Realizable Learning is All You Need

Talk by: Gaurav Mahajan
(UCSD)

Joint with:


Max Hopkins


Daniel Kane


Shachar Lovett

Overview

1. Background
   - Realizable PAC Learning
   - Agnostic Learning
   - Realizable $\iff$ Agnostic Learning
2. The Reduction
   - Algorithm and Analysis
   - Application: Distribution Dependent Learning
   - Application: General Loss Functions
3. Beyond Agnostic Learning
   - Property Generalization
   - Application: Semi-Private Learning
4. Open Problems!!

Table of Contents

Learning (Setup)

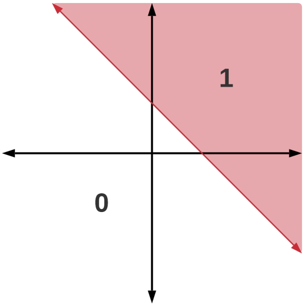
- Let X be a set (e.g. $\mathbb{R}^d$)

Learning (Setup)

- Let X be a set (e.g. $\mathbb{R}^d$)
- Let H be a family of binary classifiers (e.g. halfspaces)

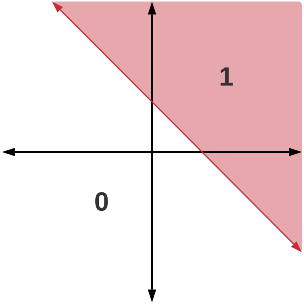
Learning (Setup)

- Let X be a set (e.g. $\mathbb{R}^d$)
- Let H be a family of binary classifiers (e.g. halfspaces)



Learning (Setup)

- Let X be a set (e.g. $\mathbb{R}^d$)
- Let H be a family of binary classifiers (e.g. halfspaces)



- We will be interested in the "learnability" of classes (X, H)
  - Given random labeled samples $(x, h(x))$, can we identify h?

Probably Approximately Correct (PAC) Learning

- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:

Probably Approximately Correct (PAC) Learning

- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$

Probably Approximately Correct (PAC) Learning

- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, h(x))$ with $x \sim D$

Probably Approximately Correct (PAC) Learning

- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, h(x))$ with $x \sim D$
  3. Based on received samples S, $\mathcal{L}$ outputs a guess for h

Probably Approximately Correct (PAC) Learning

- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, h(x))$ with $x \sim D$
  3. Based on received samples S, $\mathcal{L}$ outputs a guess for h

- $\mathcal{L}$ **wins the game** if their output $\mathcal{L}(S)$ is close to h:

Probably Approximately Correct (PAC) Learning

- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, h(x))$ with $x \sim D$
  3. Based on received samples S, $\mathcal{L}$ outputs a guess for h

- $\mathcal{L}$ **wins the game** if their output $\mathcal{L}(S)$ is close to h:

- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of D, h

PAC Learning (Formal)

- Fix 'accuracy' and 'confidence' parameters $\varepsilon, \delta > 0$
- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, h(x))$ with $x \sim D$
  3. Based on received samples S, $\mathcal{L}$ outputs a guess for h
- $\mathcal{L}$ **wins the game** if their output $\mathcal{L}(S)$ is close to h:
- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of D, h:

PAC Learning (Formal)

- Fix 'accuracy' and 'confidence' parameters $\varepsilon, \delta > 0$
- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, h(x))$ with $x \sim D$
  3. Based on received samples S, $\mathcal{L}$ outputs a guess for h

- $\mathcal{L}$ **wins the game** if their output $\mathcal{L}(S)$ is close to h:

$$\text{err}_{D,h}(\mathcal{L}(S)) := \Pr_{x \sim D}[\mathcal{L}(S)(x) \neq h(x)] \leq \varepsilon$$

- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of D, h:

PAC Learning (Formal)

- Fix 'accuracy' and 'confidence' parameters $\varepsilon, \delta > 0$
- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, h(x))$ with $x \sim D$
  3. Based on received samples S, $\mathcal{L}$ outputs a guess for h

- $\mathcal{L}$ **wins the game** if their output $\mathcal{L}(S)$ is close to h:

$$\mathsf{err}_{D,h}(\mathcal{L}(S)) := \Pr_{x \sim D}[\mathcal{L}(S)(x) \neq h(x)] \leq \varepsilon$$

- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of D, h:

$$\forall D, h : \ \Pr_{S}[\mathsf{err}_{D,h}(\mathcal{L}(S)) \leq \varepsilon] \geq 1 - \delta$$

PAC Learning (Formal)

- Fix 'accuracy' and 'confidence' parameters $\varepsilon, \delta > 0$
- PAC-learning is a **game** between a "Learner" $\mathcal{L}$ and "Adversary" $\mathcal{A}$:
  1. First, $\mathcal{A}$ secretly picks a distribution D over X, and $h \in H$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, h(x))$ with $x \sim D$
  3. Based on received samples S, $\mathcal{L}$ outputs a guess for h

- $\mathcal{L}$ **wins the game** if their output $\mathcal{L}(S)$ is close to h:

$$\text{err}_{D,h}(\mathcal{L}(S)) := \Pr_{x \sim D}[\mathcal{L}(S)(x) \neq h(x)] \leq \varepsilon$$

- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of D, h:

$$\forall D, h : \ \Pr_{S}[\text{err}_{D,h}(\mathcal{L}(S)) \leq \varepsilon] \geq 1 - \delta$$

Definition (Realizable PAC-Learning)

(X, H) is Realizably learnable with "sample complexity" $n(\varepsilon, \delta)$ if $\forall \varepsilon, \delta > 0$, $\mathcal{L}$ has a winning strategy using at most $n(\varepsilon, \delta)$ samples

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H
- A more realistic model drops this assumption:

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H
- A more realistic model drops this assumption:
  - The adversary can pick any labeling at all...
  - but the learner only needs to be close to the best hypothesis in H

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H
- A more realistic model drops this assumption:
  - The adversary can pick any labeling at all...
  - but the learner only needs to be close to the best hypothesis in H

- This model is called "Agnostic" learning:

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H
- A more realistic model drops this assumption:
  - The adversary can pick any labeling at all...
  - but the learner only needs to be close to the best hypothesis in H

- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0, 1\}$

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H
- A more realistic model drops this assumption:
  - The adversary can pick any labeling at all...
  - but the learner only needs to be close to the best hypothesis in H

- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0, 1\}$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, y) \sim \bar{D}$

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H
- A more realistic model drops this assumption:
  - The adversary can pick any labeling at all...
  - but the learner only needs to be close to the best hypothesis in H

- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0,1\}$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, y) \sim \bar{D}$
  3. Based on the received sample S, $\mathcal{L}$ outputs a guess for closest in H to $\bar{D}$

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H
- A more realistic model drops this assumption:
  - The adversary can pick any labeling at all...
  - but the learner only needs to be close to the best hypothesis in H

- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0,1\}$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x,y) \sim \bar{D}$
  3. Based on the received sample S, $\mathcal{L}$ outputs a guess for closest in H to $\bar{D}$

- $\mathcal{L}$ **wins** if their output $\mathcal{L}(S)$ is a good guess for $\bar{D}$

Agnostic Learning (Informal)

