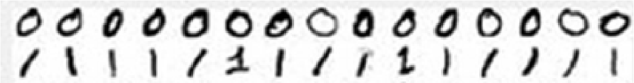
Sample Complexity Bounds for Robustly Learning Decision Lists against Evasion Attacks

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

University of Oxford

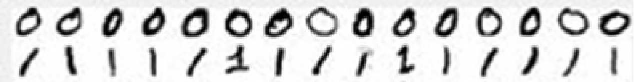
Evasion Attacks

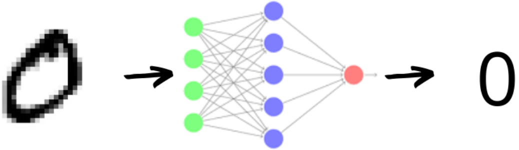
Example: distinguishing between handwritten 0's and 1's:



Evasion Attacks

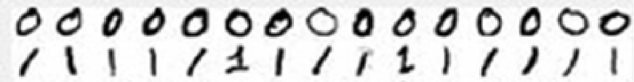
Example: distinguishing between handwritten 0's and 1's:

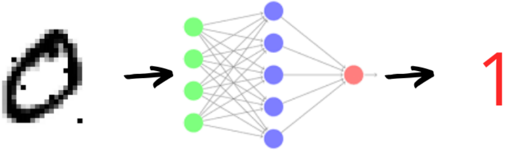




Evasion Attacks

Example: distinguishing between handwritten 0's and 1's:



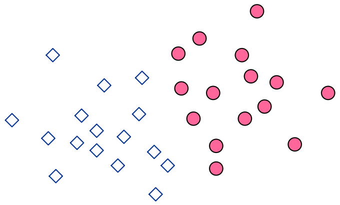


Question: How much data is needed for robust learning against evasion attacks?

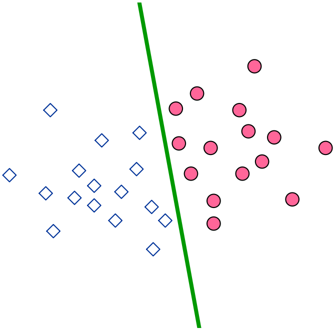
Question: How much data is needed for robust learning against evasion attacks?

Spoiler: The *adversarial budget* is a fundamental quantity in the sample complexity of robust learning against evasion attacks

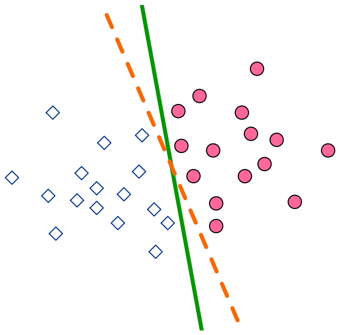
Problem Setting



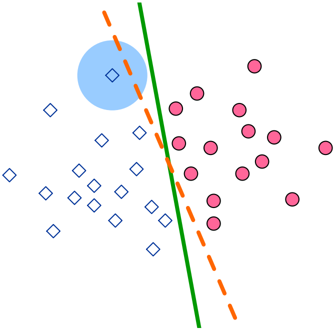
Problem Setting



Problem Setting



Problem Setting



Goal: learn a function that will be *exact-in-the-ball* robust against an adversary who can perturb inputs

Question: How much data is needed for robust learning against evasion attacks?

Question: How much data is needed for robust learning against evasion attacks?

Result #1: Monotone conjunctions require $\Omega(2^{\rho})$ examples to be robustly learned under the uniform distribution.

Question: How much data is needed for robust learning against evasion attacks?

Result #1: Monotone conjunctions require $\Omega(2^\rho)$ examples to be robustly learned under the uniform distribution.

(adversary can perturb ρ bits)

Monotone Conjunctions

"AND" of Boolean variables:

$$\text{thesis} \wedge \text{sleep deprivation} \wedge \text{caffeine}$$

Monotone Conjunctions

"AND" of Boolean variables:

$$\text{thesis} \wedge \text{sleep deprivation} \wedge \text{caffeine}$$

Concept classes that subsume MON-CONJ :

- Decision lists

Monotone Conjunctions

"AND" of Boolean variables:

$$\text{thesis} \wedge \text{sleep deprivation} \wedge \text{caffeine}$$

Concept classes that subsume MON-CONJ :

- Decision lists
- Decision trees

Monotone Conjunctions

"AND" of Boolean variables:

$$\text{thesis} \wedge \text{sleep deprivation} \wedge \text{caffeine}$$

Concept classes that subsume MON-CONJ :

- Decision lists
- Decision trees
- Linear classifiers

Monotone Conjunctions

"AND" of Boolean variables:

$$\text{thesis} \wedge \text{sleep deprivation} \wedge \text{caffeine}$$

Concept classes that subsume MON-CONJ :

- Decision lists
- Decision trees
- Linear classifiers

Monotone Conjunctions

"AND" of Boolean variables:

$$\text{thesis} \wedge \text{sleep deprivation} \wedge \text{caffeine}$$

Concept classes that subsume MON-CONJ :

- Decision lists
- Decision trees
- Linear classifiers

A sample complexity lower bound for MON-CONJ holds for these classes as well.

