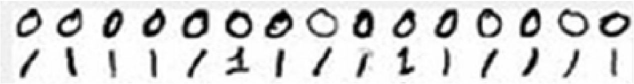
On the Hardness of Robust Classification

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

University of Oxford

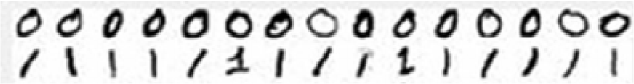
Overview

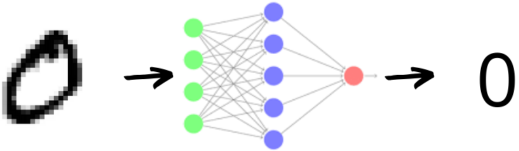
Example: distinguishing between handwritten 0's and 1's:



Overview

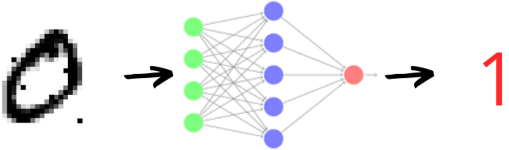
Example: distinguishing between handwritten 0's and 1's:





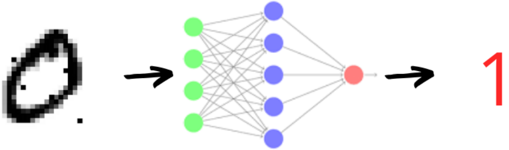
Overview

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



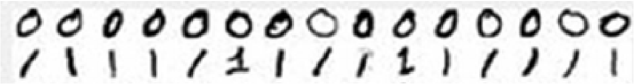
Overview

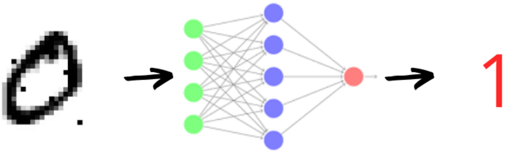
Example: distinguishing between handwritten 0's and 1's:



Overview

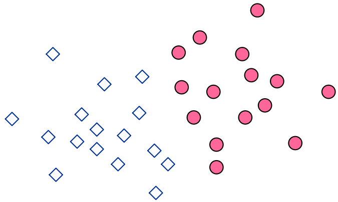
Example: distinguishing between handwritten 0's and 1's:



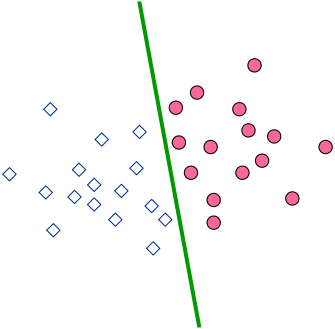


Question: how much computational resources and data are needed in robust learning?

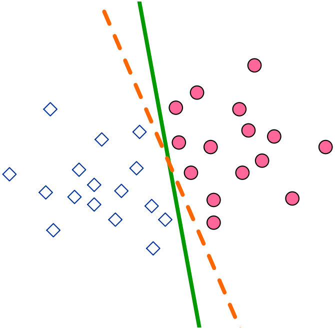
Problem Setting



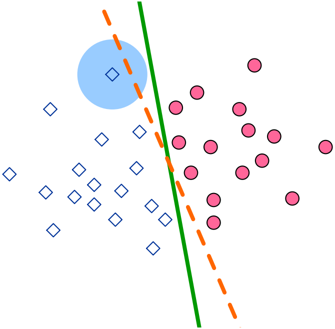
Problem Setting



Problem Setting



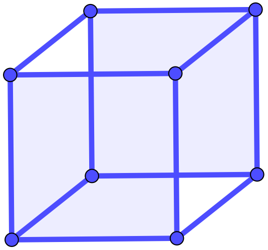
Problem Setting



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

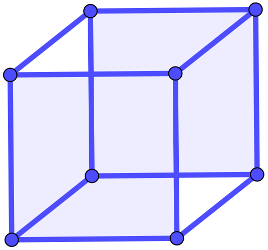
Sample Complexity

Setting: binary feature vectors, binary classification.



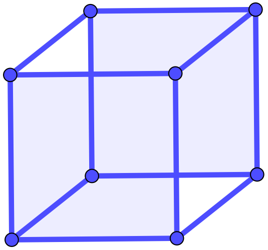
Sample Complexity

Setting: binary feature vectors, binary classification.
Requirement: *polynomial* sample complexity (*efficient robust learning*).



