Learnability Beyond Uniform Convergence

Shai Shalev-Shwartz

School of CS and Engineering,
The Hebrew University of Jerusalem

"Mathematical and Computational Foundations of Learning Theory",
Dagstuhl 2011

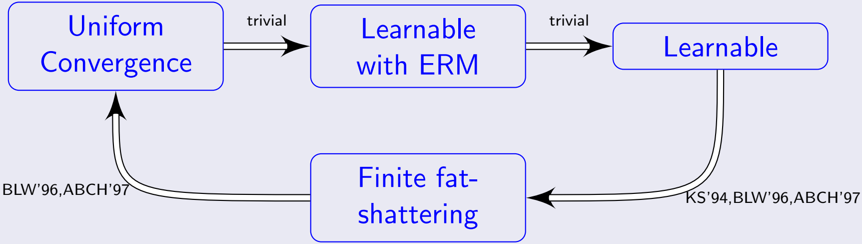
Joint work with:
N. Srebro, O. Shamir, K. Sridharan (COLT'09,JMLR'11)
A. Daniely, S. Sabato, S. Ben-David (COLT'11)

The Fundamental Theorem of Learning Theory

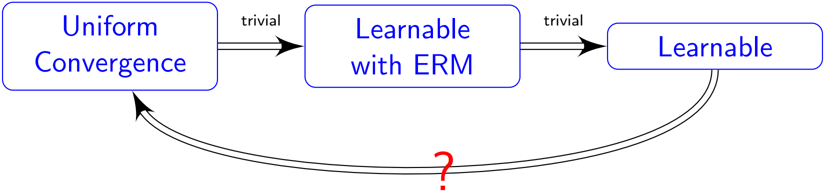
For Binary Classification

The Fundamental Theorem of Learning Theory

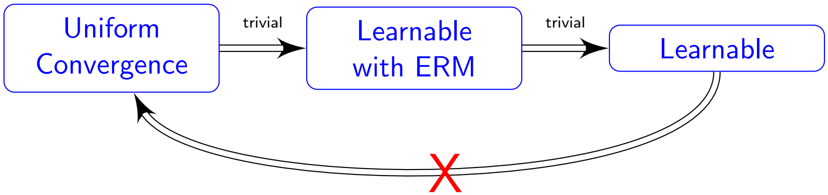
For Regression



For general learning problems?



For general learning problems?



- Not true even in multiclass classification !
- What is learnable ? How to learn ?

Outline

The General Learning Setting

Vapnik's General Learning Setting

- Hypothesis class $\mathcal{H}$
- Instance space $\mathcal{Z}$ with unknown distribution $\mathcal{D}$
- Loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$

Given: Training set $S \sim \mathcal{D}^m$

Goal: Probably approximately solve

$$\min_{h \in \mathcal{H}} L(h) \quad \text{where} \quad L(h) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}} [\ell(h, z)]$$

Examples

- Binary classification:
  - $\mathcal{Z} = \mathcal{X} \times \{0,1\}$
  - $h \in \mathcal{H}$ is a predictor $h : \mathcal{X} \to \{0,1\}$
  - $\ell(h,(x,y)) = \mathbf{1}[h(x) \neq y]$
- Multiclass categorization:
  - $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
  - $h \in \mathcal{H}$ is a predictor $h : \mathcal{X} \to \mathcal{Y}$
  - $\ell(h,(x,y)) = \mathbf{1}[h(x) \neq y]$
- k-means clustering:
  - $\mathcal{Z} = \mathbb{R}^d$
  - $\mathcal{H} \subset (\mathbb{R}^d)^k$ specifies k cluster centers
  - $\ell((\mu_1,\ldots,\mu_k),z) = \min_j \|\mu_j - z\|$
- Density Estimation:
  - h is a parameter of a density $p_h(z)$
  - $\ell(h,z) = -\log p_h(z)$

- **Uniform Convergence**:
  For $m \geq m_{\mathrm{UC}}(\epsilon, \delta)$,

$$\underset{S \sim \mathcal{D}^m}{\mathbb{P}} \left[\forall h \in \mathcal{H},\ |L_S(h) - L(h)| \leq \epsilon \right] \geq 1 - \delta$$

Learnability, ERM, Uniform convergence

- **Uniform Convergence**:
  For $m \geq m_{\mathrm{UC}}(\epsilon, \delta)$,

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left[\forall h \in \mathcal{H},\ |L_S(h) - L(h)| \leq \epsilon\right] \geq 1 - \delta$$