Sample Complexity Lower Bound

Theorem

For sufficiently large input dimension n, any $\rho(n)$-robust learning algorithm for MON-CONJ has a sample complexity lower bound of $\Omega(2^{\rho(n)})$ under the uniform distribution.

Sample Complexity Lower Bound

Theorem

For sufficiently large input dimension n, any $\rho(n)$-robust learning algorithm for MON-CONJ has a sample complexity lower bound of $\Omega(2^{\rho(n)})$ under the uniform distribution.

Proof Idea.

- Two disjoint monotone conjunctions c_1, c_2 of length 2ρ have robust risk $R_\rho(c_1, c_2)$ bounded below by a constant

$\square$

Sample Complexity Lower Bound

Theorem

For sufficiently large input dimension n, any $\rho(n)$-robust learning algorithm for MON-CONJ has a sample complexity lower bound of $\Omega(2^{\rho(n)})$ under the uniform distribution.

Proof Idea.

- Two disjoint monotone conjunctions c_1, c_2 of length 2ρ have robust risk $R_\rho(c_1, c_2)$ bounded below by a constant
- A random sample of size $m = \Omega(2^\rho)$ won't be able to distinguish c_1 from c_2 w.p. $> 1/2$

$\square$

Question: How much data is needed for robust learning against evasion attacks?

Question: How much data is needed for robust learning against evasion attacks?

Result #2: Decision lists are efficiently $\log(n)$-robustly learnable under smooth distributions.

Question: How much data is needed for robust learning against evasion attacks?

Result #2: Decision lists are efficiently $\log(n)$-robustly learnable under smooth distributions.

- A *polynomial* number of examples is enough to return a hypothesis with small robust risk (with high probability).

Question: How much data is needed for robust learning against evasion attacks?

Result #2: Decision lists are efficiently $\log(n)$-robustly learnable under smooth distributions.

- A *polynomial* number of examples is enough to return a hypothesis with small robust risk (with high probability).
- Smooth = log-Lipschitz (e.g. uniform distribution, product distribution, etc.)

Open Problem[1]

Is a sample-efficient PAC-learning algorithm for concept class $\mathcal{C}$ also a sample-efficient $\log(n)$-robust learning algorithm for $\mathcal{C}$ under the uniform distribution?

[1] From *On the Hardness of Robust Classification*, PG, VK, MK, JW, Journal of Machine Learning Research, 2021.

Open Problem[1]

Is a sample-efficient PAC-learning algorithm for concept class $\mathcal{C}$ also a sample-efficient $\log(n)$-robust learning algorithm for $\mathcal{C}$ under the uniform distribution?

Result #2 adds to the body of positive evidence for this problem.

[1] From *On the Hardness of Robust Classification*, PG, VK, MK, JW, Journal of Machine Learning Research, 2021.

Robust Learnability of Decision Lists

What is a decision list?

Robust Learnability of Decision Lists

What is a decision list?

$$\begin{tikzcd} x_3 \arrow[r] \arrow[d] & x_1 \wedge \bar{x_5} \arrow[r] \arrow[d] & x_2 \wedge \bar{x_3} \arrow[r] \arrow[d] & \text{true} \arrow[d] \\ 1 & 0 & 1 & 0 \end{tikzcd}$$

Robust Learnability of Decision Lists

What is a decision list?

$$
\begin{tikzcd}
x_3 \arrow[r] \arrow[d] & x_1 \wedge \bar{x_5} \arrow[r] \arrow[d] & x_2 \wedge \bar{x_3} \arrow[r] \arrow[d] & \text{true} \arrow[d] \\
1 & 0 & 1 & 0
\end{tikzcd}
$$

k-DL: $k =$ size of conjunction in a node

Robust Learnability of Decision Lists

What is a decision list?

