CSE 417T
Introduction to Machine Learning

Lecture 5
Instructor: Chien-Ju (CJ) Ho

Logistics: Homework 1

- Due: **September 23 (Friday), 2022**
    - http://chienjuho.com/courses/cse417t/hw1.pdf

    - Two submission links: Report and Code (The links will be up over the weekend)
        - Report: Answer all questions, including the implementation question
            - **Grades are based on the report**
        - Code: Complete and submit **hw1.py** for Problem 2
            - The code will only be used for correctness checking (when in doubts) and plagiarism checking

    - Reserve time if you never used Gradescope.
        - Make sure to **specify the pages for each problem**. You **won't get points** otherwise

Logistics: TA Office Hours

- Tentative schedule of TA office hours

Monday	9:30am Asher Baraban	3pm Qihang Zhao	
Tuesday	10am Di Huang	1pm Andrew Ruttenberg	4pm Quinn Wai Wong
Wednesday	1pm Wenxuan Zhu	3pm William Sepesi	4:30pm Sylvia Tang
Thursday	11:30am Yuan Liu	4pm Elyse Tang	6:30pm Fankun Zen
Friday	11am Riggie Kong	3pm Nan Huang	5:30pm Weiwei Ma
Sunday	Noon Jonathan Ma	1:30pm Kenneth Li	

  - 60 minutes per session; In-person office hours are highlighted in orange
  - Please follow **Piazza** for additional information (location, zoom link, etc)

- Recommendation: Try to utilize the office hour early (way ahead of deadlines), you are likely to get more of TAs' time this way

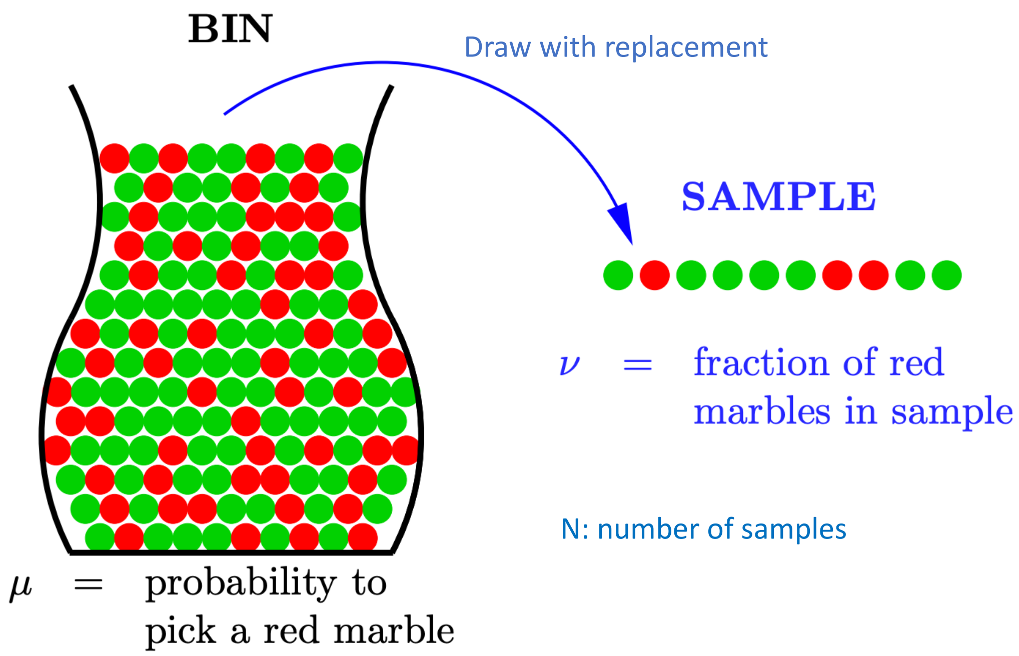
Logistics: This Thursday Lecture

- A shortened lecture this Thursday (Sep 15)
  - The lecture starts at 3:10pm
  - I need to attend a meeting that I cannot skip

- As a make-up
  - I'll host an additional one-hour office hour next Tuesday
    - 5:30pm – 6:30pm at McKelvey 2010A
  - My (tentative) regular office hour
    - 5:30pm – 6:30pm Thursday at McKelvey 2010A

Recap

Hoeffding’s Inequality



ν = fraction of red marbles in sample

N: number of samples

μ = probability to pick a red marble

$$\mathbf{Pr}[|\mu - \nu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

Define $\delta = \Pr[|\mu - \nu| > \epsilon]$

- Fix δ, ϵ decreases as N increases
- Fix ϵ, δ decreases as N increases
- Fix N, δ decreases as ϵ increases

Informal intuitions of notations
N: # sample
δ: probability of “bad” event
ϵ: error of estimation