- Realizable learning forces the adversary to pick from H
- A more realistic model drops this assumption:
  - The adversary can pick any labeling at all...
  - but the learner only needs to be close to the best hypothesis in H

- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0,1\}$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x,y) \sim \bar{D}$
  3. Based on the received sample S, $\mathcal{L}$ outputs a guess for closest in H to $\bar{D}$

- $\mathcal{L}$ **wins** if their output $\mathcal{L}(S)$ is a good guess for $\bar{D}$

- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of $\bar{D}$

Agnostic Learning (Formalized)

- Fix an 'accuracy' and 'confidence' parameters $\varepsilon, \delta > 0$
- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0,1\}$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x, y) \sim \bar{D}$
  3. Based on the received sample S, $\mathcal{L}$ outputs a guess for closest in H to $\bar{D}$
- $\mathcal{L}$ **wins** if their output $\mathcal{L}(S)$ is a good guess for $\bar{D}$
- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of $\bar{D}$

Agnostic Learning (Formalized)

- Fix an 'accuracy' and 'confidence' parameters $\varepsilon, \delta > 0$
- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0,1\}$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x,y) \sim \bar{D}$
  3. Based on the received sample S, $\mathcal{L}$ outputs a guess for closest in H to $\bar{D}$
- $\mathcal{L}$ **wins** if their output $\mathcal{L}(S)$ is a good guess for $\bar{D}$:

$$\mathsf{err}_{\bar{D}}(\mathcal{L}(S)) := \Pr_{(x,y)\sim\bar{D}}[\mathcal{L}(S)(x) \neq y] \leq OPT + \varepsilon$$

$$OPT := \min_{h \in H}\{\mathsf{err}_{\bar{D}}(h)\}$$

- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of $\bar{D}$

Agnostic Learning (Formalized)

- Fix an 'accuracy' and 'confidence' parameters $\varepsilon, \delta > 0$
- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0,1\}$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x,y) \sim \bar{D}$
  3. Based on the received sample S, $\mathcal{L}$ outputs a guess for closest in H to $\bar{D}$

- $\mathcal{L}$ **wins** if their output $\mathcal{L}(S)$ is a good guess for $\bar{D}$:

$$\mathsf{err}_{\bar{D}}(\mathcal{L}(S)) := \Pr_{(x,y)\sim\bar{D}}[\mathcal{L}(S)(x) \neq y] \leq OPT + \varepsilon$$

$$OPT := \min_{h \in H}\{\mathsf{err}_{\bar{D}}(h)\}$$

- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of $\bar{D}$

$$\forall \bar{D} : \Pr_{S\sim\bar{D}}[\mathsf{err}_{\bar{D}}(\mathcal{L}(S)) \leq OPT + \varepsilon] \geq 1 - \delta$$

Agnostic Learning (Formalized)

- Fix an 'accuracy' and 'confidence' parameters $\varepsilon, \delta > 0$
- This model is called "Agnostic" learning:
  1. First, $\mathcal{A}$ picks a Joint Distribution $\bar{D}$ over $X \times \{0,1\}$
  2. Second, $\mathcal{L}$ draws *labeled* samples $(x,y) \sim \bar{D}$
  3. Based on the received sample S, $\mathcal{L}$ outputs a guess for closest in H to $\bar{D}$
- $\mathcal{L}$ **wins** if their output $\mathcal{L}(S)$ is a good guess for $\bar{D}$:

$$\text{err}_{\bar{D}}(\mathcal{L}(S)) := \Pr_{(x,y)\sim\bar{D}}[\mathcal{L}(S)(x) \neq y] \leq OPT + \varepsilon$$

$$OPT := \min_{h \in H}\{\text{err}_{\bar{D}}(h)\}$$

- $\mathcal{L}$ has a **winning strategy** if they win whp for any choice of $\bar{D}$

$$\forall \bar{D} : \Pr_{S\sim\bar{D}}[\text{err}_{\bar{D}}(\mathcal{L}(S)) \leq OPT + \varepsilon] \geq 1 - \delta$$

Definition (Agnostic PAC-Learning)

(X, H) is Agnostically learnable with "sample complexity" $m(\varepsilon, \delta)$ if $\forall \varepsilon, \delta > 0$, $\mathcal{L}$ has a winning strategy using at most $m(\varepsilon, \delta)$ samples

Equivalence of Realizable and Agnostic Learning

- Agnostic learning seems much harder than Realizable learning...

Equivalence of Realizable and Agnostic Learning

- Agnostic learning seems much harder than Realizable learning...
  - Adversary has strictly more power!

Equivalence of Realizable and Agnostic Learning

- Agnostic learning seems much harder than Realizable learning...
  - Adversary has strictly more power!
- But it turns out they're equivalent!

- Agnostic learning seems much harder than Realizable learning...
  - Adversary has strictly more power!
- But it turns out they're equivalent!

Theorem (Blumer, Ehrenfeucht, Haussler, Warmuth '89, Haussler '92)

(X, H) *is Realizably learnable* $\iff$ (X, H) *is Agnostically learnable*

Equivalence of Realizable and Agnostic Learning

- Agnostic learning seems much harder than Realizable learning...
  - Adversary has strictly more power!
- But it turns out they're equivalent!

Theorem (Blumer, Ehrenfeucht, Haussler, Warmuth '89, Haussler '92)

(X, H) *is Realizably learnable* $\iff$ (X, H) *is Agnostically learnable*

- Proof relies on uniform convergence
  - (Empirical error approaches true error for all $h \in H$ *simultaneously*)

Equivalence of Realizable and Agnostic Learning

- Agnostic learning seems much harder than Realizable learning...
  - Adversary has strictly more power!
- But it turns out they're equivalent!

Theorem (Blumer, Ehrenfeucht, Haussler, Warmuth '89, Haussler '92)

(X, H) *is Realizably learnable* $\iff$ (X, H) *is Agnostically learnable*

- Proof relies on uniform convergence
  - (Empirical error approaches true error for all $h \in H$ *simultaneously*)
- Unfortunately, uniform convergence fails beyond the PAC-model
  - e.g. distribution-dependent learning; general loss functions...

Main Question

Is the equivalence of the realizable and agnostic models a fundamental property of learnability?

Main Question

Is the equivalence of the realizable and agnostic models a fundamental property of learnability?

- Despite no uniform convergence, equivalence always seems to hold!
  - Distribution-dependent learning [BI91]
  - Regression [BLW96]
  - Private learning [BNS14]
  - Multi-class learning [DMY16]
  - Robust learning [MHS19]
  - Semi-private learning [ABM19]
  - Private prediction [DF20]
  - Stable learning [DF20]
  - Partial learning [AHHM21]

Main Question

Is the equivalence of the realizable and agnostic models a fundamental property of learnability?

- Despite no uniform convergence, equivalence always seems to hold!
  - Distribution-dependent learning [BI91]
  - Regression [BLW96]
  - Private learning [BNS14]
  - Multi-class learning [DMY16]
  - Robust learning [MHS19]
  - Semi-private learning [ABM19]
  - Private prediction [DF20]
  - Stable learning [DF20]
  - Partial learning [AHHM21]

Can we explain this phenomenon more generally?

