

CS480/680: Introduction to Machine Learning

Lec 17: Robustness

Yaoliang Yu

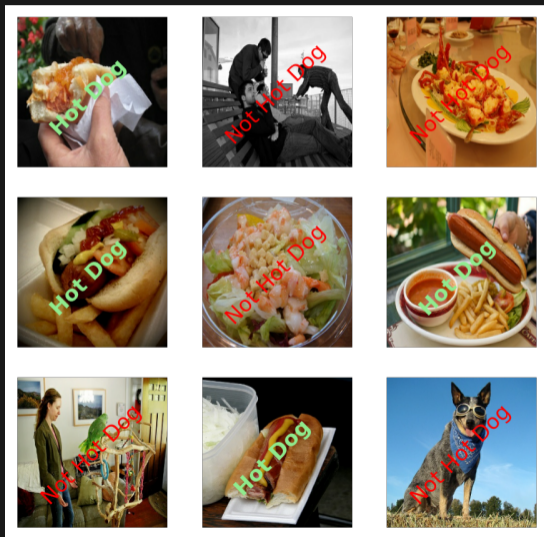


UNIVERSITY OF
WATERLOO

FACULTY OF MATHEMATICS
**DAVID R. CHERITON SCHOOL
OF COMPUTER SCIENCE**

March 26, 2024

example results



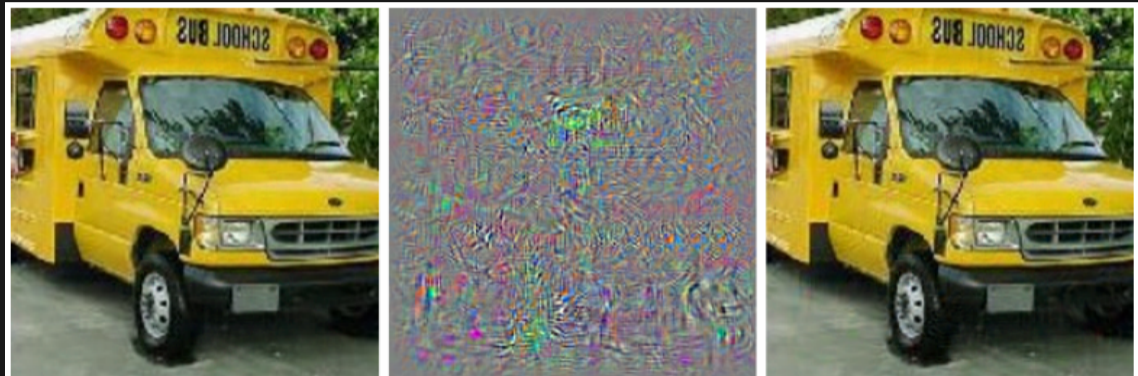
Formally

- Given a training set of pairs of examples $(\mathbf{x}_i, y_i) \in \mathbb{X} \times \mathbb{Y}$
- Return a function (classifier) $f : \mathbb{X} \rightarrow \mathbb{Y}$
- On an unseen test example \mathbf{x} , output $f(\mathbf{x})$
- The goal is to **do well** on unseen test data
 - usually do not care about performance on training data

Performance Metric

- Accuracy (top-1, top-10 error, precision, recall, etc.)
- Training time
- Memory consumption
- Test (inference) time
- Robustness
- Privacy
- Fairness
- Interpretability
- Etc.

And then the Surprise



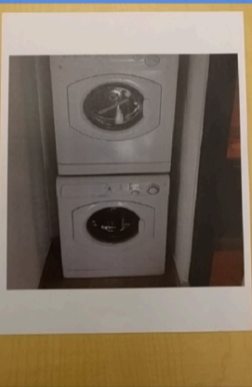
C. Szegedy et al. "Intriguing properties of neural networks". In: *International Conference on Learning Representations (ICLR)*. 2014.



washer: 0.5398173



safe: 0.34602574
washer: 0.22088042



safe: 0.3719305
loudspeaker: 0.24184975





A. Athalye et al. "Synthesizing Robust Adversarial Examples". In: *ICML*. 2018.

Why Should We Care?





