The Bias-Complexity Tradeoff

Lecture 7

Last Time

- ***Probably approximately correct (PAC) learnability*** is a property of a hypothesis class $\mathcal{H}$. If it holds, there's some function that gives a number of i.i.d. training examples m that are sufficient to guarantee that $L_{\mathcal{D}}(h_S) \leq \epsilon$ with probability at least $1 - \delta$ (for arbitrary ϵ and δ, and some algorithm)

- We've shown that any finite, realizable $\mathcal{H}$ is PAC learnable via ERM with 0-1 loss, with sample complexity

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil$$

- Textbook: chapters 2.3, 3

This Class

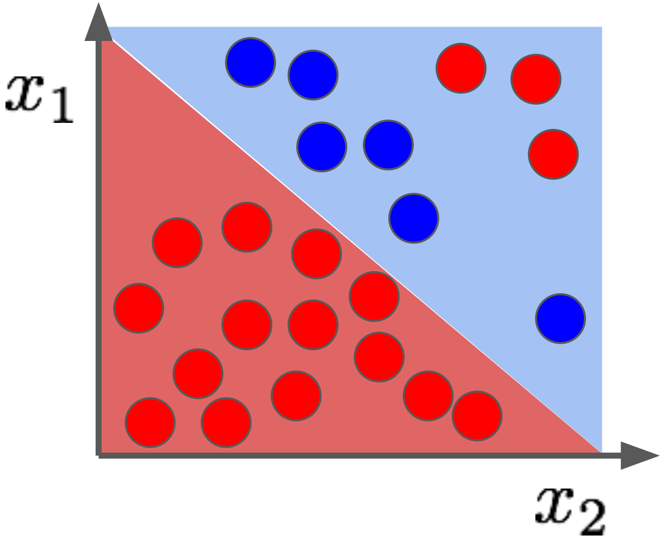
- Can we hope to find a universal learner that makes all hypothesis classes PAC learnable?

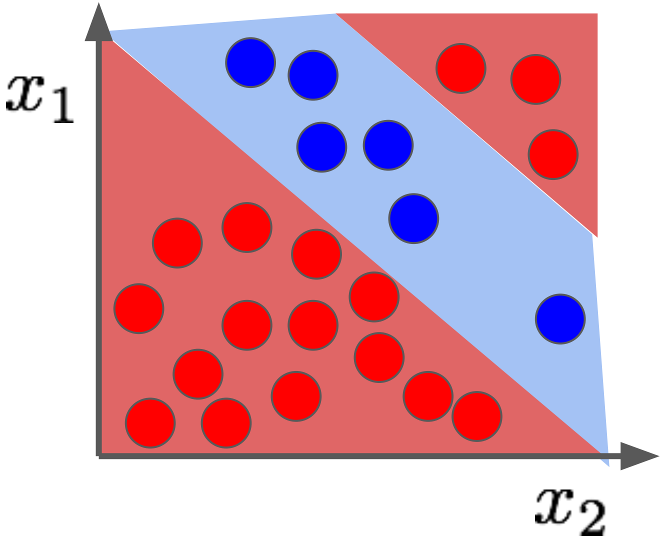
- If not, what tradeoffs must we make when selecting a learning algorithm?

Textbook: chapter 5

Motivation

Which Would You Choose?





Do We Have to Choose?

- Intersection of halfspaces might overfit, i.e., $L_{\mathcal{D}}(h_S)$ is large relative to best possible hypothesis in class

- But on another problem, where $\mathcal{D}$ is actually defined by an intersection of halfspaces, it'd be great!

- Does an algorithm exist that would successfully learn in all cases?

The No-Free-Lunch Theorem

The No-Free-Lunch Theorem

- For every learning algorithm for binary classification with 0-1 loss, there exists a task on which it fails

- Even though that task can be successfully learned by another algorithm

Formal Statement

THEOREM 5.1 (No-Free-Lunch) *Let A be any learning algorithm for the task of binary classification with respect to the $0-1$ loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that:*

1. *There exists a function $f : \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$.*
2. *With probability of at least $1/7$ over the choice of $S \sim \mathcal{D}^m$ we have that $L_{\mathcal{D}}(A(S)) \geq 1/8$.*

Understanding Machine Learning.
Shalev-Shwartz and Ben-David, 2014.

