Machine Learning Coms-4771

PAC model, VC-dimension

Lecture 16

(following Avrim Blum's notes)

Basic setting:

- We are given a sample set $S = \{(x, y)\}$.
  - assume that x's come from some fixed (but unknown) probability distribution D over the instance space
  - assume that labels y come from some target function $f \in C$
- Q: How big does S have to be to guarantee that the learned h does well on new examples from D? The error rate of h is

$$e_D(h) = \mathbf{Pr}_{x \sim D}[h(x) \neq f(x)].$$

- Probably Approximately Correct (PAC) model:
  - (Approximately correct) We can't hope to recover f exactly since D might place low probability on some part of instance space. The goal is to output h with $e_D(h) \leq \epsilon$
  - (Probably correct) We can't necessarily guarantee low error rate since we might get a non-representative sample S. So, our goal is to get an approximation with high probability
- An algorithm PAC-learns concept class C by hypothesis class H if for any $f \in C$, any D, and any $\epsilon, \delta > 0$, the algorithm takes $m = \text{poly}(1/\epsilon, 1/\delta)$ examples from D and, with probability $1 - \delta$ (over the draw of the sample), produces $h \in H$ with $e_D(h) \leq \epsilon$.

Basic sample complexity bound recap:

- If $|S| \geq \frac{1}{\epsilon}(\ln(|C|) + \ln(1/\delta))$, then with probability $\geq 1 - \delta$, all $h \in C$ with $e_D(h) > \epsilon$ have $e_S(h) > 0$. So a consistent h with $e_S(h) = 0$ is guaranteed to have $e_D(h) \leq \epsilon$ with high probability.

The argument is very simple:

- Probability that a "bad" h (with $e_D(h) > \epsilon$) is consistent with $m = |S|$ examples drawn from D is at most $(1 - \epsilon)^m$.
- Since there are $|C|$ possible h, the probability that there exists a bad h consistent with m examples is at most $|C|(1 - \epsilon)^m$ (by the union bound). Set to δ and solve for m.
- We want the smallest m such that $(1 - \epsilon)^m \leq \delta/|C|$. Use approximation $(1 - \epsilon)^m \leq e^{-\epsilon m}$ (from $(1 + \frac{1}{x})^x \leq e$).
- So it suffices to have $e^{-\epsilon m} \leq \delta/|C|$ or equivalently $e^{\epsilon m} \geq |C|/\delta$.
- Taking logs, $m \geq \frac{1}{\epsilon}(\ln(|C|) + \ln(1/\delta))$.

Agnostic learning:

- What if there is no perfect $h \in C$?
- Without making any assumptions about the target function, can we say that with high probability all $h \in C$ satisfy $|e_D(h) - e_S(h)| \leq \epsilon$? (called "uniform convergence")
- Recall the Occam's Razor Bound from John's lecture: For all $h \in C$, for all D, for all $\delta \in (0, 1]$,

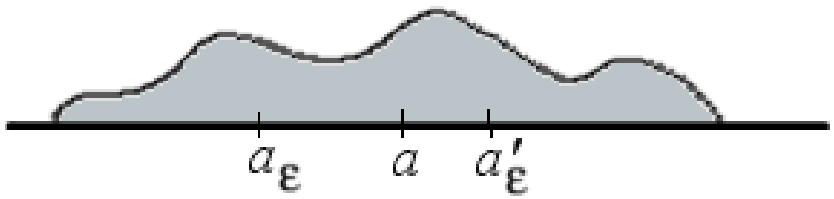
$$\mathbf{Pr}_{S \sim D^m}\left[e_D(h) \leq e_S(h) + \underbrace{\sqrt{\frac{\ln(|C|) + \ln(1/\delta)}{2m}}}_{\epsilon} \right] \geq 1 - \delta.$$

  So $m \geq \frac{1}{\epsilon^2}(\ln(|C|) + \ln(1/\delta))$ (basically an application of Hoeffding or Chernoff bounds)
- worse than previous bound $1/\epsilon$ is now $1/\epsilon^2$ but we are not making any assumption about the target function

Effective size of C

We don't want to depend on the explicit size of C (e.g., it can be infinite). Example:

- $C = \{[0, a] : 0 \leq a \leq 1\}$. Define a_ϵ so that $\mathbf{Pr}([a_\epsilon, a]) = \epsilon$ and a'_ϵ so that $\mathbf{Pr}([a, a'_\epsilon]) = \epsilon$.



- We just need $(1 - \epsilon)^m \leq \delta/2$ (since it's enough to get at least one example in each of the two ϵ-intervals), so it suffices to have $m \geq (1/\epsilon) \ln(2/\delta)$.
- Can we generalize this to arbitrary C?

Effective size of C

Define $C[m]$ as the maximum number of ways to split m points using concepts in C.

- What is $C[m]$ for "initial intervals": $C = \{[0, a] : 0 \leq a \leq 1\}$?
- What about $C = \{[a, b] : 0 \leq a < b \leq 1\}$?
- What about linear separators in $\mathbb{R}^2$?