Connection to Learning

- Given dataset $D = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\}$
  - $\boldsymbol{E_{in}(h)} \overset{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}[h(\vec{x}_n) \neq f(\vec{x}_n)]$ [In-sample error, analogy to ν]
  - $\boldsymbol{E_{out}(h)} \overset{\text{def}}{=} \Pr_{\vec{x} \sim P(\vec{x})} [h(\vec{x}) \neq f(\vec{x})]$ [Out-of-sample error, analogy to μ]

- Learning bounds
  - Fixed h (verification)

$$\Pr[|E_{out}(h) - E_{in}(h)| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

  - Finite hypothesis set: learn $g \in \{h_1, \dots, h_M\}$

$$\Pr[|E_{out}(g) - E_{in}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

Dealing with Infinite Hypothesis Set: $M \to \infty$

- Most of the practical cases involve $M \to \infty$

- Instead of # hypothesis, counting "effective" # hypothesis

- Dichotomies
  - Informally, consider a dichotomy as "data-dependent" hypothesis
  - Characterized by both H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

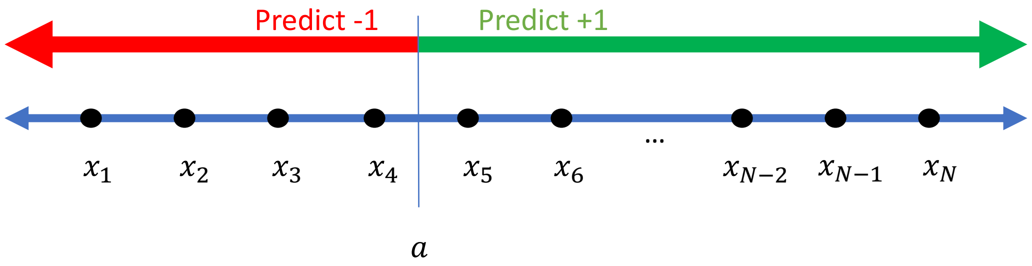
  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

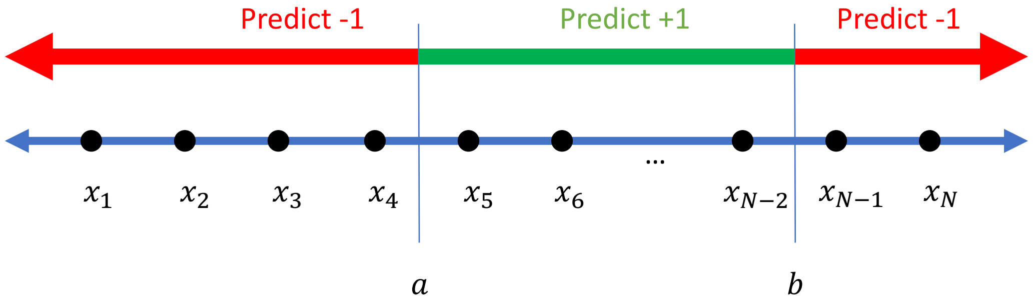
$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

Examples on Growth Functions

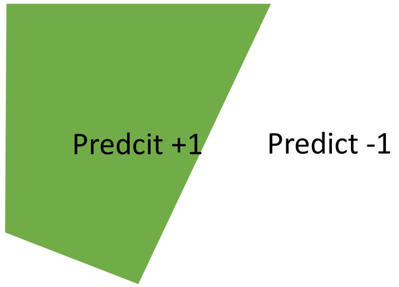
- H = Positive rays
  - $m_H(N) = N + 1$



- H = Positive intervals
  - $m_H(N) = \binom{N+1}{2} + 1 = \frac{N^2}{2} + \frac{N}{2} + 1$



- H = Convex sets
  - $m_H(N) = 2^N$



- For all H and for all N
  - $m_H(N) \leq 2^N$

Why Growth Function?

- Growth function $m_H(N)$
    - Largest number of "effective" hypothesis H can induce on N data points
    - A more precise "complexity" measure for H
    - Goal: Replace M in finite-hypothesis analysis with $m_H(N)$
        - With prob at least $1-\delta, E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{1}{2N} ln \frac{2M}{\delta}}$

- VC Generalization Bound (VC Inequality, 1971)

  With prob at least $1-\delta$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4m_H(2N)}{\delta}}$$

[We will skip the proof of the VC bound. You can find it in the textbook appendix if you are interested.]

Today’s Lecture

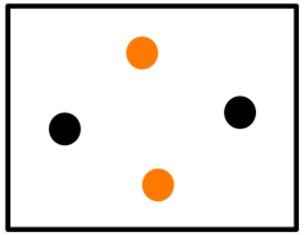
The notes are not intended to be comprehensive. They should be accompanied by lectures and/or textbook.

Let me know if you spot errors.

