Robustness and Generalization

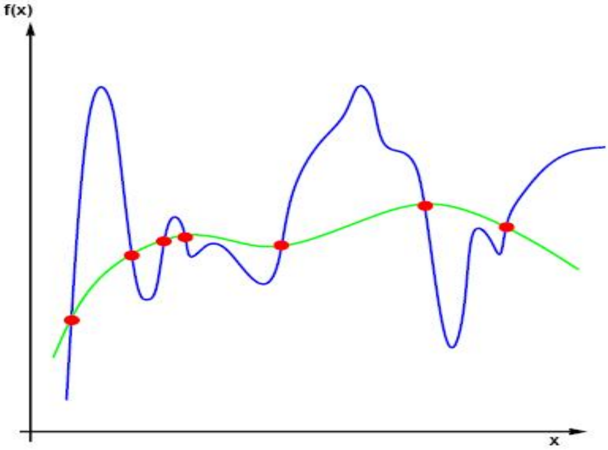
Huan Xu

The University of Texas at Austin
Department of Electrical and Computer Engineering

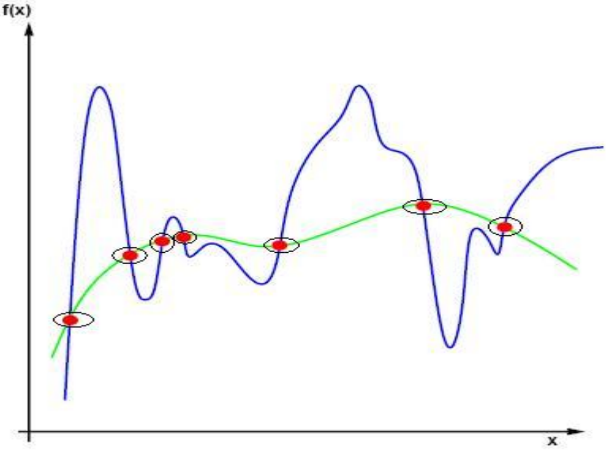
COLT, June 29, 2010

Joint work with Shie Mannor

What is Robustness?

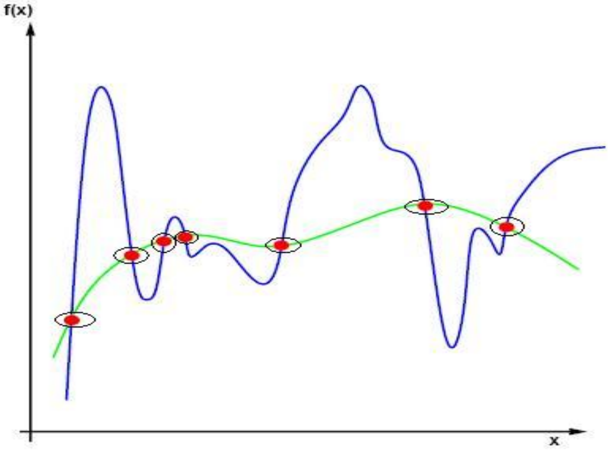


What is Robustness?



What is Robustness?

- Robustness is the property that tested on a training sample and on a similar testing sample, the performance is close.



What is Robustness?

- Robust decision making/optimization:
  - Consider a general decision problem: find v such that $\ell(v, \xi)$ is small.
  - If for $\xi' \approx \xi$, $\ell(v, \xi')$ is also small, then v is robust to the perturbation of parameter.
  - Robust optimization: $\min_v \max_{\xi' \approx \xi} \ell(v, \xi')$

What is Robustness?