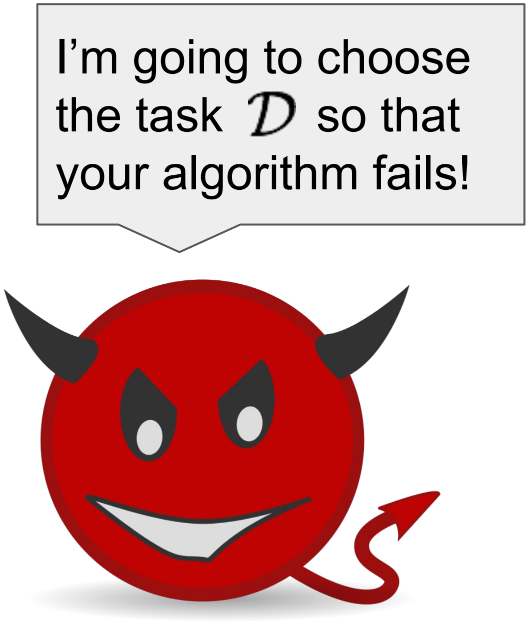
Proof Intuition

- Let C be a subset of $\mathcal{X}$ of size $2m$
- Any learning algorithm A that only observes half of the examples in C has no information about the other half
- There always exists a high probability possible world where A makes a lot of mistakes on the other half
- Full proof in section 5.1

Intuition: Adversary that wants learning to fail

- After you pick a learning algorithm and a training set size, an “adversary” chooses the task so tha $\mathcal{D}$ $L_{\mathcal{D}}(A(S))$ is high with high probability

- Not really how learning works (usually), but useful way to think about proving the existence of such a task $\mathcal{D}$



Example: Cute or Not?

Training Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Spider	**T**	**F**	**F**	**T**
Jellyfish	**F**	**F**	**F**	**T**
Shark	**F**	**T**	**T**	**F**

Test Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Cow	**T**	**T**	**F**	
Skunk	**T**	**T**	**T**	

If you say both are cute...

then only animals that do not have 2 eyes and not sharp teeth are cute!

Training Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Spider	**T**	**F**	**F**	**T**
Jellyfish	**F**	**F**	**F**	**T**
Shark	**F**	**T**	**T**	**F**

Test Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Cow	**T**	**T**	**F**	**F**
Skunk	**T**	**T**	**T**	**F**

If you say only cows are cute...

then only animals that do not have 2 eyes are cute, unless they are also furry and have sharp teeth!

Training Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Spider	T	F	F	T
Jellyfish	F	F	F	T
Shark	F	T	T	F
Test Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Cow	T	T	F	F
Skunk	T	T	T	T

If you say only skunks are cute...

then only animals that do not have sharp teeth are cute!

Training Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Spider	T	F	F	T
Jellyfish	F	F	F	T
Shark	F	T	T	F

Test Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Cow	T	T	F	T
Skunk	T	T	T	F

If you say neither are cute...

then only animals with sharp teeth are cute, unless they are furry and have 2 eyes!

Training Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Spider	**T**	**F**	**F**	**T**
Jellyfish	**F**	**F**	**F**	**T**
Shark	**F**	**T**	**T**	**F**

Test Data	Furry?	Two Eyes?	Sharp Teeth?	**Cute?**
Cow	**T**	**T**	**F**	**F**
Skunk	**T**	**T**	**T**	**F**

Takeaways

- In this example, for any hypothesis you pick, an adversary can choose data so that loss on training data is 0 and loss on test data is 1

- In general, if we see less than half of the possible examples, then we can make true loss high with high probability by choosing $\mathcal{D}$ appropriately, *even if there exists a function that classifies everything perfectly*

