Learning Theory I

CSci 5525: Machine Learning

Instructor: Nicholas Johnson

November 3, 2020

Announcements

- HW3 due in 1 week (Nov 10)
- Project progress report due in 2 weeks (Nov 17)

Early Term Survey Results

- 70% of total enrolled responded
- Question 1: How are the lectures? What could be done to make them better? Are the videos working well?
  - Most think lectures are going well and enjoy having the videos
  - Action item: No changes

Early Term Survey Results

- Question 2: How are the open QA sessions and office hours? What could be done to make them better?
  - Most like having both QA session and office hours
  - QA sessions overlap with watching lecture
  - Office hour appointments are too short
  - Action items: Post lecture videos earlier (around 10 AM CDT), shorten QA session to 11:30-12:30 PM CDT, increase office hours 5-6:30 PM CDT and appointment times to 15 minutes

Early Term Survey Results

- Question 3: Are you getting enough time to speak with me and the TAs? If not, what suggestions do you have to make this better (for example, more office hours, longer office hour appointments, other type of meeting format, etc.)?
  - Most are getting enough time to speak with me and the TAs
  - Action items: Previous question action items apply here

Early Term Survey Results

- Question 4: How can the course be modified to get closer to the quality/experience of in-person courses? What can I do so that you're getting what you want out of the course in general?
  - Pace of course too fast
  - Homework needs to be more clear
  - More time spent on equations and showing working code
  - Action items: Schedule changes to spread out assignments better, focus on clarifying homeworks, spend more time on equations and show more code examples

Early Term Survey Results

- Question 5: Any other general comments? (For example, what else could be done to make the course better for you? More examples of working code in lectures? More rigorous mathematical proofs? etc.).
  - More time spent on theory/math proofs
  - More time spent showing code
  - Homeworks are too hard/time consuming
  - Grading concerns
  - Action items: spend more time on math proofs and show working code, reduce homework length

Early Term Survey Results

- Action Item Summary:
  - Post lecture videos earlier (around 10 AM CDT)
  - Shorten QA session to 11:30-12:30 PM CDT
  - Increase office hours 5-6:30 PM CDT and appointment times to 15 minutes
  - Schedule changes to spread out assignments better (already done)
  - Focus on clarifying homeworks
  - Spend more time on math/proofs and working code
  - Reduce homework length

Learning Theory

- Seen many methods for supervised learning
  - Linear regression, logistic regression, SVM, perceptron, neural networks, etc.
- Mostly discussed solving empirical risk minimization problem
  - Given finite dataset, find model in some class to minimize loss
- Have not really talked about how these learned models will perform on new data/underlying distribution
- Next two lectures will study several theoretical tools for analyzing generalization error of learning algorithms

Bias-Variance Tradeoff Revisited

- Data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn i.i.d. from distribution P over $\mathcal{X} \times \mathcal{Y}$
- Consider regression setting with $y \in \mathbb{R}$ and squared loss
- Expected label for x is

$$\bar{y}(x) = E_P[Y|X = x] = \int_y yP(y|x)dy$$

- If P is known then to minimize squared loss given x predict $\bar{y}(x)$
- This is the Bayes optimal predictor
- Now consider algorithm that predicts f_D
- Expected test error for f_D

$$E_{(x,y)\sim P}[(f_D(x) - y)^2] = \int_x \int_y (f_D(x) - y)^2 P(x, y)dydx$$

Bias-Variance Tradeoff Revisited

- Should also take into account that dataset D is drawn randomly from P and randomness of predictor f_D
- Expected classifier (for algorithm $\mathcal{A}$)

$$\bar{f} = E_{D \sim P^n}[f_D^{\mathcal{A}}] = \int_D f_D P(D) dD$$

- Can use fact that f_D is random variable to compute expected test error given $\mathcal{A}$
- Expected test error given $\mathcal{A}$

$$E_{\substack{(x,y) \sim P \\ D \sim P^n}}[(f_D(x) - y)^2] = \int_D \int_x \int_y (f_D(x) - y)^2 P(D, x, y)$$

- D are training points and (x, y) pairs are test points
- Quantity measures expected accuracy of algorithm $\mathcal{A}$ - we want to analyze this