Bounding Growth Function

- What we know so far
  - H = Positive rays: $m_H(N) = N + 1$
  - H = Positive intervals: $m_H(N) = \binom{N+1}{2} + 1$
  - H = Convex sets: $m_H(N) = 2^N$

- What about H = 2-D Perceptron?
  - $m_H(3) = 8$
  - $m_H(4) = 14$
  - $m_H(5) = ?$



- Generally hard to write down the growth function exactly
  - Goal: “bound” the growth function using some proxy

Bounding Growth Function

- More definitions....
  - Shatter:
    - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
    - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
  - Break point
    - k is a **break point** for H if no data set of size k can be shattered by H

- A peek at the key result (take this as a fact for now)
  - If there are no break points for H, $m_H(N) = 2^N$
  - If k is a break point for H, $m_H(N)$ is polynomial in N.

    In particular, $m_H(N) = O(N^{k-1})$ →

A bit more accurately:

- $m_H(N) \leq \sum_{i=1}^{k-1} \binom{N}{i}$, or
- $m_H(N) \leq N^{k-1} + 1$

Practice

- Dichotomies
  - Informally, consider a dichotomy as a “data-dependent” hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

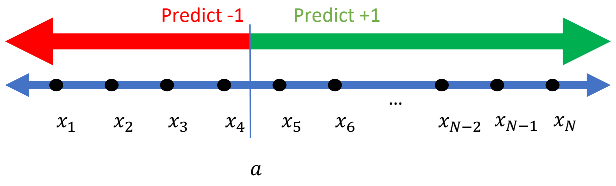
- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

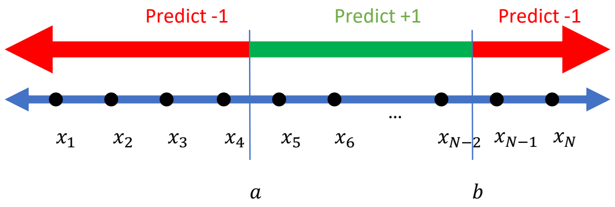
- Shatter:
  - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
  - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
- Break point
  - k is a **break point** for H if no data set of size k can be shattered by H

- What are the break points for

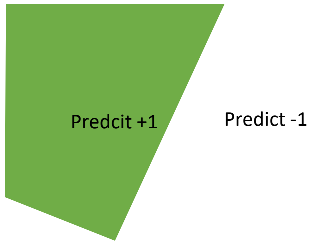
1. Positive Rays



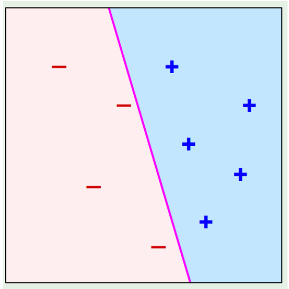
2. Positive Intervals



3. Convex Sets



4. 2-D Perceptron



Practice

- <u>Dichotomies</u>
  - Informally, consider a dichotomy as a “data-dependent” hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- <u>Growth function</u>
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

- <u>Shatter:</u>
  - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
  - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
- <u>Break point</u>
  - k is a **break point** for H if no data set of size k can be shattered by H

$\boldsymbol{m_H(N)}$				$\boldsymbol{m_H(N)}$			
		N=1	N=2	N=3	N=4	N=5	Break Points
$N+1$	Positive Rays						
$\frac{N^2}{2}+\frac{N}{2}+1$	Positive Intervals						
2^N	Convex Sets						
	2D Perceptron						

Practice

- Dichotomies
  - Informally, consider a dichotomy as a “data-dependent” hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \ldots, \vec{x}_N)$

$$H(\vec{x}_1, \ldots \vec{x}_N) = \{(h(\vec{x}_1), \ldots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \ldots, \vec{x}_N$

- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \ldots, \vec{x}_N)} |H(\vec{x}_1, \ldots, \vec{x}_N)|$$

- 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

				$\boldsymbol{m_H(N)}$			
$\boldsymbol{m_H(N)}$		N=1	N=2	N=3	N=4	N=5	Break Points
$N+1$	Positive Rays	2	3	4	5	6	$k = 2,3,4, \ldots$
$\frac{N^2}{2}+\frac{N}{2}+1$	Positive Intervals						
2^N	Convex Sets						
	2D Perceptron						

Practice

- <u>Dichotomies</u>
  - Informally, consider a dichotomy as a “data-dependent” hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- <u>Growth function</u>
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

- <u>Shatter:</u>
  - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
  - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
- <u>Break point</u>
  - k is a **break point** for H if no data set of size k can be shattered by H

$\boldsymbol{m_H(N)}$

$\boldsymbol{m_H(N)}$		N=1	N=2	N=3	N=4	N=5	Break Points
$N+1$	Positive Rays	2	3	4	5	6	$k = 2,3,4, \dots$
$\frac{N^2}{2} + \frac{N}{2} + 1$	Positive Intervals	2	4	7	11	16	$k = 3,4,5, \dots$
2^N	Convex Sets	2	4	8	16	32	None
	2D Perceptron	2	4	8	14	?	$k = 4,5,6, \dots$

Why Break Points?