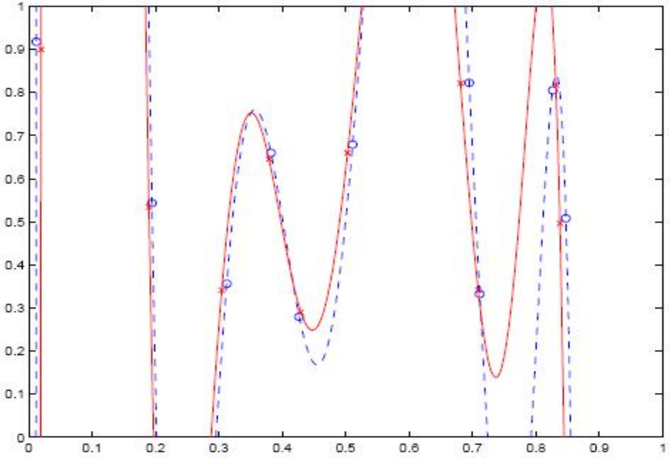
- Robust decision making/optimization:
  - Consider a general decision problem: find v such that $\ell(v, \xi)$ is small.
  - If for $\xi' \approx \xi$, $\ell(v, \xi')$ is also small, then v is robust to the perturbation of parameter.
  - Robust optimization: $\min_v \max_{\xi' \approx \xi} \ell(v, \xi')$
- Robustness in machine learning
  - Robust optimization was introduced to machine learning to handle observation noise (e.g., [Lanckriet *et al* 2003]; [Lebret and El Ghaoui 1997]; [Shivaswamy *et al* 2006]).
  - It is then discovered that SVM and Lasso can both be rewritten as robust optimization (of empirical loss), and the RO formulation implies consistency [HX, Caramanis and SM 2009; 2010].

What is Robustness?

- Robust decision making/optimization:
  - Consider a general decision problem: find v such that $\ell(v, \xi)$ is small.
  - If for $\xi' \approx \xi$, $\ell(v, \xi')$ is also small, then v is robust to the perturbation of parameter.
  - Robust optimization: $\min_v \max_{\xi' \approx \xi} \ell(v, \xi')$
- Robustness in machine learning
  - Robust optimization was introduced to machine learning to handle observation noise (e.g., [Lanckriet *et al* 2003]; [Lebret and El Ghaoui 1997]; [Shivaswamy *et al* 2006]).
  - It is then discovered that SVM and Lasso can both be rewritten as robust optimization (of empirical loss), and the RO formulation implies consistency [HX, Caramanis and SM 2009; 2010].
- This paper formalizes this observation to general learning algorithms.

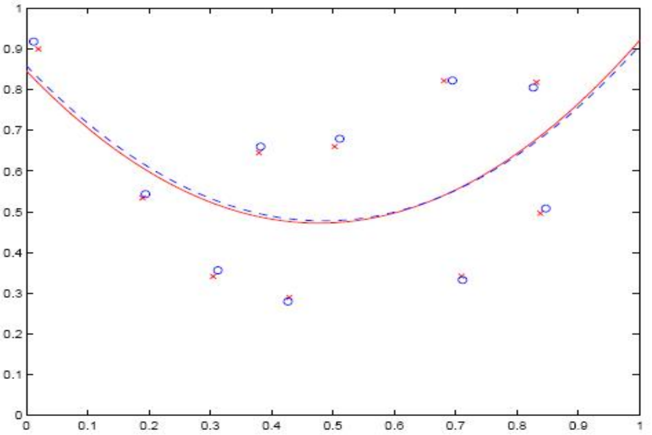
Difference with Stabiilty

Non-stable algorithm:



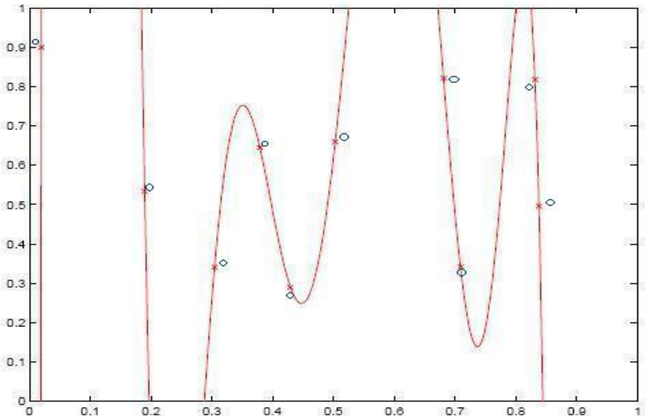
Difference with Stabiilty

Stable algorithm:



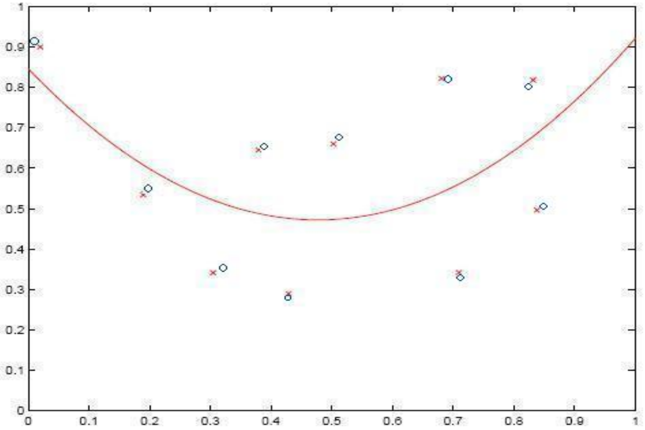
Difference with Stabiilty

Non-robust algorithm:



Difference with Stabiilty

Robust algorithm:



Outline

1. Algorithmic Robustness and Generalization Bound
2. Robust Algorithms
3. (Weak) Robustness is Necessary and Sufficient to (Asymptotic) Generalizability

Outline

1. **Algorithmic Robustness and Generalization Bound**
2. Robust Algorithms
3. (Weak) Robustness is Necessary and Sufficient to (Asymptotic) Generalizability

Notations

- Training sample set **s** of n training samples $(s_1, \cdots, s_n)$.
- $\mathcal{Z}$ and $\mathcal{H}$ are the set from which each sample is drawn, and the hypothesis set.
- $\mathcal{A}_{\mathbf{s}}$ is the hypothesis learned given training set **s**.
- For each hypothesis $h \in \mathcal{H}$ and a point $z \in \mathcal{Z}$, there is an associated loss $\ell(h, z) \in [0, M]$.

Notations

- Training sample set $\mathbf{s}$ of n training samples $(s_1, \cdots, s_n)$.
- $\mathcal{Z}$ and $\mathcal{H}$ are the set from which each sample is drawn, and the hypothesis set.
- $\mathcal{A}_{\mathbf{s}}$ is the hypothesis learned given training set $\mathbf{s}$.
- For each hypothesis $h \in \mathcal{H}$ and a point $z \in \mathcal{Z}$, there is an associated loss $\ell(h, z) \in [0, M]$.
- In supervised learning, we decompose $\mathcal{Z} = \mathcal{Y} \times \mathcal{X}$, and use $_{|x}$ and $_{|y}$ to denote the x-component and y-component of a point.

Notations

- Training sample set $\mathbf{s}$ of n training samples $(s_1, \cdots, s_n)$.
- $\mathcal{Z}$ and $\mathcal{H}$ are the set from which each sample is drawn, and the hypothesis set.
- $\mathcal{A}_{\mathbf{s}}$ is the hypothesis learned given training set $\mathbf{s}$.
- For each hypothesis $h \in \mathcal{H}$ and a point $z \in \mathcal{Z}$, there is an associated loss $\ell(h, z) \in [0, M]$.
- In supervised learning, we decompose $\mathcal{Z} = \mathcal{Y} \times \mathcal{X}$, and use $_{|x}$ and $_{|y}$ to denote the x-component and y-component of a point.
- The covering number of a metric space T: $\mathcal{N}(\epsilon, T, \rho)$

Motivating example 1: Large Margin Classifier

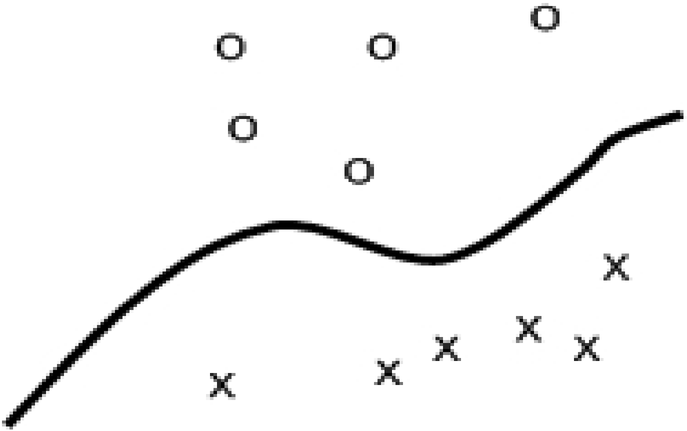
An algorithm $\mathcal{A}_{\mathbf{s}}$ has a margin γ if for $j = 1, \cdots, n$

$$\mathcal{A}_{\mathbf{s}}(x) = \mathcal{A}_{\mathbf{s}}(s_{j|x}); \quad \forall x : \|x - s_{j|x}\|_2 < \gamma.$$