Bias-Variance Tradeoff Revisited

- Decomposition of expected test error

$$
\begin{aligned}
& E_{x,y,D}[(f_D(x) - y)^2] \\
&= E_{x,y,D}\left[((f_D(x) - \bar{f}(x)) + (\bar{f}(x) - y)^2\right] \\
&= E_{x,D}\left[(f_D(x) - \bar{f}(x))^2\right] + 2E_{x,y,D}\left[(f_D(x) - \bar{f}(x))(\bar{f}(x) - y)\right] + E_{x,y}\left[(\bar{f}(x) - y)^2\right]
\end{aligned}
$$

- Middle term is 0 so we focus on the first and last terms

$$E_{x,y,D}[(f_D(x) - y)^2] = E_{x,D}\left[(f_D(x) - \bar{f}(x))^2\right] + E_{x,y}\left[(\bar{f}(x) - y)^2\right]$$

variance

- We can break down second term above

Bias-Variance Tradeoff Revisited

- Breaking down the second term we get

$$
\begin{aligned}
& E_{x,y}\left[(\bar{f}(x)-y)^2\right] \\
&= E_{x,y}\left[((\bar{f}(x)-\bar{y}(x))+(\bar{y}(x)-y))^2\right] \\
&= E_{x,y}\left[(\bar{y}(x)-y)^2\right] + E_x\left[(\bar{f}(x)-\bar{y}(x))^2\right] + 2E_{x,y}\left[(\bar{f}(x)-\bar{y}(x))(\bar{y}(x)-y)\right]
\end{aligned}
$$

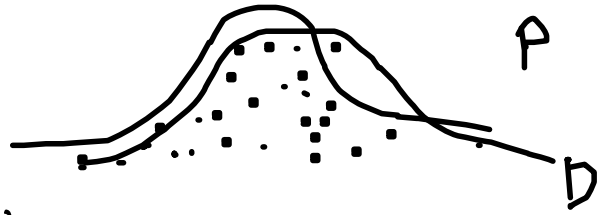
noise; bias^2

- Third term is 0 so we finally get

$$
\begin{aligned}
& E_{x,y,D}[(f_D(x)-y)^2] \\
&= E_{x,D}\left[(f_D(x)-\bar{f}(x))^2\right] + E_{x,y}\left[(\bar{y}(x)-y)^2\right] + E_x\left[(\bar{f}(x)-\bar{y}(x))^2\right]
\end{aligned}
$$

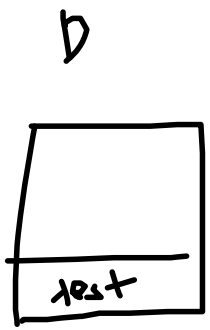
variance; noise; bias^2

Types/Models of Learning



- Basic model:
  - Fixed unknown distribution P over $\mathcal{X} \times \mathcal{Y}$
  - The *true* error rate of a classifier $h : \mathcal{X} \mapsto \mathcal{Y}$

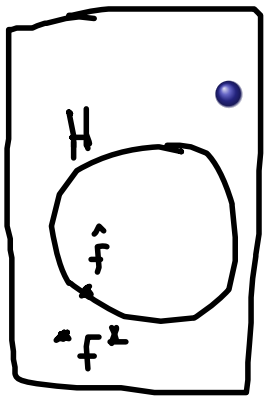
$$\mathcal{R}_P(h) = E_{(\mathbf{x},y)\sim P}[\mathbb{1}(h(\mathbf{x}) \neq y)]$$



dataset

  - The training set has an empirical distribution D
  - The error rate on the training set is

$$\hat{\mathcal{R}}(h) = E_{(\mathbf{x},y)\sim D}[\mathbb{1}(h(\mathbf{x}) \neq y)] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(h(\mathbf{x}) \neq y)$$



- Learning from a Hypothesis Space $\mathcal{H}$
  - Realizable/Consistent learning: The "best function" $f^* \in \mathcal{H}$
  - Agnostic learning: The best function may be outside $\mathcal{H}$

General Setting

- Loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto [0, \infty)$
- Loss on specific example $\ell(h(x), y)$
- Training set $D = (x_1, y_1), \ldots, (x_n, y_n)$
- The (true) risk and empirical risk

$$\mathcal{R}(h) = E_{(x,y)\sim P}[\ell(h(x), y)] \qquad \hat{\mathcal{R}}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i), y_i)$$