Table of Contents

A Direct Reduction in Two Steps

- Let $\mathcal{L}$ be a realizable learner for H on $n(\varepsilon, \delta)$ samples

A Direct Reduction in Two Steps

- Let $\mathcal{L}$ be a realizable learner for H on $n(\varepsilon, \delta)$ samples
- We'll build an agnostic learner for H in two main steps:

A Direct Reduction in Two Steps

- Let $\mathcal{L}$ be a realizable learner for H on $n(\varepsilon, \delta)$ samples
- We'll build an agnostic learner for H in two main steps:
  1. **Step 1: Build a "cover" of H**

A Direct Reduction in Two Steps

- Let $\mathcal{L}$ be a realizable learner for H on $n(\varepsilon, \delta)$ samples
- We'll build an agnostic learner for H in two main steps:
  1. **Step 1: Build a "cover" of H**
     - Using unlabeled samples and the learner $\mathcal{L}$...
     - Construct a small (finite) subset that "approximates" H

A Direct Reduction in Two Steps

- Let $\mathcal{L}$ be a realizable learner for H on $n(\varepsilon, \delta)$ samples
- We'll build an agnostic learner for H in two main steps:

1. **Step 1: Build a "cover" of H**
   - Using unlabeled samples and the learner $\mathcal{L}$...
   - Construct a small (finite) subset that "approximates" H
2. **Step 2: Learn the cover**

A Direct Reduction in Two Steps

- Let $\mathcal{L}$ be a realizable learner for H on $n(\varepsilon, \delta)$ samples
- We'll build an agnostic learner for H in two main steps:
  1. **Step 1: Build a "cover" of H**
     - Using unlabeled samples and the learner $\mathcal{L}$...
     - Construct a small (finite) subset that "approximates" H
  2. **Step 2: Learn the cover**
     - Using labeled samples, output a good hypothesis in the cover

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H) if for every $h \in H$, there exists $h' \in C$ such that

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \; \exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon$$

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \; \exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon$$

- We instead construct a a non-uniform (ε, δ)-cover for (D, H)

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \ \exists h' \in C\colon \mathrm{err}_{D,h}(h') \leq \varepsilon$$

- We instead construct a a non-uniform (ε, δ)-cover for (D, H)
  a finite set C with the following guarantee:

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \ \exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon$$

- We instead construct a a non-uniform (ε, δ)-cover for (D, H)
  a finite set C with the following guarantee:
  for every **fixed** hypothesis $h \in H$,

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \ \exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon$$

- We instead construct a a non-uniform (ε, δ)-cover for (D, H)
  a finite set C with the following guarantee:
  for every **fixed** hypothesis $h \in H$,
  C contains h' close to h with probability $1 - \delta$

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \; \exists h' \in C \colon \mathsf{err}_{D,h}(h') \le \varepsilon$$

- We instead construct a a non-uniform (ε, δ)-cover for (D, H)
  a finite set C with the following guarantee:
  for every **fixed** hypothesis $h \in H$,
  C contains h' close to h with probability $1 - \delta$

$$\forall h \in H \;\; \Pr_{C}[\exists h' \in C \colon \mathsf{err}_{D,h}(h') \le \varepsilon] \ge 1 - \delta \, .$$

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \; \exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon$$

- We instead construct a a non-uniform (ε, δ)-cover for (D, H)
  a finite set C with the following guarantee:
  for every **fixed** hypothesis $h \in H$,
  C contains h' close to h with probability $1 - \delta$

$$\forall h \in H \; \Pr_C[\exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon] \geq 1 - \delta \, .$$

- Note this does **not** mean C is an ε-cover for (D, H) with high probability!

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \; \exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon$$

- We instead construct a a non-uniform (ε, δ)-cover for (D, H)
  a finite set C with the following guarantee:
  for every **fixed** hypothesis $h \in H$,
  C contains h' close to h with probability $1 - \delta$

$$\forall h \in H \;\; \Pr_C[\exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon] \geq 1 - \delta \, .$$

- Note this does **not** mean C is an ε-cover for (D, H) with high probability!
  - C is likely to miss some hypotheses each time

Step 1: Non-Uniform Covering

- A set C is an ε-cover for (D, H)
  if for every $h \in H$, there exists $h' \in C$ such that
  h' is close to h under D

$$\forall h \in H \ \exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon$$

- We instead construct a a non-uniform (ε, δ)-cover for (D, H)
  a finite set C with the following guarantee:
  for every **fixed** hypothesis $h \in H$,
  C contains h' close to h with probability $1 - \delta$

$$\forall h \in H \ \Pr_C[\exists h' \in C \colon \mathsf{err}_{D,h}(h') \leq \varepsilon] \geq 1 - \delta \,.$$

- Note this does **not** mean C is an ε-cover for (D, H) with high probability!
  - C is likely to miss some hypotheses each time
  - Covering all hypotheses **simultaneously** requires additional samples

- Recall Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)

Step 2: Non-Uniform Covering $\rightarrow$ Agnostic Learning

- Recall Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
- Why care about bounded non-uniform covers?
  - For every fixed $h \in H$, C contains h' close to h whp

Step 2: Non-Uniform Covering $\rightarrow$ Agnostic Learning

- Recall Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
- Why care about bounded non-uniform covers?
  - For every fixed $h \in H$, C contains h' close to h whp
  - This is true for h_{OPT}, the optimal hypothesis! ($\text{err}_{\bar{D}}(h_{OPT}) = OPT$)

Step 2: Non-Uniform Covering $\rightarrow$ Agnostic Learning

- Recall Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
- Why care about bounded non-uniform covers?
  - For every fixed $h \in H$, C contains h' close to h whp
  - This is true for h_{OPT}, the optimal hypothesis! ($\text{err}_{\bar{D}}(h_{OPT}) = OPT$)
- In other words, C probably contains h^* close to h_{OPT} satisfying:

$$\Pr_{x \sim \bar{D}_X}[h^*(x) \neq h_{OPT}(x)] \leq \varepsilon/2$$

Step 2: Non-Uniform Covering → Agnostic Learning

- Recall Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
- Why care about bounded non-uniform covers?
  - For every fixed $h \in H$, C contains h' close to h whp
  - This is true for h_{OPT}, the optimal hypothesis! ($\text{err}_{\bar{D}}(h_{OPT}) = OPT$)
- In other words, C probably contains h^* close to h_{OPT} satisfying:

$$\Pr_{x \sim \bar{D}_X}[h^*(x) \neq h_{OPT}(x)] \leq \varepsilon/2 \implies \text{err}_{\bar{D}}(h^*) \leq OPT + \varepsilon/2$$

Step 2: Non-Uniform Covering → Agnostic Learning

- Recall Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
- Why care about bounded non-uniform covers?
  - For every fixed $h \in H$, C contains h' close to h whp
  - This is true for h_{OPT}, the optimal hypothesis! ($\text{err}_{\bar{D}}(h_{OPT}) = OPT$)
- In other words, C probably contains h^* close to h_{OPT} satisfying:

$$\Pr_{x \sim \bar{D}_X}[h^*(x) \neq h_{OPT}(x)] \leq \varepsilon/2 \implies \text{err}_{\bar{D}}(h^*) \leq OPT + \varepsilon/2$$

