Supervised Learning with Massart Noise

Ankur Moitra (MIT)

Simons Institute Bootcamp Tutorial, Part 3

In 1984, Valiant introduced the **PAC Learning Model**:

(1) Given samples (X, Y) where the distribution on X is arbitrary and Y is a label that is +1 or -1

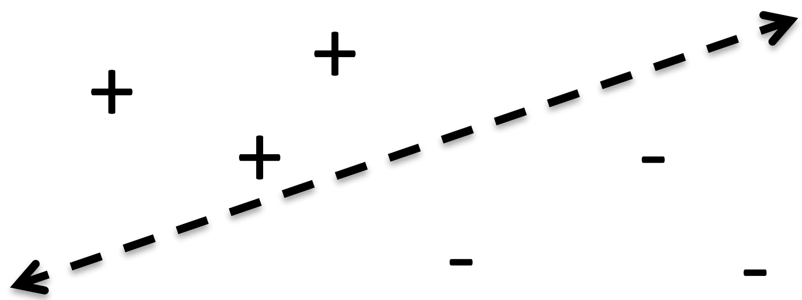
(2) Assume Y = h(X) for some unknown hypothesis h that is in a known class H

In 1984, Valiant introduced the **PAC Learning Model**:

(1) Given samples (X, Y) where the distribution on X is arbitrary and Y is a label that is +1 or -1

(2) Assume Y = h(X) for some unknown hypothesis h that is in a known class H

e.g. the class of **halfspaces** $Y = \mathrm{sgn}(\langle w^*, X \rangle + b)$

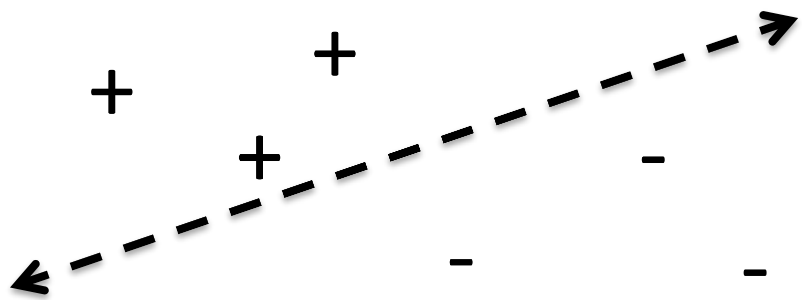


In 1984, Valiant introduced the **PAC Learning Model**:

(1) Given samples (X, Y) where the distribution on X is arbitrary and Y is a label that is +1 or -1

(2) Assume Y = h(X) for some unknown hypothesis h that is in a known class H

e.g. the class of **halfspaces** $Y = \mathrm{sgn}(\langle w^*, X \rangle + b)$



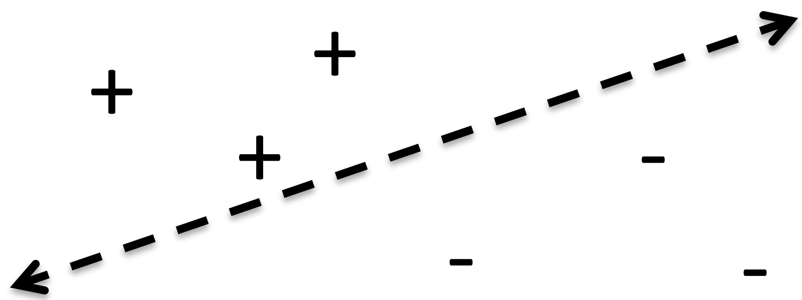
Goal: Estimate h approximately

In 1984, Valiant introduced the **PAC Learning Model**:

(1) Given samples (X, Y) where the distribution on X is arbitrary and Y is a label that is +1 or -1

(2) Assume Y = h(X) for some unknown hypothesis h that is in a known class H

e.g. the class of **halfspaces** $Y = \mathrm{sgn}(\langle w^*, X\rangle + b)$



Goal: Estimate h approximately

Probably **A**pproximately **C**orrect

MODELS FOR NOISE

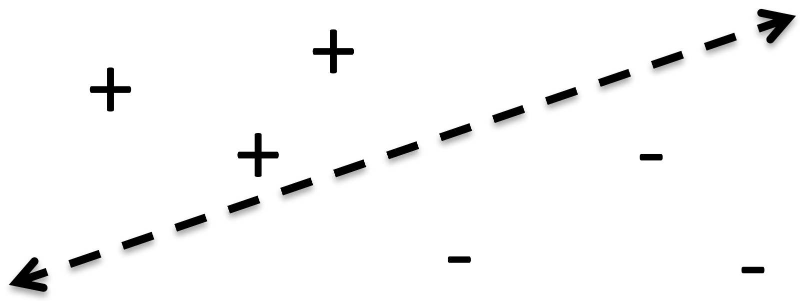
What if there is no simple hypothesis that fits the data *exactly*?

MODELS FOR NOISE

What if there is no simple hypothesis that fits the data *exactly*?

Standard frameworks:

Random Classification Noise: Each label is flipped with some fixed probability

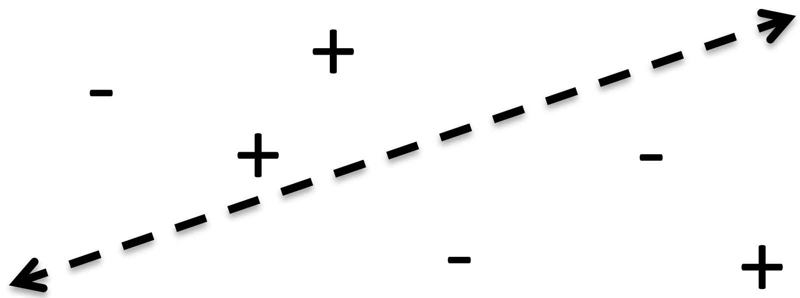


MODELS FOR NOISE

What if there is no simple hypothesis that fits the data *exactly*?

Standard frameworks:

Random Classification Noise: Each label is flipped with some fixed probability

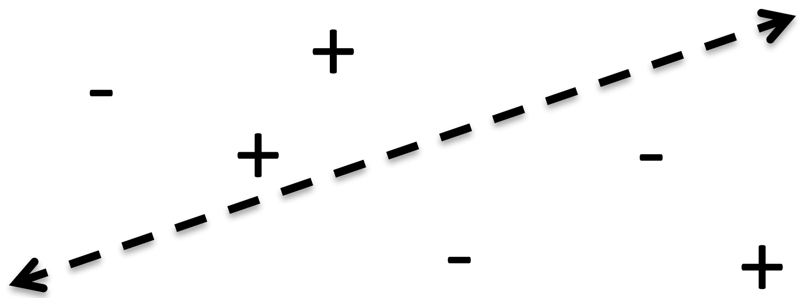


MODELS FOR NOISE

What if there is no simple hypothesis that fits the data *exactly*?

Standard frameworks:

Random Classification Noise: Each label is flipped with some fixed probability



[Blum et al. '98]: There is a polynomial time algorithm for learning halfspaces under random classification noise

MODELS FOR NOISE

What if there is no simple hypothesis that fits the data *exactly*?

MODELS FOR NOISE

What if there is no simple hypothesis that fits the data *exactly*?

Standard frameworks:

Agnostic Noise: No assumption about the structure of the noise, still want to find approximately best agreement in the class

MODELS FOR NOISE

What if there is no simple hypothesis that fits the data *exactly*?

Standard frameworks:

Agnostic Noise: No assumption about the structure of the noise, still want to find approximately best agreement in the class

Unfortunately, agnostic learning is generally hard without further assumptions!

MODELS FOR NOISE

What if there is no simple hypothesis that fits the data *exactly*?

Standard frameworks:

Agnostic Noise: No assumption about the structure of the noise, still want to find approximately best agreement in the class

Unfortunately, agnostic learning is generally hard without further assumptions!

[Kalai et al. '05], **[Awasthi et al. '18]**: There is a polynomial time algorithm for agnostic learning when X is Gaussian

MODELS FOR NOISE

What if there is no simple hypothesis that fits the data *exactly*?

Standard frameworks:

Agnostic Noise: No assumption about the structure of the noise, still want to find approximately best agreement in the class

Unfortunately, agnostic learning is generally hard without further assumptions!

[Kalai et al. '05], **[Awasthi et al. '18]**: There is a polynomial time algorithm for agnostic learning when X is Gaussian

[Daniely '16]: Distribution-independent **weak** agnostic learning of halfspaces is **hard**

Are there challenging noise models where we can learn without making distributional assumptions on X?

Are there challenging noise models where we can learn without making distributional assumptions on X?

In this talk, we'll be interested in:

Massart Noise: The label of each point x is flipped independently with some probability $\eta(x) \leq \eta < 1/2$

Are there challenging noise models where we can learn without making distributional assumptions on X?

In this talk, we'll be interested in:

Massart Noise: The label of each point x is flipped independently with some probability $\eta(x) \leq \eta < 1/2$

Interpretation #1: Each label is flipped independently with prob. η but an adversary can choose to **unflip** it

Are there challenging noise models where we can learn without making distributional assumptions on X?

In this talk, we'll be interested in:

Massart Noise: The label of each point x is flipped independently with some probability $\eta(x) \leq \eta < 1/2$

Interpretation #1: Each label is flipped independently with prob. η but an adversary can choose to **unflip** it

Interpretation #2 (sort of): An adversary can arbitrarily control a **random** η fraction of the data

Are there challenging noise models where we can learn without making distributional assumptions on X?

In this talk, we'll be interested in:

Massart Noise: The label of each point x is flipped independently with some probability $\eta(x) \leq \eta < 1/2$

Interpretation #1: Each label is flipped independently with prob. η but an adversary can choose to **unflip** it

Interpretation #2 (sort of): An adversary can arbitrarily control a **random** η fraction of the data

Are there distribution-independent algorithms for learning with Massart noise?

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- A Two-Player Game
- The Algorithm and Convergence

Part III: Experiments and Fairness

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- **Recent Results**

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- A Two-Player Game
- The Algorithm and Convergence

Part III: Experiments and Fairness

RECENT RESULTS

Theorem [Diakonikolas, Gouleakis, Tzamos '19]: There is a polynomial time algorithm for **improperly** learning halfpsaces under Massart noise with error $\eta + \epsilon$

RECENT RESULTS

Theorem [Diakonikolas, Gouleakis, Tzamos '19]: There is a polynomial time algorithm for **improperly** learning halfpsaces under Massart noise with error $\eta + \epsilon$

The algorithm outputs a partition of space into a polynomial number of regions, with a different halfspace on each

RECENT RESULTS

Theorem [Diakonikolas, Gouleakis, Tzamos '19]: There is a polynomial time algorithm for **improperly** learning halfpsaces under Massart noise with error $\eta + \epsilon$

The algorithm outputs a partition of space into a polynomial number of regions, with a different halfspace on each

Is there a proper learning algorithm?