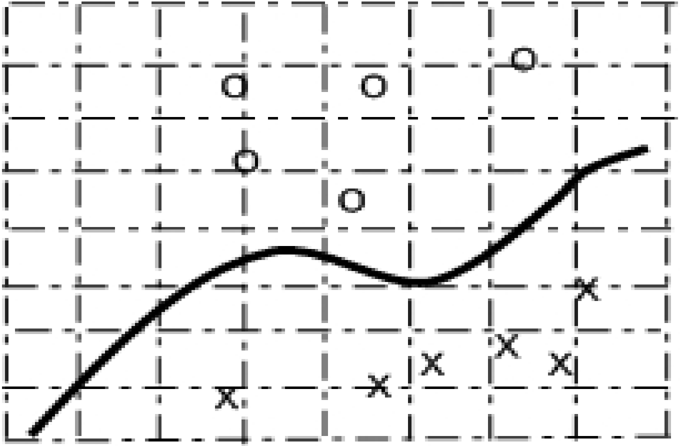
Example

Fix $\gamma > 0$ and put $K = 2\mathcal{N}(\gamma/2, \mathcal{X}, \|\cdot\|_2)$. If $\mathcal{A}_{\mathbf{s}}$ has a margin γ, then $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^K$, such that if s_j and $z \in \mathcal{Z}$ belong to a same C_i, then $|\ell(\mathcal{A}_{\mathbf{s}}, s_j) - \ell(\mathcal{A}_{\mathbf{s}}, z)| = 0$.

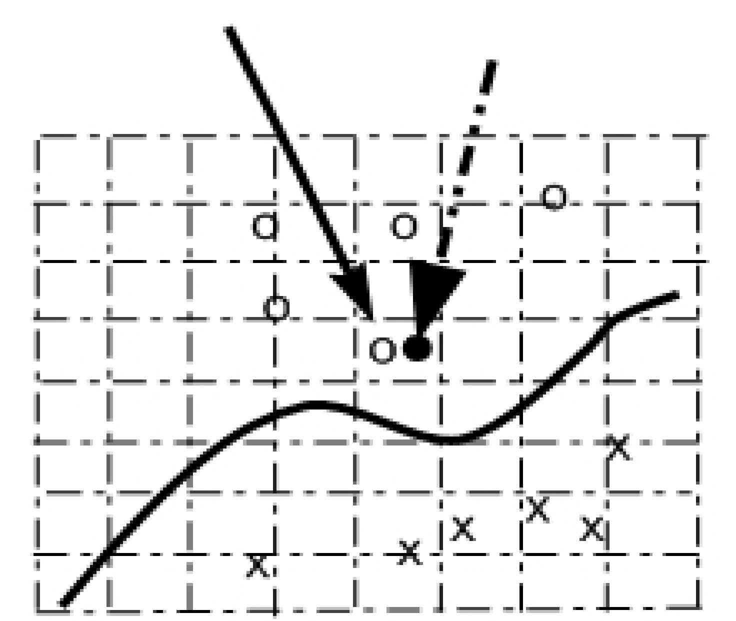
Motivating example 1: Large Margin Classifier



Motivating example 1: Large Margin Classifier



Motivating example 1: Large Margin Classifier



Motivating example 2: Linear Regression

The norm-constrained linear regression algorithm is

$$\mathcal{A}_{\mathbf{s}} = \min_{w \in \mathbb{R}^m : \|w\|_2 \leq c} \sum_{i=1}^{n} |s_{i|y} - w^\top s_{i|x}|, \qquad (0.1)$$

Example

Fix $\epsilon > 0$ and let $K = \mathcal{N}(\epsilon/2, \mathcal{X}, \|\cdot\|_2) \times \mathcal{N}(\epsilon/2, \mathcal{Y}, |\cdot|)$. Consider the norm-constrained linear regression algorithm as in (0.1). The set $\mathcal{Z}$ can be partitioned into K disjoint sets, such that if s_j and $z \in \mathcal{Z}$ belong to a same C_i, then

$$|\ell(\mathcal{A}_{\mathbf{s}}, s_j) - \ell(\mathcal{A}_{\mathbf{s}}, z)| \leq (c+1)\epsilon.$$

Algorithmic Robustness

Definition (Algorithmic Robustness)

