CS480/680: Introduction to Machine Learning

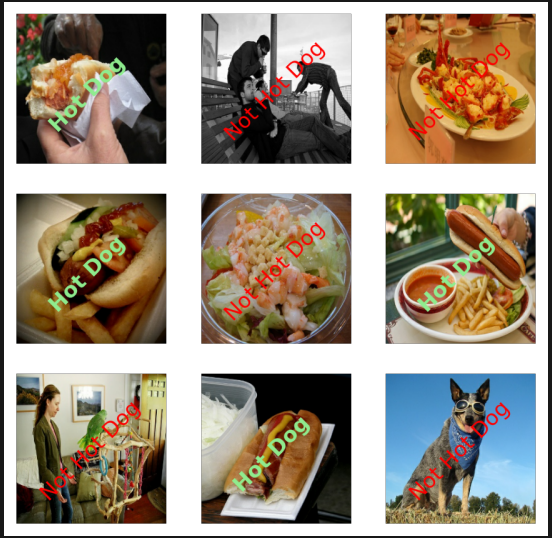
Lec 20: Robustness

Yaoliang Yu

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

July 17, 2024

example results



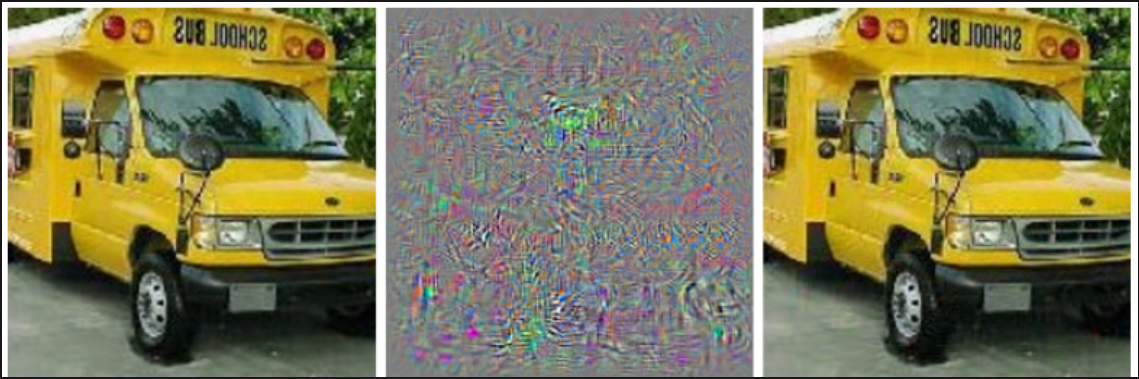
Formally

- Given a training set of pairs of examples $(\mathbf{x}_i, \mathrm{y}_i) \in \mathbb{X} \times \mathbb{Y}$
- Return a function (classifier) $f : \mathbb{X} \to \mathbb{Y}$
- 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

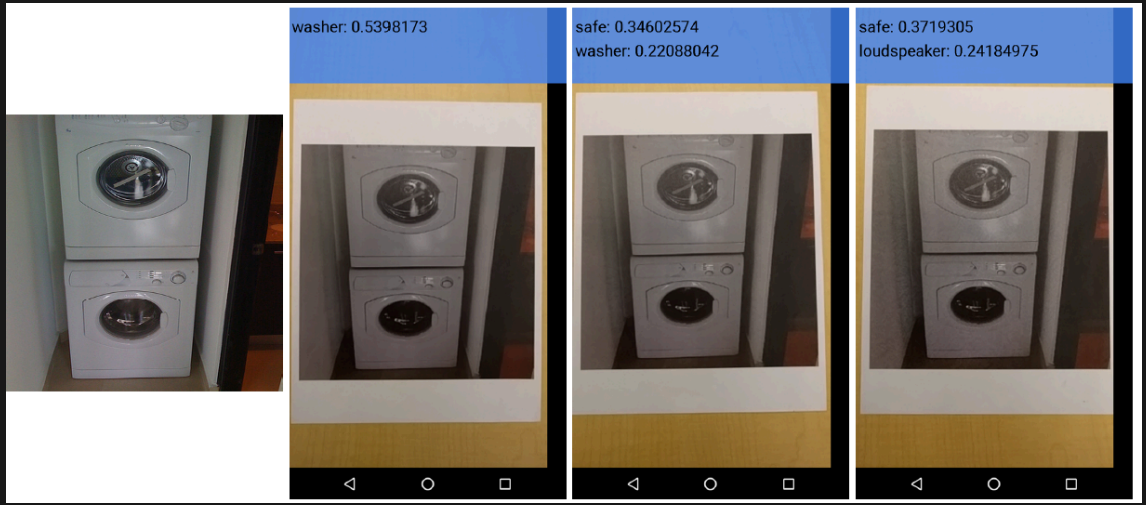
Performance Metric

- 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.