- **Learnable**:
  $\exists \mathcal{A}$ s.t. for $m \geq m_{\mathrm{PAC}}(\epsilon, \delta)$,

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left[L(\mathcal{A}(S)) \leq \min_{h \in \mathcal{H}} L(h) + \epsilon\right] \geq 1 - \delta$$

Learnability, ERM, Uniform convergence

- **Uniform Convergence**:
  For $m \geq m_{\mathrm{UC}}(\epsilon, \delta)$,
  $$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} [\forall h \in \mathcal{H},\ |L_S(h) - L(h)| \leq \epsilon] \geq 1 - \delta$$
- **Learnable**:
  $\exists \mathcal{A}$ s.t. for $m \geq m_{\mathrm{PAC}}(\epsilon, \delta)$,
  $$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left[L(\mathcal{A}(S)) \leq \min_{h \in \mathcal{H}} L(h) + \epsilon \right] \geq 1 - \delta$$
- **ERM**:
  An algorithm that returns $\mathcal{A}(S) \in \mathrm{argmin}_{h \in \mathcal{H}} L_S(h)$

Learnability, ERM, Uniform convergence

- **Uniform Convergence**:
  For $m \geq m_{\mathrm{UC}}(\epsilon, \delta)$,

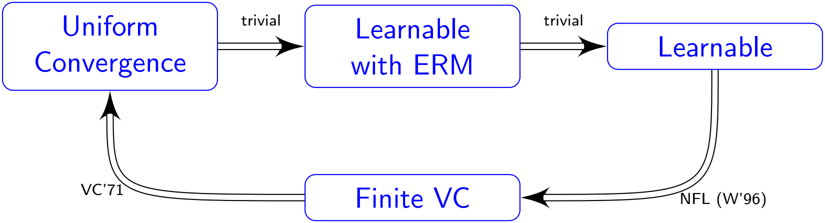
$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left[\forall h \in \mathcal{H},\ |L_S(h) - L(h)| \leq \epsilon\right] \geq 1 - \delta$$

- **Learnable**:
  $\exists \mathcal{A}$ s.t. for $m \geq m_{\mathrm{PAC}}(\epsilon, \delta)$,

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left[L(\mathcal{A}(S)) \leq \min_{h \in \mathcal{H}} L(h) + \epsilon\right] \geq 1 - \delta$$

- **ERM**:
  An algorithm that returns $\mathcal{A}(S) \in \mathrm{argmin}_{h \in \mathcal{H}} L_S(h)$
- **Learnable by arbitrary ERM**:
  Like "Learnable" but $\mathcal{A}$ should be an ERM.
  Denote sample complexity by $m_{\mathrm{ERM}}(\epsilon, \delta)$

For Binary Classification



$$m_{\mathrm{UC}}(\epsilon,\delta) \;\approx\; m_{\mathrm{ERM}}(\epsilon,\delta) \;\approx\; m_{\mathrm{PAC}}(\epsilon,\delta) \;\approx\; \frac{\mathrm{VC}(\mathcal{H})\log(1/\delta)}{\epsilon^2}$$

Outline

First (trivial) Counter Example

Minorizing function:

- Let $\mathcal{H}'$ be a class of binary classifiers with infinite VC dimension
- Let $\mathcal{H} = \mathcal{H}' \cup \{h_0\}$
- Let $\ell(h,(x,y)) = \begin{cases} 1 & \text{if } h \neq h_0 \wedge h(x) \neq y \\ 1/2 & \text{if } h \neq h_0 \wedge h(x) = y \\ 0 & \text{if } h = h_0 \end{cases}$
- No uniform convergence ($m_{\mathrm{UC}} = \infty$)
- Learnable by ERM ($m_{\mathrm{ERM}} = 0$)

From Vapnik's book ...

This example shows that there exist trivial cases of consistency that depend on whether a given set of functions contains a minorizing function.