Algorithm $\mathcal{A}$ is $(K, \epsilon(\mathbf{s}))$ robust if

- $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^{K}$;
- such that $\forall s \in \mathbf{s}$,

$$s, z \in C_i, \quad \Longrightarrow \quad |\ell(\mathcal{A}_{\mathbf{s}}, s) - \ell(\mathcal{A}_{\mathbf{s}}, z)| \leq \epsilon(\mathbf{s}). \tag{0.2}$$

Algorithmic Robustness

Definition (Algorithmic Robustness)

Algorithm $\mathcal{A}$ is $(K, \epsilon(\mathbf{s}))$ robust if

- $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^{K}$;
- such that $\forall s \in \mathbf{s}$,

$$s, z \in C_i, \quad \Longrightarrow \quad |\ell(\mathcal{A}_{\mathbf{s}}, s) - \ell(\mathcal{A}_{\mathbf{s}}, z)| \leq \epsilon(\mathbf{s}). \tag{0.2}$$

Remark:

- The definition requires that the difference between a testing sample "similar to" a training sample is small.
- The property jointly depends on the solution to the algorithm and the training set.

Generalization property of robust algorithms – the main theorem

Theorem

Let $\hat{\ell}(\cdot)$ and $\ell_{\mathrm{emp}}(\cdot)$ denote the expected loss and the training loss. If $\mathbf{s}$ consists of n i.i.d. samples, and $\mathcal{A}$ is $(K, \epsilon(\mathbf{s}))$-robust, then for any $\delta > 0$, with probability at least $1 - \delta$,

$$\left|\hat{\ell}(\mathcal{A}_{\mathbf{s}}) - \ell_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})\right| \leq \epsilon(s) + M\sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}.$$

Generalization property of robust algorithms – the main theorem

Theorem

Let $\hat{\ell}(\cdot)$ and $\ell_{\text{emp}}(\cdot)$ denote the expected loss and the training loss. If **s** *consists of n i.i.d. samples, and $\mathcal{A}$ is $(K, \epsilon(\mathbf{s}))$-robust, then for any $\delta > 0$, with probability at least $1 - \delta$,*

$$\left|\hat{\ell}(\mathcal{A}_{\mathbf{s}}) - \ell_{\text{emp}}(\mathcal{A}_{\mathbf{s}})\right| \leq \epsilon(s) + M\sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}.$$

Remark:

The bounds depends on the partitioning of the sample space.

Proof of the Main Theorem

- Let N_i be the set of index of points of $\mathbf{s}$ that fall into C_i. Then $(|N_1|, \cdots, |N_K|)$ is an IID multinomial random variable with parameters n and $(\mu(C_1), \cdots, \mu(C_K))$.

Proof of the Main Theorem

- Let N_i be the set of index of points of $\mathbf{s}$ that fall into C_i. Then $(|N_1|, \cdots, |N_K|)$ is an IID multinomial random variable with parameters n and $(\mu(C_1), \cdots, \mu(C_K))$.
- Breteganolle-Huber-Carol inequality gives

$$\Pr\left\{\sum_{i=1}^{K}\left|\frac{|N_i|}{n} - \mu(C_i)\right| \geq \lambda\right\} \leq 2^K \exp(\frac{-n\lambda^2}{2}).$$

Proof of the Main Theorem

- Let N_i be the set of index of points of **s** that fall into C_i. Then $(|N_1|, \cdots, |N_K|)$ is an IID multinomial random variable with parameters n and $(\mu(C_1), \cdots, \mu(C_K))$.
- Breteganolle-Huber-Carol inequality gives

$$\Pr\left\{\sum_{i=1}^{K}\left|\frac{|N_i|}{n} - \mu(C_i)\right| \geq \lambda\right\} \leq 2^K \exp(\frac{-n\lambda^2}{2}).$$

- Hence, with probability at least $1-\delta$,

$$\sum_{i=1}^{K}\left|\frac{|N_i|}{n} - \mu(C_i)\right| \leq \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}. \tag{0.3}$$

Proof of the Main Theorem (Cont.)

Furthermore,

$$\left|\hat{\ell}(\mathcal{A}_{\mathbf{s}}) - \ell_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})\right| = \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\mu(C_i) - \frac{1}{n}\sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}}, s_i)\right|$$

$$\leq \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\frac{|N_i|}{n} - \frac{1}{n}\sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}}, s_i)\right|$$

$$+ \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\mu(C_i) - \sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\frac{|N_i|}{n}\right|$$

Proof of the Main Theorem (Cont.)