Relationship to PAC Learning

Relationship to PAC Learning

- If $m < |\mathcal{X}|/2$, then there is at least half of the possible examples that we have no information about

- A lower bound on the sample complexity of PAC learning for binary classification and 0-1 loss:

$$\frac{|\mathcal{X}|}{2} \leq m_{\mathcal{H}}\left(\frac{1}{8}, \frac{1}{7}\right)$$

Relationship to PAC Learning

Note that

$$\frac{|\mathcal{X}|}{2} \leq m_{\mathcal{H}}\left(\frac{1}{8}, \frac{1}{7}\right)$$

does not contradict our upper bound

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil$$

because for the latter we assumed realizability

Relationship to PAC Learning

COROLLARY 5.2 *Let $\mathcal{X}$ be an infinite domain set and let $\mathcal{H}$ be the set of all functions from $\mathcal{X}$ to $\{0, 1\}$. Then, $\mathcal{H}$ is not PAC learnable.*

Understanding Machine Learning.
Shalev-Shwartz and Ben-David, 2014.

Proof (By Contradiction)

- Assume that such an $\mathcal{H}$ is PAC learnable, and choose some $\epsilon < 1/8$ and $\delta < 1/7$ to use as max. error and max. probability of having more error

- Since we assumed $\mathcal{H}$ is PAC learnable, there must be an algorithm A and an integer $m = m(\epsilon, \delta)$ such that for every $\mathcal{D}$, if there exists f such that $L_{\mathcal{D}}(f) = 0$, then with probability greater than $1 - \delta$, $L_{\mathcal{D}}(A(S)) \leq \epsilon$

- However, by the No-Free-Lunch theorem, since $|\mathcal{X}| > 2m$, there exists $\mathcal{D}$ such that with probability greater than 1/7, $L_{\mathcal{D}}(A(S)) > 1/8$, which is the desired contradiction

Question

We’ll Have to Wait for the Question

The Need for Prior Knowledge

- As we’ve also seen informally, we need to reduce $\mathcal{H}$ using prior knowledge

- Our choice of $\mathcal{H}$ captures our beliefs about how the observed examples could relate to the unobserved ones, also called our ***inductive bias***

- Example: halfspace hypothesis class captures assumption that increasing meal price can only increase or decrease probability that a meal is tasty

Error Decomposition

Error

- What is error?
- The true error is the expected loss on the data distribution
- Recall: for 0-1 loss, it is probability that hypothesis does not predict the correct label on a random data point generated by the underlying distribution

Decomposing Error

Can decompose ERM error into two different categories:

- Approximation error (bias, quality of prior knowledge) ϵ_{app}
- Estimation error (overfitting) ϵ_{est}

$$L_D(h_S) = \epsilon_{app} + \epsilon_{est}$$

Approximation Error (Bias)

- The minimum risk achievable by a predictor in the hypothesis class
- Measures how much risk we have because we restrict ourselves to a specific class or how much inductive bias we have
- Under the realizability assumption, the approximation error is zero. In general, the approximation error can be large

$$\epsilon_{app} = \min_{h \in H} L_D(h)$$

Estimation Error

- The difference between the approximation error and the error achieved by the ERM predictor.

- The quality of this estimation depends on the training set size and on the size, or complexity, of the hypothesis class. For a finite hypothesis class, ε_{est} increases (logarithmically) with |H| and decreases with m.

$$\epsilon_{est} = L_D(h_S) - \epsilon_{app}$$

The Bias-Complexity Tradeoff

The Bias-Complexity Tradeoff

- One needs to balance approximation and estimation error to pick a good hypothesis class *H*

- Choosing *H* to be a very rich class decreases the approximation error but might increase the estimation error (overfitting)

- Choosing *H* to be a very limited class reduces the estimation error but might increase the approximation error (underfitting)

The Bias-Complexity Tradeoff

More Complexity →

← **More Bias**

- Higher approximation error
- Possible underfitting

- Higher estimation error
- Possible overfitting

Question

We'll Have to Wait for the Question

Example: Google Flu Trends

- Used search trends to predict flu epidemics in 25 different countries
- Paper reported that model predicted outbreaks up to 10 days before CDC models
- Massively overestimated flu outbreaks and missed others
- What could have caused such a significant difference between testing and live deployment?

