Learning Theory

Nicholas Ruozzi
University of Texas at Dallas

Based on the slides of Vibhav Gogate and David Sontag

Announcements

- TA: Baoye Xue
  - Office hours: Monday and Wednesday 5pm-6pm in the Clark Center CN 1.202D
  - Email: bxx140230@utdallas.edu

Learning Theory

- So far, we've been focused only on algorithms for finding the best hypothesis in the hypothesis space
  - How do we know that the learned hypothesis will perform well on the test set?
  - How many samples do we need to make sure that we learn a good hypothesis?
  - In what situations is learning possible?

Learning Theory

- If the training data was linearly separable, we saw that perceptron/SVMs will always perfectly classify the training data
  - This does not mean that it will perfectly classify the test data
  - Intuitively, if the true distribution of samples is linearly separable, then seeing more data should help us do better

Problem Complexity

- Complexity of a learning problem depends on
  - Size/expressiveness of the hypothesis space
  - Accuracy to which a target concept must be approximated
  - Probability with which the learner must produce a successful hypothesis
  - Manner in which training examples are presented, e.g. randomly or by query to an oracle

Problem Complexity

- Measures of complexity
  - Sample complexity
    - How much data you need in order to (with high probability) learn a good hypothesis
  - Computational complexity
    - Amount of time and space required to accurately solve (with high probability) the learning problem
    - Higher sample complexity means higher computational complexity

PAC Learning

- Probably approximately correct (PAC)
  - Developed by Leslie Valiant
  - The only reasonable expectation of a learner is that with high probability it learns a close approximation to the target concept
  - Specify two small parameters, ϵ and δ, and require that with probability at least $(1-\delta)$ a system learn a concept with error at most ϵ

Consistent Learners

- Imagine a simple setting
  - The hypothesis space is finite (i.e., $|H| = c$)
  - The true distribution of the data is $p(\vec{x})$, no noisy labels
  - We learned a perfect classifier on the training set, let's call it $h \in \mathrm{H}$
    - A learner is said to be consistent if it always outputs a perfect classifier on the training data assuming that one exists
  - Want to compute the error of the classifier

Notions of Error

- Training error of $h \in H$
  - The error on the training data
  - Number of samples incorrectly classified divided by the total number of samples
- True error of $h \in H$
  - The error over all possible future random samples
  - Probability that h misclassifies a random data point

$$p(h(x) \neq y)$$

Learning Theory

- Let $\left(x^{(1)}, y_1\right), \ldots, \left(x^{(m)}, y_m\right)$ be m labelled data points sampled independently according to p
- Let C_i^h be a random variable that indicates whether or not the i^{th} data point is correctly classified
- The probability that h misclassifies the i^{th} data point is

$$p\left(C_i^h = 0\right) = \sum_{(x,y)} p(x, y)\, 1_{h(x) \neq y} = \epsilon_h$$

Learning Theory

- Let $(x^{(1)}, y_1), \ldots, (x^{(m)}, y_m)$ be m labelled data points sampled independently according to p

- Let C_i^h be a random variable that indicates whether or not the i^{th} data point is correctly classified

- The probability that h misclassifies the i^{th} data point is

$$p(C_i^h = 0) = \sum_{(x,y)} p(x, y)\, 1_{h(x) \neq y} = \epsilon_h$$

This is the true error of h

Learning Theory

- Probability that all data points classified correctly?

$$p\left(C_1^h = 1, \dots, C_m^h = 1\right) = \prod_{i=1}^{m} p(C_i^h = 1) = (1 - \epsilon_h)^m$$

- Probability that a hypothesis $h \in \mathrm{H}$ whose true error is at least ϵ correctly classifies the m data points is then

$$p\left(C_1^h = 1, \dots, C_m^h = 1\right) \le (1 - \epsilon)^m \le e^{-\epsilon m}$$

for $\epsilon \le 1$

Learning Theory

- The version space (set of consistent hypotheses) is said to be ϵ-exhausted if and only if every consistent hypothesis has true error less than ϵ
  - Enough samples to guarantee that every consistent hypothesis has error at most ϵ
- We'll show that w.h.p. every hypothesis with true error at least ϵ is not consistent with the data

The Union Bound

- Let $H_{BAD} \subseteq H$ be the set of all hypotheses that have true error at least ϵ
- From before for each $h \in H_{BAD}$,

$$p(h \text{ correctly classifies all } m \text{ data points}) \leq e^{-\epsilon m}$$

- So, the probability that *some* $h \in H_{BAD}$ correctly classifies all of the data points is

$$p\left(\bigvee_{h \in H_{BAD}} \left(C_1^h = 1, \ldots, C_m^h = 1\right)\right) \leq \sum_{h \in H_{BAD}} p\left(C_1^h = 1, \ldots, C_m^h = 1\right)$$
$$\leq |H_{BAD}| e^{-\epsilon m}$$
$$\leq |H| e^{-\epsilon m}$$

Haussler, 1988

- What we just proved:
  - **Theorem:** For a finite hypothesis space, H, with m i.i.d. samples, and $0 < \epsilon < 1$, the probability that the version space is not ϵ-exhausted is at most $|H|e^{-\epsilon m}$
- We can turn this into a sample complexity bound