Therefore, any theory of consistency that uses the classical definition needs

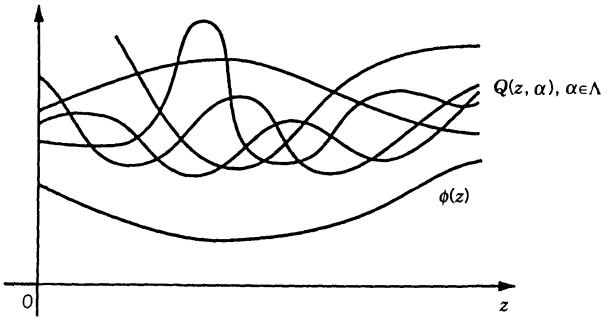


FIGURE 3.2. A case of trivial consistency. The ERM method is inconsistent on the set of functions $Q(z, \alpha)$, $\alpha \in \Lambda$, and is consistent on the set of functions $\phi(z) \bigcup Q(z, \alpha)$, $\alpha \in \Lambda$.

Second Counter Example — Multiclass

- $\mathcal{X}$ – a set, $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$.
- $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ where

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

Second Counter Example — Multiclass

- $\mathcal{X}$ – a set, $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$.
- $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ where

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

- Claim: No uniform convergence: $m_{\mathrm{UC}} \geq |\mathcal{X}|/\epsilon$
  - Target function is $h_\emptyset$
  - For any training set S, take $T = \mathcal{X} \setminus S$
  - $L_S(h_T) = 0$ but $L(h_T) = \mathbb{P}[T]$

Second Counter Example — Multiclass

- $\mathcal{X}$ – a set, $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$.
- $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ where

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

- Claim: $\mathcal{H}$ is Learnable: $m_{\text{PAC}} \leq \frac{1}{\epsilon}$
  - Let T be the target
  - $\mathcal{A}(S) = h_T$ if $(x, T) \in S$
  - $\mathcal{A}(S) = h_\emptyset$ if $S = \{(x_1, *), \ldots, (x_m, *)\}$
  - In the 1st case, $L(\mathcal{A}(S)) = 0$.
  - In the 2nd case, $L(\mathcal{A}(S)) = \mathbb{P}[T]$
  - With high probability, if $\mathbb{P}[T] > \epsilon$ then we'll be in the 1st case

Second Counter Example — Multiclass

Corollary

- $\frac{m_{UC}}{m_{PAC}} \approx |\mathcal{X}|$.
- *If* $|\mathcal{X}| \to \infty$ *then the problem is learnable but there is no uniform convergence!*

Third Counter Example — Stochastic Convex Optimization

Consider the family of problems:

- $\mathcal{H}$ is a convex set with $\max_{h \in \mathcal{H}} \|h\| \leq 1$
- For all z, $\ell(h, z)$ is convex and Lipschitz w.r.t. h

Third Counter Example — Stochastic Convex Optimization

Consider the family of problems:

- $\mathcal{H}$ is a convex set with $\max_{h \in \mathcal{H}} \|h\| \leq 1$
- For all z, $\ell(h, z)$ is convex and Lipschitz w.r.t. h

Claim:

- Problem is learnable by the rule:

$$\underset{h \in \mathcal{H}}{\operatorname{argmin}} \frac{\lambda_m}{2} \|h\|^2 + \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i)$$

- No uniform convergence
- Not learnable by ERM

Third Counter Example — Stochastic Convex Optimization

Proof (of "not learnable by arbitrary ERM")

- 1-Mean + missing features

Third Counter Example — Stochastic Convex Optimization

Proof (of "not learnable by arbitrary ERM")

- 1-Mean + missing features
- $z = (\alpha, x)$, $\alpha \in \{0,1\}^d$, $x \in \mathbb{R}^d$, $\|x\| \le 1$
- $\ell(h, (\alpha, x)) = \sqrt{\sum_i \alpha_i (h_i - x_i)^2}$
- Take $\mathbb{P}[\alpha_i = 1] = 1/2$, $\mathbb{P}[x = \mu] = 1$
- Let $h^{(i)}$ be s.t.

$$h_j^{(i)} = \begin{cases} 1 - \mu_j & \text{if } j = i \\ \mu_j & \text{o.w.} \end{cases}$$

- If d is large enough, exists i such that $h^{(i)}$ is an ERM
- But $L(h^{(i)}) \ge 1/\sqrt{2}$

Third Counter Example — Stochastic Convex Optimization

Proof (of "not even learnable by a unique ERM")

Perturb the loss a little bit:

$$\ell(h,(\alpha,x)) = \sqrt{\sum_i \alpha_i (h_i - x_i)^2} + \epsilon \sum_i 2^{-i}(h_i - 1)^2$$