- Now if we can agnostically learn C to $\varepsilon/2$ error, we get h_{out}:

$$\text{err}_{\bar{D}}(h_{out}) \leq OPT + \varepsilon$$

Step 2: Non-Uniform Covering → Agnostic Learning

- Recall Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
- Why care about bounded non-uniform covers?
  - For every fixed $h \in H$, C contains h' close to h whp
  - This is true for h_{OPT}, the optimal hypothesis! ($\text{err}_{\bar{D}}(h_{OPT}) = OPT$)
- In other words, C probably contains h^* close to h_{OPT} satisfying:

$$\Pr_{x \sim \bar{D}_X}[h^*(x) \neq h_{OPT}(x)] \leq \varepsilon/2 \implies \text{err}_{\bar{D}}(h^*) \leq OPT + \varepsilon/2$$

- Now if we can agnostically learn C to $\varepsilon/2$ error, we get h_{out}:

$$\text{err}_{\bar{D}}(h_{out}) \leq OPT + \varepsilon$$

- Since C is finite, we can use Empirical Risk Minimization:
  - For any fixed $h \in H$, empirical error approaches true error
  - Union bounding over C, true for all $h \in C$ **simultaneously**

Step 1(b): Realizable Learning → Non-Uniform Covering

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?

Step 1(b): Realizable Learning → Non-Uniform Covering

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!
- Consider the following two-step algorithm for constructing C:

Step 1(b): Realizable Learning $\rightarrow$ Non-Uniform Covering

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!
- Consider the following two-step algorithm for constructing C:
  1. Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$

Step 1(b): Realizable Learning → Non-Uniform Covering

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!

- Consider the following two-step algorithm for constructing C:
  1. Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
  2. Run $\mathcal{L}$ on all possible labelings of S_U to get:

$$C := \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

Step 1(b): Realizable Learning → Non-Uniform Covering

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!

- Consider the following two-step algorithm for constructing C:
  1. Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
  2. Run $\mathcal{L}$ on all possible labelings of S_U to get:

$$C := \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

- **Claim 1:** $|C| < \infty$

Step 1(b): Realizable Learning $\rightarrow$ Non-Uniform Covering

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!

- Consider the following two-step algorithm for constructing C:
  1. Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
  2. Run $\mathcal{L}$ on all possible labelings of S_U to get:

$$C := \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

- **Claim 1:** $|C| < \infty$
  - There are at most $2^{|S_U|}$ labelings of S_U

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!

- Consider the following two-step algorithm for constructing C:
  1. Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
  2. Run $\mathcal{L}$ on all possible labelings of S_U to get:

$$C := \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

- **Claim 1:** $|C| < \infty$
  - There are at most $2^{|S_U|}$ labelings of S_U

- **Claim 2:** C is a Non-Uniform $(\varepsilon/2, \delta/2)$-cover:

Step 1(b): Realizable Learning → Non-Uniform Covering

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!

- Consider the following two-step algorithm for constructing C:
  1. Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
  2. Run $\mathcal{L}$ on all possible labelings of S_U to get:

$$C := \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

- **Claim 1:** $|C| < \infty$
  - There are at most $2^{|S_U|}$ labelings of S_U

- **Claim 2:** C is a Non-Uniform $(\varepsilon/2, \delta/2)$-cover:
  - Fix any $h \in H$, realizable learning promises that:

$$\Pr_{S_U \sim \bar{D}_X^n}[\mathrm{err}_{\bar{D},h}(\mathcal{L}(S_U, h(S_U))) \leq \varepsilon/2] \geq 1 - \delta/2$$

Step 1(b): Realizable Learning → Non-Uniform Covering

- So how can we build a bounded non-uniform $(\varepsilon/2, \delta/2)$-cover?
  - It turns out realizable learning is all you need!

- Consider the following two-step algorithm for constructing C:
  1. Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
  2. Run $\mathcal{L}$ on all possible labelings of S_U to get:

$$C := \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

- **Claim 1:** $|C| < \infty$
  - There are at most $2^{|S_U|}$ labelings of S_U

- **Claim 2:** C is a Non-Uniform $(\varepsilon/2, \delta/2)$-cover:
  - Fix any $h \in H$, realizable learning promises that:

$$\Pr_{S_U \sim \bar{D}_X^n}[\mathrm{err}_{\bar{D},h}(\mathcal{L}(S_U, h(S_U))) \leq \varepsilon/2] \geq 1 - \delta/2$$

  - C contains $\mathcal{L}(S_U, h(S_U))$ for each $h \in H$, so we're done!

Putting it All Together

- Let's review the full algorithm:

Putting it All Together

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)

Putting it All Together

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
  - Given a realizable learner $\mathcal{L}$ w/ sample complexity $n(\varepsilon, \delta)$ we...

Putting it All Together

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
  - Given a realizable learner $\mathcal{L}$ w/ sample complexity $n(\varepsilon, \delta)$ we...

1. Step 1: Build a Non-Uniform Cover

Putting it All Together

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
  - Given a realizable learner $\mathcal{L}$ w/ sample complexity $n(\varepsilon, \delta)$ we...

1. Step 1: Build a Non-Uniform Cover
   - Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2, \delta/2)}$

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
  - Given a realizable learner $\mathcal{L}$ w/ sample complexity $n(\varepsilon, \delta)$ we...

1. Step 1: Build a Non-Uniform Cover
   - Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
   - Run $\mathcal{L}$ on all possible labelings of S_U:

$$C \coloneqq \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

Putting it All Together

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
  - Given a realizable learner $\mathcal{L}$ w/ sample complexity $n(\varepsilon, \delta)$ we...

1. Step 1: Build a Non-Uniform Cover
   - Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2, \delta/2)}$
   - Run $\mathcal{L}$ on all possible labelings of S_U:

$$C \coloneqq \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

2. Step 2: Learn the Non-Uniform Cover

Putting it All Together

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
  - Given a realizable learner $\mathcal{L}$ w/ sample complexity $n(\varepsilon, \delta)$ we...

1. Step 1: Build a Non-Uniform Cover
   - Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
   - Run $\mathcal{L}$ on all possible labelings of S_U:

$$C \coloneqq \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

2. Step 2: Learn the Non-Uniform Cover
   - Draw a *labeled* sample $S_L \sim \bar{D}^m$, $m \approx \log(|C|/\delta)/\varepsilon^2$

Putting it All Together

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
  - Given a realizable learner $\mathcal{L}$ w/ sample complexity $n(\varepsilon, \delta)$ we...

1. Step 1: Build a Non-Uniform Cover
   - Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
   - Run $\mathcal{L}$ on all possible labelings of S_U:

$$C := \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

2. Step 2: Learn the Non-Uniform Cover
   - Draw a *labeled* sample $S_L \sim \bar{D}^m$, $m \approx \log(|C|/\delta)/\varepsilon^2$
   - Return hypothesis in C with minimum empirical error over S_L

Putting it All Together

- Let's review the full algorithm:
  - Adversary picks joint distribution $\bar{D}$ (w/ marginal $\bar{D}_X$)
  - Given a realizable learner $\mathcal{L}$ w/ sample complexity $n(\varepsilon, \delta)$ we...