A. Kurakin, I. Goodfellow, and S. Bengio. "Adversarial examples in the physical world". In: *International Conference on Learning Representations (ICLR)*. 2017.

A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. "Synthesizing Robust Adversarial Examples". In: *ICML*. 2018.

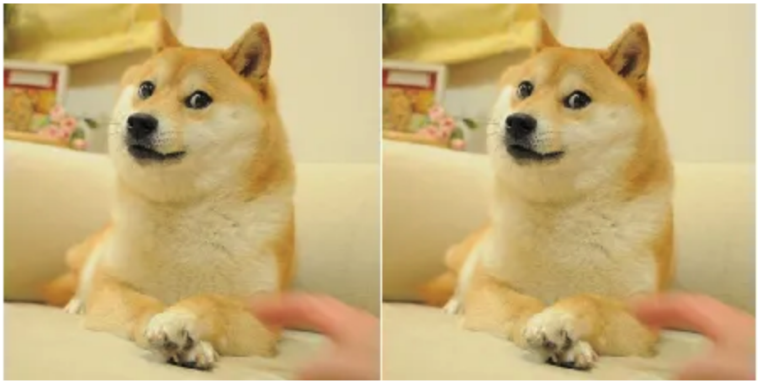
Why Should We Care?

I. Goodfellow, P. McDaniel, and N. Papernot. "Making Machine Learning Robust Against Adversarial Inputs". *Communications of the ACM*, vol. 61, no. 7 (2018), pp. 56–66, J. Gilmer et al. "Motivating the Rules of the Game for Adversarial Example Research". 2018.


STOP

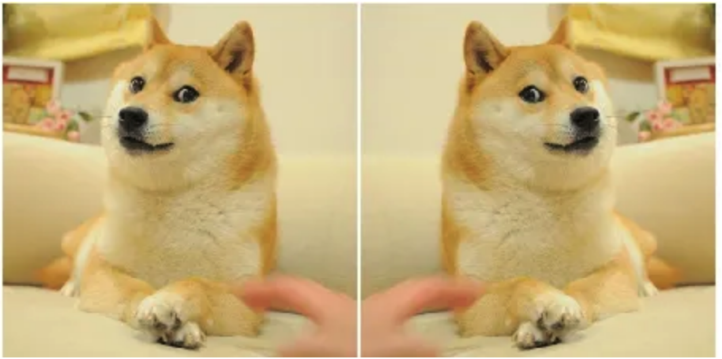
Apple's NeuralHash

- Standard hash: sensitive to small changes



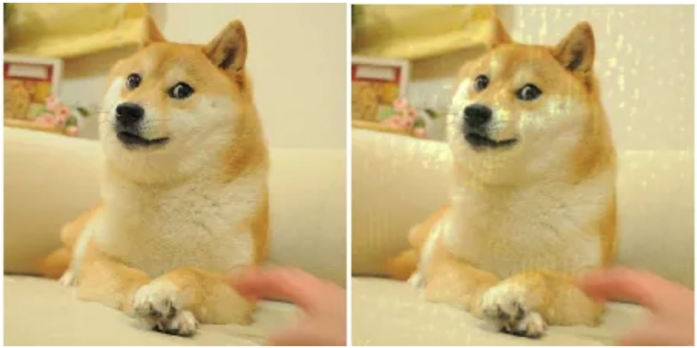
(**Left**) Original Doge meme, MD5: 53facff91ec83f60a88235ab628590bb | (**Right**) Image cropped by author, MD5: da25273f33c4ec95f71984075079bd16

- Neural hash: contrastive training



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

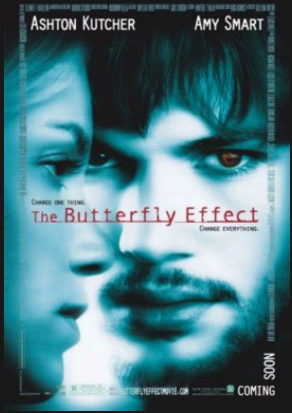
- Adversarial attack against Neural hash