- Theorem statement (Again, take it as a fact for now)
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, i.e., if $m_H(k) < 2^k$ for some value k, then

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

- The above theorem can be rephrased below:
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, the following statements are true
    - $m_H(N) \leq N^{k-1} + 1$ [Can be proven using induction. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N
- We can "bound" the growth function without knowing it exactly.
  - Find break point!

Recall the VC Generalization Bound

- VC Generalization Bound

  With prob $1-\delta$

$$E_{out}(g) \le E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4m_H(2N)}{\delta}}$$

- In the following discussion, we treat δ as a constant [i.e., with high probability, the following is true]

$$E_{out}(g) \le E_{in}(g) + O\left(\sqrt{\frac{1}{N}\ln m_H(N)}\right)$$

[For example, we can set δ to be a small constant, say 0.01. Then every time we wrote the above inequality, we mean that it is true with probability at least 99%.]

Applying Break Points in VC Bound

- VC Bound:

$$E_{out}(g) \le E_{in}(g) + O\left(\sqrt{\frac{1}{N}\ln m_H(N)}\right)$$

- Rephrase the above theorem
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, the following statements are true
    - $m_H(N) \le N^{k-1} + 1$ [Can be proven using induction. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N

- If there are no break point ($m_H(N) = 2^N$)

  $E_{out}(g) \le E_{in}(g) + O(1)$

  (This implies that we can't infer E_{out} from E_{in} even when $N \to \infty$)

- If k is a break point for H, i.e., $m_H(N) = O(N^{k-1})$

$$E_{out}(g) \le E_{in}(g) + O\left(\sqrt{(k-1)\frac{\ln N}{N}}\right)$$

H is Either Good or Bad

- Rephrase the above theorem
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, the following statements are true
    - $m_H(N) \leq N^{k-1} + 1$ [Can be proven using induction. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N

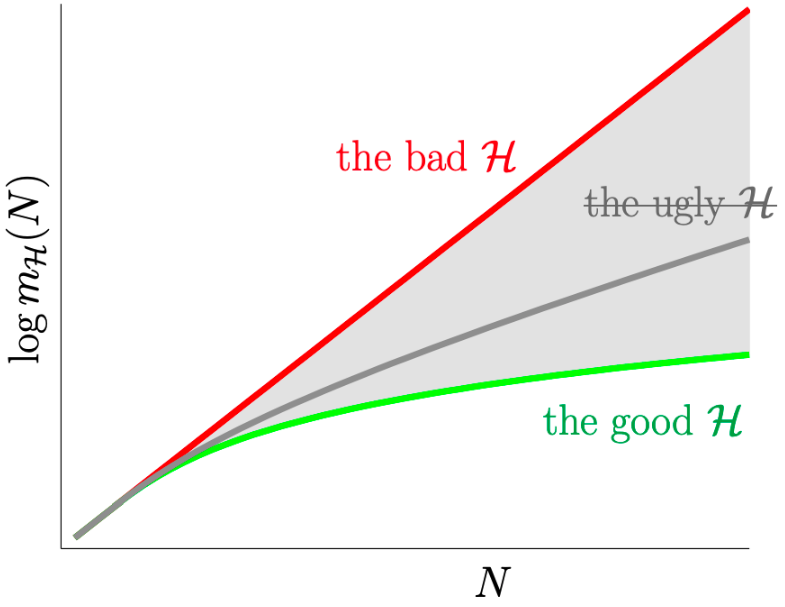
- The growth function of H is either one of the two
  - Without break points, $m_H(N) = 2^N$
  - With some break point, $m_H(N)$ is polynomial in N (it can be bounded more tightly using the theorem)
  - There is nothing in between!

- Bad hypothesis set

$$E_{out}(g) \leq E_{in}(g) + O(1)$$

- Good hypothesis set $m_H(N) = O(N^{k-1})$

$$E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{(k-1)\frac{\ln N}{N}}\right)$$



VC Dimension

- VC Dimension of H: $d_{vc}(H)$ or d_{vc}
  - The VC dimension of H is the **largest N such that $m_H(N) = 2^{\mathrm{N}}$.**
    - $d_{vc}(H) = \infty$ if $m_H(N) = 2^{\mathrm{N}}$ for all N.
  - Or, let k^* be the smallest break point for H, the VC dimension of H is $k^* - 1$

			$\boldsymbol{m_H(N)}$				
	N=1	N=2	N=3	N=4	N=5	Break Points	VC Dimension
Positive Rays	2	3	4	5	6	$k = 2,3,4, \ldots$	
Positive Intervals	2	4	7	11	16	$k = 3,4,5, \ldots$	
Convex Sets	2	4	8	16	32	None	
2D Perceptron	2	4	8	14	?	$k = 4,5,6, \ldots$	

VC Dimension

- VC Dimension of H: $d_{vc}(H)$ or d_{vc}
  - The VC dimension of H is the **largest N such that $m_H(N) = 2^N$.**
    - $d_{vc}(H) = \infty$ if $m_H(N) = 2^N$ for all N.
  - Or, let k^* be the smallest break point for H, the VC dimension of H is $k^* - 1$