1. Step 1: Build a Non-Uniform Cover
   - Draw an unlabeled sample $S_U \sim \bar{D}_X^{n(\varepsilon/2,\delta/2)}$
   - Run $\mathcal{L}$ on all possible labelings of S_U:

$$C := \{\mathcal{L}(S_U, h(S_U)) : h \in H\}$$

2. Step 2: Learn the Non-Uniform Cover
   - Draw a *labeled* sample $S_L \sim \bar{D}^m$, $m \approx \log(|C|/\delta)/\varepsilon^2$
   - Return hypothesis in C with minimum empirical error over S_L

- Outputs h_{out} satisfies $\mathrm{err}_{\bar{D}}(h_{out}) \leq OPT + \varepsilon$ w/ high probability!

Unifying Framework

- This reduction uses no model-specific properties at all!
  - No reliance on uniform convergence, sample compression, etc.

Unifying Framework

- This reduction uses no model-specific properties at all!
  - No reliance on uniform convergence, sample compression, etc.

- This allows for a unifying framework for many models:
  - Distribution-dependent learning
  - Regression/Lipschitz loss
  - Robust learning
  - Semi-private learning
  - Private prediction
  - Stable learning
  - Partial learning
  - Statistical Query model
  - Fairness

Unifying Framework

- This reduction uses no model-specific properties at all!
  - No reliance on uniform convergence, sample compression, etc.

- This allows for a unifying framework for many models:
  - Distribution-dependent learning
  - Regression/Lipschitz loss
  - Robust learning
  - Semi-private learning
  - Private prediction
  - Stable learning
  - Partial learning
  - Statistical Query model
  - Fairness

These results were mostly known: how about some new applications?

Learning with Arbitrary Distributional Assumptions

- In practice, PAC-Learning is often considered too worst-case
  - One common relaxation is to make distributional assumptions on X

Learning with Arbitrary Distributional Assumptions

- In practice, PAC-Learning is often considered too worst-case
  - One common relaxation is to make distributional assumptions on X
- We can model this generally by the learnability of triples $(\mathscr{D}, X, H)$
  - Here $\mathscr{D}$ is a fixed family of distributions over X
  - The Adversary may only pick distributions from $\mathscr{D}$

Learning with Arbitrary Distributional Assumptions

- In practice, PAC-Learning is often considered too worst-case
  - One common relaxation is to make distributional assumptions on X
- We can model this generally by the learnability of triples $(\mathscr{D}, X, H)$
  - Here $\mathscr{D}$ is a fixed family of distributions over X
  - The Adversary may only pick distributions from $\mathscr{D}$

Uniform Convergence does not characterize learnability in this model.

Learning with Arbitrary Distributional Assumptions

- In practice, PAC-Learning is often considered too worst-case
  - One common relaxation is to make distributional assumptions on X
- We can model this generally by the learnability of triples $(\mathscr{D}, X, H)$
  - Here $\mathscr{D}$ is a fixed family of distributions over X
  - The Adversary may only pick distributions from $\mathscr{D}$

Uniform Convergence does not characterize learnability in this model.

Proposition (Benedek and Itai '91)

There exists a learnable class (D, X, H) over binary labels and classification loss without the uniform convergence property.

Learning with Arbitrary Distributional Assumptions

- In practice, PAC-Learning is often considered too worst-case
  - One common relaxation is to make distributional assumptions on X
- We can model this generally by the learnability of triples $(\mathscr{D}, X, H)$
  - Here $\mathscr{D}$ is a fixed family of distributions over X
  - The Adversary may only pick distributions from $\mathscr{D}$

Uniform Convergence does not characterize learnability in this model.

Proposition (Benedek and Itai '91)

There exists a learnable class (D, X, H) over binary labels and classification loss without the uniform convergence property.

- $X = [0, 1]$, $Y = \{0, 1\}$, D be the uniform distribution over X.

Learning with Arbitrary Distributional Assumptions

- In practice, PAC-Learning is often considered too worst-case
  - One common relaxation is to make distributional assumptions on X
- We can model this generally by the learnability of triples $(\mathscr{D}, X, H)$
  - Here $\mathscr{D}$ is a fixed family of distributions over X
  - The Adversary may only pick distributions from $\mathscr{D}$

Uniform Convergence does not characterize learnability in this model.

Proposition (Benedek and Itai '91)

There exists a learnable class (D, X, H) over binary labels and classification loss without the uniform convergence property.

- $X = [0, 1]$, $Y = \{0, 1\}$, D be the uniform distribution over X.
- $H =$ indicator functions for all finite sets $S \subset X$ and X

Learning with Arbitrary Distributional Assumptions

- In practice, PAC-Learning is often considered too worst-case
  - One common relaxation is to make distributional assumptions on X

- We can model this generally by the learnability of triples $(\mathscr{D}, X, H)$
  - Here $\mathscr{D}$ is a fixed family of distributions over X
  - The Adversary may only pick distributions from $\mathscr{D}$

Uniform Convergence does not characterize learnability in this model.

Proposition (Benedek and Itai '91)

There exists a learnable class (D, X, H) over binary labels and classification loss without the uniform convergence property.

- $X = [0, 1]$, $Y = \{0, 1\}$, D be the uniform distribution over X.
- $H =$ indicator functions for all finite sets $S \subset X$ and X
- Learn in single sample

Learning with Arbitrary Distributional Assumptions

- In practice, PAC-Learning is often considered too worst-case
  - One common relaxation is to make distributional assumptions on X

- We can model this generally by the learnability of triples $(\mathscr{D}, X, H)$
  - Here $\mathscr{D}$ is a fixed family of distributions over X
  - The Adversary may only pick distributions from $\mathscr{D}$

Uniform Convergence does not characterize learnability in this model.

Proposition (Benedek and Itai '91)

There exists a learnable class (D, X, H) over binary labels and classification loss without the uniform convergence property.

- $X = [0, 1]$, $Y = \{0, 1\}$, D be the uniform distribution over X.
- $H =$ indicator functions for all finite sets $S \subset X$ and X
- Learn in single sample
- Bad empirical estimate: hypothesis whose support is given by sample.

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ *when* $\mathscr{D}$ *is family of distributions?*

Open Problem: Learnability with Arbitrary Distributional Assumptions

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ when $\mathscr{D}$ is family of distributions?

- Initial motivation for this work, very little is known!

Open Problem: Learnability with Arbitrary Distributional Assumptions

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ when $\mathscr{D}$ is family of distributions?

- Initial motivation for this work, very little is known!
  - Uniform Convergence does not characterize learnability in this model.

Open Problem: Learnability with Arbitrary Distributional Assumptions

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ when $\mathscr{D}$ is family of distributions?

- Initial motivation for this work, very little is known!
  - Uniform Convergence does not characterize learnability in this model.
  - UBME: Finite ε-cover for (D, H) for every distribution $D \in \mathscr{D}$

Open Problem: Learnability with Arbitrary Distributional Assumptions

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ *when* $\mathscr{D}$ *is family of distributions?*

- Initial motivation for this work, very little is known!
  - Uniform Convergence does not characterize learnability in this model.
  - UBME: Finite ε-cover for (D, H) for every distribution $D \in \mathscr{D}$
    - Necessary for learnability

Open Problem: Learnability with Arbitrary Distributional Assumptions

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ when $\mathscr{D}$ is family of distributions?