Furthermore,

$$
\begin{aligned}
\left|\hat{\ell}(\mathcal{A}_{\mathbf{s}}) - \ell_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})\right| &= \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\mu(C_i) - \frac{1}{n}\sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}}, s_i)\right| \\
&\leq \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\frac{|N_i|}{n} - \frac{1}{n}\sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}}, s_i)\right| \\
&\quad + \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\mu(C_i) - \sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\frac{|N_i|}{n}\right|
\end{aligned}
$$

- The first term is bounded by $\left|\frac{1}{n}\sum_{i=1}^{K}\sum_{j\in N_i} \max_{z_2 \in C_i} |\ell(\mathcal{A}_{\mathbf{s}}, s_j) - \ell(\mathcal{A}_{\mathbf{s}}, z_2)|\right| \leq \epsilon(s)$.

Proof of the Main Theorem (Cont.)

Furthermore,

$$\left|\hat{\ell}(\mathcal{A}_{\mathbf{s}}) - \ell_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})\right| = \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\mu(C_i) - \frac{1}{n}\sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}}, s_i)\right|$$

$$\leq \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\frac{|N_i|}{n} - \frac{1}{n}\sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}}, s_i)\right|$$

$$+ \left|\sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\mu(C_i) - \sum_{i=1}^{K} \mathbb{E}\big(\ell(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\frac{|N_i|}{n}\right|$$

- The first term is bounded by $\left|\frac{1}{n}\sum_{i=1}^{K}\sum_{j\in N_i} \max_{z_2\in C_i} |\ell(\mathcal{A}_{\mathbf{s}}, s_j) - \ell(\mathcal{A}_{\mathbf{s}}, z_2)|\right| \leq \epsilon(s)$.
- The second term is bounded by $\left|\max_{z\in\mathcal{Z}} |\ell(\mathcal{A}_{\mathbf{s},z})| \sum_{i=1}^{K} \left|\frac{|N_i|}{n} - \mu(C_i)\right|\right| \leq M\sum_{i=1}^{K} \left|\frac{|N_i|}{n} - \mu(C_i)\right|$.

Additional Results: Pseudo Robustness

- Robustness – “similar performace” around each training sample.

Additional Results: Pseudo Robustness

- Robustness – “similar performace” around each training sample.
- Pseudo robustness – “similar performace” around some training sample:

Additional Results: Pseudo Robustness

- Robustness – “similar performace” around each training sample.
- Pseudo robustness – “similar performace” around some training sample:

Definition (Pseudo robustness:)

Algorithm $\mathcal{A}$ is $(K, \epsilon(\mathbf{s}), \hat{n}(\mathbf{s}))$ *pseudo robust* if

- $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted as $\{C_i\}_{i=1}^{K}$,
- and there exists a subset of training samples $\hat{\mathbf{s}}$ with $|\hat{\mathbf{s}}| = \hat{n}(\mathbf{s})$;
- such that $\forall s \in \hat{\mathbf{s}}$,

$$s, z \in C_i, \quad \Longrightarrow \quad |\ell(\mathcal{A}_{\mathbf{s}}, s) - \ell(\mathcal{A}_{\mathbf{s}}, z)| \leq \epsilon(\mathbf{s}).$$

Additional Results: Pseudo Robustness

Theorem

If $\mathbf{s}$ *consists of* n *i.i.d. samples, and* $\mathcal{A}$ *is* $(K, \epsilon(\mathbf{s}), \hat{n}(\mathbf{s}))$ *pseudo robust, then for any* $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\left|\hat{\ell}(\mathcal{A}_{\mathbf{s}}) - \ell_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})\right| \leq \frac{\hat{n}(\mathbf{s})}{n}\epsilon(s) + M\left(\frac{n - \hat{n}(\mathbf{s})}{n} + \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}\right).$$

Additional Results: Pseudo Robustness

Theorem

If $\mathbf{s}$ *consists of* n *i.i.d. samples, and* $\mathcal{A}$ *is* $(K, \epsilon(\mathbf{s}), \hat{n}(\mathbf{s}))$ *pseudo robust, then for any* $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\left|\hat{\ell}(\mathcal{A}_{\mathbf{s}}) - \ell_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})\right| \leq \frac{\hat{n}(\mathbf{s})}{n}\epsilon(s) + M\left(\frac{n - \hat{n}(\mathbf{s})}{n} + \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}\right).$$

- An additional term due to “non-robust” traninig samples.