			$m_H(N)$				
	N=1	N=2	N=3	N=4	N=5	Break Points	VC Dimension
Positive Rays	2	3	4	5	6	$k = 2,3,4, \ldots$	1
Positive Intervals	2	4	7	11	16	$k = 3,4,5, \ldots$	2
Convex Sets	2	4	8	16	32	None	∞
2D Perceptron	2	4	8	14	?	$k = 4,5,6, \ldots$	3

VC Dimension

- If there are no break point ($m_H(N) = 2^N$)

  $E_{out}(g) \leq E_{in}(g) + O(1)$

  (This implies that we can't infer E_{out} from E_{in} even when $N \to \infty$)

- If k is a break point for H, i.e., $m_H(N) = O(N^{k-1})$

  $E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{(k-1)\frac{\ln N}{N}}\right)$

- VC Dimension of H: $d_{vc}(H)$ or d_{vc}
  - The VC dimension of H is the **largest N such that $m_H(N) = 2^{\mathrm{N}}$.**
    - $d_{vc}(H) = \infty$ if $m_H(N) = 2^{\mathrm{N}}$ for all N.
  - Or, let k^* be the smallest break point for H, the VC dimension of H is $k^* - 1$

- Plug the definition into VC Generalization Bound

$$E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{VC}\frac{\ln N}{N}}\right)$$

Discussion on the VC Theory

All models are wrong

but some are useful

George E.P. Box

Discussion on the VC Theory

- VC Bound

$$E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{VC}\frac{\ln N}{N}}\right)$$

- Built on top of the i.i.d. data assumption
- The bound is “loose”
  - Depends only on H and N
  - The analysis is loose in many places
- However, it qualitatively characterizes the practice reasonably well
  - (the bound is roughly equally loose for every H)

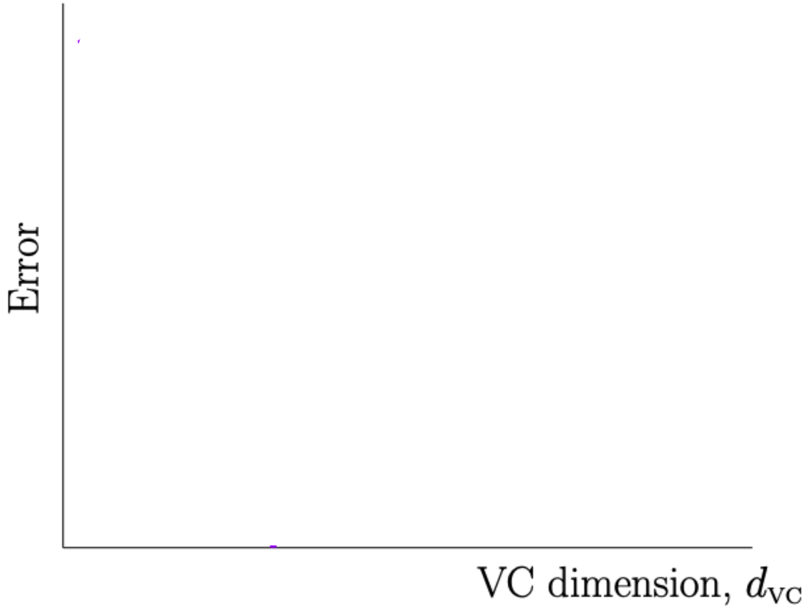
$$E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{VC}\frac{\ln N}{N}}\right)$$

- Goal of learning: Minimize $E_{out}(g)$
- How to achieve that
  - Minimize $E_{in}(g)$
    - Choose a hypothesis set with large d_{VC} (complex hypothesis likely fit data better)
  - Minimize generalization error
    - Choose a hypothesis with small d_{VC}
    - Have a lot of data points to train on (N is large)
- Think about the high-level tradeoff of choosing d_{VC} and its dependency on N

Discussion on the VC Theory

- It establishes the feasibility of learning for infinite hypothesis set
- It provides nice intuitions on what’s happening underneath ML
  - A single parameter to characterize complexity of H

