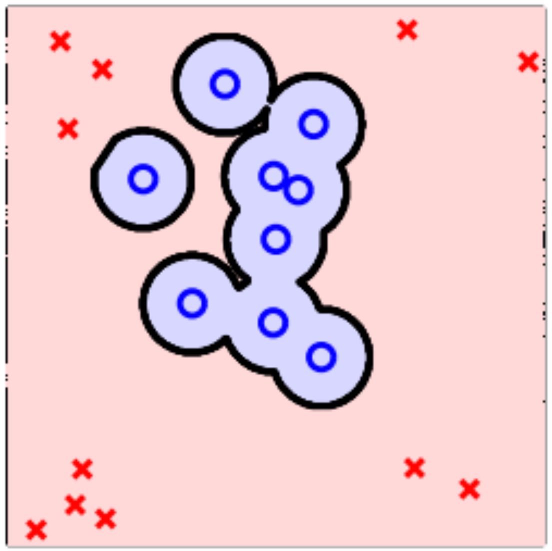


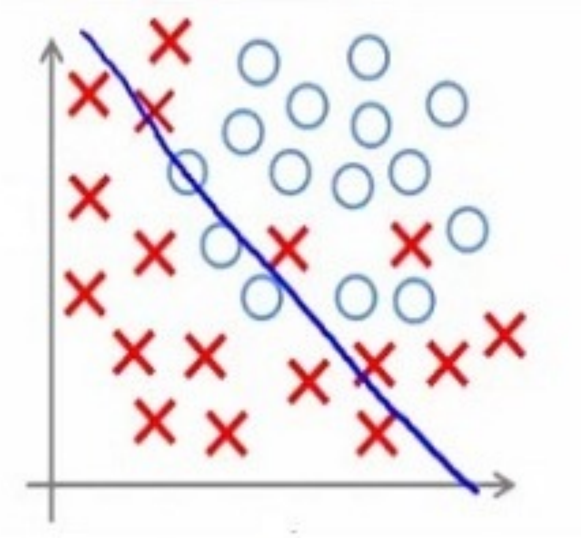
The simplest model that fits the data is also the most plausible

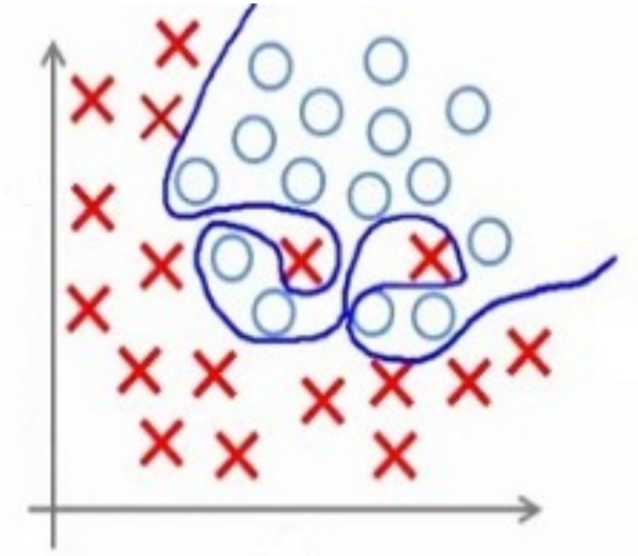
What does it mean to be simple?

Why is simple better?









Simple Model?

- For a hypothesis set H to be simple
  - # dichotomies it can generate is small
  - VC Dimension is small

- For a hypothesis h to be simple
  - lower order polynomial
  - smaller weights (think about the regularization)
  - easy to describe?
  - fewer number of parameters (fewer bits to describe)

Simple Model?