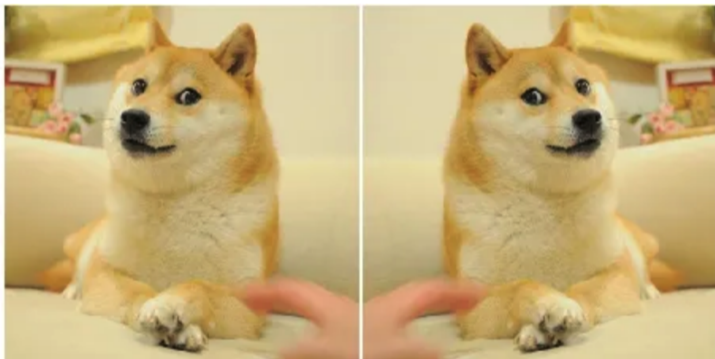
Apple's NeuralHash

- Standard hash: sensitive to small changes



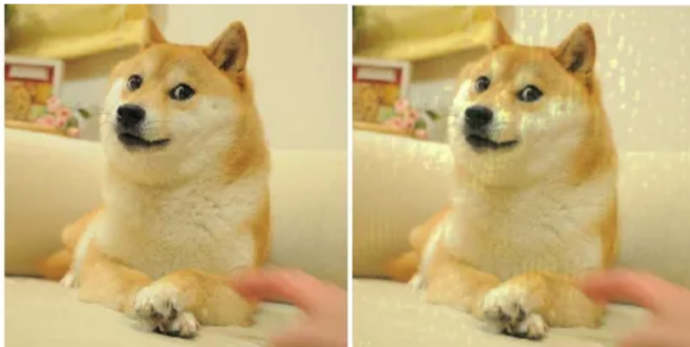
(Left) Original Doge meme, MD5: 53facff91ec83f60a88235ab628590bb | **(Right)** Image cropped by author, MD5: da25273f33c4ec95f71984075079bd16

- Neural hash: contrastive training



(Left) Original Doge meme, NeuralHash: 11d9b097ac960bd2c6c131fa | **(Right)** Image flipped by author, NeuralHash: 20d8f097ac960ad2c7c231fe

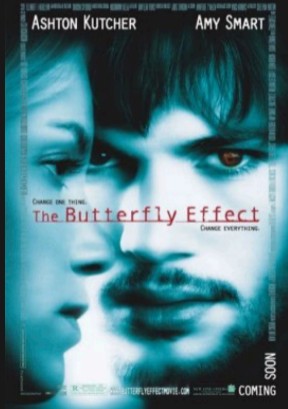
- Adversarial attack against Neural hash



(Left) Original Doge meme, NeuralHash: 11d9b097ac960bd2c6c131fa | **(Right)** Image generated by author, NeuralHash: f8d1b897a45e0bf2f7e1b0fe

Formally

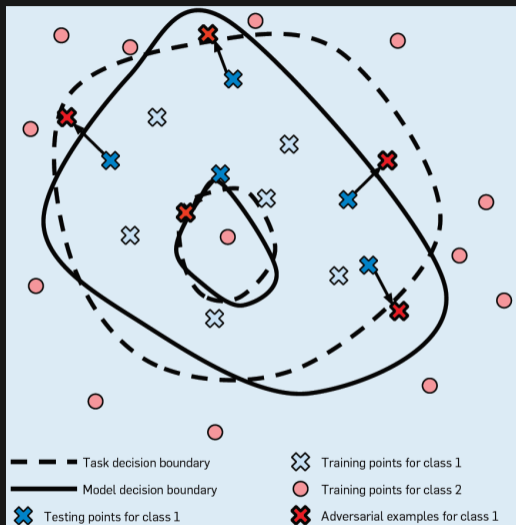
- Exist small $\Delta\mathbf{x}$ such that $f(\mathbf{x} + \Delta\mathbf{x}) \neq y(\mathbf{x})$
- Practically, exist small $\Delta\mathbf{x}$ such that $f(\mathbf{x} + \Delta\mathbf{x}) \neq f(\mathbf{x})$
 - similar if f is very accurate
 - such examples $\mathbf{x} + \Delta\mathbf{x}$ are called “adversarial”
- Intuitively, f is not sufficiently smooth (continuous)
- Or in fancier words, f is not robust
 - small perturbation should lead to small change



The Deep Challenge

$$\mathbf{x} \mapsto \varphi(\mathbf{x}) \mapsto \langle \varphi(\mathbf{x}), \mathbf{w} \rangle =: \hat{y}$$

- φ and \mathbf{w} are learned **jointly**



I. Goodfellow et al. "Making Machine Learning Robust Against Adversarial Inputs". *Communications of the ACM*, vol. 61, no. 7 (2018), pp. 56–66.