Sample Complexity

Setting: binary feature vectors, binary classification.
Requirement: *polynomial* sample complexity (*efficient robust learning*).



Theorem
Under the exact-in-the-ball definition of robustness, only trivial concepts can be robustly learned.

Sample Complexity

Setting: binary feature vectors, binary classification.
Requirement: *polynomial* sample complexity (*efficient robust learning*).

Theorem

Under the exact-in-the-ball definition of robustness, only trivial concepts can be robustly learned.

Sample Complexity

Setting: binary feature vectors, binary classification.
Requirement: *polynomial* sample complexity (*efficient robust learning*).

Theorem

Under the exact-in-the-ball definition of robustness, only trivial concepts can be robustly learned.

Sample Complexity

Setting: binary feature vectors, binary classification.
Requirement: *polynomial* sample complexity (*efficient robust learning*).

Theorem

Under the exact-in-the-ball definition of robustness, only trivial concepts can be robustly learned.

- Distributional assumptions are *essential* !

A Robustness Threshold

Question: How much perturbation budget ρ can we give an adversary and still ensure efficient robust learnability?

A Robustness Threshold

Question: How much perturbation budget ρ can we give an adversary and still ensure efficient robust learnability?

Our paper: Monotone conjunctions

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

A Robustness Threshold

Question: How much perturbation budget ρ can we give an adversary and still ensure efficient robust learnability?

Our paper: Monotone conjunctions

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

Theorem

Under smooth distributions, the threshold to efficiently robustly learn monotone conjunctions is $\rho = O(\log n)$.

A Robustness Threshold

Question: How much perturbation budget ρ can we give an adversary and still ensure efficient robust learnability?

Our paper: Monotone conjunctions

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

Theorem

Under smooth distributions, the threshold to efficiently robustly learn monotone conjunctions is $\rho = O(\log n)$.

$\boldsymbol{\rho = O(\log n)}$: there is a sample-efficient algorithm.

A Robustness Threshold

Question: How much perturbation budget ρ can we give an adversary and still ensure efficient robust learnability?

Our paper: Monotone conjunctions

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

Theorem

Under smooth distributions, the threshold to efficiently robustly learn monotone conjunctions is $\rho = O(\log n)$.

$\boldsymbol{\rho = O(\log n)}$: there is a sample-efficient algorithm.

$\boldsymbol{\rho = \omega(\log n)}$: no sample-efficient learning algorithm exists.

A Robustness Threshold

Question: How much perturbation budget ρ can we give an adversary and still ensure efficient robust learnability?

Our paper: Monotone conjunctions

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

Theorem

Under smooth distributions, the threshold to efficiently robustly learn monotone conjunctions is $\rho = O(\log n)$.

$\boldsymbol{\rho = O(\log n)}$: there is a sample-efficient algorithm.

$\boldsymbol{\rho = \omega(\log n)}$: no sample-efficient learning algorithm exists.

Information-theoretic result: even when simply considering sample complexity, robust learning can be hard.

Computational Hardness

Question: *Can an information-theoretically easy robust learning problem still be computationally hard?*

Computational Hardness

Question: *Can an information-theoretically easy robust learning problem still be computationally hard?* Yes!

Computational Hardness

Question: *Can an information-theoretically easy robust learning problem still be computationally hard?* Yes!

Simple proof of the result of Bubeck et al. (2018)

Come see our poster!

Take Away

- *Inadequacies* of widely-used and natural definitions of robustness surface under a learning theory perspective.

Take Away

- *Inadequacies* of widely-used and natural definitions of robustness surface under a learning theory perspective.
- Easy proof for computational hardness of robust learning.

Take Away

- *Inadequacies* of widely-used and natural definitions of robustness surface under a learning theory perspective.
- Easy proof for computational hardness of robust learning.
- It may be possible to only solve "easy" robust learning problems with strong *distributional assumptions*.

Take Away

- *Inadequacies* of widely-used and natural definitions of robustness surface under a learning theory perspective.
- Easy proof for computational hardness of robust learning.
- It may be possible to only solve “easy” robust learning problems with strong *distributional assumptions*.
- Other learning models, e.g. active learning.

Thank you!

Paper (arxiv version)

Poster session: Today 10:45 – 12:45 (Learning Theory)