Connection:

A hypothesis set with *simple* hypotheses should be *simple*

Consider a hypothesis h can be specified by ℓ bits

$\Rightarrow H$ contains all such h

$\Rightarrow$ The size of H is 2^ℓ

Simple: small model complexity / VC dimension / size of hypothesis set

Why is Simple Better?

simple -> small VC dimension -> good generalization, less overfitting, …

Simple $\mathcal{H}$

$\Rightarrow$ small growth function $m_{\mathcal{H}}(N)$

$\Rightarrow$ if data labels are generated randomly, the probability of fitting perfectly is?

$$\frac{m_{\mathcal{H}}(N)}{2^N}$$

$\Rightarrow$ more significant when fit really happens

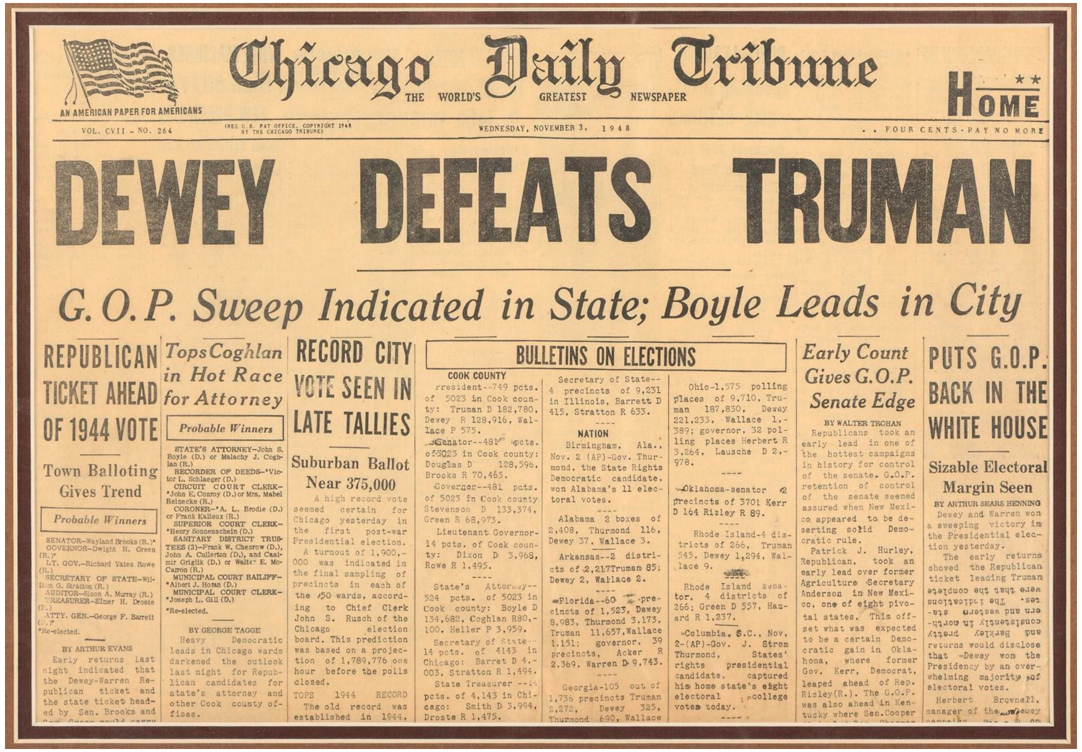
Occam’s Razor

Sampling Bias

Data Snooping

1948 US Presidential Election

- Truman vs. Dewey
- Chicago Daily Tribune decided to run a phone poll of how people <u>voted</u>



Chicago Daily Tribune

THE WORLD'S GREATEST NEWSPAPER

AN AMERICAN PAPER FOR AMERICANS

HOME

WEDNESDAY, NOVEMBER 3, 1948

DEWEY DEFEATS TRUMAN

G. O. P. Sweep Indicated in State; Boyle Leads in City

REPUBLICAN TICKET AHEAD OF 1944 VOTE

Town Balloting Gives Trend

Probable Winners

Tops Coghlan in Hot Race for Attorney

Probable Winners

RECORD CITY VOTE SEEN IN LATE TALLIES

Suburban Ballot Near 375,000

BULLETINS ON ELECTIONS

Early Count Gives G.O.P. Senate Edge

PUTS G.O.P. BACK IN THE WHITE HOUSE

Sizable Electoral Margin Seen

Truman


Chicago Daily Tribune
DEWEY DEFEATS TRUMAN
G.O.P. Sweep Indicated in State; Boyle Leads in

What happened?

One explanation: we cannot claim anything for certain.

However, there are bigger issues here…

- Phones are expensive in 1948…
- Dewey was more favored in rich populations

- Imagine you are polling from people in DC/Texas/NY to predict who will win the presidential election…

Sampling Bias

If the data is sampled in a biased way, learning will produce a similarly biased outcome.