CS639: Introduction to Computational Learning Theory **Scribe:** Changho Shin
Lecture 6: VC-Dimension, Lower Bound on Mistake Bound of Any Online Learning Algorithms
Lecturer: Ilias Diakonikolas

6.1 VC-Dimension

- Combinatorial parameter at a concept class
- Roughly speaking, "How complicated the concept class is"
- $\Rightarrow$ How many mistakes we need to learn it

Definition 6.1.1 *Let C be a concept class over $\mathcal{X}$. We say that $S \subseteq X$ is shattered by C if for every $T \subseteq S$, there exist some $c \in C$ such that $c \cap S = T$.*

Recall every boolean function c over $\mathcal{X}$ is equivalent to $\{x \in \mathcal{X} : c(x) = 1\}$. In other words, S is shattered if C induces all possible dichotomies on S.

Example

- $\mathcal{X} = \{1, 2, 3, 4, 5\}$
- C has 6 concepts: $c_1 = \{1, 2, 3\}, c_2 = \{2, 4, 5\}, c_3 = \{3, 4\}, c_4 = \{1, 2, 5\}, c_5 = \{1, 3, 5\}, c_6 = \{5\}$.
- Then C shatters {2, 4}, since
  - $\emptyset = c_6 \cap S$
  - $\{2\} = c_4 \cap S$
  - $\{4\} = c_3 \cap S$
  - $\{2, 4\} = c_2 \cap S$

$VCDim(C)$ is the size of the largest set $S \subseteq \mathcal{X}$.

Definition 6.1.2 *VC Dimension of a concept class C is the size of the largest shattered set by C, i.e., smallest integer d such that*

- *There exists a set of size d that is shattered.*
- *No set of size $d + 1$ is shattered.*

Note that VC dim can be ∞.

Example

- $\mathcal{X} = \{1, 2, 3, 4, 5\}, C = \{c_1, ..., c_6\}$ as defined above.
- We can say that $VCDim(C) \geq 2$.
- How about upper bound?
- Claim: $VCDim(C) \leq 2$.
- We have only 6 concepts in class C and need at least 8 concepts to shatter any 3 elements set.

Example

- $\mathcal{X} = \mathbb{R}, C =$all closed intervals $[a, b]$.
- Claim: $VCDim(C)$=2
- Note that even though C is infinite, it has sufficient structure so that VC Dim is bounded.
- Lower bound proof: $VCDim(C) \geq 2$. Consider $S = \{1, 2\}$.
- Upper bound proof: $VCDim(C) \leq 2$. We need to show that there is no set of size 3 that is shattered by C. Consider $S = \{x, y, z\}(x < y < z)$ and $c(x) = c(z) = 1, c(y) = 0$. This particular dichotomy cannot be obtained with any concept in C. If an interval $[a, b] \in c$ contains x and z, then it also contains y
- As a result, $VCDim(C) = 2$

Example

- $\mathcal{X} = \mathbb{R}^2, C =$all halfspaces over $\mathbb{R}^2$. Obviously, $VCDim(C) \geq 3$. Also, $VCDim \leq 3$, which can be shown by the fact that there is no set of 4 points that is shattered by c.
  - If 3 points lie on a line, then this set is not shattered.
  - Otherwise, either one point inside triangle formed by other three points. Or, all four points are vertices of quadrilateral, where we can induce a similar argument.

The following is the more general theorem.

Theorem 6.1.3 *Halfpsaces in* $\mathbb{R}^n$ *have* $VCDim(C) = n + 1$.

Example

- $X = \{0, 1\}^n$
- $C =$all monotone conjunctions.
- **cClaim**: $VCDim(C) = n$

- Upper bound: $VCdim(C) \leq n$. Proof: By definition, for any C, $VCDim(C) \leq \log_2 |C|$, and $|C| = 2^n$
- Lower bound: $VCdim(C) \geq n$. Proof: Suffices to find a set of size n that is shattered. Consider the set of examples $S = \{\mathbb{1} - e_i, 1 \leq i \leq n\}$. For S, the learning rule is including x_i if $c(\mathbb{1} - e_i) = -$, where c is the true concept.

Digression: What is the complexity of computing the $VCDim$ of a given concept class C?
$\rightarrow$ Given a binary matrix that represents a concept class, finding VC dim of the class is "log NP complete" (Not computationally easy)

Theorem 6.1.4 *Suppose C has $VCDim(C) = d$. then no online algorithm for C can have mistake bound $\leq d$*

Proof: $VCDim(C) = d$ implies that some set $S = \{x^1, ..., x^d\}$ is shattered by C. Adversary gives x' to our algorithm. Our algorithm outputs predictions y'. Important point adversary can say that our algorithms prediction "wrong" still have complete freedom has response to predictions on $x^2, ..., x^d$ because S is shattered. Adversary's limitation is that if they cause an algorithm to make a mistake, there needs to be a function $c \in C$, which is consistent with adversary's choices. No matther what the adversary decides for $\{x^1, ..., x^d\}$ to be, this is possible to achieve by some function in C because S is shattered. ■

Corollary 6.1.5 *Elimination algorithm has the optimal mistake bound for monotone conjunctions.*

Note: Lower bound of VC dim for M.B is not always tight. One can construct examples of concept class whose $VCDim = CONSTANT$, but the mistake bound of best online algorithm is very large. There exists a more complicated combinatorial parameter that exactly characterized the mistake bound "Littlestone dim" in Littleston's paper.

6.2 Weighted Majority Algorithm (=Noise Tolerant Halving Algorithm)

- Setting
  - Pool of N "experts"
  - Sequence of trials
- At each trial, each expert makes binary predictions. The weighted majority voting algorithm has some parameter θ $(0 < \theta < 1)$.
- Each expert i has weight w_i, which represents how much we trust.
- Initially, each $w_i = 1$.
- At each trial, each expert i predicts $z_i \in \{0, 1\}$.
- Weighted Majority Voting
  - Let $q_0 = \sum_{i \text{ s.t.} z_i = 0} w_i$, and $q_1 = \sum_{i \text{ s.t.} z_i = 1} w_i$.

– Predict $z = \begin{cases} 0, & \text{if } q_0 \geq q_1 \\ 1, & \text{if } q_0 < q_1 \end{cases}$

- Get true outcome of trial.
- For each $i \in \{1, ..., N\}$ s.t. z_i is wrong, $w_i \leftarrow \theta w_i$, where $0 < \theta < 1$.
- Observation: If $\theta = 0$, this is just Halving Algorithm.
- Comparison of Halving Algorithm and Weighted Majority Voting Algorithm

Halving Algorithm	Weighted Majority Voting
i-th concept in C	Expert i
output of i-th concept in $j-th$ example	Prediction of $i-th$ expert on j-th trial
$\|C\|$	$N = \#$ of experts

Ideally, would like to compete with best single expert $\approx \log_2 |C| + m$ ($\#$ mistakes of best expert).

Theorem 6.2.1 *For any sequence of trials, if best expert in pool makes m mistakes, then the weighted majority voting algorithm with parameter θ makes at most $\frac{\log N + m \log 1/\theta}{\log 2/(1+\theta)}$*

- $\theta = 1/2 \rightarrow 2.41(m + \log N)$
- $\theta = 3/4 \rightarrow 2.2m + 5.2 \log N$
- $\theta = (1 - \epsilon) \rightarrow \approx 2m + \frac{2}{\epsilon} \log N$