Effective size of C

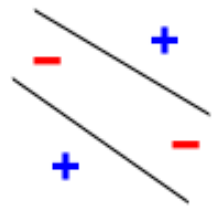
Define $C[m]$ as the maximum number of ways to split m points using concepts in C.

- What is $C[m]$ for "initial intervals": $C = \{[0, a] : 0 \leq a \leq 1\}$?
- What about $C = \{[a, b] : 0 \leq a < b \leq 1\}$?
- What about linear separators in $\mathbb{R}^2$?

- It turns out that we can roughly replace $|C|$ with $C[2m]$.
- Theorem: For any class C, distribution D, if $m > (2/\epsilon)[\log(2C[2m]) + \log(1/\delta)]$, then with probability $1 - \delta$, all $h \in C$ with error rate at least ϵ are inconsistent with the sample (assuming the target is in C).
- $C[m]$ can be hard to calculate exactly, but can get a good bound using the notion of the VC-dimension.

Shattering

- Definition: A set of points S is shattered by C if there are concepts in C that split S in all of the $2^{|S|}$ possible ways.
  - All possible ways of classifying points in S are achievable using concepts in C, i.e., it looks like C contains all functions.
- Example: Any 3 non-collinear points in $\mathbb{R}^2$ can be shattered by linear separators, but no set of 4 points can.



- Example: No set of 2 points can be shattered by $C = \{[0, a] : 0 \leq a \leq 1\}$. No set of 3 points can be shattered by $C = \{[a, b] : 0 \leq a < b \leq 1\}$.

VC dimension

- Definition: The VC-dimension of a concept class C is the size of the largest set of points that can be shattered by C.
- Another way to put it: If the VC-dimension is d, there exists a set of d points that can be shattered, but there is no set of $d + 1$ points that can be shattered.
- Examples:
  - VCdim (Linear threshold functions in $\mathbb{R}^2$) $= 3$

VC dimension

- Definition: The VC-dimension of a concept class C is the size of the largest set of points that can be shattered by C.
- Another way to put it: If the VC-dimension is d, there exists a set of d points that can be shattered, but there is no set of $d+1$ points that can be shattered.
- Examples:
  - VCdim (Linear threshold functions in $\mathbb{R}^2$) $= 3$
  - VCdim (Linear threshold functions in $\mathbb{R}^k$) $= ?$
  - VCdim (initial intervals $[0, a]$) $= 1$
  - VCdim (intervals $[a, b]$) $= 2$

VC dimension

- Definition: The VC-dimension of a concept class C is the size of the largest set of points that can be shattered by C.
- Another way to put it: If the VC-dimension is d, there exists a set of d points that can be shattered, but there is no set of $d+1$ points that can be shattered.
- Examples:
  - VCdim (Linear threshold functions in $\mathbb{R}^2$) $= 3$
  - VCdim (Linear threshold functions in $\mathbb{R}^k$) $=$?
  - VCdim (initial intervals $[0, a]$) $= 1$
  - VCdim (intervals $[a, b]$) $= 2$
  - VCdim (collection of all finite subsets of $\mathbb{R}$) $=$?

Upper and lower bound theorems

- Sauer's lemma:

$$C[m] \leq \sum_{i=0}^{\mathrm{VCdim}(C)} \binom{m}{i} = O(m^{\mathrm{VCdim}(C)})$$

- Theorem: For any class C, distribution D, if

$$m > O((1/\epsilon)[\mathrm{VCdim}(C)\log(1/\epsilon) + \log(1/\delta)],$$

then with probability $1-\delta$, all $h \in C$ with error rate at least ϵ are inconsistent with the sample (assuming the target is in C).

- Lower bound: For any algorithm A, there exists a distribution D and target in C such that if $m < (\mathrm{VCdim}(C)-1)/8\epsilon$ then $\mathbf{E}[e_D(A)] \geq \epsilon$.

Proof of the lower bound

Lower bound: For any algorithm A, there exists a distribution D and target in C such that if $m < (\text{VCdim}(C) - 1)/8\epsilon$ then $\mathbf{E}[e_D(A)] \geq \epsilon$.

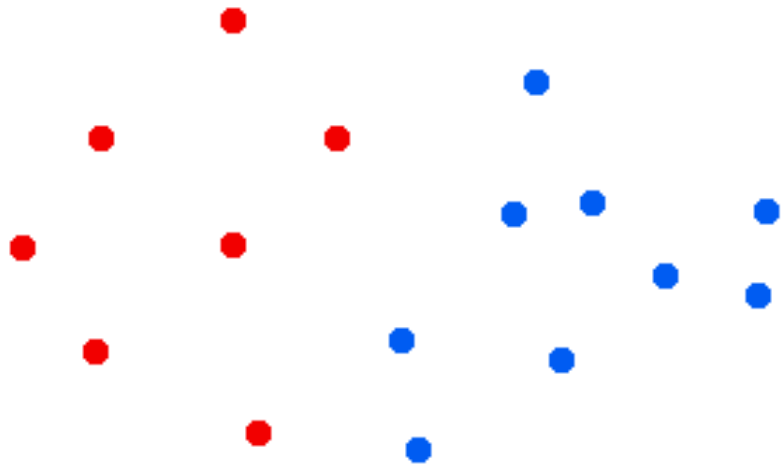
Proof:

- Consider $d = \text{VCdim}(C)$ points that can be shattered.
- Define D to put $1 - 4\epsilon$ probability mass on one of the points, and distribution the remaining 4ϵ equally among the other $d - 1$ points (called rare points). Let the target be a random labeling of these points.
- After seeing m examples, we expect only $4\epsilon m$ to be rare ones. This is at most $(d - 1)/2$ since $m < (d - 1)/8\epsilon$.
- So there are still at least $(d - 1)/2$ points which we haven't seen in the sample, and thus 2ϵ chance that the next example is new. Since the target is random, we can't do better than random guessing, so the expected error is at least ϵ.

Active Learning

Supervised Learning

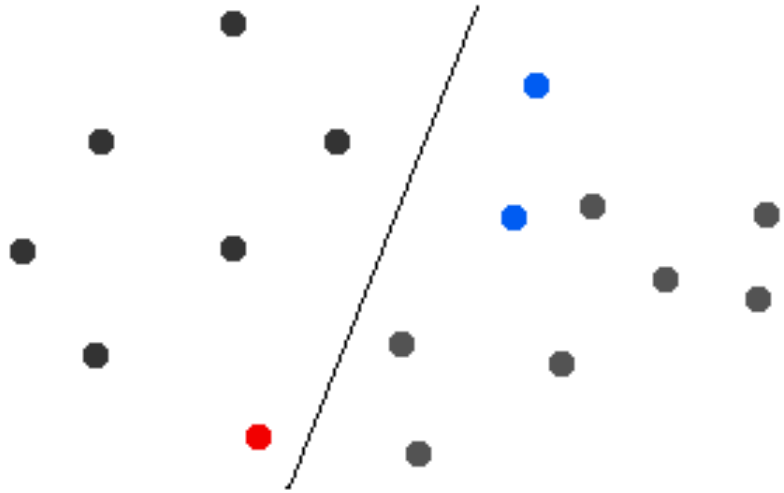
Given access to a labeled sample (drawn iid from an unknown distribution D), we want to learn a classifier $h \in H$ with $e_D(h) \leq epsilon$.



VC theory: need m to be roughly $\text{VCdim}(H)/\epsilon$, in the *realizable* case (when all examples are consistent with some target function in H)

Active Learning

In many situations unlabeled data is cheap and easy to collect, but labeling it is very expensive (e.g., requires a hired human). Idea: let the classifier pick which examples it wants labeled.



The hope is that by directing the labeling process, we can pick a good classifier at low cost.

Membership queries (Angluin 1992)

- You can ask for the label of any point (no unlabeled data)
- What is the smallest number of "membership queries" (does the point belong to the target concept? is the label 1 or not?) needed to identify the target $h^* \in H$ exactly?
- Synthesize highly informative queries. Each query cuts the version space in half.
- Example: to learn monotone conjunctions of n boolean variables, ask for the label of n examples $(1, \ldots, 1, 0, 1, \ldots, 1) \in \{0, 1\}^n$, with 0 in position i for $1 \leq i \leq n$.
- Many positive results in this framework even for complicated hypothesis classes
- Problem: Created synthetic example are impossible to label in practice!
- [Baum and Lang, 1991] Tried it for handwritten character recognition. [Lewis and Gale, 1992] Natural language classification. Synthesized images and text are incomprehensible to humans.

Selective sampling

A better, PAC-like model [Cohn, Atlas, and Ladner, 1992]

- Underlying distribution D on the $X \times Y$
- The learner can:
  - draw an unlabeled example from D
  - ask for the label of *one of these examples*
- PAC-type guarantee: find an ϵ-good classifier with high probability using a small number of queries.

Why should we hope for success?

- Simple hypothesis class: threshold functions on the real line: $H = \{h_w(x) = \mathbf{1}[x > w] : w \in [0, 1]\}$.
- Assume realizable case. Start with $m = O(1/\epsilon)$ unlabeled points.
- We just need $O(\log m) = O(\log 1/\epsilon)$ labels to find the threshold point. An exponential improvement in sample complexity compared to passive learning!

We will continue on Thursday

Tentative Plan:

- Mar 27: Active Learning (Alina)
- Apr 1: Large-scale learning and online optimization (John)
- Apr 3: Multi-armed bandits (Alina)
- Apr 8: Algorithms for Nearest Neighbor Search and applications in learning (Alina)
- Apr 10: Modular learning (John)
- Apr 15: Reinforcement learning (John)
- Apr 17—29 (four lectures): Graphical Models and Hidden Markov Models (Tony)
- May 1: final (25% of the grade)
- Projects (announced next week): remaining 25% of the grade + bonus points to recover from the exams