WORLD MAP

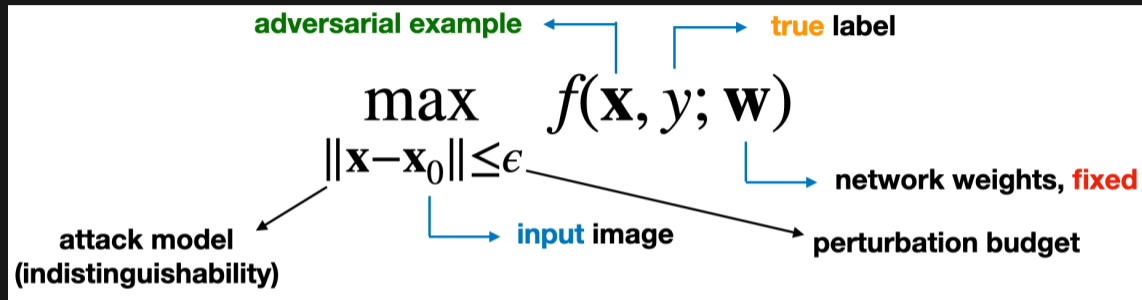


Inevitable

“Adversarial” examples exist for any (non-constant) classifiers.

- Existence is not surprising
- Ubiquity is
- How do we find adversarial examples?

Attack Formulation



- Untargeted attack, e.g. $f(\mathbf{x}, y; \mathbf{w}) = -\log p_y(\mathbf{x}; \mathbf{w})$
- Targeted attack, e.g. $f(\mathbf{x}, y; \mathbf{w}) = \log p_{\tilde{y}}(\mathbf{x}; \mathbf{w})$, for a target label $\tilde{y} \neq y$

Attack Algorithms

Algorithm 1: Fast Gradient Sign Method (FGSM)

Input: model w ; data x ; label y

```
1 for  $t = 0, 1, \dots$  do
2    $x \leftarrow x + \epsilon \cdot \text{sign}(\nabla_x f(x, y; w))$            // sign grad ascent
3    $x \leftarrow P(x)$                                            // projection to stay close and valid
```

Algorithm 2: Projected Gradient Method (PGM)

Input: model w ; data x ; label y

```
1 for  $t = 0, 1, \dots$  do
2    $x \leftarrow x + \eta \cdot \nabla_x f(x, y; w)$                  // grad ascent
3    $x \leftarrow P(x)$                                            // projection to stay close and valid
```