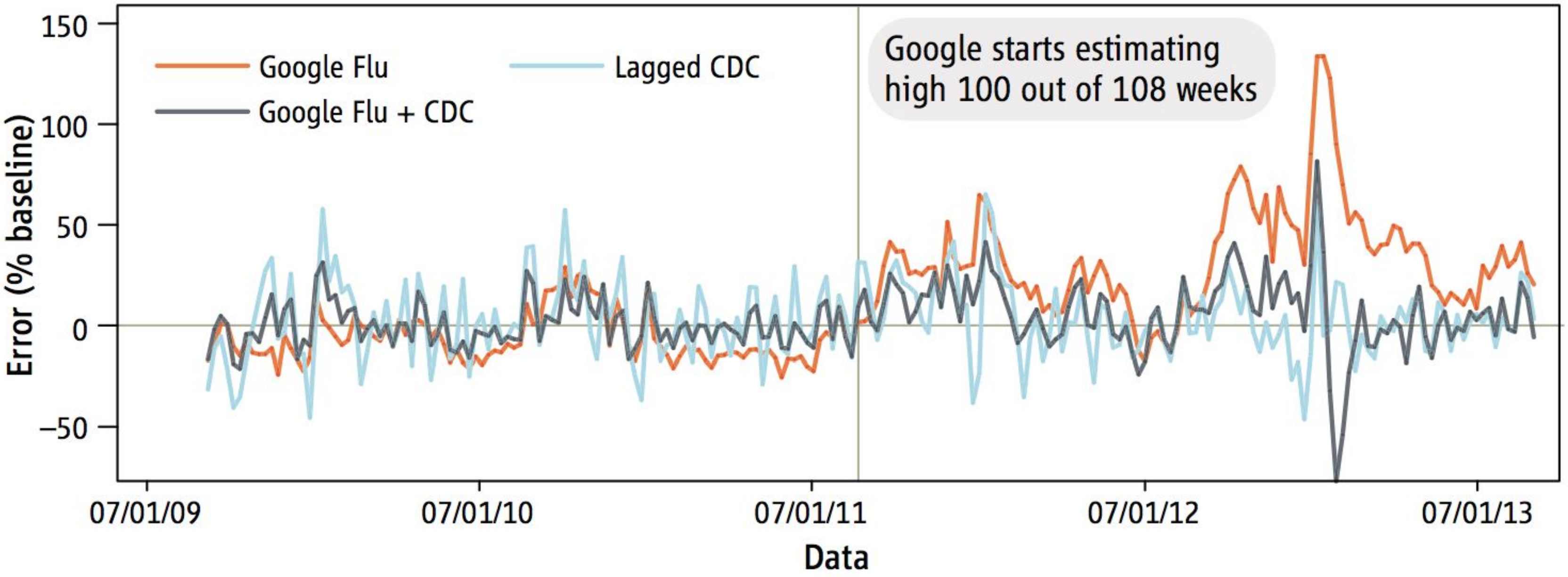
FINAL FINAL

BIG DATA

The Parable of Google Flu: Traps in Big Data Analysis

David Lazer,[1,2*] Ryan Kennedy,[1,3,4] Gary King,[3] Alessandro Vespignani[3,5,6]

Big Data Hubris



The Most Important Things

- The ***no-free-lunch theorem*** tells us that there is no universal learning algorithm that will work best on all problems.
- Further, for every algorithm, there is a problem it fails on, even though another succeeds
- Instead, for every learning problem we must balance the bias-complexity tradeoff using prior knowledge
- Textbook: chapter 5

Next Time

- How do we balance the bias-complexity tradeoff in practice?
- Textbook: chapters 11.0, 11.2, 11.3, 13.0, 13.1, 13.4