- Initial motivation for this work, very little is known!
  - Uniform Convergence does not characterize learnability in this model.
  - UBME: Finite ε-cover for (D, H) for every distribution $D \in \mathscr{D}$
    - Necessary for learnability
    - Sufficient for learnability when $\mathscr{D}$ is set of all distributions or singleton.

Open Problem: Learnability with Arbitrary Distributional Assumptions

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ when $\mathscr{D}$ is family of distributions?

- Initial motivation for this work, very little is known!
  - Uniform Convergence does not characterize learnability in this model.
  - UBME: Finite ε-cover for (D, H) for every distribution $D \in \mathscr{D}$
    - Necessary for learnability
    - Sufficient for learnability when $\mathscr{D}$ is set of all distributions or singleton.
    - However, not sufficient when $\mathscr{D}$ is an arbitrary distribution family [Dudley, Kulkarni, Richardson, Zeitouni '94]

Open Problem: Learnability with Arbitrary Distributional Assumptions

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ when $\mathscr{D}$ is family of distributions?

- Initial motivation for this work, very little is known!
  - Uniform Convergence does not characterize learnability in this model.
  - UBME: Finite ε-cover for (D, H) for every distribution $D \in \mathscr{D}$
    - Necessary for learnability
    - Sufficient for learnability when $\mathscr{D}$ is set of all distributions or singleton.
    - However, not sufficient when $\mathscr{D}$ is an arbitrary distribution family [Dudley, Kulkarni, Richardson, Zeitouni '94]
- Can we use our tools to say more about this model?

Open Problem: Learnability with Arbitrary Distributional Assumptions

Open Problem

What characterizes learnability of $(\mathscr{D}, X, H)$ when $\mathscr{D}$ is family of distributions?

- Initial motivation for this work, very little is known!
  - Uniform Convergence does not characterize learnability in this model.
  - UBME: Finite ε-cover for (D, H) for every distribution $D \in \mathscr{D}$
    - Necessary for learnability
    - Sufficient for learnability when $\mathscr{D}$ is set of all distributions or singleton.
    - However, not sufficient when $\mathscr{D}$ is an arbitrary distribution family [Dudley, Kulkarni, Richardson, Zeitouni '94]
- Can we use our tools to say more about this model?

Our reduction still works perfectly well!

Theorem

$(\mathscr{D}, X, H)$ *is Realizably learnable* $\iff$ $(\mathscr{D}, X, H)$ *is Agnostically learnable*

General Loss Functions

- Many interesting learning problems have more involved notions of loss

General Loss Functions

- Many interesting learning problems have more involved notions of loss
  - In regression, error is measured wrt ℓ_p-loss
  - In robust learning, error is measured wrt robust loss

General Loss Functions

- Many interesting learning problems have more involved notions of loss
  - In regression, error is measured wrt ℓ_p-loss
  - In robust learning, error is measured wrt robust loss

- For classification loss and finite label space, learnability is characterized by uniform convergence.

General Loss Functions

- Many interesting learning problems have more involved notions of loss
  - In regression, error is measured wrt ℓ_p-loss
  - In robust learning, error is measured wrt robust loss

- For classification loss and finite label space, learnability is characterized by uniform convergence.
  - Realizably learnable $\iff$ Agnostically learnable

General Loss Functions

- Many interesting learning problems have more involved notions of loss
  - In regression, error is measured wrt ℓ_p-loss
  - In robust learning, error is measured wrt robust loss
- For classification loss and finite label space, learnability is characterized by uniform convergence.
  - Realizably learnable $\iff$ Agnostically learnable
  - Not true for general loss functions!

General Loss Functions

- Many interesting learning problems have more involved notions of loss
  - In regression, error is measured wrt ℓ_p-loss
  - In robust learning, error is measured wrt robust loss
- For classification loss and finite label space, learnability is characterized by uniform convergence.
  - Realizably learnable $\iff$ Agnostically learnable
  - Not true for general loss functions!

Proposition

There exists a realizably learnable class (X, H, ℓ) over a finite label space Y which is not agnostically learnable.

General Loss Functions

- Many interesting learning problems have more involved notions of loss
  - In regression, error is measured wrt ℓ_p-loss
  - In robust learning, error is measured wrt robust loss
- For classification loss and finite label space, learnability is characterized by uniform convergence.
  - Realizably learnable $\iff$ Agnostically learnable
  - Not true for general loss functions!

Proposition

There exists a realizably learnable class (X, H, ℓ) over a finite label space Y which is not agnostically learnable.

- $X =$ natural numbers, $Y = \{0,1\}^2$.

General Loss Functions

- Many interesting learning problems have more involved notions of loss
  - In regression, error is measured wrt ℓ_p-loss
  - In robust learning, error is measured wrt robust loss
- For classification loss and finite label space, learnability is characterized by uniform convergence.
  - Realizably learnable $\iff$ Agnostically learnable
  - Not true for general loss functions!

Proposition

There exists a realizably learnable class (X, H, ℓ) over a finite label space Y which is not agnostically learnable.

- $X =$ natural numbers, $Y = \{0, 1\}^2$.
- $H =$ all functions which output the first bit as 0.

General Loss Functions

- Many interesting learning problems have more involved notions of loss
  - In regression, error is measured wrt ℓ_p-loss
  - In robust learning, error is measured wrt robust loss

- For classification loss and finite label space, learnability is characterized by uniform convergence.
  - Realizably learnable $\iff$ Agnostically learnable
  - Not true for general loss functions!

Proposition

There exists a realizably learnable class (X, H, ℓ) over a finite label space Y which is not agnostically learnable.

- $X =$ natural numbers, $Y = \{0, 1\}^2$.
- $H =$ all functions which output the first bit as 0.
- loss function $\ell : Y \times Y \to \{0, 1, c\}$ as

$$\ell((b_1, r_1), (b_2, r_2)) = \begin{cases} 0 & b_1 = b_2 \\ 1 & b_1 \neq b_2 \text{ and } r_1 = r_2 \\ c & \text{otherwise.} \end{cases}$$

- Loss function in this example $\ell : Y \times Y \to \mathbb{R}$ does not satisfy
  - Identity of indiscernibles: $\ell(y_1, y_2) = 0 \iff y_1 = y_2$.

- Loss function in this example $\ell : Y \times Y \to \mathbb{R}$ does not satisfy
  - Identity of indiscernibles: $\ell(y_1, y_2) = 0 \iff y_1 = y_2$.

This is the only case where the equivalence fails!

- Loss function in this example $\ell : Y \times Y \to \mathbb{R}$ does not satisfy
  - Identity of indiscernibles: $\ell(y_1, y_2) = 0 \iff y_1 = y_2$.

This is the only case where the equivalence fails!

Theorem

Suppose ℓ satisfies the identity of indiscernibles and Y is a finite label space. Then, $(\mathscr{D}, X, H, \ell)$ is Realizable learnable $\implies$ $(\mathscr{D}, X, H, \ell)$ is agnostically learnable.

General Loss Functions

- Loss function in this example $\ell : Y \times Y \to \mathbb{R}$ does not satisfy
  - Identity of indiscernibles: $\ell(y_1, y_2) = 0 \iff y_1 = y_2$.