Algorithm 3: Feed-forward MLP trained with backpropogation

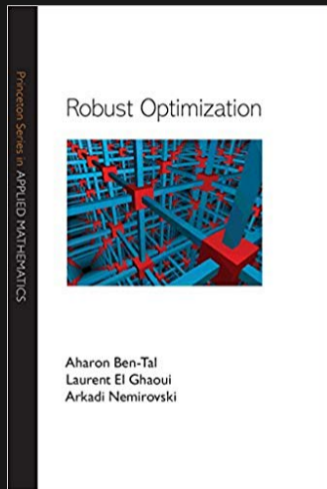
Input: $\mathbf{x} \in \mathbb{R}^{d_0}$, activation $\sigma : \mathbb{R} \rightarrow \mathbb{R}$, loss $\ell : \mathbb{R}^c \rightarrow \mathbb{R}$, regularizer r

```
1  $\mathbf{h}_0 \leftarrow \mathbf{x}$  // initialize with input data
2 for  $k = 1, \dots, l$  do // feature map: layer by layer
3    $\mathbf{z}_k \leftarrow W_k \mathbf{h}_{k-1} + \mathbf{b}_k$  //  $W_k \in \mathbb{R}^{d_k \times d_{k-1}}, \mathbf{b}_k \in \mathbb{R}^{d_k}$ 
4    $\mathbf{h}_k \leftarrow \sigma(\mathbf{z}_k)$  // element-wise
5  $\hat{\mathbf{y}} \leftarrow W_{l+1} \mathbf{h}_l + \mathbf{b}_{l+1}$  //  $W_{l+1} \in \mathbb{R}^{c \times d_l}, \mathbf{b}_{l+1} \in \mathbb{R}^c$ 
6  $\hat{\mathbf{p}} \leftarrow \text{softmax}(\hat{\mathbf{y}})$  //  $\text{softmax}(\mathbf{a}) = \exp(\mathbf{a}) / \langle \exp(\mathbf{a}), \mathbf{1} \rangle$ 
7  $\frac{\partial \ell}{\partial W_{l+1}} \leftarrow \frac{\partial \hat{\mathbf{y}}}{\partial W_{l+1}} \cdot \frac{\partial \hat{\mathbf{p}}}{\partial \hat{\mathbf{y}}} \cdot \frac{\partial \ell}{\partial \hat{\mathbf{p}}} + \frac{\partial r}{\partial W_{l+1}}$  // we absorb  $\mathbf{b}_k$  into  $W_k$ 
8  $\frac{\partial \ell}{\partial \mathbf{h}_l} \leftarrow \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}_l} \cdot \frac{\partial \hat{\mathbf{p}}}{\partial \hat{\mathbf{y}}} \cdot \frac{\partial \ell}{\partial \hat{\mathbf{p}}}$  // initialize
9 for  $k = l, \dots, 1$  do // backward: accumulate derivatives
10   $\frac{\partial \ell}{\partial \mathbf{z}_k} \leftarrow \frac{\partial \mathbf{h}_k}{\partial \mathbf{z}_k} \cdot \frac{\partial \ell}{\partial \mathbf{h}_k}$  //  $\frac{\partial \mathbf{h}_k}{\partial \mathbf{z}_k} = \text{diag}(\sigma'(\mathbf{z}_k))$ 
11   $\frac{\partial \ell}{\partial W_k} \leftarrow \frac{\partial \mathbf{z}_k}{\partial W_k} \cdot \frac{\partial \ell}{\partial \mathbf{z}_k} + \frac{\partial r}{\partial W_k}$  //  $\frac{\partial \mathbf{z}_k}{\partial W_k} = \sum_j [\mathbf{h}_{k-1}^\top \otimes \mathbf{e}_j] \otimes \mathbf{e}_j$ 
12   $\frac{\partial \ell}{\partial \mathbf{h}_{k-1}} \leftarrow \frac{\partial \mathbf{z}_k}{\partial \mathbf{h}_{k-1}} \cdot \frac{\partial \ell}{\partial \mathbf{z}_k}$  //  $\frac{\partial \mathbf{z}_k}{\partial \mathbf{h}_{k-1}} = W_k^\top$ 
```

Adversarial Training

$$\min_{\mathbf{w}} \hat{\mathbb{E}} \left[\underbrace{\max_{\|\Delta \mathbf{x}\| \leq \epsilon} \ell(\mathbf{x} + \Delta \mathbf{x}; \mathbf{w})}_{\ell_{\epsilon}(\mathbf{x}; \mathbf{w})} \right]$$

- $\epsilon = 0$ reduces to the usual training
- $\epsilon \neq 0$ amounts to a change in the loss
- A min-max game between the defender (model \mathbf{w}) and attacker (adversarial example $\mathbf{x} + \Delta \mathbf{x}$)
- SGD on \mathbf{w} where the inner max is solved by an attack algorithm



Lasso Revisited

- Consider the linear regression problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \|X\mathbf{w} - \mathbf{y}\|_2, \quad \text{where } X \in \mathbb{R}^{n \times d}, \mathbf{y} \in \mathbb{R}^n$$

- Suppose we perturb each *feature* independently, and we are interested in solving the robust linear regression problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \max_{\forall j, \|\mathbf{z}_j\|_2 \leq \lambda} \|(X + Z)\mathbf{w} - \mathbf{y}\|_2, \quad \text{where } Z = [\mathbf{z}_1, \dots, \mathbf{z}_d] \in \mathbb{R}^{n \times d}$$

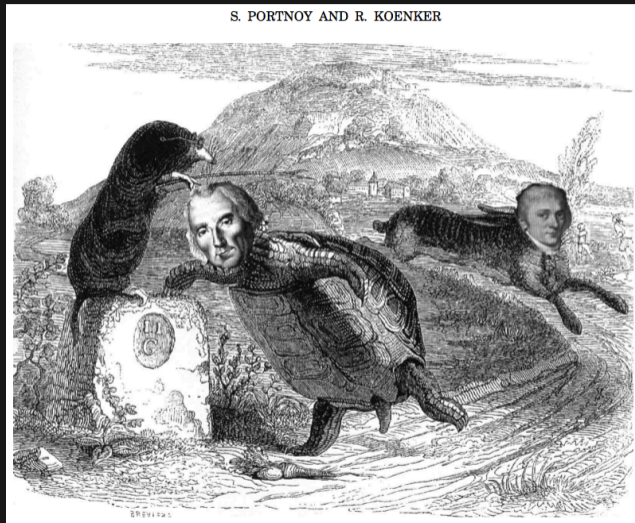
- Prove that robust linear regression is exactly equivalent to (square-root) Lasso:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \|X\mathbf{w} - \mathbf{y}\|_2 + \lambda \|\mathbf{w}\|_1, \quad \text{where } \|\mathbf{w}\|_1 = \sum_j |w_j|$$