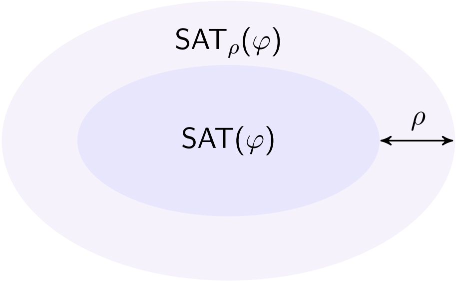
$$\begin{tikzcd} x_3 \arrow[r] \arrow[d] & x_1 \wedge \bar{x_5} \arrow[r] \arrow[d] & x_2 \wedge \bar{x_3} \arrow[r] \arrow[d] & \text{true} \arrow[d] \\ 1 & 0 & 1 & 0 \end{tikzcd}$$

k-DL: $k =$ size of conjunction in a node

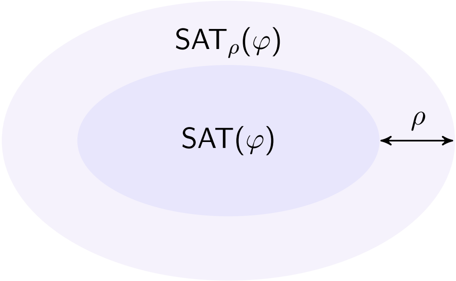
Theorem

Decision lists are efficiently $\log(n)$*-robustly learnable under smooth distributions.*

A Unifying Result

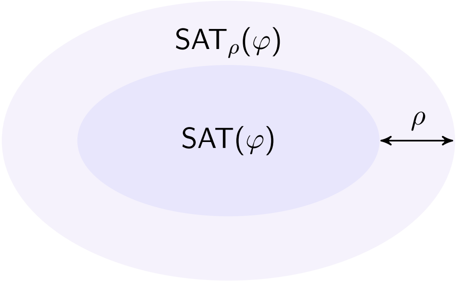


A Unifying Result



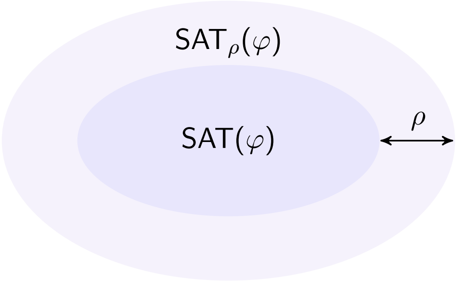
- $\varphi \in k$-CNF: $\varphi(x) = \bigwedge_{i \in I} \bigvee_{1 \leq j \leq k} l_{ij}$

A Unifying Result



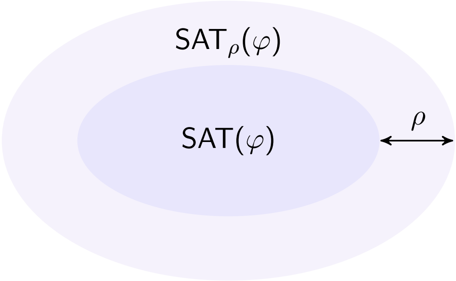
- $\varphi \in k\text{-CNF}$: $\varphi(x) = \bigwedge_{i \in I} \bigvee_{1 \leq j \leq k} l_{ij}$
- $\rho(n) = \log n$

A Unifying Result



- $\varphi \in k\text{-CNF}$: $\varphi(x) = \bigwedge_{i \in I} \bigvee_{1 \leq j \leq k} l_{ij}$
- $\rho(n) = \log n$
- $\mathrm{SAT}(\varphi) = \{x \in \mathcal{X} \mid \varphi(x) = 1\}$

A Unifying Result



- $\varphi \in k$-CNF: $\varphi(x) = \bigwedge_{i \in I} \bigvee_{1 \le j \le k} l_{ij}$
- $\rho(n) = \log n$
- $\mathrm{SAT}(\varphi) = \{x \in \mathcal{X} \mid \varphi(x) = 1\}$
- $|\mathrm{SAT}(\varphi)| \le \mathrm{poly}(\varepsilon, 1/n) \implies |\mathrm{SAT}_{\log n}(\varphi)| \le \varepsilon$

Proof Idea

Theorem

Decision lists are efficiently $\log(n)$*-robustly learnable under smooth distributions.*

Proof Idea

Theorem

Decision lists are efficiently $\log(n)$*-robustly learnable under smooth distributions.*

Proof Idea.

1. Express the event of an exit at depths d_1, d_2 for two k-DL as a k-**CNF formula** $\varphi = \bigwedge_i \bigvee_{j=1}^{k} z_{ij}$

$\square$

Proof Idea

Theorem

Decision lists are efficiently $\log(n)$*-robustly learnable under smooth distributions.*

Proof Idea.