This is the only case where the equivalence fails!

Theorem

Suppose ℓ satisfies the identity of indiscernibles and Y is a finite label space. Then, $(\mathscr{D}, X, H, \ell)$ is Realizable learnable $\implies$ $(\mathscr{D}, X, H, \ell)$ is agnostically learnable.

- We prove variants of equivalence for infinite labels:

General Loss Functions

- Loss function in this example $\ell : Y \times Y \to \mathbb{R}$ does not satisfy
  - Identity of indiscernibles: $\ell(y_1, y_2) = 0 \iff y_1 = y_2$.

This is the only case where the equivalence fails!

Theorem

Suppose ℓ satisfies the identity of indiscernibles and Y is a finite label space. Then, $(\mathscr{D}, X, H, \ell)$ is Realizable learnable $\implies$ $(\mathscr{D}, X, H, \ell)$ is agnostically learnable.

- We prove variants of equivalence for infinite labels:
  - Loss functions bounded from above and below

General Loss Functions

- Loss function in this example $\ell : Y \times Y \to \mathbb{R}$ does not satisfy
  - Identity of indiscernibles: $\ell(y_1, y_2) = 0 \iff y_1 = y_2$.

This is the only case where the equivalence fails!

Theorem

Suppose ℓ satisfies the identity of indiscernibles and Y is a finite label space. Then, $(\mathscr{D}, X, H, \ell)$ is Realizable learnable $\implies (\mathscr{D}, X, H, \ell)$ is agnostically learnable.

- We prove variants of equivalence for infinite labels:
  - Loss functions bounded from above and below
  - Loss functions satisfying an approximate triangle inequality

General Loss Functions

- Loss function in this example $\ell : Y \times Y \to \mathbb{R}$ does not satisfy
  - Identity of indiscernibles: $\ell(y_1, y_2) = 0 \iff y_1 = y_2$.

This is the only case where the equivalence fails!

Theorem

Suppose ℓ satisfies the identity of indiscernibles and Y is a finite label space. Then, $(\mathscr{D}, X, H, \ell)$ is Realizable learnable $\implies$ $(\mathscr{D}, X, H, \ell)$ is agnostically learnable.

- We prove variants of equivalence for infinite labels:
  - Loss functions bounded from above and below
  - Loss functions satisfying an approximate triangle inequality
- Basic technique involves *discretizing* before applying reduction

Table of Contents

Property Generalization

- Let P denote a "property of learning algorithm"
  - e.g. noise-tolerance, privacy, robustness

Property Generalization

- Let P denote a "property of learning algorithm"
  - e.g. noise-tolerance, privacy, robustness

Definition (Finitely-Satisfiable Properties)

We call P finitely-satisfiable if there exists a learner with property P for every finite class (X, H)

Property Generalization

- Let P denote a "property of learning algorithm"
  - e.g. noise-tolerance, privacy, robustness

Definition (Finitely-Satisfiable Properties)

We call P finitely-satisfiable if there exists a learner with property P for every finite class (X, H)

- Agnostic learning is a finitely-satisfiable property by ERM

Property Generalization

- Let P denote a "property of learning algorithm"
  - e.g. noise-tolerance, privacy, robustness

Definition (Finitely-Satisfiable Properties)

We call P finitely-satisfiable if there exists a learner with property P for every finite class (X, H)

- Agnostic learning is a finitely-satisfiable property by ERM
- Realizable vs agnostic learning is part of a more general phenomenon:

- Let P denote a "property of learning algorithm"
  - e.g. noise-tolerance, privacy, robustness

Definition (Finitely-Satisfiable Properties)

We call P finitely-satisfiable if there exists a learner with property P for every finite class (X, H)

- Agnostic learning is a finitely-satisfiable property by ERM
- Realizable vs agnostic learning is part of a more general phenomenon:

Informal Meta-Theorem (Property Generalization)

Let P be a finitely-satisfiable property and $\mathcal{L}$ a realizable learner for (X, H). Then $\mathcal{L}$ can be used as a subroutine to build a learner for (X, H) satisfying (a variant of) property P.

Property Generalization

- Let P denote a "property of learning algorithm"
  - e.g. noise-tolerance, privacy, robustness

Definition (Finitely-Satisfiable Properties)

We call P finitely-satisfiable if there exists a learner with property P for every finite class (X, H)

- Agnostic learning is a finitely-satisfiable property by ERM
- Realizable vs agnostic learning is part of a more general phenomenon:

Informal Meta-Theorem (Property Generalization)

Let P be a finitely-satisfiable property and $\mathcal{L}$ a realizable learner for (X, H). Then $\mathcal{L}$ can be used as a subroutine to build a learner for (X, H) satisfying (a variant of) property P.

- Main idea: replace ERM with finite learner for property P

Application: Privacy

- An algorithm is **private** if it is unlikely to change on similar samples

Application: Privacy

- An algorithm is **private** if it is unlikely to change on similar samples
- Privacy is a classic example of a finitely-satisfiable property
  - Any finite class (X, H) can be privately learned [McSherry and Talwar '07]

- An algorithm is **private** if it is unlikely to change on similar samples
- Privacy is a classic example of a finitely-satisfiable property
  - Any finite class (X, H) can be privately learned [McSherry and Talwar '07]
  - Algorithm is called the *exponential mechanism* (EM)

Application: Privacy

- An algorithm is **private** if it is unlikely to change on similar samples
- Privacy is a classic example of a finitely-satisfiable property
  - Any finite class (X, H) can be privately learned [McSherry and Talwar '07]
  - Algorithm is called the *exponential mechanism* (EM)
- If we learn our cover C using EM, we get a **semi-private learner**

Application: Privacy

- An algorithm is **private** if it is unlikely to change on similar samples
- Privacy is a classic example of a finitely-satisfiable property
  - Any finite class (X, H) can be privately learned [McSherry and Talwar '07]
  - Algorithm is called the *exponential mechanism* (EM)
- If we learn our cover C using EM, we get a **semi-private learner**
  - In this model, we have access to *public unlabeled data*

Application: Privacy

- An algorithm is **private** if it is unlikely to change on similar samples
- Privacy is a classic example of a finitely-satisfiable property
  - Any finite class (X, H) can be privately learned [McSherry and Talwar '07]
  - Algorithm is called the *exponential mechanism* (EM)
- If we learn our cover C using EM, we get a **semi-private learner**
  - In this model, we have access to *public unlabeled data*
  - Goal is to minimize amount of public data used (harder to gather)

Application: Privacy

- An algorithm is **private** if it is unlikely to change on similar samples
- Privacy is a classic example of a finitely-satisfiable property
  - Any finite class (X, H) can be privately learned [McSherry and Talwar '07]
  - Algorithm is called the *exponential mechanism* (EM)
- If we learn our cover C using EM, we get a **semi-private learner**
  - In this model, we have access to *public unlabeled data*
  - Goal is to minimize amount of public data used (harder to gather)

Theorem (Realizable $\iff$ Semi-Private Learning)

If (X, H) is Realizably learnable, it is possible to privately learn (X, H) to any ε-accuracy using $O(1/\varepsilon)$ public (unlabeled) samples