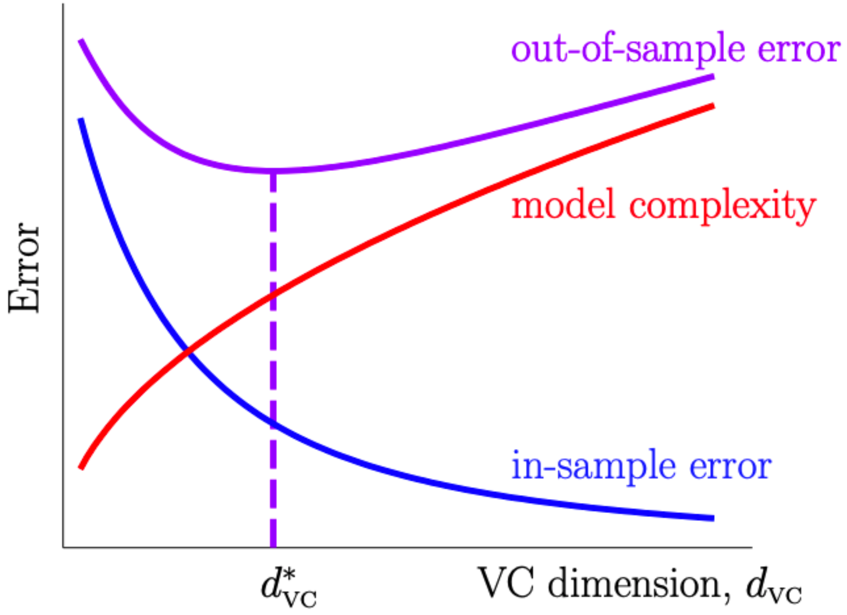
$$E_{out}(g) \le E_{in}(g) + O\left(\sqrt{d_{VC}\frac{\ln N}{N}}\right)$$



Discussion on the VC Theory

- It establishes the feasibility of learning for infinite hypothesis set.
- It provides nice intuitions on what’s happening underneath ML.
  - A single parameter to characterize complexity of H

$$E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{VC}\frac{\ln N}{N}}\right)$$



Sample Complexity

- Sample complexity:
  - Analogy to time/space complexity
  - How many data points do we need to achieve generalization error less than ϵ with prob $1-\delta$?

- Recall the (full) VC Bound:

  With prob at least $1-\delta,\ E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4((2N)^{d_{vc}}+1)}{\delta}}$

- How to determine the sample complexity?
  - Set $\sqrt{\frac{8}{N} ln \frac{4((2N)^{d_{vc}}+1)}{\delta}} \leq \epsilon$
  - We get $N \geq \frac{8}{\epsilon^2} ln \left(\frac{4\left(1+(2N)^{d_{VC}}\right)}{\delta}\right)$

- $N \propto 1/\epsilon^2$
- $N = O(d_{vc} \ln N)$
  - Empirically, people roughly $N \propto d_{vc}$

Test Set

- Goal of learning: Minimize $E_{out}(g)$

- Can we estimate E_{out} directly?
  - Reserve a test set (D_{test}) before learning
  - Ensure D_{test} is not used at all in any way for learning
  - For D_{test}, g is a “fixed” hypothesis and standard Hoeffding’s inequality is valid
  - Let $E_{test}(g)$ be the error in the test set

$$P\{|E_{test}(g) - E_{out}(g)| > \epsilon\} \leq 2e^{-2\epsilon^2 N_{test}} \text{ where } N_{test} = |D_{test}|$$

Test Set

- Test set is great: we can obtain an unbiased estimate of E_{out}
- At what cost?
  - We have a finite amount of data
  - Data points in test set cannot be involved in learning at all
  - More points in test set
    - Better estimate of E_{out}
    - Less data points in training set -> often leads to worse learned hypothesis

- Practical rule of thumb (i.e., a common heuristic, not really a gold rule)
  - 80% for training, 20% for testing

Proof: Bounding Growth Functions

Recall: Theorem in Bounding Growth Function

- Theorem statement:
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, i.e., if $m_H(k) < 2^k$ for some value k, then

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

- You were asked to take this as a fact
- Will provide proof sketch now

Proof Sketch

[See LFD Section 2.1.2 for the formal proof]

[Safe to Skip] (This proof won’t appear in exams/homework)

[Safe to Skip]

Key Intuitions

- When there exist a break point k
  - No datasets of size k can be shattered
  - It also imposes strong constraints on dataset of size $k' > k$
    - No subset of data with size k can be shattered
  - This leads to the bound $m_H(N) = O(N^{k-1})$

[Safe to Skip]

Proof Intuitions

- Max # dichotomies you can list on **2 points** when **no 2 points can be shattered**

$\vec{x}_1$	$\vec{x}_2$
+1	+1
+1	-1
-1	+1

[Safe to Skip]

Proof Intuitions

- Max # dichotomies you can list on **4 points** when **no 2 points can be shattered**

$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	$\vec{x}_4$
+1	+1	+1	+1
+1	+1	+1	-1
+1	+1	-1	+1
+1	-1	+1	+1
-1	+1	+1	+1

Can you add an additional dichotomy?

[Safe to Skip]

Proof Intuitions

- How **“no 2 points are shattered”** impacts the scenario with **4 points**?