RECENT RESULTS

Theorem [Diakonikolas, Gouleakis, Tzamos '19]: There is a polynomial time algorithm for **improperly** learning halfpsaces under Massart noise with error $\eta + \epsilon$

The algorithm outputs a partition of space into a polynomial number of regions, with a different halfspace on each

Is there a proper learning algorithm?

It gets error $\eta + \epsilon$, rather than the optimal agreement within the class

RECENT RESULTS

Theorem [Diakonikolas, Gouleakis, Tzamos '19]: There is a polynomial time algorithm for **improperly** learning halfpsaces under Massart noise with error $\eta + \epsilon$

The algorithm outputs a partition of space into a polynomial number of regions, with a different halfspace on each

Is there a proper learning algorithm?

It gets error $\eta + \epsilon$, rather than the optimal agreement within the class

Can we achieve OPT efficiently?

RECENT RESULTS

Theorem [Chen, Koehler, Moitra, Yau '20]: There is a polynomial time algorithm for **properly** learning halfspaces under Massart noise with error $\eta + \epsilon$

RECENT RESULTS

Theorem [Chen, Koehler, Moitra, Yau '20]: There is a polynomial time algorithm for **properly** learning halfspaces under Massart noise with error $\eta + \epsilon$

General framework, independently discovered by **[Diakonikolas, Kontonis, Tzamos, Zarifis '20]** for learning with Tsybakov noise

RECENT RESULTS

Theorem [Chen, Koehler, Moitra, Yau '20]: There is a polynomial time algorithm for **properly** learning halfspaces under Massart noise with error $\eta + \epsilon$

General framework, independently discovered by **[Diakonikolas, Kontonis, Tzamos, Zarifis '20]** for learning with Tsybakov noise

Theorem [Chen, Koehler, Moitra, Yau '20]: There is a polynomial time algorithm for learning **generalized linear models** under Massart noise

i.e $\mathbb{E}[Y|X] = \sigma(\langle w^*, X\rangle + b)$

link function: monotone, Lipschitz

RECENT RESULTS

Theorem [Chen, Koehler, Moitra, Yau '20]: There is a polynomial time algorithm for **properly** learning halfspaces under Massart noise with error $\eta + \epsilon$

General framework, independently discovered by **[Diakonikolas, Kontonis, Tzamos, Zarifis '20]** for learning with Tsybakov noise

Theorem [Chen, Koehler, Moitra, Yau '20]: There is a polynomial time algorithm for learning **generalized linear models** under Massart noise

i.e $\mathbb{E}[Y|X] = \sigma(\langle w^*, X\rangle + b)$

link function: monotone, Lipschitz

In particular, this includes noisy logistic regression as a special case

LOWER BOUNDS

Moreover there is a surprisingly unnoticed connection between learning with Massart noise and Valiant's **evolvability**

LOWER BOUNDS

Moreover there is a surprisingly unnoticed connection between learning with Massart noise and Valiant's **evolvability**

In particular, we show

Lower bounds for Evolvability → **Lower bounds for learning under Massart noise**

LOWER BOUNDS

Moreover there is a surprisingly unnoticed connection between learning with Massart noise and Valiant's **evolvability**

In particular, we show

Lower bounds for Evolvability → **Lower bounds for learning under Massart noise**

Theorem [Chen, Koehler, Moitra, Yau '20]: Any statistical query algorithm for learning under Massart noise to error $\mathrm{OPT} + \epsilon$ must make a superpolynomial number of queries

LOWER BOUNDS

Moreover there is a surprisingly unnoticed connection between learning with Massart noise and Valiant's **evolvability**

In particular, we show

Lower bounds for Evolvability → **Lower bounds for learning under Massart noise**

Theorem [Chen, Koehler, Moitra, Yau '20]: Any statistical query algorithm for learning under Massart noise to error $\mathrm{OPT} + \epsilon$ must make a superpolynomial number of queries

Additionally can give new distribution-dependent evolutionary algorithms that are resilient to drift from this connection

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- A Two-Player Game
- The Algorithm and Convergence

Part III: Experiments and Fairness

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- **Loss Functions and Convex Surrogates**
- A Two-Player Game
- The Algorithm and Convergence

Part III: Experiments and Fairness

LOSS FUNCTIONS

Typically we want to measure the **0/1 Loss**:

$$\mathbb{P}[Y \neq \mathrm{sgn}(\langle w, X \rangle)]$$

LOSS FUNCTIONS

Typically we want to measure the **0/1 Loss**:

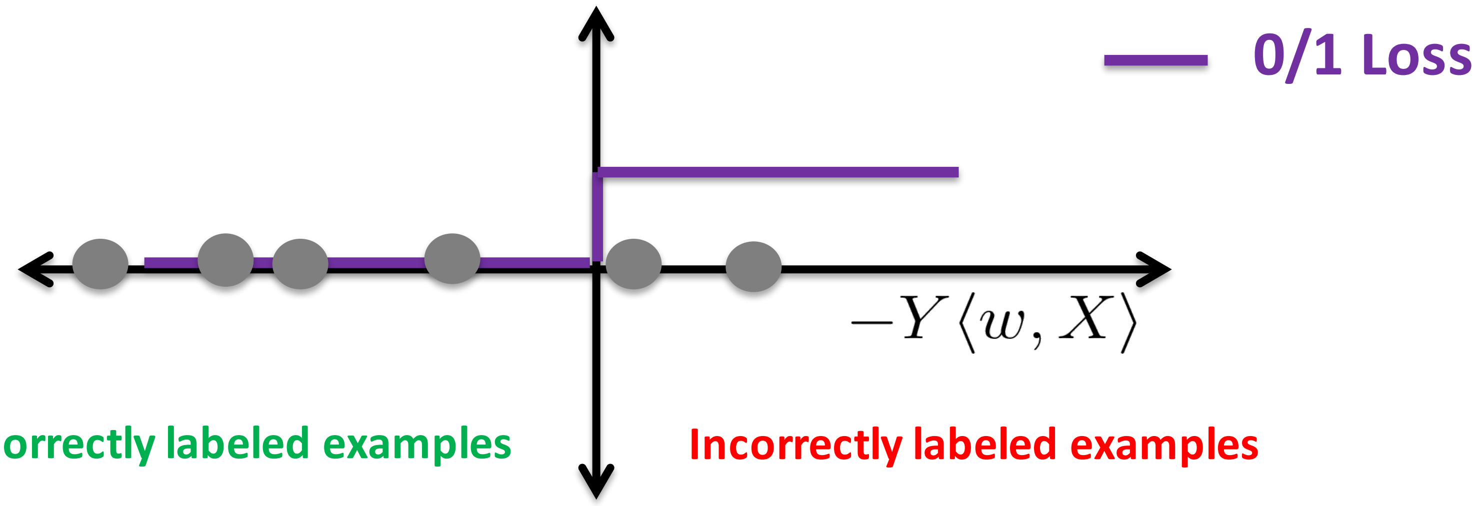
$$\mathbb{P}[Y \neq \mathrm{sgn}(\langle w, X \rangle)] \longleftrightarrow \mathbb{E}[\mathbf{1}[-Y\langle w, X \rangle \geq 0]]$$

LOSS FUNCTIONS

Typically we want to measure the **0/1 Loss**:

$$\mathbb{P}[Y \neq \mathrm{sgn}(\langle w, X \rangle)] \longleftrightarrow \mathbb{E}[\mathbf{1}[-Y\langle w, X\rangle \geq 0]]$$

Now we can visualize what's happening

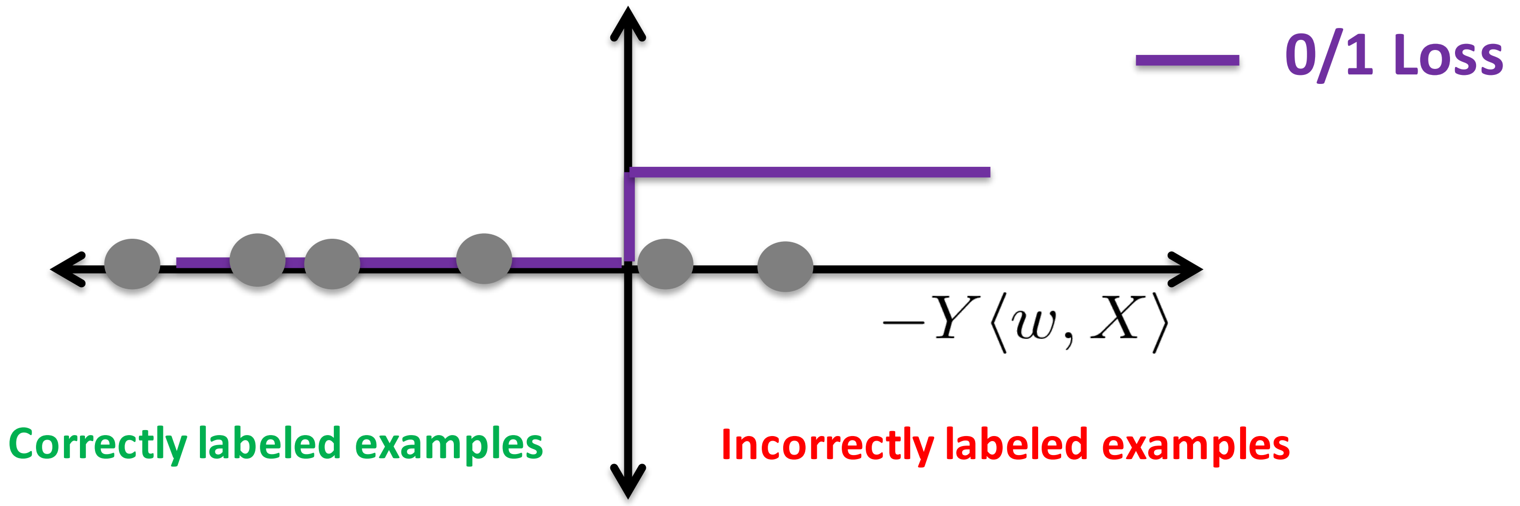


LOSS FUNCTIONS

Typically we want to measure the **0/1 Loss**:

$$\mathbb{P}[Y \neq \mathrm{sgn}(\langle w, X \rangle)] \longleftrightarrow \mathbb{E}[\mathbf{1}[-Y\langle w, X\rangle \geq 0]]$$

Now we can visualize what's happening



The trouble is, the loss is **nonconvex** as a function of w

CONVEX SURROGATES

A standard approach is to work with a **convex surrogate**

CONVEX SURROGATES

A standard approach is to work with a **convex surrogate**

For example, the **ReLU Loss**:

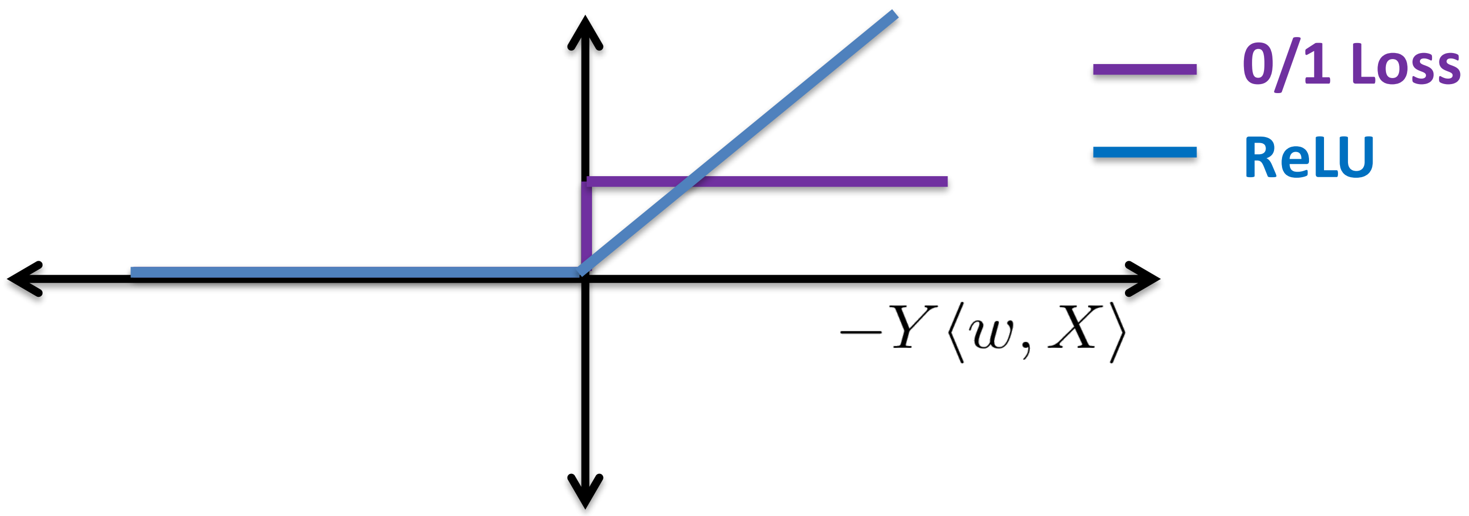
$$\mathbb{E}[|\langle w, X\rangle|\mathbf{1}[-Y\langle w, X\rangle \geq 0]]$$

CONVEX SURROGATES

A standard approach is to work with a **convex surrogate**

For example, the **ReLU Loss**:

$$\mathbb{E}[|\langle w, X\rangle|\mathbf{1}[-Y\langle w, X\rangle \geq 0]]$$

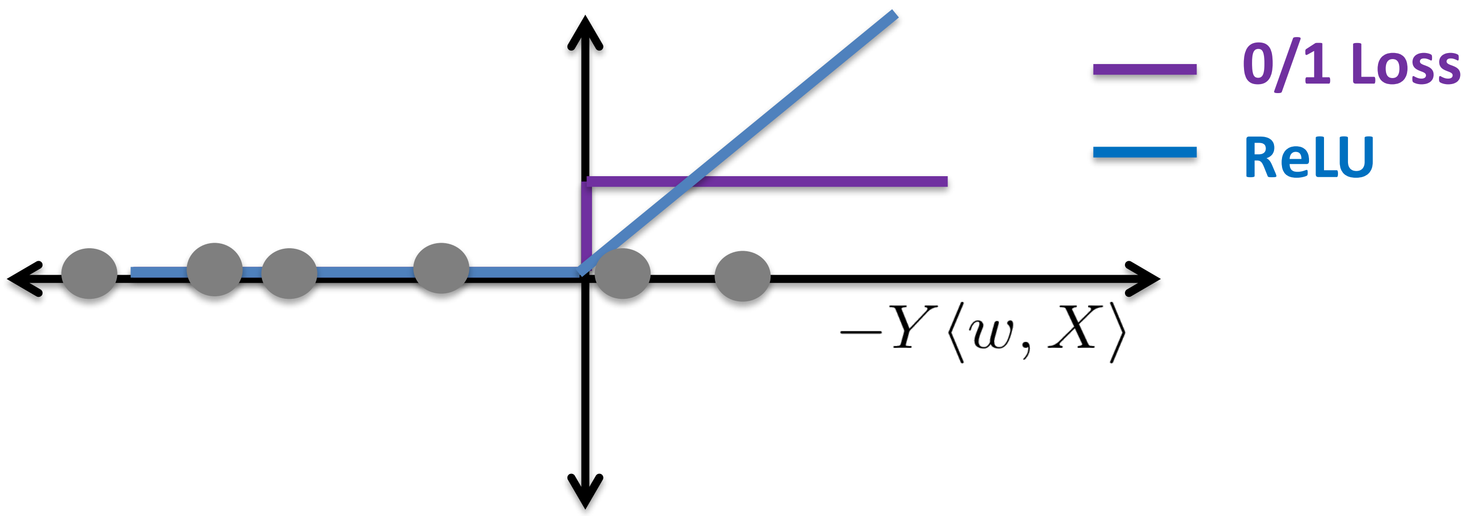


CONVEX SURROGATES

A standard approach is to work with a **convex surrogate**

For example, the **ReLU Loss**:

$$\mathbb{E}[|\langle w, X\rangle|\mathbf{1}[-Y\langle w, X\rangle \geq 0]]$$

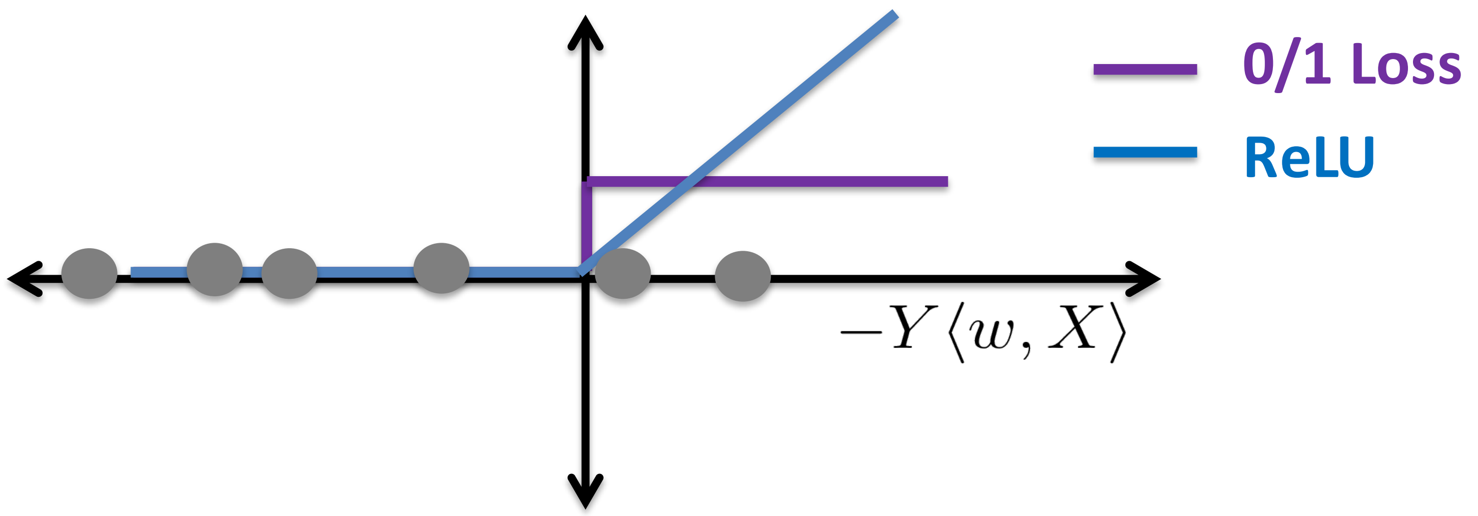


CONVEX SURROGATES

A standard approach is to work with a **convex surrogate**

For example, the **ReLU Loss**:

$$\mathbb{E}[|\langle w, X\rangle|\mathbf{1}[-Y\langle w, X\rangle \geq 0]]$$



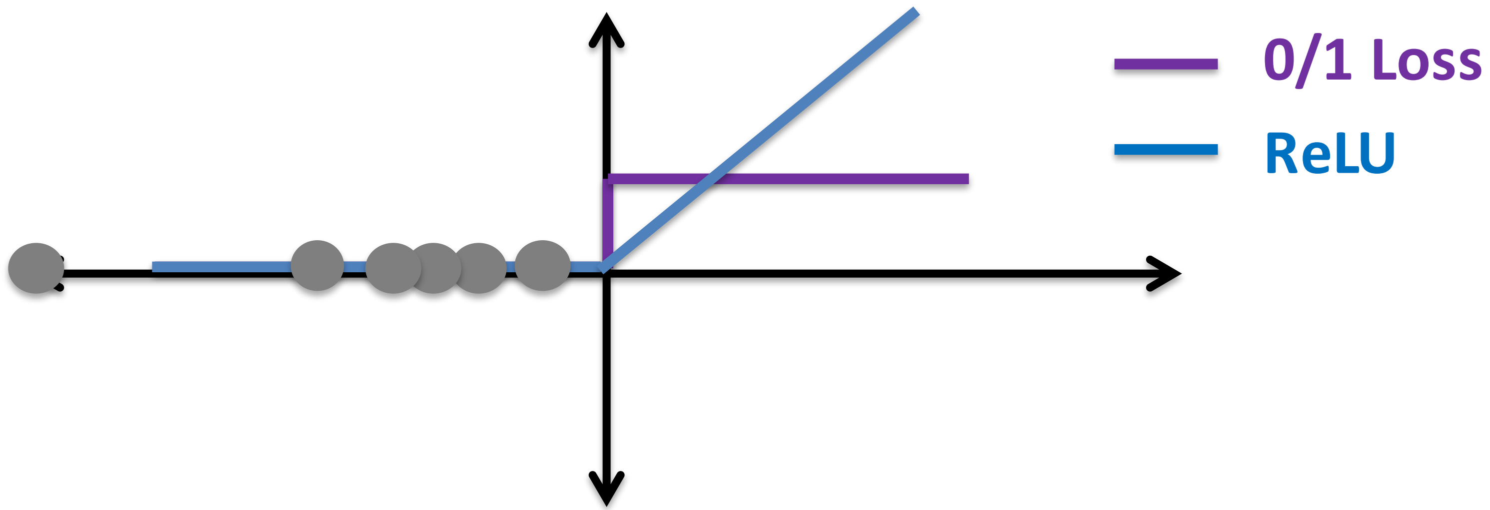
The loss function is convex, and achieving zero loss is equivalent to fitting the samples exactly

CONVEX SURROGATES, CONTINUED

What happens when we add noise?