Application: Privacy

- An algorithm is **private** if it is unlikely to change on similar samples
- Privacy is a classic example of a finitely-satisfiable property
  - Any finite class (X, H) can be privately learned [McSherry and Talwar '07]
  - Algorithm is called the *exponential mechanism* (EM)
- If we learn our cover C using EM, we get a **semi-private learner**
  - In this model, we have access to *public unlabeled data*
  - Goal is to minimize amount of public data used (harder to gather)

Theorem (Realizable $\iff$ Semi-Private Learning)

If (X, H) is Realizably learnable, it is possible to privately learn (X, H) to any ε-accuracy using $O(1/\varepsilon)$ public (unlabeled) samples

- Result is tight when (X, H) cannot be privately learned [Alon, Bassily, Moran '19]

Application: Privacy

- An algorithm is **private** if it is unlikely to change on similar samples
- Privacy is a classic example of a finitely-satisfiable property
  - Any finite class (X, H) can be privately learned [McSherry and Talwar '07]
  - Algorithm is called the *exponential mechanism* (EM)
- If we learn our cover C using EM, we get a **semi-private learner**
  - In this model, we have access to *public unlabeled data*
  - Goal is to minimize amount of public data used (harder to gather)

Theorem (Realizable $\iff$ Semi-Private Learning)

If (X, H) is Realizably learnable, it is possible to privately learn (X, H) to any ε-accuracy using $O(1/\varepsilon)$ public (unlabeled) samples

- Result is tight when (X, H) cannot be privately learned [Alon, Bassily, Moran '19]
  - Improves over [ABM19] by avoiding uniform convergence
  - Build a "uniform" cover and then learns the cover using EM.

Thanks!

- New blackbox reduction from agnostic to realizable learning
  - Provides unifying framework by avoiding model-specific assumptions
  - New results for models w/ no known characterizations
  - Proof goes through new notion of "non-uniform" covers

Thanks!

- New blackbox reduction from agnostic to realizable learning
  - Provides unifying framework by avoiding model-specific assumptions
  - New results for models w/ no known characterizations
  - Proof goes through new notion of "non-uniform" covers
- Open Problems
  - Characterizing learnability w/ arbitrary distributional assumptions

Thanks!

- New blackbox reduction from agnostic to realizable learning
  - Provides unifying framework by avoiding model-specific assumptions
  - New results for models w/ no known characterizations
  - Proof goes through new notion of "non-uniform" covers
- Open Problems
  - Characterizing learnability w/ arbitrary distributional assumptions
  - There are a few models our techniques can't handle yet...
    e.g. Private learning

Thanks!

- New blackbox reduction from agnostic to realizable learning
  - Provides unifying framework by avoiding model-specific assumptions
  - New results for models w/ no known characterizations
  - Proof goes through new notion of "non-uniform" covers
- Open Problems
  - Characterizing learnability w/ arbitrary distributional assumptions
  - There are a few models our techniques can't handle yet...
    e.g. Private learning
  - Connections between non-uniform covers and other randomized coverings

Max Hopkins

Daniel Kane

Shachar Lovett

Table of Contents

Uniform vs Non-Uniform Covers

- ABM19 builds a "uniform" cover and then learns the cover using EM.

Uniform vs Non-Uniform Covers

- ABM19 builds a "uniform" cover and then learns the cover using EM.

Definition (Uniform (ε, δ)-cover)

A distribution μ over the power set $P(H)$ is a uniform (ε, δ)-cover if $C \sim \mu$ covers H with high probability

$$\Pr_{C \sim \mu}[C \text{ is an } \varepsilon \text{ -cover for } (D, X, H)] \geq 1 - \delta$$

.

Uniform vs Non-Uniform Covers

- ABM19 builds a "uniform" cover and then learns the cover using EM.

Definition (Uniform (ε, δ)-cover)

A distribution μ over the power set $P(H)$ is a uniform (ε, δ)-cover if $C \sim \mu$ covers H with high probability

$$\Pr_{C \sim \mu}[C \text{ is an } \varepsilon \text{ -cover for } (D, X, H)] \geq 1 - \delta$$

.

- Uniform: C covers h for every $h \in H$ simultaneously whp.

Uniform vs Non-Uniform Covers

- ABM19 builds a "uniform" cover and then learns the cover using EM.

Definition (Uniform (ε, δ)-cover)

A distribution μ over the power set $P(H)$ is a uniform (ε, δ)-cover if $C \sim \mu$ covers H with high probability

$$\Pr_{C \sim \mu}[C \text{ is an } \varepsilon \text{ -cover for } (D, X, H)] \geq 1 - \delta$$

.

- Uniform: C covers h for every $h \in H$ simultaneously whp.
- Non-uniform: C covers h whp for every $h \in H$.

Uniform vs Non-Uniform Covers

- ABM19 builds a "uniform" cover and then learns the cover using EM.

Definition (Uniform (ε, δ)-cover)

A distribution μ over the power set $P(H)$ is a uniform (ε, δ)-cover if $C \sim \mu$ covers H with high probability

$$\Pr_{C \sim \mu}[C \text{ is an } \varepsilon \text{ -cover for } (D, X, H)] \geq 1 - \delta$$

.

- Uniform: C covers h for every $h \in H$ simultaneously whp.
- Non-uniform: C covers h whp for every $h \in H$.

Building proper uniform cover is strictly harder than proper non-uniform cover!

Uniform vs Non-Uniform Covers

- ABM19 builds a "uniform" cover and then learns the cover using EM.

Definition (Uniform (ε, δ)-cover)

A distribution μ over the power set $P(H)$ is a uniform (ε, δ)-cover if $C \sim \mu$ covers H with high probability

$$\Pr_{C \sim \mu}[C \text{ is an } \varepsilon \text{ -cover for } (D, X, H)] \geq 1 - \delta$$

.

- Uniform: C covers h for every $h \in H$ simultaneously whp.
- Non-uniform: C covers h whp for every $h \in H$.

Building proper uniform cover is strictly harder than proper non-uniform cover!

Proposition

There exists triple $(\mathscr{D}, X, H)$ such that

- *Proper finite uniform cover requires at least $\Omega(1/\varepsilon \cdot \log(1/\varepsilon))$ samples.*
- *Proper finite non-uniform cover in at most $O(1/\varepsilon)$ samples.*

- ABM19 builds a "uniform" cover and then learns the cover using EM.

Definition (Uniform (ε, δ)-cover)

A distribution μ over the power set $P(H)$ is a uniform (ε, δ)-cover if $C \sim \mu$ covers H with high probability

$$\Pr_{C \sim \mu}[C \text{ is an } \varepsilon \text{ -cover for } (D, X, H)] \geq 1 - \delta$$

.

- Uniform: C covers h for every $h \in H$ simultaneously whp.
- Non-uniform: C covers h whp for every $h \in H$.

Building proper uniform cover is strictly harder than proper non-uniform cover!

Proposition

There exists triple $(\mathscr{D}, X, H)$ such that

- *Proper finite uniform cover requires at least $\Omega(1/\varepsilon \cdot \log(1/\varepsilon))$ samples.*
- *Proper finite non-uniform cover in at most $O(1/\varepsilon)$ samples.*

- Open Problem: Does this gap also exist for improper covers?