$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	$\vec{x}_4$
+1	+1	+1	+1
+1	+1	+1	-1
+1	+1	-1	+1
+1	-1	+1	+1
-1	+1	+1	+1

$(\vec{x}_1, \vec{x}_2, \vec{x}_3)$ appear twice, with different $\vec{x}_4$

No 1 point is shattered

$(\vec{x}_1, \vec{x}_2, \vec{x}_3)$ appear once (including one in each of the pair above)

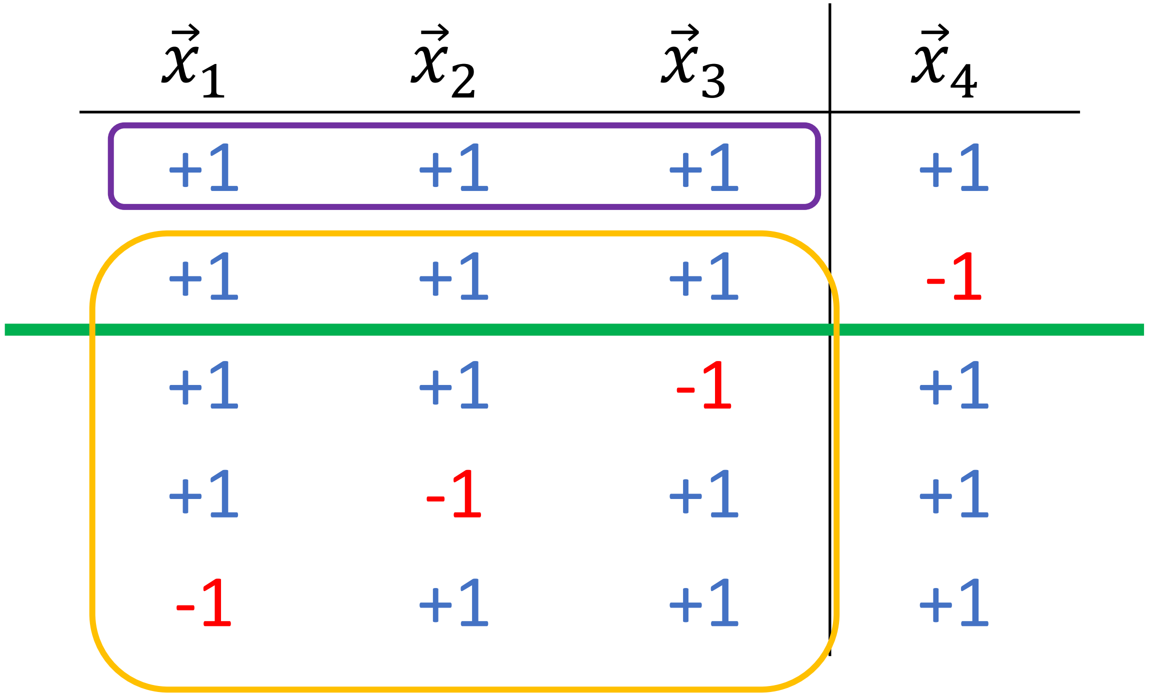
No 2 points are shattered

[Safe to Skip]

Proof Intuitions

- Max # dichotomies you can list on **4 points** when **no 2 points can be shattered**

No 1 point is shattered



$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	$\vec{x}_4$
+1	+1	+1	+1
+1	+1	+1	-1
+1	+1	-1	+1
+1	-1	+1	+1
-1	+1	+1	+1

No 2 points are shattered

$\boldsymbol{B(N,k)}$: max # dichotomies on N points when no k points are shattered

A recursive definition:

$$\boldsymbol{B(N,k)} \le \boldsymbol{B(N-1,k)} + \boldsymbol{B(N-1,k-1)}$$

Sauer's Lemma: $\boldsymbol{B(N,k)} \le \sum_{i=0}^{k-1} \binom{N}{i}$

Can be proved by induction

$\boldsymbol{B(N,k)} \le \sum_{i=0}^{k-1} \binom{N}{i}$ is the bound of $m_H(N)$ for H with break point k

Summary: Bounding Growth Functions

- Theorem statement:
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, i.e., if $m_H(k) < 2^k$ for some value k, then

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

- Rephrase the 2nd statement of the above theorem
  - If k is a break point for H, the following statements are true
    - $m_H(N) \leq N^{k-1} + 1$ [Can be proven using induction from above. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N
  - If d_{vc} is the VC dimension of H, then
    - $m_H(N) \leq \sum_{i=0}^{d_{vc}} \binom{N}{i}$
    - $m_H(N) \leq N^{d_{vc}} + 1$
    - $m_H(N) = O(N^{d_{vc}})$

If d_{vc} is the VC dimension of H, $d_{vc} + 1$ is a break point for H

Summary: Vapnik–Chervonenkis (VC) Bound

- VC Generalization Bound

  With prob at least $1-\delta$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4m_H(2N)}{\delta}}$$

- Let d_{vc} be the VC dimension of H, we have $\boldsymbol{m_H(N) \leq N^{d_{vc}} + 1}$.