Outline

1. Algorithmic Robustness and Generalization Bound
2. **Robust Algorithms**
3. (Weak) Robustness is Necessary and Sufficient to (Asymptotic) Generalizability

Which algorithms are robust?

Example (Majority Voting)

Let $\mathcal{Y} = \{-1, +1\}$. Partition $\mathcal{X}$ to $\mathcal{C}_1, \cdots, \mathcal{C}_K$, and use $\mathcal{C}(x)$ to denote the set to which x belongs. A new sample $x_a \in \mathcal{X}$ is labeled by

$$\mathcal{A}_{\mathbf{s}}(x_a) \triangleq \begin{cases} 1, & \text{if } \sum_{s_i \in \mathcal{C}(x_a)} \mathbf{1}(s_{i|y} = 1) \geq \sum_{s_i \in \mathcal{C}(x_a)} \mathbf{1}(s_{i|y} = -1); \\ -1, & \text{otherwise.} \end{cases}$$

If the loss function is $l(\mathcal{A}_s, z) = f(z_{|y}, \mathcal{A}_{\mathbf{s}}(z_{|x}))$ for some function f, then MV is $(2K, 0)$ robust.

Which algorithms are robust?

Theorem

Fix $\gamma > 0$ and metric ρ of $\mathcal{Z}$. Suppose $\mathcal{A}$ satisfies

$$|\ell(\mathcal{A}_{\mathbf{s}}, z_1) - \ell(\mathcal{A}_{\mathbf{s}}, z_2)| \leq \epsilon(\mathbf{s}), \quad \forall z_1, z_2 : z_1 \in \mathbf{s},\ \rho(z_1, z_2) \leq \gamma,$$

and $\mathcal{N}(\gamma/2, \mathcal{Z}, \rho) < \infty$. Then $\mathcal{A}$ is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \rho),\ \epsilon(\mathbf{s})\big)$-robust.

Which algorithms are robust?

Theorem

Fix $\gamma > 0$ and metric ρ of $\mathcal{Z}$. Suppose $\mathcal{A}$ satisfies

$$|\ell(\mathcal{A}_{\mathbf{s}}, z_1) - \ell(\mathcal{A}_{\mathbf{s}}, z_2)| \leq \epsilon(\mathbf{s}), \quad \forall z_1, z_2 : z_1 \in \mathbf{s},\ \rho(z_1, z_2) \leq \gamma,$$

and $\mathcal{N}(\gamma/2, \mathcal{Z}, \rho) < \infty$. Then $\mathcal{A}$ is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \rho),\ \epsilon(\mathbf{s})\big)$-robust.

Example (Lipschitz continuous functions)

If $\mathcal{Z}$ is compact w.r.t. metric ρ, $\ell(\mathcal{A}_{\mathbf{s}}, \cdot)$ is Lipschitz continuous with Lipschitz constant $c(\mathbf{s})$, i.e.,

$$|l(\mathcal{A}_{\mathbf{s}}, z_1) - l(\mathcal{A}_{\mathbf{s}}, z_2)| \leq c(\mathbf{s})\rho(z_1, z_2), \quad \forall z_1, z_2 \in \mathcal{Z},$$

then $\mathcal{A}$ is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \rho),\ c(\mathbf{s})\gamma\big)$-robust for all $\gamma > 0$.

Which algorithms are robust?

Theorem

Fix $\gamma > 0$ and metric ρ of $\mathcal{Z}$. Suppose $\mathcal{A}$ satisfies

$$|\ell(\mathcal{A}_{\mathbf{s}}, z_1) - \ell(\mathcal{A}_{\mathbf{s}}, z_2)| \leq \epsilon(\mathbf{s}), \quad \forall z_1, z_2 : z_1 \in \mathbf{s}, \; \rho(z_1, z_2) \leq \gamma,$$

and $\mathcal{N}(\gamma/2, \mathcal{Z}, \rho) < \infty$. Then $\mathcal{A}$ is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \rho), \epsilon(\mathbf{s})\big)$-robust.

