When are Local Queries Useful for Robust Learning?

P. Gourdeau, V. Kanade, M. Kwiatkowska and J. Worrell

University of Oxford

Robust Learning with Random Examples

$$\mathcal{A}$$

Robust Learning with Random Examples

$$S_c \sim D^m \longrightarrow \mathcal{A}$$

Robust Learning with Random Examples

$$S_c \sim D^m \longrightarrow \boxed{\mathcal{A}} \longrightarrow \begin{array}{c} h \in \mathcal{H} \\ \text{s.t. } \mathrm{err}(h, c) \leq \varepsilon \end{array}$$

Robust Learning with Random Examples

$$S_c \sim D^m \longrightarrow \boxed{\mathcal{A}} \longrightarrow \begin{array}{c} h \in \mathcal{H} \\ \text{s.t. } \mathrm{R}_\rho(h, c) \leq \varepsilon \end{array}$$

Exact-in-the-ball: *hypothesis = target in perturbation region*

- ρ = adversary's budget at test time

Robust Learning with Random Examples

$$S_c \sim D^m \longrightarrow \boxed{\mathcal{A}} \longrightarrow h \in \mathcal{H} \text{ s.t. } \mathrm{R}_\rho(h, c) \leq \varepsilon$$

Exact-in-the-ball: *hypothesis = target in perturbation region*

- $\rho =$ adversary's budget at test time

Previous work:

Robust Learning with Random Examples

$$S_c \sim D^m \longrightarrow \mathcal{A} \longrightarrow h \in \mathcal{H} \text{ s.t. } \mathrm{R}_\rho(h, c) \leq \varepsilon$$

Exact-in-the-ball: *hypothesis = target in perturbation region*

- $\rho =$ adversary's budget at test time

Previous work:

- Any distribution $\implies$ can only robustly learn *trivial* concepts

Robust Learning with Random Examples

$$S_c \sim D^m \longrightarrow \boxed{\mathcal{A}} \longrightarrow h \in \mathcal{H} \text{ s.t. } \mathsf{R}_\rho(h, c) \leq \varepsilon$$

Exact-in-the-ball: *hypothesis = target in perturbation region*

- ρ = adversary's budget at test time

Previous work:

- Any distribution $\implies$ can only robustly learn *trivial* concepts
- Uniform distribution $\implies$ conjunctions ($f(x) = \bigwedge_{i \in I} x_i$) and superclasses need $\Omega(2^\rho)$ sample points

Robust Learning with Random Examples

$$S_c \sim D^m \longrightarrow \boxed{\mathcal{A}} \longrightarrow \begin{array}{c} h \in \mathcal{H} \\ \text{s.t. } \mathsf{R}_\rho(h, c) \leq \varepsilon \end{array}$$

Exact-in-the-ball: *hypothesis* $=$ *target in perturbation region*

- $\rho =$ adversary's budget at test time

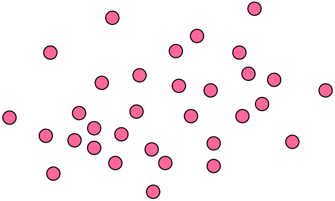
Previous work:

- Any distribution $\implies$ can only robustly learn *trivial* concepts
- Uniform distribution $\implies$ conjunctions $(f(x) = \bigwedge_{i \in I} x_i)$ and superclasses need $\Omega(2^\rho)$ sample points

What happens when we give more power to the learner?

Local Queries

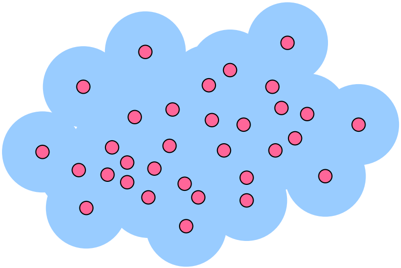
General idea:



- $(\mathcal{X}, d)$ metric space

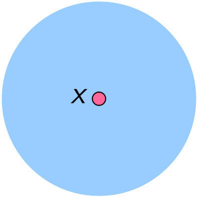
Local Queries

General idea:

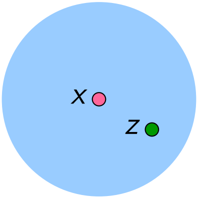


- $(\mathcal{X}, d)$ metric space
- Query region: $B_\lambda(x) = \{z \in \mathcal{X} \mid d(x, z) \le \lambda\}$

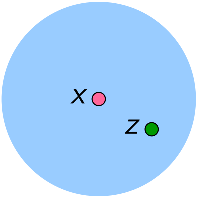
Local Membership Queries (LMQ) [Awasthi et al. 2013]



Local Membership Queries (LMQ) [Awasthi et al. 2013]

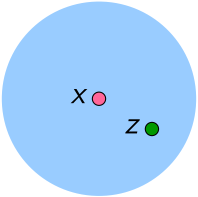


Local Membership Queries (LMQ) [Awasthi et al. 2013]



$c(z)$?

Local Membership Queries (LMQ) [Awasthi et al. 2013]

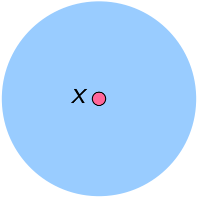


$c(z)$?

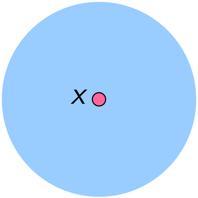
Theorem

Even when adding LMQs, robustly learning conjunctions still needs $\Omega(2^\rho)$ joint sample and LMQ complexity under the uniform distribution

Local Equivalence Queries

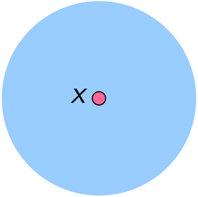


Local Equivalence Queries



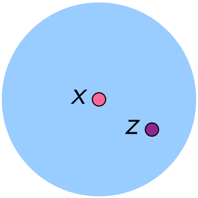
$$c = h?$$

Local Equivalence Queries



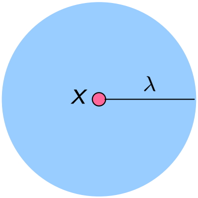
$c \neq h$

Local Equivalence Queries



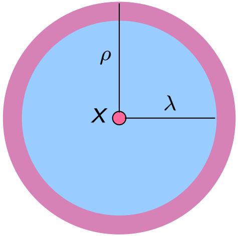
$$c(z) \neq h(z)$$

Local Equivalence Queries



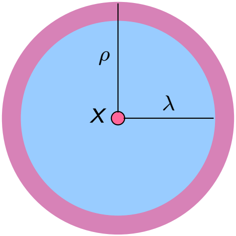
λ-LEQ;

Local Equivalence Queries



λ-LEQ; ρ-adversary

Local Equivalence Queries

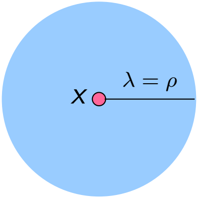


λ-LEQ; ρ-adversary

Theorem

$\lambda < \rho \implies$ *robust learning is impossible for stable functions, including monotone conjunctions*

Local Equivalence Queries



λ-LEQ; ρ-adversary

Theorem

$\lambda = \rho \implies$ *robust learning is possible with a number of random examples m linear in the robust VC dimension (RVC) and a number of LEQ* $r = m \cdot M$*, where M is a mistake bound in the online model*

Take Away

- LMQs don't help robust learning with conjunctions and superclasses

Take Away

- LMQs don't help robust learning with conjunctions and superclasses
- LEQs enable robust learning iff $\lambda \geq \rho$ for many classes of functions

Take Away

- LMQs don't help robust learning with conjunctions and superclasses
- LEQs enable robust learning iff $\lambda \geq \rho$ for many classes of functions
- We get (improved) bounds for specific classes:

Take Away

- LMQs don't help robust learning with conjunctions and superclasses
- LEQs enable robust learning iff $\lambda \geq \rho$ for many classes of functions
- We get (improved) bounds for specific classes:
  - conjunctions, linear classifiers

Take Away

- LMQs don't help robust learning with conjunctions and superclasses
- LEQs enable robust learning iff $\lambda \geq \rho$ for many classes of functions
- We get (improved) bounds for specific classes:
  - conjunctions, linear classifiers
- Full picture isn't clear yet: many open problems!

Thank you!

Paper (arxiv version)