$h^* = \operatorname{argmin}_h \mathcal{R}(h)$

predict

- Usually, ℓ is 0-1 loss
- Analysis can be generalized to other losses with some work
- Excess risk of h is $\mathcal{R}(h) - \hat{\mathcal{R}}(h)$

Risk

ERM

- We have seen algorithms to compute $\hat{h} = \min_h \hat{\mathcal{R}}(h)$
- Let h^* be minimizer of true risk then

$$\begin{aligned} \mathcal{R}(\hat{h}) - \mathcal{R}(h^*) &= \hat{\mathcal{R}}(h^*) - \mathcal{R}(h^*) && \text{sampling error} \\ &+ \hat{\mathcal{R}}(\hat{h}) - \hat{\mathcal{R}}(h^*) && \text{approximation error} \\ &+ \mathcal{R}(\hat{h}) - \hat{\mathcal{R}}(\hat{h}) && \text{generalization error} \end{aligned}$$

- How do we bound these?
  - Sampling error is easiest; Use law of large numbers
  - Approximation error depends on how good ERM algorithm is
  - Generalization error is tricky; tempting to apply law of large numbers but $\hat{h}$ is a random variable that depends on data (need to consider complexity of predictor function)

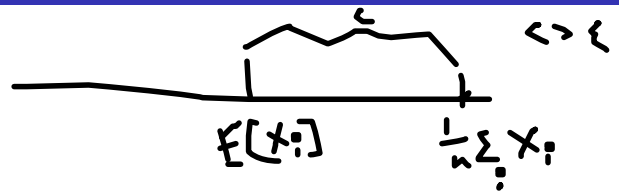
Bounding Sampling Error

- We will use 0-1 loss which gives

$$\mathcal{R}(h) = E_{(x,y)\sim P}[\mathbb{1}(h(x) \neq y)] \qquad \hat{\mathcal{R}}(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(h(x_i) \neq y_i)$$

- How fast does $\hat{\mathcal{R}}(h)$ converge to $\mathcal{R}(h)$ as a function of n? Use concentration inequalities!

Chernoff/Hoeffding's Inequality



- Let $X_1, \ldots, X_n \in [a, b]$ be i.i.d. real-valued random variables. Then

$$P\left(\frac{1}{n}\sum_i X_i - E[X_i] \geq \epsilon\right) \leq \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

- Equivalently, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$

$$E[X] \leq \left(\frac{1}{n}\sum_i X_i\right) + (b-a)\sqrt{\frac{\ln(1/\delta)}{2n}}$$

- Other useful inequalities: Markov's, Bernstein's, McDiarmid's

Bounding Sampling Error

- Let each $X_i = \mathbb{1}[h(x_i) \neq y_i]$ then we get the following
- With probability at least $1 - \delta$

$$\mathcal{R}(h) \leq \hat{\mathcal{R}}(h) + \sqrt{\frac{\log 1/\delta}{2n}}$$

- Bound holds for *any* h, does not hold for all h simultaneously
- A generalization bound should hold for all h simultaneously
- This is called uniform convergence

Finite Hypothesis Spaces

- Consider event $Z_h = \{(x, y) \in D^n : \mathcal{R}(h) > \hat{\mathcal{R}}_{(x,y)}(h) + \epsilon\}$
- Samples of size n give empirical risk ϵ more than true risk
- By union bound over $\mathcal{H} = \{h_1, \ldots, h_m\}$

$$P\left(\bigcup_h Z_h\right) \leq \sum_{i=1}^{m} P(Z_{h_i}) = m \exp(-2n\epsilon^2)$$

- Given 'tolerance' δ, want $m \exp(-2n\epsilon^2) \leq \delta$
- With probability at least $1 - \delta$

$$\mathcal{R}(h) \leq \hat{\mathcal{R}}(h) + \sqrt{\frac{\log |\mathcal{H}| + \log 1/\delta}{2n}}$$

- Union bound for general function classes (i.e., infinite-sized function classes)?
- Preview: replace $\log(|\mathcal{H}|)$ with complexity measure of $\mathcal{H}$