Example (Lipschitz continuous functions)

If $\mathcal{Z}$ is compact w.r.t. metric ρ, $\ell(\mathcal{A}_{\mathbf{s}}, \cdot)$ is Lipschitz continuous with Lipschitz constant $c(\mathbf{s})$, i.e.,

$$|l(\mathcal{A}_{\mathbf{s}}, z_1) - l(\mathcal{A}_{\mathbf{s}}, z_2)| \leq c(\mathbf{s})\rho(z_1, z_2), \quad \forall z_1, z_2 \in \mathcal{Z},$$

then $\mathcal{A}$ is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \rho), c(\mathbf{s})\gamma\big)$-robust for all $\gamma > 0$.

- Similarly, SVM, Lasso, feed-forward neural network and PCA are robust.

Which algorithms are robust?

A large margin classifier is a classification rule such that most of the training samples are "far away" from the classification boundary. We denote the *distance* of a point x to a classification rule Δ by $\mathcal{D}(x, \Delta)$.

Example (Large-margin classifier)

If there exist γ and $\hat{n}$ such that

$$\sum_{i=1}^{n} \mathbf{1}\big(\mathcal{D}(s_{i|x}, \mathcal{A}_s) > \gamma\big) \geq \hat{n},$$

then algorithm $\mathcal{A}$ is $(2\mathcal{N}(\gamma/2, \mathcal{X}, \rho), 0, \hat{n})$ pseudo robust, provided that $\mathcal{N}(\gamma/2, \mathcal{X}, \rho) < \infty$.

Outline

1. Algorithmic Robustness and Generalization Bound
2. Robust Algorithms
3. **(Weak) Robustness is Necessary and Sufficient to (Asymptotic) Generalizability**

(Asymptotic) generalizability

Finite sample bound

(Asymptotic) generalizability

~~Finite sample bound~~ asymptotic property

(Asymptotic) generalizability

~~Finite sample bound~~ asymptotic property

Definition

1. A learning algorithm $\mathcal{A}$ *generalizes w.r.t.* **s** if

$$\limsup_n \left\{ \mathbb{E}_t \Big(\ell(\mathcal{A}_{\mathbf{s}(n)}, t) \Big) - \frac{1}{n} \sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}(n)}, s_i) \right\} \leq 0.$$

2. A learning algorithm $\mathcal{A}$ *generalize w.p. 1* if it generalize w.r.t. almost every **s**.

Weak robustness

Robustness

Weak robustness

~~Robustness~~ weak robustness

Weak robustness

~~Robustness~~ weak robustness

- Robustness requires that the sample space can be partitioned into disjoint subsets such that if a testing sample belongs to the same partitioning set of a training sample, then they have similar loss.
- Weak robustness generalizes such notion by considering the average loss of testing samples and training samples: if for a large (in the probabilistic sense) subset of $\mathcal{Z}^n$, the testing error is close to the training error, then the algorithm is weakly robust.

Weak robustness (cont.)

Definition

1. A learning algorithm $\mathcal{A}$ is *weakly robust w.r.t* $\mathbf{s}$ if there exists a sequence of $\{\mathcal{D}_n \subseteq \mathcal{Z}^n\}$ such that $\Pr(\mathbf{t}(n) \in \mathcal{D}_n) \to 1$, here $\mathbf{t}(n)$ are n i.i.d. testing samples, and

$$\limsup_n \left\{ \max_{\hat{\mathbf{s}}(n) \in \mathcal{D}_n} \Big[\frac{1}{n} \sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}(n)}, \hat{s}_i) - \frac{1}{n} \sum_{i=1}^{n} \ell(\mathcal{A}_{\mathbf{s}(n)}, s_i) \Big] \right\} \leq 0.$$

2. A learning algorithm $\mathcal{A}$ is *a.s. weakly robust* if it is robust w.r.t. almost every $\mathbf{s}$.

All Learning is Robust !

Theorem

1. *An algorithm $\mathcal{A}$ generalizes w.r.t.* $\mathbf{s}$ *if and only if it is weakly robust w.r.t.* $\mathbf{s}$*.*
2. *An algorithm $\mathcal{A}$ generalizes w.p. 1 if and only if it is a.s. weakly robust.*

Conclusion

Summary:

- Propose *Algorithmic Robustness*.
- Present finite sample bound based on algorithmic robustness.
- Show that weak robustness is necessary and sufficient for generalizability.

Conclusion

Summary:

- Propose *Algorithmic Robustness*.
- Present finite sample bound based on algorithmic robustness.
- Show that weak robustness is necessary and sufficient for generalizability.

Future Direction:

- Adaptive partition?
- Other robust algorithms?
- Better rate?