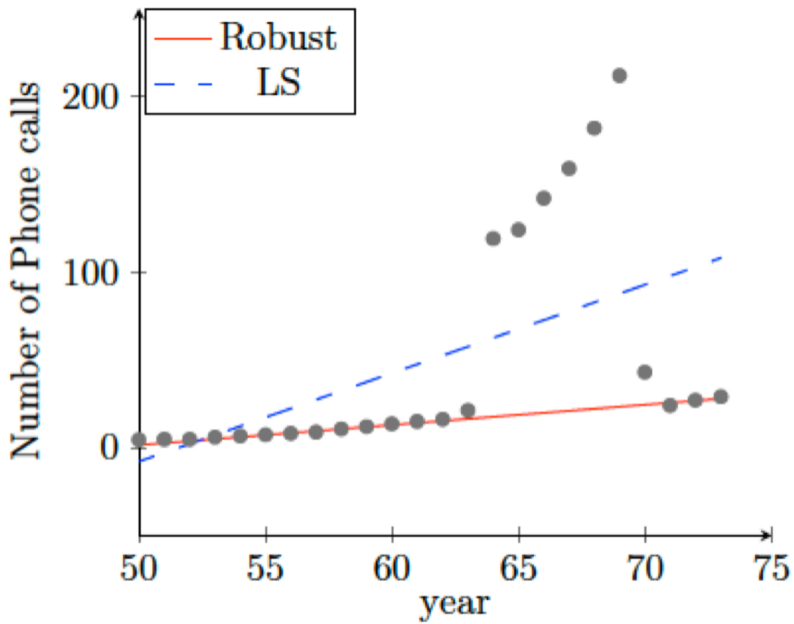
Robustness More Broadly

$$\|X\mathbf{w} - \mathbf{y}\|_2^2 \quad \text{vs.} \quad \|X\mathbf{w} - \mathbf{y}\|_1$$

- Least-squares (Gauss) vs. least absolute deviation (Laplace)



S. Portnoy and R. Koenker. "The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators". *Statistical Science*, vol. 12, no. 4 (1997), pp. 279–300.



WILEY SERIES IN PROBABILITY AND STATISTICS

Robust Statistics

Second Edition



Peter J. Huber
Elvezio M. Ronchetti

WILEY

WILEY

Robust Statistics The Approach Based on Influence Functions

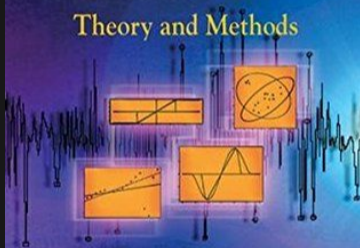
FRANK R. HAMPEL
ELVEZIO M. RONCHETTI
PETER J. ROUSSEEUW
WERNER A. STAHEL

WILEY SERIES IN PROBABILITY AND STATISTICS

WILEY

Robust Statistics

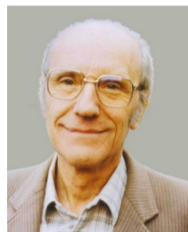
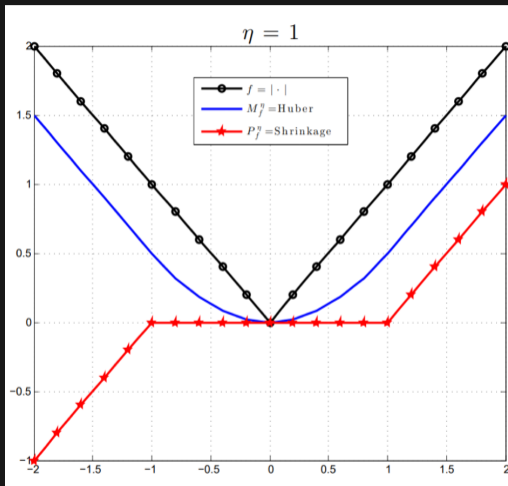
Theory and Methods