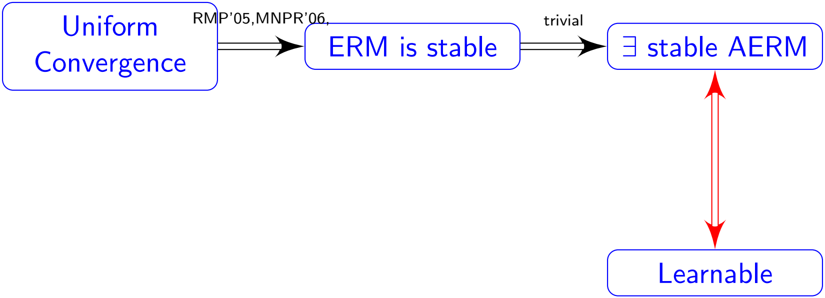
- Now loss is strictly convex — unique ERM
- But the unique ERM does not generalize (as before)

Outline

Characterizing Learnability using Stability

Theorem

A sufficient and necessary condition for learnability is the existence of Asymptotic ERM (AERM) which is stable.



More formally

Definition (Stability)

We say that A is $\epsilon_{\text{stable}}(m)$-uniform-replace-one stable if for all $\mathcal{D}$,

$$\mathop{\mathbb{E}}_{S,z',i} |\ell(\mathcal{A}(S^{(i)}); z') - \ell(\mathcal{A}(S); z')| \le \epsilon_{\text{stable}}(m).$$

More formally

Definition (Stability)

We say that A is $\epsilon_{\text{stable}}(m)$-uniform-replace-one stable if for all $\mathcal{D}$,

$$\mathop{\mathbb{E}}_{S,z',i} |\ell(\mathcal{A}(S^{(i)}); z') - \ell(\mathcal{A}(S); z')| \leq \epsilon_{\text{stable}}(m).$$

Definition (AERM)

We say that $\mathcal{A}$ is an *AERM (Asymptotic Empirical Risk Minimizer)* with rate $\epsilon_{\text{erm}}(m)$ if for all $\mathcal{D}$:

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m} [L_S(\mathcal{A}(S)) - \min_{h \in \mathcal{H}} L_S(h)] \leq \epsilon_{\text{erm}}(m)$$

Proof sketch: (Stable AERM is sufficient and necessary for Learnability)

Sufficient:

- For AERM: stability $\Rightarrow$ generalization
- AERM+generalization $\Rightarrow$ consistency

Necessary:

- $\exists$ consistent $\mathcal{A} \Rightarrow$
  $\exists$ consistent and generalizing A' (using subsampling)
- Consistent+generalizing $\Rightarrow$ AERM
- AERM+generalizing $\Rightarrow$ stable

Intermediate Summary

- Learnability $\iff$ $\exists$ stable AERM
- But, how do we find one?
- And, is there a combinatorial notion of learnability (like VC dimension) ?

Outline

1. Definitions
2. Learnability without uniform convergence
3. Characterizing Learnability using Stability
4. **Characterizing Multiclass Learnability**
5. Open Questions

Why multiclass learning

- Practical relevance
- A simple twist of binary classification
- In a sense, captures the essence of difficulty of the General Learning Setting

The Graph Dimension

S is G-shattered by $\mathcal{H}$ if $\exists f \in H$ s.t. for every $T \subseteq S$ exists $h \in \mathcal{H}$ with

$$\begin{aligned} h(x) &= f(x) \quad \text{if } x \in T \\ h(x) &\neq f(x) \quad \text{if } x \in S \setminus T \end{aligned}$$

The Graph Dimension

S is G-shattered by $\mathcal{H}$ if $\exists f \in H$ s.t. for every $T \subseteq S$ exists $h \in \mathcal{H}$ with

$$h(x) = f(x) \ \text{ if } x \in T$$
$$h(x) \neq f(x) \ \text{ if } x \in S \setminus T$$

Graph dimension: Maximal size of G-shattered set

The Graph Dimension

S is G-shattered by $\mathcal{H}$ if $\exists f \in H$ s.t. for every $T \subseteq S$ exists $h \in \mathcal{H}$ with

$$\begin{aligned} h(x) &= f(x) \;\; \text{if } x \in T \\ h(x) &\neq f(x) \;\; \text{if } x \in S \setminus T \end{aligned}$$

Graph dimension: Maximal size of G-shattered set
Remark: When $|\mathcal{Y}| = 2$, Graph dimension equals to VC dimension

Example

- Consider again our counter example: $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$ and $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ with

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

Example

- Consider again our counter example: $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$ and $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ with

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

- Claim: Graph dimension of $\mathcal{H}$ is $|\mathcal{X}|$
- Proof: Take $f = h_\emptyset$ and $S = \mathcal{X}$. For each $T \subset S$ take h_{T^c}. So, for $x \in T$, $h_{T^c}(x) = * = f(x)$ and for $x \notin T$, $h_{T^c}(x) = T \neq *$.