Sample Complexity

- Let δ be an upper bound on the desired probability of not ϵ-exhausting the sample space
  - The probability that the version space is not ϵ-exhausted is at most $|H|e^{-\epsilon m} \leq \delta$
  - Solving for m yields

$$m \geq -\frac{1}{\epsilon}\log\frac{\delta}{|H|}$$
$$= \left(\log|H| + \log\frac{1}{\delta}\right)/\epsilon$$

Sample Complexity

- Let δ be an upper bound on the desired probability of not ϵ-exhausting the sample space
  - The probability that the version space is not ϵ-exhausted is at most $|H|e^{-\epsilon m} \leq \delta$
  - Solving for m yields

$$m \geq -\frac{1}{\epsilon}\log\frac{\delta}{|H|}$$

$$= \left(\log|H| + \log\frac{1}{\delta}\right)/\epsilon$$

This is sufficient, but not necessary (union bound is quite loose)

Decision Trees

- Suppose that we want to learn an arbitrary Boolean function given n Boolean features

- Hypothesis space consists of all decision trees

  – Size of this space $= ?$

- How many samples are sufficient?

Decision Trees

- Suppose that we want to learn an arbitrary Boolean function given n Boolean features

- Hypothesis space consists of all decision trees
  - Size of this space $= 2^{2^n}$ = number of Boolean functions on n inputs

- How many samples are sufficient?

$$m \geq \left(\log 2^{2^n} + \log \frac{1}{\delta} \right) / \epsilon$$

Generalizations

- How do we handle the case the there is no perfect classifier?
  - Pick the hypothesis with the lowest error on the training set
- What do we do if the hypothesis space isn't finite?
  - Infinite sample complexity?
  - Next time…

Chernoff Bounds

- Chernoff bound: Suppose $Y_1, \ldots, Y_m$ are i.i.d. random variables taking values in $\{0, 1\}$ such that $E_p[Y_i] = y$. For $\epsilon > 0$,

$$p\left(y - \frac{1}{m}\sum_i Y_i \geq \epsilon\right) \leq e^{-2m\epsilon^2}$$

- Applying this to $1 - C_1^h, \ldots, 1 - C_m^h$ gives

$$p\left(\epsilon_h - \frac{1}{m}\sum_i (1 - C_i^h) \geq \epsilon\right) \leq e^{-2m\epsilon^2}$$

Chernoff Bounds

- Chernoff bound: Suppose $Y_1, \ldots, Y_m$ are i.i.d. random variables taking values in $\{0, 1\}$ such that $E_p[Y_i] = y$. For $\epsilon > 0$,

$$p\left(y - \frac{1}{m}\sum_i Y_i \geq \epsilon\right) \leq e^{-2m\epsilon^2}$$

- Applying this to $1 - C_1^h, \ldots, 1 - C_m^h$ gives

$$p\left(\epsilon_h - \frac{1}{m}\sum_i (1 - C_i^h) \geq \epsilon\right) \leq e^{-2m\epsilon^2}$$

This is the training error

PAC Bounds

- **Theorem:** For a finite hypothesis space H finite, m i.i.d. samples, and $0 < \epsilon < 1$, the probability that true error of any of the best classifiers (i.e., lowest training error) is larger than its training error plus ϵ is at most $|H|e^{-2m\epsilon^2}$
  – Sample complexity (for desired $\delta \geq |H|e^{-2m\epsilon^2}$)

$$m \geq \left(\log|H| + \log\frac{1}{\delta}\right)/2\epsilon^2$$

PAC Bounds

- If we require that the previous error is bounded above by δ, then with probability $(1-\delta)$, for all $h \in H$

$$\underbrace{\epsilon_h \le \epsilon_h^{train}}_{\text{“bias”}} + \underbrace{\sqrt{\frac{1}{2m}\left(\log|H| + \log\frac{1}{\delta}\right)}}_{\text{“variance”}}$$

– For small $|H|$
  - High bias (may not be enough hypotheses to choose from)
  - Low variance

PAC Bounds

- If we require that the previous error is bounded above by δ, then with probability $(1-\delta)$, for all $h \in H$

$$\underbrace{\epsilon_h \le \epsilon_h^{train}}_{\text{"bias"}} + \underbrace{\sqrt{\frac{1}{2m}\left(\log|H| + \log\frac{1}{\delta}\right)}}_{\text{"variance"}}$$

“bias” “variance”

  - For large $|H|$
    - Low bias (lots of good hypotheses)
    - High variance

PAC Learning

- Given:
  - Set of data X
  - Hypothesis space H
  - Set of target concepts C
  - Training instances from unknown probability distribution over X of the form $(x, c(x))$
- Goal:
  - Learn the target concept $c \in \mathrm{C}$

PAC Learning

- Given:
  - A concept class C over n instances from the set X
  - A learner L with hypothesis space H
  - Two constants, $\epsilon, \delta \in (0, \frac{1}{2})$
- C is said to be PAC learnable by L using H iff for all distributions over X, learner L by sampling n instances, will with probability at least $1 - \delta$ output a hypothesis $h \in \mathrm{H}$ such that
  - $\epsilon_h \leq \epsilon$
  - Running time is polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}, n, size(c)$

PAC Learning

- PAC concerned about computational resources required for learning
  - In practice, we are often only concerned about the number of training examples required
  - The two are related
    - The computational limitation also imposes a polynomial constraint on the training set size, since a learner can process at most polynomial data in polynomial time
    - The learner must visit each example at least once