  With prob at least $1-\delta$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4((2N)^{d_{vc}}+1)}{\delta}}$$

- If we treat δ as a constant, then we can say, with high probability

$$E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{vc} \frac{\ln N}{N}}\right)$$

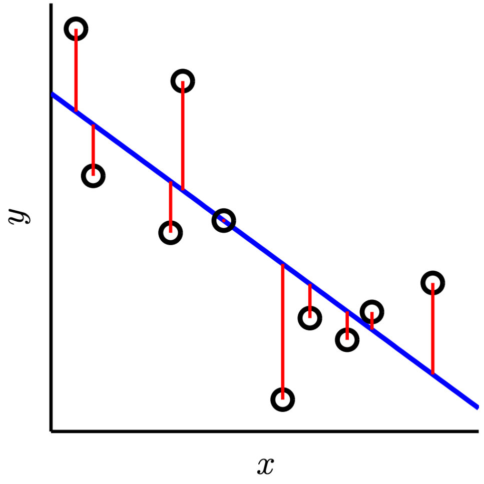
Bias-Variance Decomposition

Another theory of generalization

Real-Value Target and Squared Error

- So far, we focus on binary target function and binary error
  - Binary target function $f(\vec{x}) \in \{-1,1\}$
  - Binary error $e\left(h(\vec{x}), f(\vec{x})\right) = \mathbb{I}[h(\vec{x}) \neq f(\vec{x})]$

- Real-value functions [”**regression**”] and squared error?
  - Real-value target function $f(\vec{x}) \in \mathbb{R}$
  - Square error $e\left(h(\vec{x}), f(\vec{x})\right) = \left(h(\vec{x}) - f(\vec{x})\right)^2$



Real-Value Target and Squared Error

- Real-value functions [called ”**regression**”] and squared error
  - Real-value target function $f(\vec{x}) \in \mathbb{R}$
  - Squared error $e\big(h(\vec{x}), f(\vec{x})\big) = \big(h(\vec{x}) - f(\vec{x})\big)^2$

- Errors:
  - In-sample error: $E_{in}(g) = \frac{1}{N} \sum_{n=1}^{N} e(h(\vec{x}_n), f(\vec{x}_n)) = \frac{1}{N} \sum_{n=1}^{N} \big(h(\vec{x}_n) - f(\vec{x}_n)\big)^2$
  - Out-of-sample error: $E_{out}(g) = \mathbb{E}_{\vec{x}}[e\big(h(\vec{x}), f(\vec{x})\big)] = \mathbb{E}_{\vec{x}}[\big(g(\vec{x}) - f(\vec{x})\big)^2]$

- Theory of generalization: What can we say about $E_{out}(g)$?

- Note that g is learned by some algorithm on the dataset D
  - We’ll make the dependency on D explicit and write it as $\boldsymbol{g^{(D)}}$ here.
  - [In VC theory, we consider the worst-case D through the definition of growth function $m_H(N)$]

Define “expected” hypothesis
$\bar{g}(\vec{x}) = \mathbb{E}_D\left[g^{(D)}(\vec{x})\right]$

- $E_{out}\left(g^{(D)}\right) = \mathbb{E}_{\vec{x}}[\left(g^{(D)}(\vec{x}) - f(\vec{x})\right)^2]$
- $\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right]$

$$= \mathbb{E}_D\left[\mathbb{E}_{\vec{x}}\left[\left(g^{(D)}(\vec{x}) - f(\vec{x})\right)^2\right]\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x}) + \bar{g}(\vec{x}) - f(\vec{x})\right)^2\right]\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x}) + \bar{g}(\vec{x}) - f(\vec{x})\right)^2\right]\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2 + \left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2 + 2\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)\left(\bar{g}(\vec{x}) - f(\vec{x})\right)\right]\right]$$

- Note that $\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)\left(\bar{g}(\vec{x}) - f(\vec{x})\right)\right] = \left(\bar{g}(\vec{x}) - f(\vec{x})\right)\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)\right] = 0$

Finishing Up

- $\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right]$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2 + \left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2\right]\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2\right]\right] + \mathbb{E}_{\vec{x}}\left[\left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\text{Variance of } g^{(D)}(\vec{x}) + \text{Bias of } \bar{g}(\vec{x})\right]$$

$$= \text{Variance} + \text{Bias}$$

- Bias-Variance Decomposition

$\bar{g}(\vec{x}) = \mathbb{E}_D\left[g^{(D)}(\vec{x})\right]$

X: a random variable
μ: the mean of X

Variance of X:
$Var(X) = \mathbb{E}[(X-\mu)^2]$

Discussion

- $\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right] = \mathbb{E}_{\vec{x}}\left[\overset{\overset{\text{Bias}(\vec{x})}{\uparrow}}{\left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2}\right] + \mathbb{E}_{\vec{x}}\left[\overset{\overset{\text{Var}(\vec{x})}{\uparrow}}{\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2\right]}\right]$

- This is a **conceptual** decomposition
  - Both $\bar{g}$ and f are unknown
  - We can’t really calculate bias and variance in practice

- However, it provides a conceptual guideline in decreasing E_{out}