Example

- Consider again our counter example: $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$ and $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ with

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

- Claim: Graph dimension of $\mathcal{H}$ is $|\mathcal{X}|$
- Proof: Take $f = h_\emptyset$ and $S = \mathcal{X}$. For each $T \subset S$ take h_{T^c}. So, for $x \in T$, $h_{T^c}(x) = * = f(x)$ and for $x \notin T$, $h_{T^c}(x) = T \neq *$.
- Conclusion: Graph dimension does not characterize multiclass learnability (in fact, Graph dimension characterizes uniform convergence)

The Natarajan Dimension

S is N-shattered by $\mathcal{H}$ if $\exists f_1, f_2 \in \mathcal{H}$ s.t. $\forall x \in S,\; f_1(x) \neq f_2(x)$, and for every $T \subseteq S$ exists $h \in \mathcal{H}$ with

$$h(x) = \begin{cases} f_1(x) & \text{if } x \in T \\ f_2(x) & \text{if } x \in S \setminus T \end{cases}$$

The Natarajan Dimension

S is N-shattered by $\mathcal{H}$ if $\exists f_1, f_2 \in \mathcal{H}$ s.t. $\forall x \in S,\ f_1(x) \neq f_2(x)$, and for every $T \subseteq S$ exists $h \in \mathcal{H}$ with

$$h(x) = \begin{cases} f_1(x) & \text{if } x \in T \\ f_2(x) & \text{if } x \in S \setminus T \end{cases}$$

Natarajan dimension: Maximal size of N-shattered set

The Natarajan Dimension

S is N-shattered by $\mathcal{H}$ if $\exists f_1, f_2 \in \mathcal{H}$ s.t. $\forall x \in S,\ f_1(x) \neq f_2(x)$, and for every $T \subseteq S$ exists $h \in \mathcal{H}$ with

$$h(x) = \begin{cases} f_1(x) & \text{if } x \in T \\ f_2(x) & \text{if } x \in S \setminus T \end{cases}$$

Natarajan dimension: Maximal size of N-shattered set

Remarks:

- When $|\mathcal{Y}| = 2$, Natarajan dimension also equals to VC dimension
- Natarajan $\leq$ Graph

Example

- Consider again our counter example: $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$ and $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ with

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

Example

- Consider again our counter example: $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$ and $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ with

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

- Claim: Natarajan dimension of $\mathcal{H}$ is 1
- Proof:
  - Take $S = \{x_1, x_2\}$. The only possible labelings of S by $\mathcal{H}$ are

	h_1	h_2	h_3	h_4
x_1	1,2	1	*	*
x_2	1,2	*	2	*

  - Constraints on f_1, f_2 are that $f_1(x) \neq f_2(x)$ for all x, and exists h with $h(x_1) = f_1(x)$ and $h_2(x) = f_2(x)$.
  - No (f_1, f_2) satisfies these two constraints.

Example

- Consider again our counter example: $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$ and $\mathcal{H} = \{h_T : T \subset \mathcal{X}\}$ with

$$h_T(x) = \begin{cases} * & x \notin T \\ T & x \in T \end{cases}$$

- Claim: Natarajan dimension of $\mathcal{H}$ is 1
- Proof:
  - Take $S = \{x_1, x_2\}$. The only possible labelings of S by $\mathcal{H}$ are

	h_1	h_2	h_3	h_4
x_1	1,2	1	*	*
x_2	1,2	*	2	*

  - Constraints on f_1, f_2 are that $f_1(x) \neq f_2(x)$ for all x, and exists h with $h(x_1) = f_1(x)$ and $h_2(x) = f_2(x)$.
  - No (f_1, f_2) satisfies these two constraints.
- Does Natarajan dimension characterize multiclass learnability ?