1. Express the event of an exit at depths d_1, d_2 for two k-DL as a k-**CNF formula** $\varphi = \bigwedge_i \bigvee_{j=1}^{k} z_{ij}$
   - Represents an error between the hypothesis and ground truth

$\square$

Proof Idea

Theorem

Decision lists are efficiently $\log(n)$*-robustly learnable under smooth distributions.*

Proof Idea.

1. Express the event of an exit at depths d_1, d_2 for two k-DL as a k-**CNF formula** $\varphi = \bigwedge_i \bigvee_{j=1}^{k} z_{ij}$
   - Represents an error between the hypothesis and ground truth
2. **Induction on** k**:** the $\log(n)$-expansion of satisfying assignments of φ (i.e., the robust risk) isn't too large

$\square$

Proof Idea

Theorem

Decision lists are efficiently $\log(n)$*-robustly learnable under smooth distributions.*

Proof Idea.

1. Express the event of an exit at depths d_1, d_2 for two k-DL as a k-**CNF formula** $\varphi = \bigwedge_i \bigvee_{j=1}^{k} z_{ij}$
   - Represents an error between the hypothesis and ground truth
2. **Induction on k:** the $\log(n)$-expansion of satisfying assignments of φ (i.e., the robust risk) isn't too large
   - Unifying result from previous slide

□

Proof Idea

Theorem

Decision lists are efficiently $\log(n)$*-robustly learnable under smooth distributions.*

Proof Idea.

1. Express the event of an exit at depths d_1, d_2 for two k-DL as a k-**CNF formula** $\varphi = \bigwedge_i \bigvee_{j=1}^{k} z_{ij}$
   - Represents an error between the hypothesis and ground truth
2. **Induction on** k**:** the $\log(n)$-expansion of satisfying assignments of φ (i.e., the robust risk) isn't too large
   - Unifying result from previous slide
3. Controlling the standard risk $\implies$ controlling the **robust** risk

□

Proof Idea

Theorem

Decision lists are efficiently $\log(n)$*-robustly learnable under smooth distributions.*

Proof Idea.

1. Express the event of an exit at depths d_1, d_2 for two k-DL as a k-**CNF formula** $\varphi = \bigwedge_i \bigvee_{j=1}^{k} z_{ij}$
   - Represents an error between the hypothesis and ground truth
2. **Induction on** k**:** the $\log(n)$-expansion of satisfying assignments of φ (i.e., the robust risk) isn't too large
   - Unifying result from previous slide
3. Controlling the standard risk $\implies$ controlling the **robust** risk
   - The standard learning algorithm for k-DL is a robust learner!

$\square$

Takeaways and Future Work

- The *adversary's budget* is a fundamental quantity in determining the sample complexity of robust learning

Takeaways and Future Work

- The *adversary's budget* is a fundamental quantity in determining the sample complexity of robust learning
- We can efficiently use standard PAC learning algorithms as *black boxes* for certain robust learning problems

Takeaways and Future Work

- The *adversary's budget* is a fundamental quantity in determining the sample complexity of robust learning
- We can efficiently use standard PAC learning algorithms as *black boxes* for certain robust learning problems
  - Our paper: decision lists under smooth distributions

Takeaways and Future Work

- The *adversary's budget* is a fundamental quantity in determining the sample complexity of robust learning
- We can efficiently use standard PAC learning algorithms as *black boxes* for certain robust learning problems
  - Our paper: decision lists under smooth distributions
- Research directions:

Takeaways and Future Work

- The *adversary's budget* is a fundamental quantity in determining the sample complexity of robust learning
- We can efficiently use standard PAC learning algorithms as *black boxes* for certain robust learning problems
  - Our paper: decision lists under smooth distributions
- Research directions:
  - Tighter bounds for decision lists

Takeaways and Future Work

- The *adversary's budget* is a fundamental quantity in determining the sample complexity of robust learning
- We can efficiently use standard PAC learning algorithms as *black boxes* for certain robust learning problems
  - Our paper: decision lists under smooth distributions
- Research directions:
  - Tighter bounds for decision lists
  - Linear classifiers

Takeaways and Future Work

- The *adversary's budget* is a fundamental quantity in determining the sample complexity of robust learning
- We can efficiently use standard PAC learning algorithms as *black boxes* for certain robust learning problems
    - Our paper: decision lists under smooth distributions
- Research directions:
    - Tighter bounds for decision lists
    - Linear classifiers
    - General PAC classes

Thank you!

Paper (arxiv version)