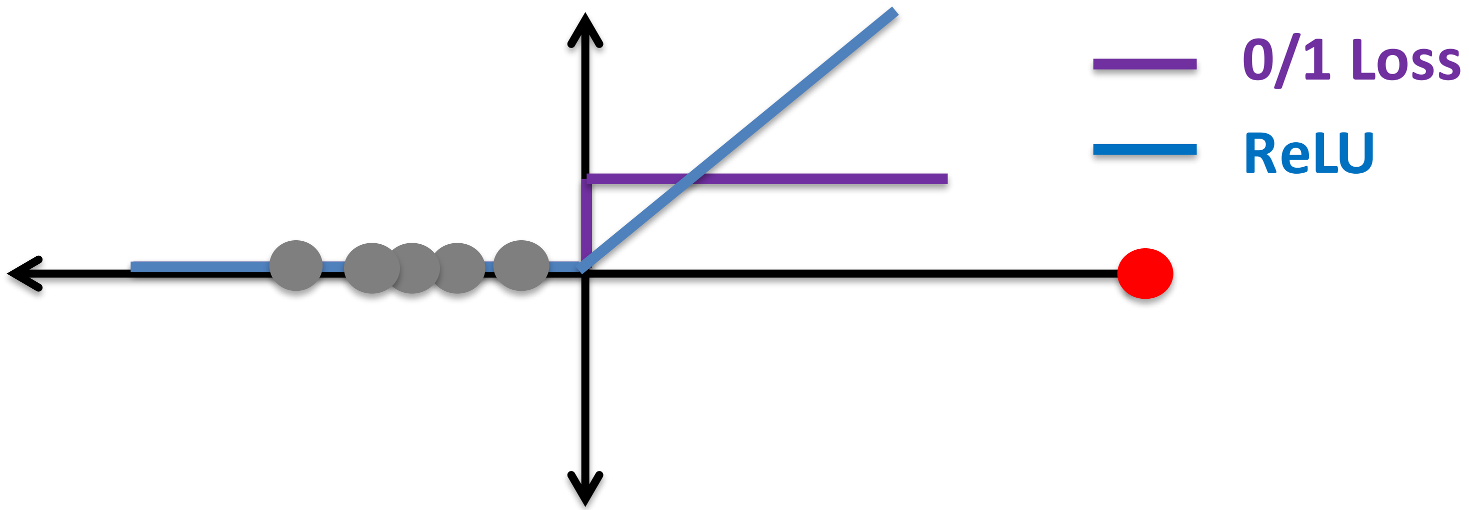
CONVEX SURROGATES, CONTINUED

What happens when we add noise?



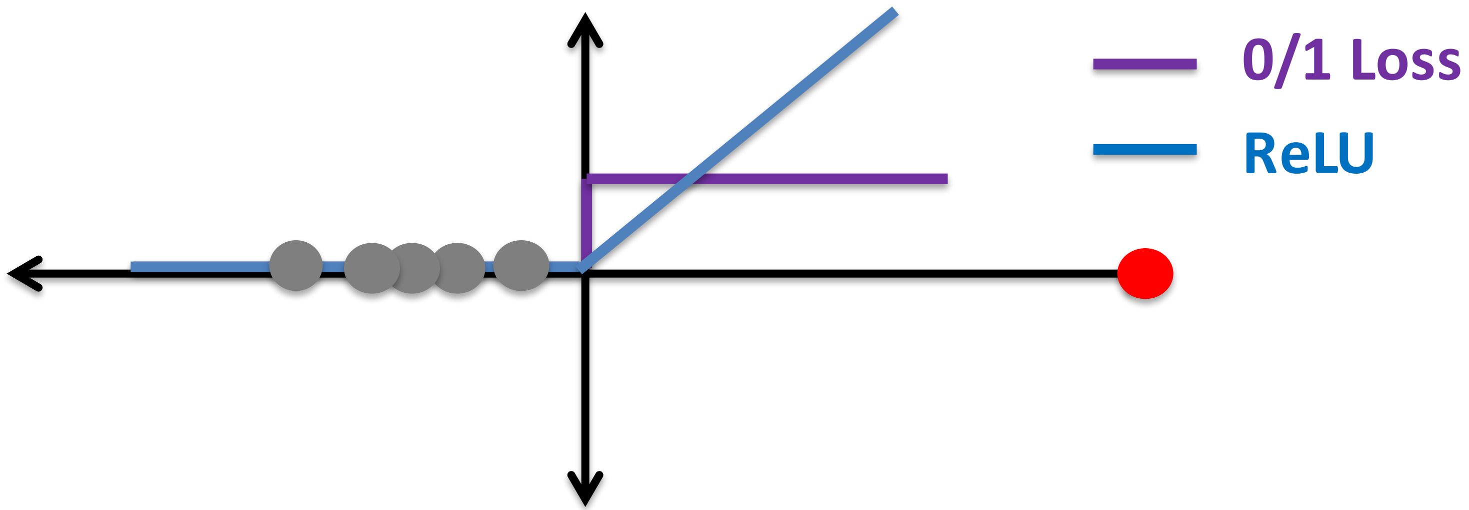
CONVEX SURROGATES, CONTINUED

What happens when we add noise?



CONVEX SURROGATES, CONTINUED

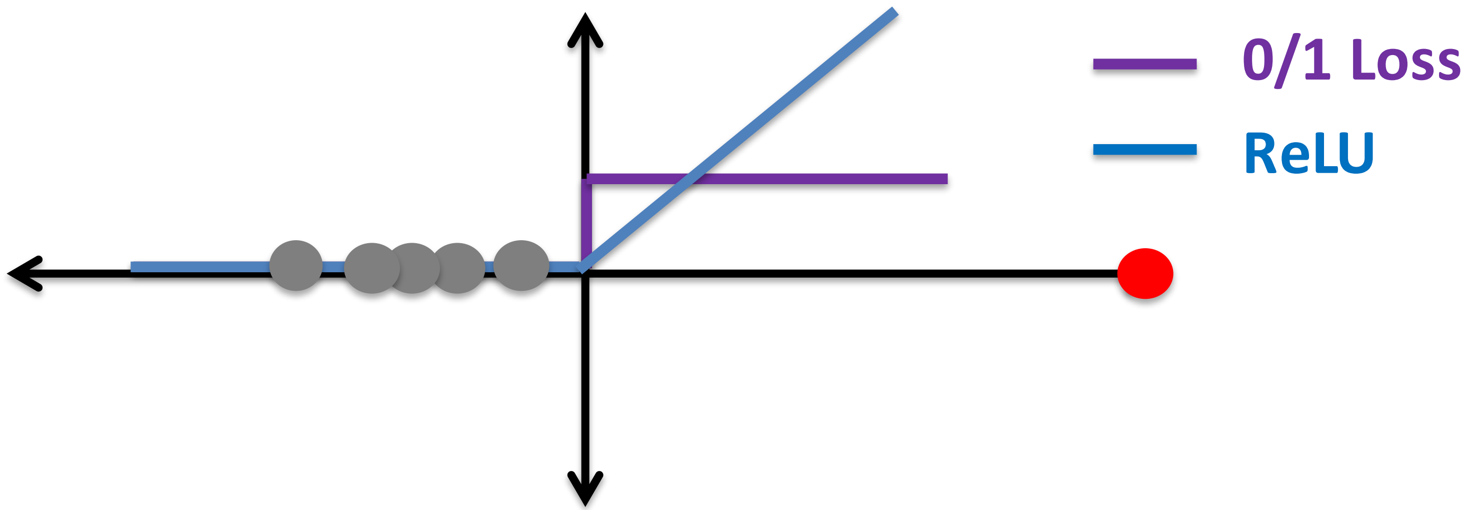
What happens when we add noise?



The ReLU loss is not representative of how many examples you are getting wrong

CONVEX SURROGATES, CONTINUED

What happens when we add noise?



The ReLU loss is not representative of how many examples you are getting wrong

You could incur a huge loss for a single mistake, if it is far from the decision boundary, or incur a tiny loss for many mistakes as long as they are close

CONVEX SURROGATES, CONTINUED

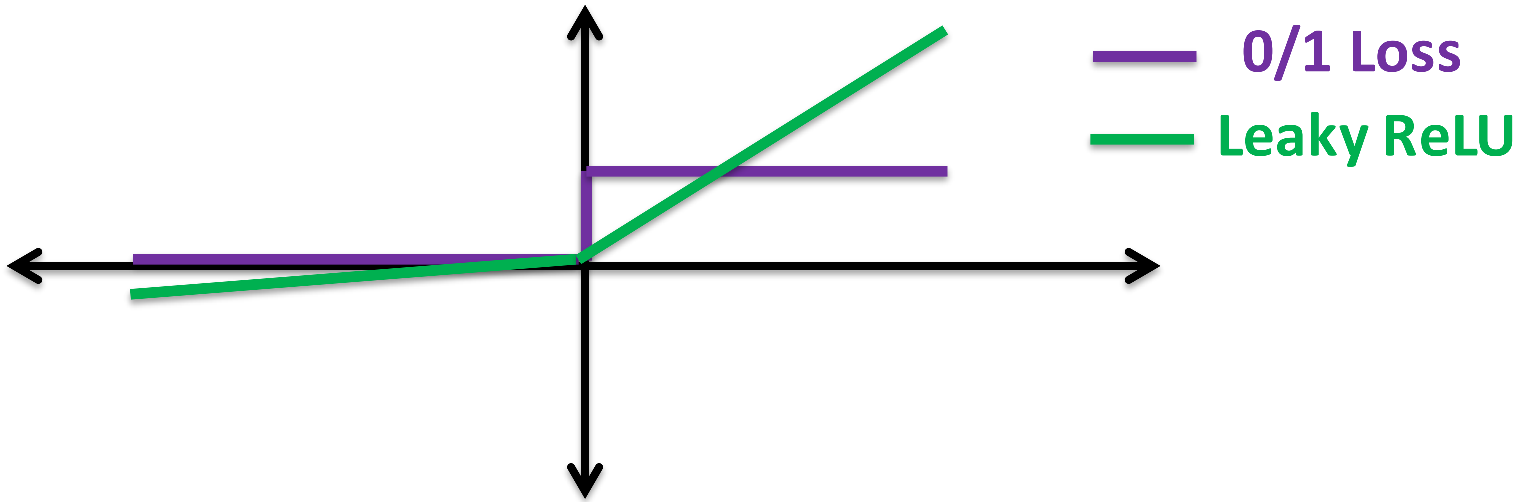
For random noise, natural approach is to use the **Leaky ReLU**:

$$\mathbb{E}[|\langle w, X\rangle|(\mathbf{1}[-Y\langle w, X\rangle \geq 0] - \lambda)]$$

CONVEX SURROGATES, CONTINUED

For random noise, natural approach is to use the **Leaky ReLU**:

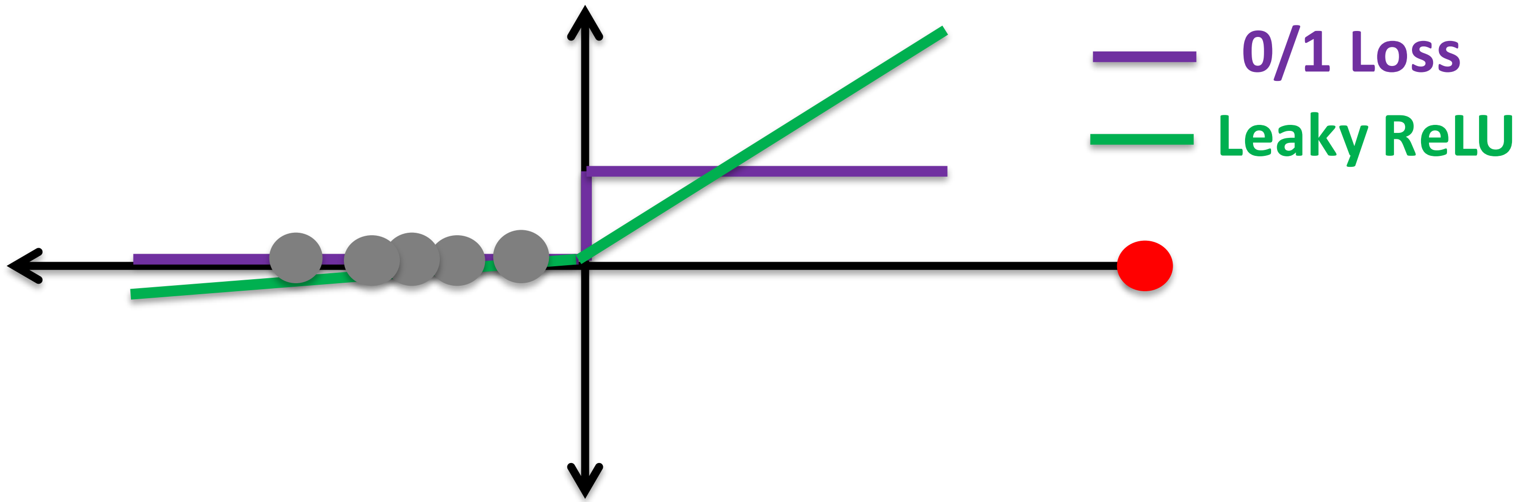
$$\mathbb{E}[|\langle w, X\rangle|(\mathbf{1}[-Y\langle w, X\rangle \geq 0] - \lambda)]$$



CONVEX SURROGATES, CONTINUED

For random noise, natural approach is to use the **Leaky ReLU**:

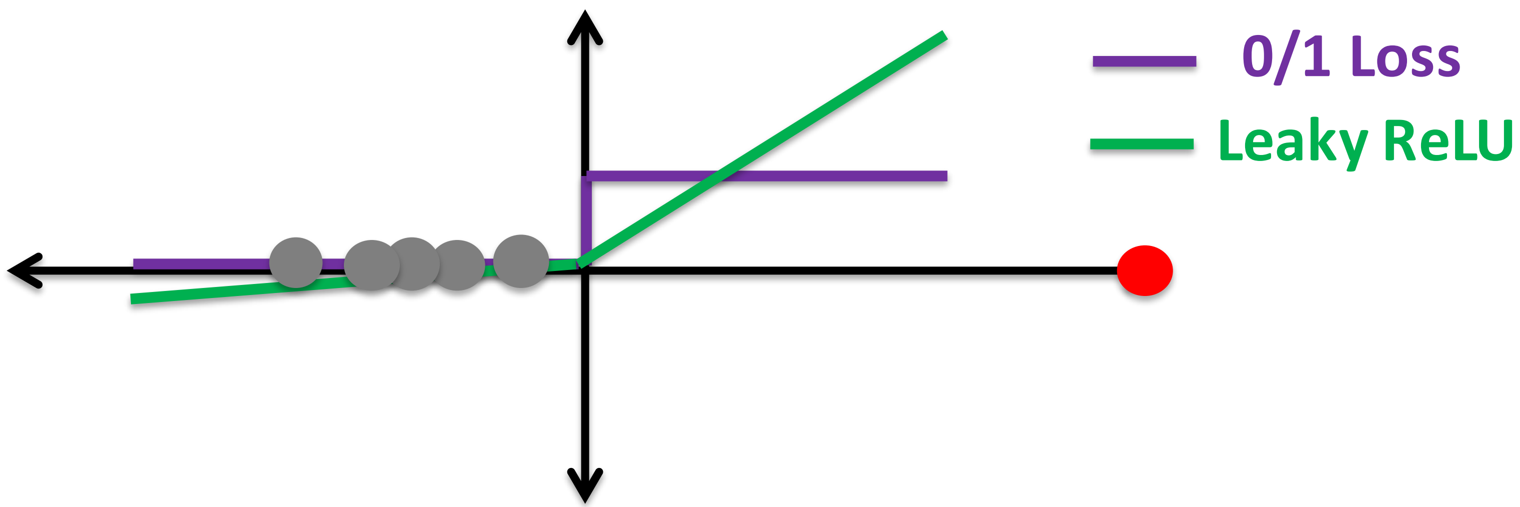
$$\mathbb{E}[|\langle w, X\rangle|(\mathbf{1}[-Y\langle w, X\rangle \geq 0] - \lambda)]$$



CONVEX SURROGATES, CONTINUED

For random noise, natural approach is to use the **Leaky ReLU**:

$$\mathbb{E}[|\langle w, X\rangle|(\mathbf{1}[-Y\langle w, X\rangle \geq 0] - \lambda)]$$



Intuition: For examples far from decision boundary, the gain when you get it right **offsets** the loss when its label is flipped (on average)

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- A Two-Player Game
- The Algorithm and Convergence

Part III: Experiments and Fairness

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- **A Two-Player Game**
- The Algorithm and Convergence

Part III: Experiments and Fairness

A GENERAL FRAMEWORK

Consider the following two-player game

$$\min_{\|w\| \leq 1} \max_{c} \mathbb{E}[c(X)\ell_\lambda(-Y\langle w, X\rangle)]$$

Leaky ReLU

where c ranges over all distributions

A GENERAL FRAMEWORK

Consider the following two-player game

$$\min_{\|w\| \leq 1} \max_{c} \mathbb{E}[c(X)\ell_\lambda(-Y\langle w, X\rangle)]$$

Leaky ReLU

where c ranges over all distributions

Intuition: The true hypothesis does well on any region of space, and the max-player looks for a region where the min-player is doing the worst

A GENERAL FRAMEWORK

Consider the following two-player game

$$\min_{\|w\| \leq 1} \max_{c} \mathbb{E}[c(X)\ell_\lambda(-Y\langle w, X\rangle)]$$

Leaky ReLU

where c ranges over all distributions

Intuition: The true hypothesis does well on any region of space, and the max-player looks for a region where the min-player is doing the worst

While you might do well overall according to the Leaky ReLU, because the adversary added less noise, the max player can always restrict to where you are doing poorly

A GENERAL FRAMEWORK

Consider the following two-player game

$$\min_{\|w\| \leq 1} \max_{c} \mathbb{E}[c(X)\ell_\lambda(-Y\langle w, X\rangle)]$$

Leaky ReLU

where c ranges over all distributions

Intuition: The true hypothesis does well on any region of space, and the max-player looks for a region where the min-player is doing the worst

Claim: The optimal solution for the min-player is w^*

A GENERAL FRAMEWORK

Consider the following two-player game

$$\min_{\|w\| \leq 1} \max_{c} \mathbb{E}[c(X)\ell_\lambda(-Y\langle w, X\rangle)]$$

Leaky ReLU

where c ranges over all distributions

Intuition: The true hypothesis does well on any region of space, and the max-player looks for a region where the min-player is doing the worst

Claim: The optimal solution for the min-player is w^*

Unfortunately, optimizing over the max-players strategies is both statistically and computationally hard

A GENERAL FRAMEWORK, CONTINUED

Instead we work with a relaxation where the max-player can only restrict the distribution to **slabs along the current w**

$$\min_{\|w\| \leq 1} \max_{r > 0} \; \mathbb{E}[\ell_\lambda(-Y\langle w, X\rangle) | -r \leq \langle w, X\rangle \leq r]$$

A GENERAL FRAMEWORK, CONTINUED

Instead we work with a relaxation where the max-player can only restrict the distribution to **slabs along the current w**

$$\min_{\|w\| \leq 1} \max_{r > 0} \mathbb{E}[\ell_\lambda(-Y\langle w, X\rangle) | -r \leq \langle w, X\rangle \leq r]$$

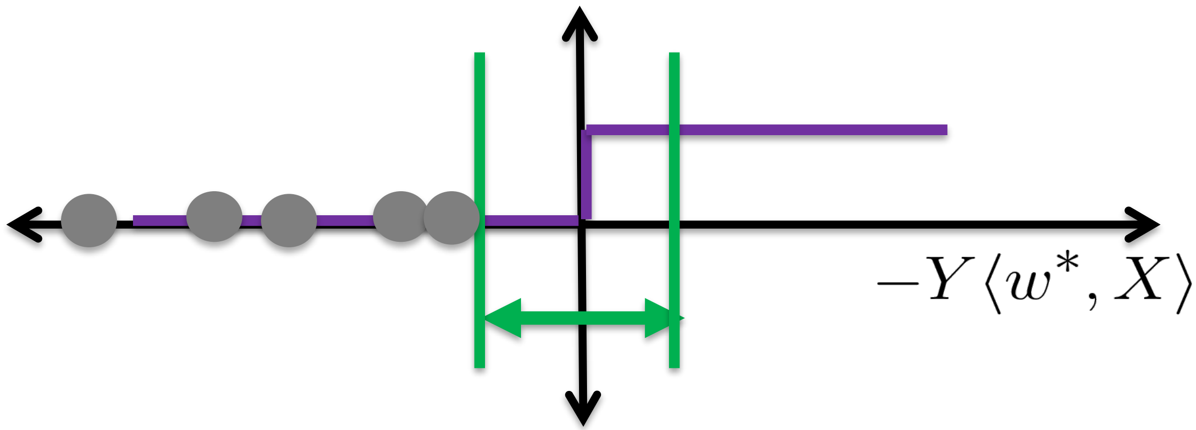
We will show that any approximate equilibrium necessarily corresponds to a hypothesis with low error

ANALYZING THE GAME

Definition: The **margin** is the smallest distance of any example from the true decision boundary

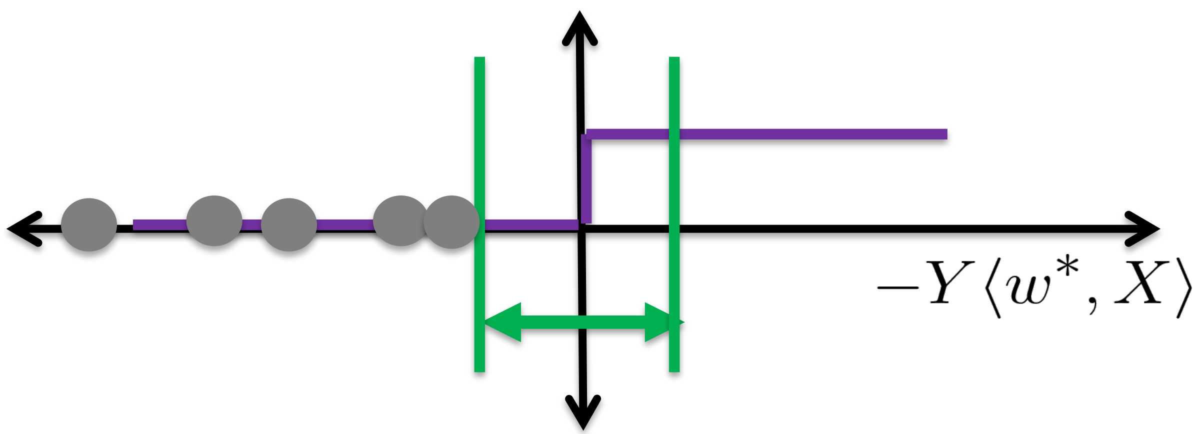
ANALYZING THE GAME

Definition: The **margin** is the smallest distance of any example from the true decision boundary, i.e.