Multiclass Learnability of Symmetric Classes

Theorem

If $\mathcal{H}$ is a class of symmetric functions with Natarajan dimension d then

$$\frac{d + \ln(1/\delta)}{\epsilon} \;\le\; m_{PAC}(\epsilon,\delta) \;\le\; \frac{d\ln(d/\epsilon) + \ln(1/\delta)}{\epsilon} \;.$$

Multiclass Learnability of Symmetric Classes

Theorem

If $\mathcal{H}$ is a class of *symmetric* *functions with Natarajan dimension d then*

$$\frac{d + \ln(1/\delta)}{\epsilon} \;\leq\; m_{PAC}(\epsilon, \delta) \;\leq\; \frac{d\ln(d/\epsilon) + \ln(1/\delta)}{\epsilon} \; .$$

Open Question

Is the above also true for non-symmetric hypotheses classes?

Proof: The Learning Algorithm

A Principle for Designing Good ERMs

A good ERM is an ERM that, for every target hypothesis, considers a small number of hypotheses

Proof: The Learning Algorithm

A Principle for Designing Good ERMs

A good ERM is an ERM that, for every target hypothesis, considers a small number of hypotheses

- Given a target hypothesis $h^\star$, let $\mathcal{S}(h^\star) = \{S : \mathrm{err}_S(h^\star) = 0\}$
- Let $\mathcal{A}(\mathcal{S}(h^\star)) = \{\mathcal{A}(S) : S \in \mathcal{S}(h^\star)\}$
- Claim: If $|\mathcal{A}(\mathcal{S}(h^\star))|$ is small then $\mathcal{A}$ is consistent.

Proof: The Learning Algorithm

A Principle for Designing Good ERMs

A good ERM is an ERM that, for every target hypothesis, considers a small number of hypotheses

- Given a target hypothesis $h^\star$, let $\mathcal{S}(h^\star) = \{S : \mathrm{err}_S(h^\star) = 0\}$
- Let $\mathcal{A}(\mathcal{S}(h^\star)) = \{\mathcal{A}(S) : S \in \mathcal{S}(h^\star)\}$
- Claim: If $|\mathcal{A}(\mathcal{S}(h^\star))|$ is small then $\mathcal{A}$ is consistent.
- Obviously, $|\mathcal{A}(\mathcal{S}(h^\star))| \leq |\mathcal{H}|$ but can be much smaller
- Example: Recall our counter example, then $|\mathcal{A}_{bad}(\mathcal{S}(\emptyset))| = 2^{|\mathcal{X}|}$ while for all $h^\star$, $|\mathcal{A}_{good}(\mathcal{S}(h^\star))| \leq 2$

Proof: Natarajan+Symmetric $\Rightarrow$ small $|\mathcal{A}(\mathcal{S}(h^\star))|$

- Lemma: $|\mathcal{A}(\mathcal{S}(h^\star))| \leq m^d \cdot (\text{Max Range})^{2d}$
- Lemma: If $\mathcal{H}$ is symmetric and has Natarajan dimension d, then the *Max Range* of each $h \in \mathcal{H}$ is at most $2d+1$.

Sample Complexity of Specific classes

- We show how to calculate sample complexity of popular hypothesis classes — particularly, multiclass-to-binary reductions
- Enables a rigorous comparison of known multiclass algorithms
  - Previous analysis (e.g. SS'01,BL'07): how the binary error translates to multiclass error
  - Our analysis: Direct calculation of the sample complexity of the multiclass classifier

Specific classes

- Multiclass-to-binary reductions:
  - 1-vs-rest
  - Linear multiclass construction: $\arg\max_i (Wx)_i$
  - Filter trees
- Use linear predictors in $\mathbb{R}^d$ as the binary classifiers

Specific classes

- Multiclass-to-binary reductions:
  - 1-vs-rest
  - Linear multiclass construction: $\arg\max_i (Wx)_i$
  - Filter trees
- Use linear predictors in $\mathbb{R}^d$ as the binary classifiers

Theorem

The Natarajan dimension of all the above classes is $\tilde{\Theta}(d\,|\mathcal{Y}|)$.

- All these reductions have the same estimation error. To compare them, one should analyze approximation error.

Summary and Open Questions

- Equivalence between uniform convergence and learnability breaks even in multiclass problems
- What characterizes multiclass learnability ?
- What is the corresponding learning rule ?
- What characterizes learnability in the general learning setting ?
- What is the corresponding learning rule ?

Summary and Open Questions

- Equivalence between uniform convergence and learnability breaks even in multiclass problems
- What characterizes multiclass learnability ?
- What is the corresponding learning rule ?
- What characterizes learnability in the general learning setting ?
- What is the corresponding learning rule ?

THANKS