Ricardo Maronna, Doug Martin
and Victor Yohai

WILEY SERIES IN PROBABILITY AND STATISTICS

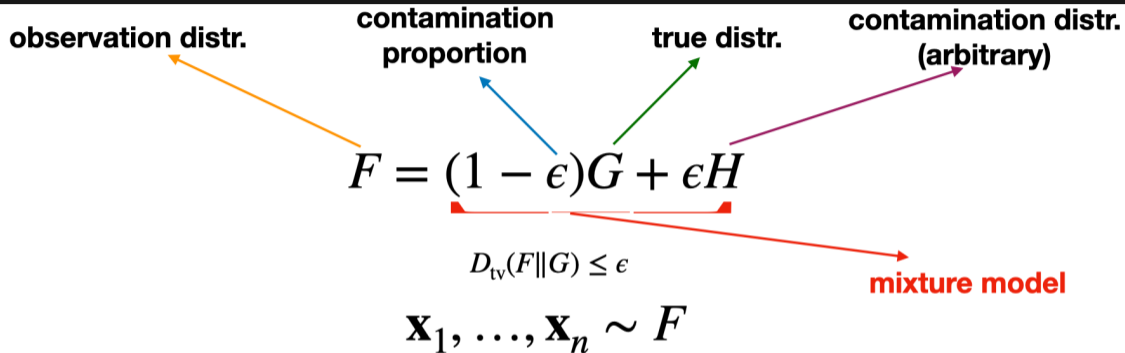
Huber's loss is Moreau's envelope



$$\begin{aligned} M_{|\cdot|}^\eta(t) &= \min_s \frac{1}{2}(s - t)^2 + \eta|s| \\ &= \begin{cases} \frac{1}{2}t^2, & |t| \leq \eta \\ \eta|t| - \frac{1}{2}\eta^2, & |t| \geq \eta \end{cases} \end{aligned}$$

J. J. Moreau. "Fonctions convexes duales et points proximaux dans un espace hilbertien". *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, vol. 255 (1962), pp. 2897–2899, P. J. Huber. "Robust estimation of a location parameter". *Annals of Statistics*, vol. 35, no. 1 (1964), pp. 73–101.

Huber's ϵ -contamination Model

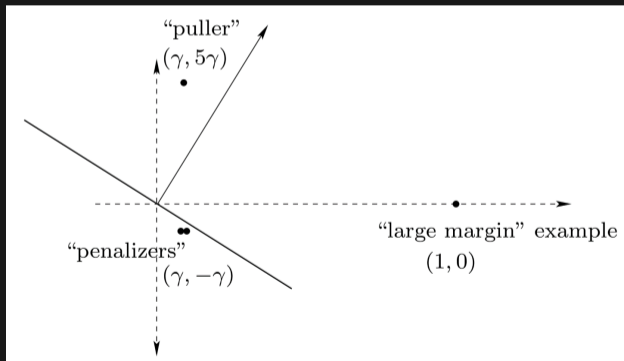


- With probability (w.p.) ϵ , \mathbf{x}_i is from contamination distribution H
- W.p. $1 - \epsilon$, \mathbf{x}_i from true distribution G
- Roughly ϵ proportion is (arbitrarily) contaminated
- A mixture of two distributions: don't know which data comes from which distr.

A Negative Result About Boosting

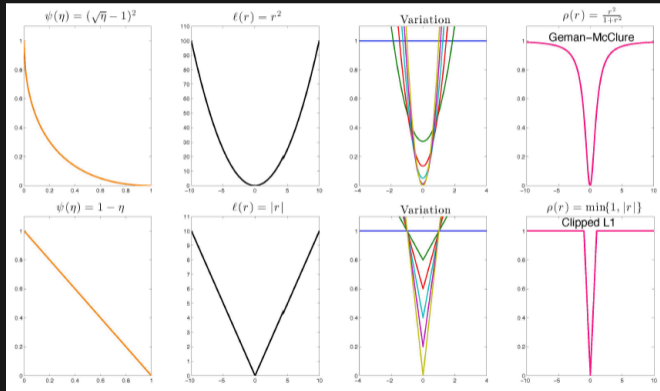
Theorem:

All convex function based boosters cannot tolerate random classification noise at rate $\epsilon \in (0, \frac{1}{2})$



Variational Loss

$$\tilde{\ell}(t) := \min_{\eta \in [0,1]} \eta \cdot \ell(t) + \psi(\eta)$$



M. J. Black and A. Rangarajan. "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision". *International Journal of Computer Vision*, vol. 19 (1996), pp. 57–91, Y. Yu et al. "A Polynomial-time Form of Robust Regression". In: *Advances in Neural Information Processing Systems 26 (NIPS)*. 2012.