ANALYZING THE GAME

Definition: The **margin** is the smallest distance of any example from the true decision boundary, i.e.



Key Lemma #1 [Diakonikolas et al.]: In the Massart noise model, for any $\lambda \geq \eta$ and distribution on X with margin γ

$$L_\lambda(w^*) \leq -\gamma(\lambda - \mathrm{err}(w^*))$$

Leaky ReLU loss on distribution

PROOF OF LEMMA 1

Proof: The key is to first condition on X, then randomness of noise

$$L_\lambda(w^*) = \mathbb{E}\left[\Big(\mathbb{P}[\mathrm{sgn}(\langle w^*, X\rangle) \neq Y | X] - \lambda\Big) |\langle w^*, X\rangle|\right]$$

PROOF OF LEMMA 1

Proof: The key is to first condition on X, then randomness of noise

$$L_\lambda(w^*) = \mathbb{E}\Big[\Big(\underbrace{\mathbb{P}[\mathrm{sgn}(\langle w^*, X\rangle) \neq Y | X]}_{\leq \eta \leq \lambda} - \lambda\Big)|\langle w^*, X\rangle|\Big]$$

PROOF OF LEMMA 1

Proof: The key is to first condition on X, then randomness of noise

$$L_\lambda(w^*) = \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w^*, X\rangle) \neq Y | X] - \lambda\Big)|\langle w^*, X\rangle|\Big]$$

$$\leq -\gamma(\lambda - \mathrm{err}(w^*))$$

PROOF OF LEMMA 1

Proof: The key is to first condition on X, then randomness of noise

$$L_\lambda(w^*) = \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w^*, X\rangle) \neq Y | X] - \lambda\Big) |\langle w^*, X\rangle|\Big]$$

$$\leq -\gamma(\lambda - \mathrm{err}(w^*)) \quad \blacksquare$$

PROOF OF LEMMA 1

Proof: The key is to first condition on X, then randomness of noise

$$L_\lambda(w^*) = \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w^*, X\rangle) \neq Y|X]-\lambda\Big)|\langle w^*, X\rangle|\Big]$$

$$\leq -\gamma(\lambda - \mathrm{err}(w^*)) \quad \blacksquare$$

Thus the true direction achieves small loss

PROOF OF LEMMA 1

Proof: The key is to first condition on X, then randomness of noise

$$L_\lambda(w^*) = \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w^*, X\rangle) \neq Y|X] - \lambda\Big)|\langle w^*, X\rangle|\Big]$$

$$\leq -\gamma(\lambda - \mathrm{err}(w^*)) \quad \blacksquare$$

Thus the true direction achieves small loss

Moreover, this is true even if we change the distribution by restricting to a part of the domain

PROOF OF LEMMA 1

Proof: The key is to first condition on X, then randomness of noise

$$L_\lambda(w^*) = \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w^*, X\rangle) \neq Y | X] - \lambda\Big)|\langle w^*, X\rangle|\Big]$$

$$\leq -\gamma(\lambda - \mathrm{err}(w^*)) \quad \blacksquare$$

Thus the true direction achieves small loss

Moreover, this is true even if we change the distribution by restricting to a part of the domain – **not true in agnostic learning**

ANALYZING THE GAME, CONTINUED

Key Lemma #2 (simplified): In the Massart noise model, suppose that $\mathrm{err}(w) \geq \lambda$. Then there is some slab $S(w, r)$ with

$$L_{\lambda}^{S(w,r)}(w) \geq 0$$

Leaky ReLU loss on distribution conditioned on being in S(w, r)

ANALYZING THE GAME, CONTINUED

Key Lemma #2 (simplified): In the Massart noise model, suppose that $\mathrm{err}(w) \geq \lambda$. Then there is some slab $S(w, r)$ with

$$L_\lambda^{S(w,r)}(w) \geq 0$$

Leaky ReLU loss on distribution conditioned on being in S(w, r)

If the current direction w does not achieve small enough error, then the max-player can do well in the game

ANALYZING THE GAME, CONTINUED

Key Lemma #2 (simplified): In the Massart noise model, suppose that $\mathrm{err}(w) \geq \lambda$. Then there is some slab $S(w, r)$ with

$$L_\lambda^{S(w,r)}(w) \geq 0$$

Leaky ReLU loss on distribution conditioned on being in S(w, r)

If the current direction w does not achieve small enough error, then the max-player can do well in the game

Thus doing well, with respect to the min-player, is equivalent to achieving small error

PROOF OF LEMMA 2

Proof: Suppose, for the sake of contradiction, there is no such slab.

PROOF OF LEMMA 2

Proof: Suppose, for the sake of contradiction, there is no such slab. Then for every r

$$0 > \mathbb{E}\left[\left(\mathbf{1}[\operatorname{sgn}(\langle w, X\rangle) \neq Y] - \lambda\right)|\langle w, X\rangle|\mathbf{1}[|\langle w, X, \rangle| \leq r]\right]$$

This is just the loss times the indicator for the the slab.

PROOF OF LEMMA 2

Proof: Suppose, for the sake of contradiction, there is no such slab. Then for every r

$$0 > \mathbb{E}\Big[\Big(\mathbf{1}[\mathrm{sgn}(\langle w, X\rangle) \neq Y] - \lambda\Big)|\langle w, X\rangle|\mathbf{1}[|\langle w, X, \rangle| \leq r]\Big]$$

This is just the loss times the indicator for the the slab. Now using

$$x = \int_0^\infty \mathbf{1}[y < x]dy \text{ for } x > 0$$

and subconditioning, the right hand side is …

PROOF OF LEMMA 2

Proof: Suppose, for the sake of contradiction, there is no such slab. Then for every r

$$0 > \mathbb{E}\Big[\Big(\mathbf{1}[\mathrm{sgn}(\langle w, X\rangle) \neq Y] - \lambda\Big)|\langle w, X\rangle|\mathbf{1}[|\langle w, X, \rangle| \leq r]\Big]$$

This is just the loss times the indicator for the the slab. Now using

$$x = \int_0^\infty \mathbf{1}[y < x]dy \text{ for } x > 0$$

and subconditioning, the right hand side is

$$= \int_0^\infty \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w, X\rangle) \neq Y|X] - \lambda\Big)\mathbf{1}[s < |\langle w, X\rangle| \leq r]\Big]ds$$

This implies that for all r there is s(r) < r with

$$0 > \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w, X\rangle) \neq Y | X] - \lambda\Big)\mathbf{1}[s(r) < |\langle w, X\rangle| \leq r]\Big]$$

This implies that for all r there is s(r) < r with

$$0 > \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w, X\rangle) \neq Y|X] - \lambda\Big)\mathbf{1}[s(r) < |\langle w, X\rangle| \leq r]\Big]$$

Rearranging and dividing by the prob. of being in the slab gives

$$\lambda > \mathbb{P}\Big[\mathrm{sgn}(\langle w, X\rangle \neq Y)\Big| s(r) < |\langle w, X\rangle| \leq r\Big]$$

This implies that for all r there is s(r) < r with

$$0 > \mathbb{E}\left[\Big(\mathbb{P}[\mathrm{sgn}(\langle w, X\rangle) \neq Y|X] - \lambda\Big)\mathbf{1}[s(r) < |\langle w, X\rangle| \leq r]\right]$$

Rearranging and dividing by the prob. of being in the slab gives

$$\lambda > \mathbb{P}\left[\mathrm{sgn}(\langle w, X\rangle \neq Y)\Big| s(r) < |\langle w, X\rangle| \leq r\right]$$

Now chaining together these regions, disjointly, implies

$$\lambda > \mathrm{err}(w)$$

This implies that for all r there is s(r) < r with

$$0 > \mathbb{E}\Big[\Big(\mathbb{P}[\mathrm{sgn}(\langle w, X\rangle) \neq Y | X] - \lambda\Big)\mathbf{1}[s(r) < |\langle w, X\rangle| \leq r]\Big]$$

Rearranging and dividing by the prob. of being in the slab gives

$$\lambda > \mathbb{P}\Big[\mathrm{sgn}(\langle w, X\rangle \neq Y)\Big| s(r) < |\langle w, X\rangle| \leq r\Big]$$

Now chaining together these regions, disjointly, implies

$$\lambda > \mathrm{err}(w)$$

which completes the proof by contradiction. $\blacksquare$

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- A Two-Player Game
- The Algorithm and Convergence

Part III: Experiments and Fairness

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- A Two-Player Game
- **The Algorithm and Convergence**

Part III: Experiments and Fairness

THE ALGORITHM

Now how do we find a good strategy for the min-player?

THE ALGORITHM

Now how do we find a good strategy for the min-player?

- **Initialize** w to a vector in the unit ball
- **Repeat**
  - **Max-Player** finds the slab $S(w, r^*)$ that maximizes the loss $L_\lambda^{S(w,r^*)}$. If the loss is $\leq \epsilon$ then **return** w
  - **Min-Player** takes a step in the direction $-g$ where

$$g = \nabla L_\lambda^{S(w,r^*)}$$

and projects back into the unit ball

THE ALGORITHM

Now how do we find a good strategy for the min-player?

- **Initialize** w to a vector in the unit ball
- **Repeat**
  - **Max-Player** finds the slab $S(w, r^*)$ that maximizes the loss $L_\lambda^{S(w,r^*)}$. If the loss is $\leq \epsilon$ then **return** w
  - **Min-Player** takes a step in the direction $-g$ where
$$g = \nabla L_\lambda^{S(w,r^*)}$$
and projects back into the unit ball

Full version needs to use the empirical loss, and restrict the max-player to search only over slabs with nonnegligible mass

BOUNDING THE NUMBER OF ITERATIONS

The key point is that by convexity we have

$$L_\lambda^{S(w,r^*)}(w) - L_\lambda^{S(w,r^*)}(w^*) \leq \langle -g, w^* - w \rangle$$

BOUNDING THE NUMBER OF ITERATIONS

The key point is that by convexity we have

$$L_{\lambda}^{S(w,r^*)}(w) - L_{\lambda}^{S(w,r^*)}(w^*) \leq \langle -g, w^* - w \rangle$$

So whenever we incur more loss than the true direction w*,
we are incurring regret in the sense of **online convex optimization***

BOUNDING THE NUMBER OF ITERATIONS

The key point is that by convexity we have

$$L_\lambda^{S(w,r^*)}(w) - L_\lambda^{S(w,r^*)}(w^*) \leq \langle -g, w^* - w \rangle$$

So whenever we incur more loss than the true direction w*, we are incurring regret in the sense of **online convex optimization** *

* **i.e. in each step we play a point x from a known convex body, an adversary plays a convex function f, and we incur loss f(x) and the goal is to compete with the best point in hindsight**

BOUNDING THE NUMBER OF ITERATIONS

The key point is that by convexity we have

$$L_\lambda^{S(w,r^*)}(w) - L_\lambda^{S(w,r^*)}(w^*) \leq \langle -g, w^* - w \rangle$$

So whenever we incur more loss than the true direction w*,
we are incurring regret in the sense of **online convex optimization***

* **i.e. in each step we play a point x from a known convex body, an adversary plays a convex function f, and we incur loss f(x) and the goal is to compete with the best point in hindsight**

Finally **[Zinkevich '03]** proved that projected gradient descent achieves low regret, so this cannot happen for too many steps

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- A Two-Player Game
- The Algorithm and Convergence

Part III: Experiments and Fairness

OUTLINE

Part I: Introduction

- Random, Agnostic and Massart Noise
- Recent Results

Part II: Properly Learning Halfspaces with Massart Noise

- Loss Functions and Convex Surrogates
- A Two-Player Game
- The Algorithm and Convergence

Part III: Experiments and Fairness

EXPERIMENTS

When is this noise model useful?

EXPERIMENTS

When is this noise model useful?

UCI Adults Dataset: 48.8k individuals, 14 attributes, goal is to predict whether income is above or below $50k

EXPERIMENTS

When is this noise model useful?

UCI Adults Dataset: 48.8k individuals, 14 attributes, goal is to predict whether income is above or below $50k

We added noise *outside* a target group, and ran off-the-shelf algorithms whose goal is to maximize overall accuracy

EXPERIMENTS

When is this noise model useful?

UCI Adults Dataset: 48.8k individuals, 14 attributes, goal is to predict whether income is above or below $50k

We added noise *outside* a target group, and ran off-the-shelf algorithms whose goal is to maximize overall accuracy

Motivation: Numerous empirical studies about how the level of noise various across demographic groups e.g. in surveys

EXPERIMENTS

When is this noise model useful?

UCI Adults Dataset: 48.8k individuals, 14 attributes, goal is to predict whether income is above or below $50k

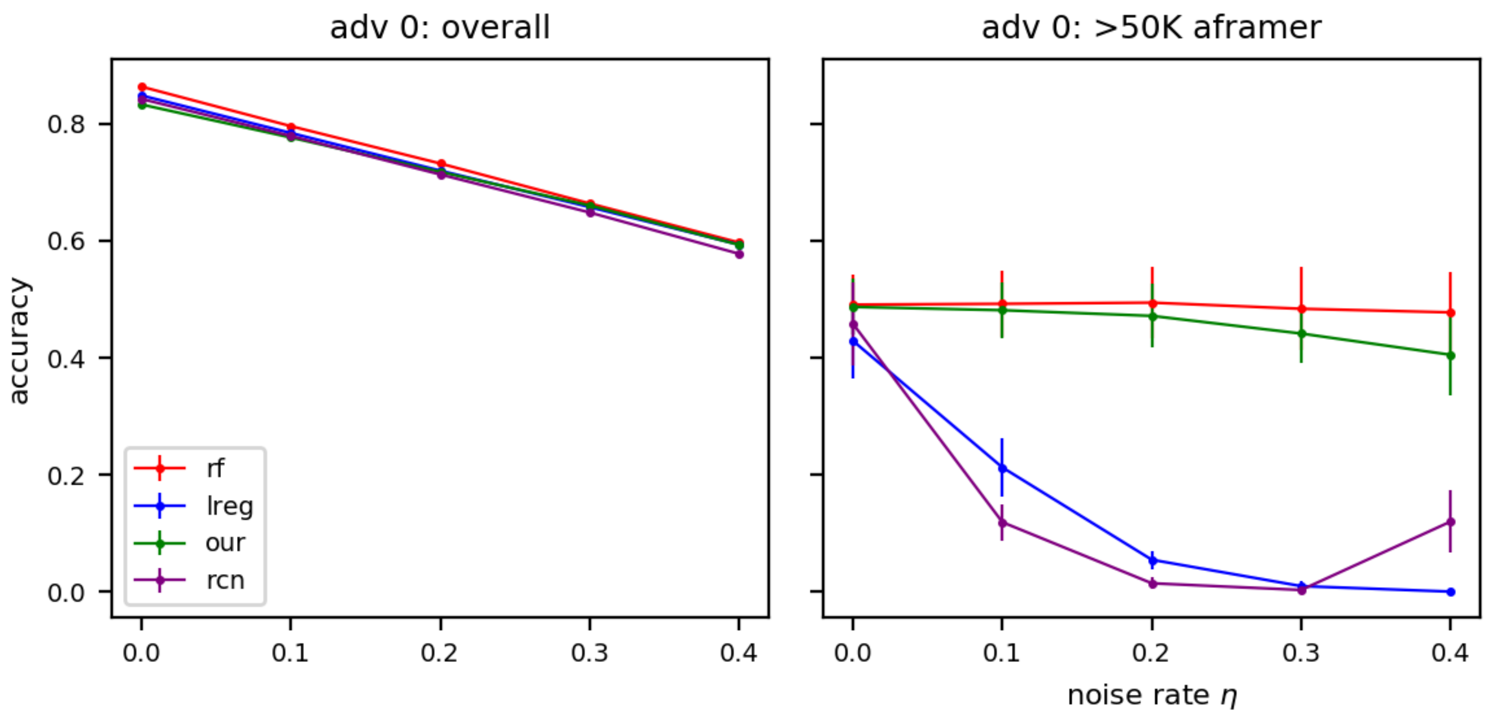
We added noise *outside* a target group, and ran off-the-shelf algorithms whose goal is to maximize overall accuracy

Motivation: Numerous empirical studies about how the level of noise various across demographic groups e.g. in surveys

We measure overall accuracy and accuracy on the part of the target group that is above $50k

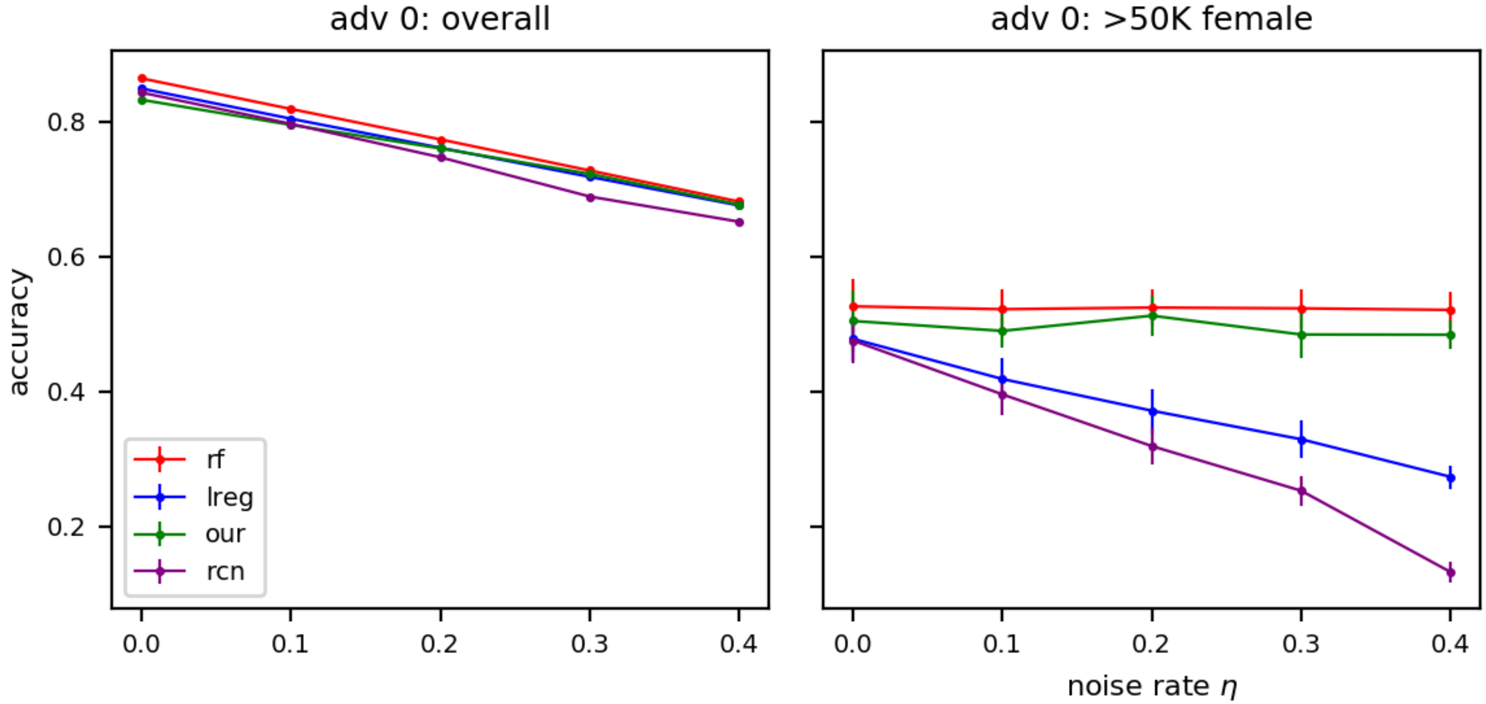
EXPERIMENTS

Target group: African Americans



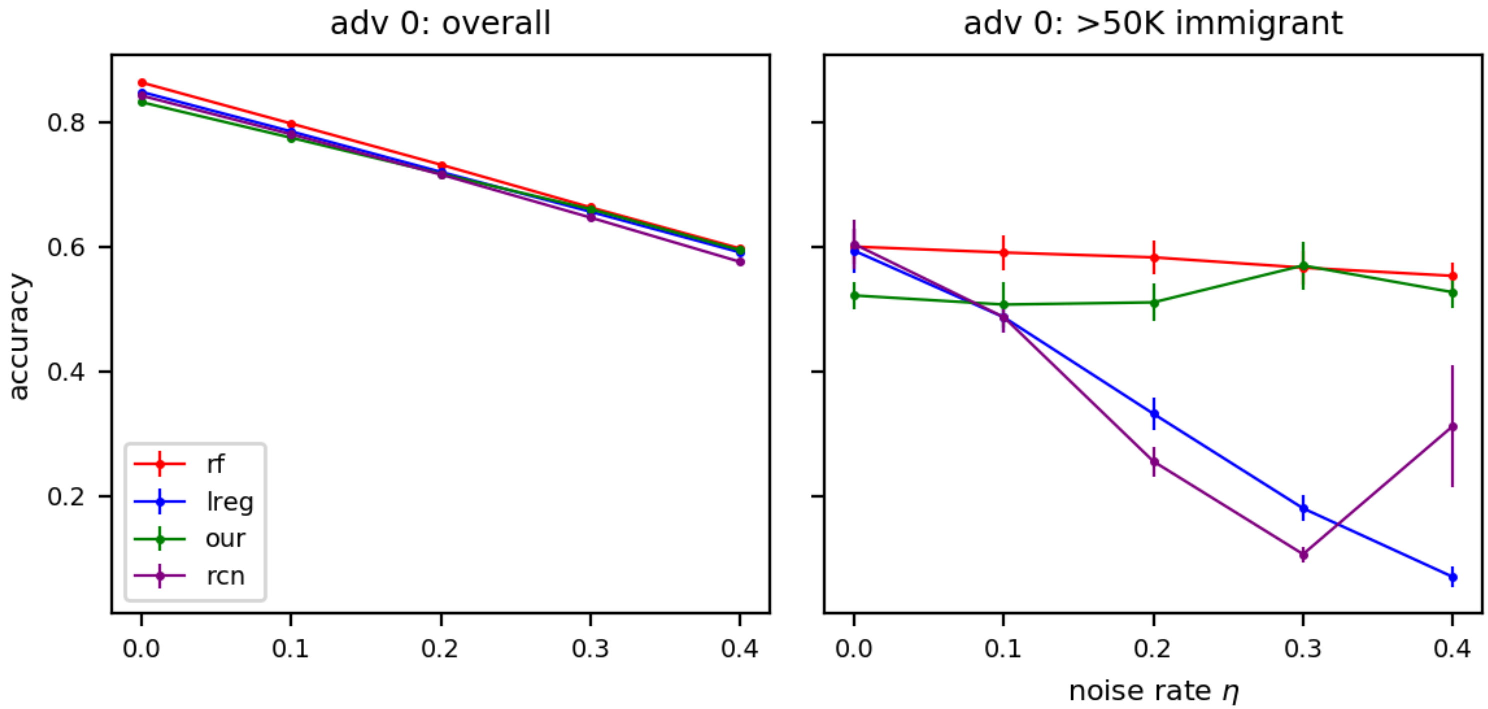
EXPERIMENTS

Target group: Female



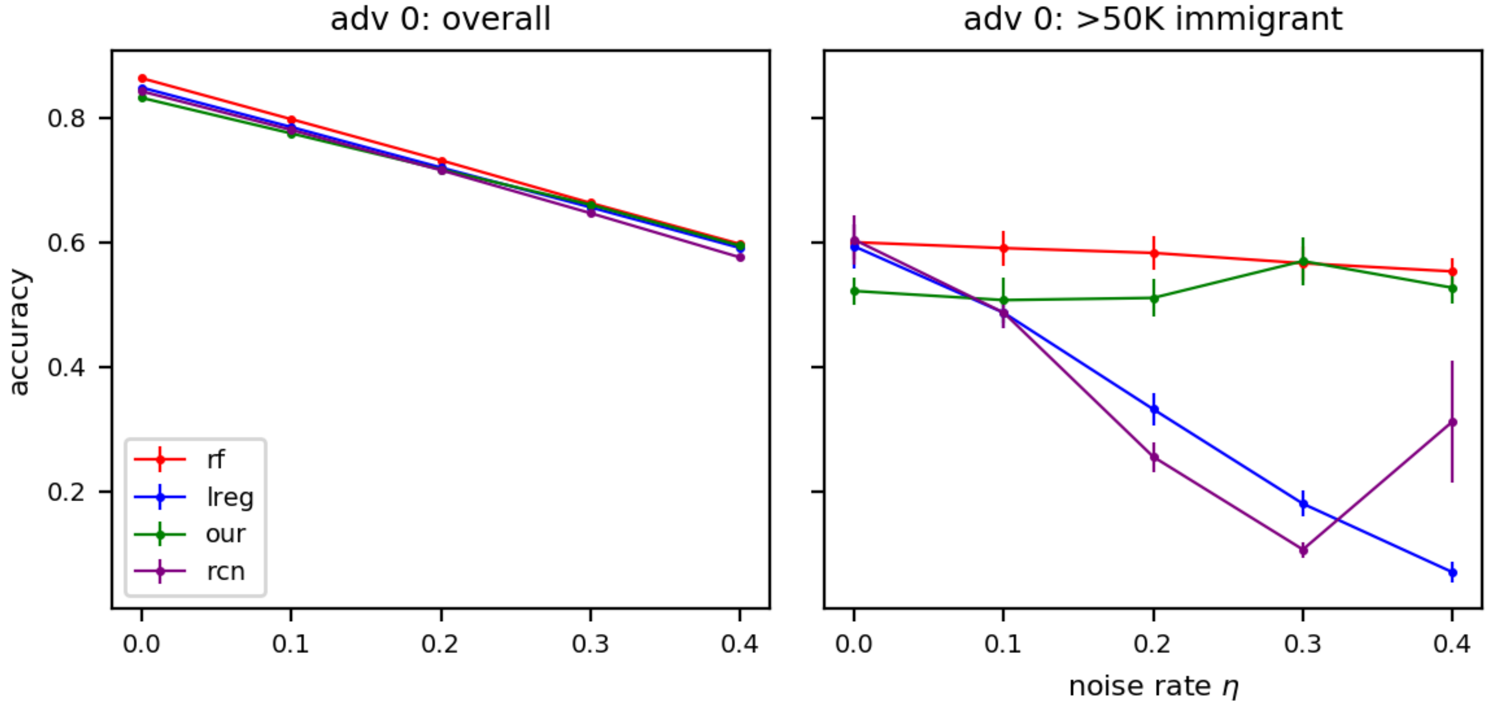
EXPERIMENTS

Target group: Immigrant



EXPERIMENTS

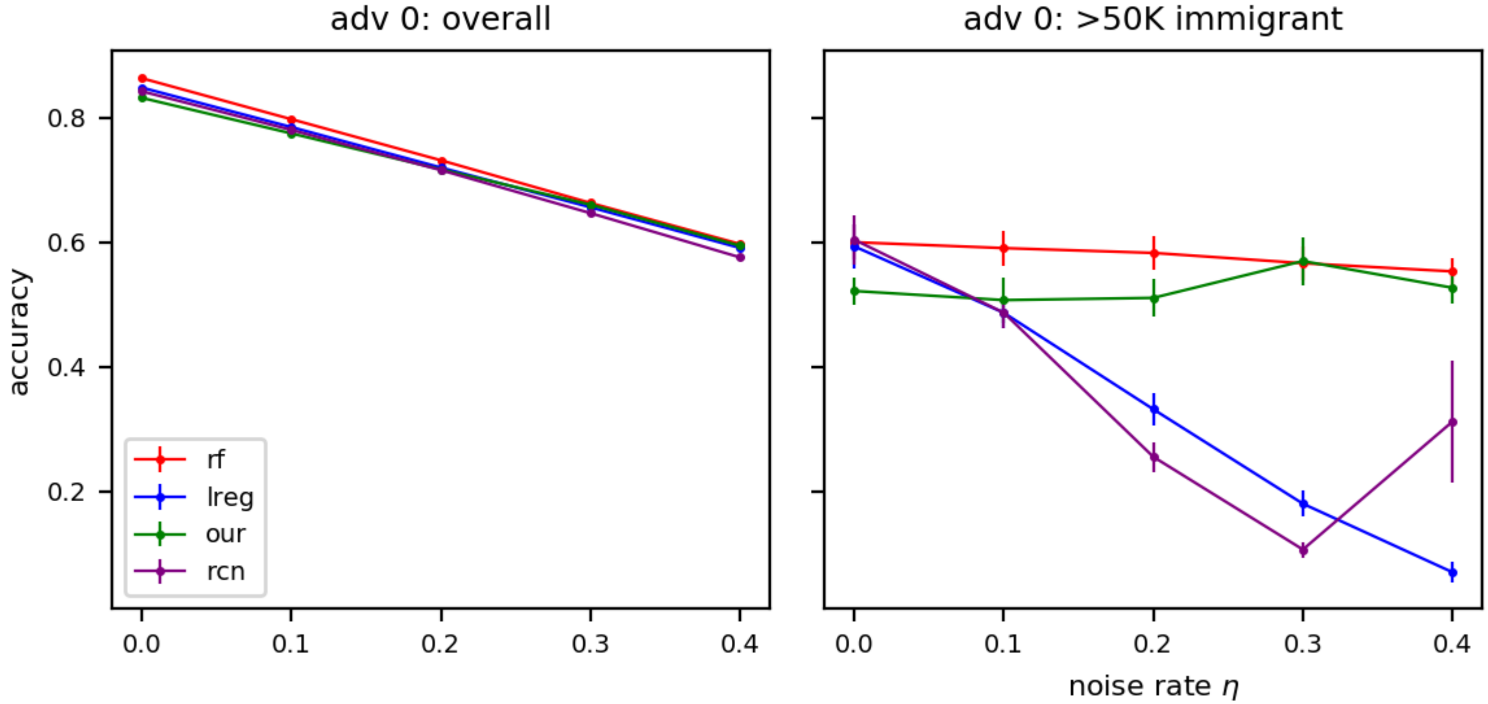
Target group: Immigrant



Many natural algorithms (e.g. logistic regression) amplify bias in the data – to achieve good overall accuracy they compromise the accuracy on various demographic groups

EXPERIMENTS

Target group: Immigrant



In contrast, our algorithm does just as well in overall accuracy minus the side effects – without knowing the identity of these protected groups

DISCUSSION

Many definitions (e.g. equalized odds, calibration) guarantee some compelling fairness criteria

DISCUSSION

Many definitions (e.g. equalized odds, calibration) guarantee some compelling fairness criteria

However they are difficult to achieve

DISCUSSION

Many definitions (e.g. equalized odds, calibration) guarantee some compelling fairness criteria

However they are difficult to achieve

From a practical standpoint, is there a sense in which making an algorithm more robust can also make it more fair?

DISCUSSION

Many definitions (e.g. equalized odds, calibration) guarantee some compelling fairness criteria

However they are difficult to achieve

From a practical standpoint, is there a sense in which making an algorithm more robust can also make it more fair?

e.g. because it can tolerate heterogenous noise

DISCUSSION

Many definitions (e.g. equalized odds, calibration) guarantee some compelling fairness criteria

However they are difficult to achieve

From a practical standpoint, is there a sense in which making an algorithm more robust can also make it more fair?

e.g. because it can tolerate heterogenous noise

Differentially private algorithms are robust, and have even been used for fairness, but our notions of robustness in learning theory tend to be quite different (not worst-case)

Summary:

- Polynomial time algorithm for learning a halfspace under Massart noise
- Extensions to Generalized Linear Models
- **Connections between Robustness and Fairness?**

Summary:

- Polynomial time algorithm for learning a halfspace under Massart noise
- Extensions to Generalized Linear Models
- **Connections between Robustness and Fairness?**

Thanks! Any Questions?