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

Formally

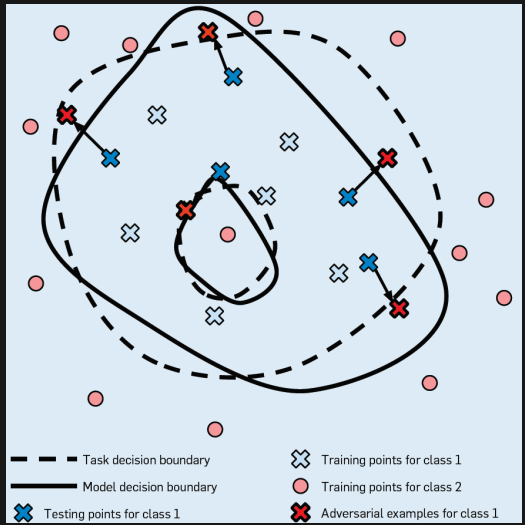
- Exist small $\Delta\mathbf{x}$ such that $f(\mathbf{x} + \Delta\mathbf{x}) \neq \mathsf{y}(\mathbf{x})$
- Practically, exist small $\Delta\mathbf{x}$ such that $f(\mathbf{x} + \Delta\mathbf{x}) \neq f(\mathbf{x})$
  - similar if f is very accurate
  - such examples $\mathbf{x} + \Delta\mathbf{x}$ are called "adversarial"
- Intuitively, f is not sufficiently smooth (continuous)
- Or in fancier words, f is not robust
  - small pertubation 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, P. McDaniel, and N. Papernot. "Making Machine Learning Robust Against Adversarial Inputs". *Communications of the ACM*, vol. 61, no. 7 (2018), pp. 56–66.

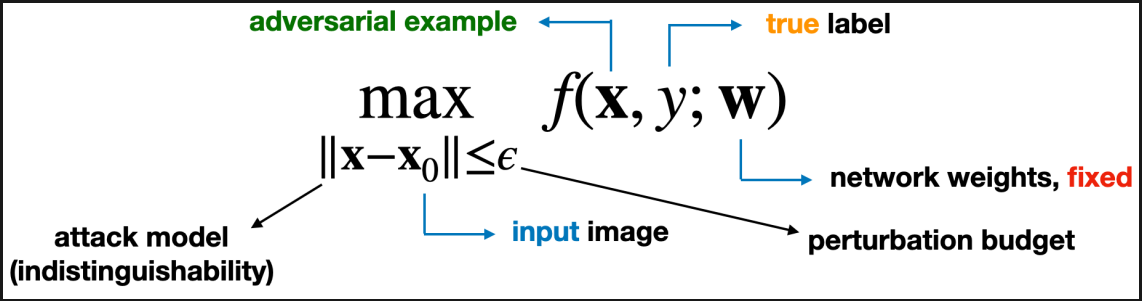
WORLD MAP

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

I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples". In: *International Conference on Learning Representations (ICLR)*. 2015.

Attack Algorithms

Algorithm 1: Fast Gradient Sign Method (FGSM)

Input: model $\mathbf{w}$; data $\mathbf{x}$; label y

for $t = 0, 1, \ldots$ **do**

$\quad \mathbf{x} \leftarrow \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}; \mathbf{w}))$ // sign grad ascent

$\quad \mathbf{x} \leftarrow \mathrm{P}(\mathbf{x})$ // projection to stay close and valid

Algorithm 2: Projected Gradient Method (PGM)

Input: model $\mathbf{w}$; data $\mathbf{x}$; label y

for $t = 0, 1, \ldots$ **do**

$\quad \mathbf{x} \leftarrow \mathbf{x} + \eta \cdot \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}; \mathbf{w})$ // grad ascent

$\quad \mathbf{x} \leftarrow \mathrm{P}(\mathbf{x})$ // projection to stay close and valid

Algorithm 3: Feed-forward MLP trained with backpropogation

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

$\mathbf{h}_0 \leftarrow \mathbf{x}$ // initialize with input data

for $k = 1, \ldots, l$ **do** // feature map: layer by layer

  $\mathbf{z}_k \leftarrow W_k \mathbf{h}_{k-1} + \mathbf{b}_k$ // $W_k \in \mathbb{R}^{d_k \times d_{k-1}}, \mathbf{b}_k \in \mathbb{R}^{d_k}$

  $\mathbf{h}_k \leftarrow \sigma(\mathbf{z}_k)$ // element-wise

$\hat{\mathbf{y}} \leftarrow W_{l+1} \mathbf{h}_l + \mathbf{b}_{l+1}$ // $W_{l+1} \in \mathbb{R}^{c \times d_l}, \mathbf{b}_{l+1} \in \mathbb{R}^c$

$\hat{\mathbf{p}} \leftarrow \mathtt{softmax}(\hat{\mathbf{y}})$ // $\mathtt{softmax}(\mathbf{a}) = \exp(\mathbf{a}) / \langle \exp(\mathbf{a}), \mathbf{1} \rangle$

$\frac{\partial \ell}{\partial W_{l+1}} \leftarrow \frac{\partial \hat{\mathbf{y}}}{\partial W_{l+1}} \cdot \frac{\partial \hat{\mathbf{p}}}{\partial \hat{\mathbf{y}}} \cdot \frac{\partial \ell}{\partial \hat{\mathbf{p}}} + \frac{\partial r}{\partial W_{l+1}}$ // we absorb $\mathbf{b}_k$ into W_k

$\frac{\partial \ell}{\partial \mathbf{h}_l} \leftarrow \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}_l} \cdot \frac{\partial \hat{\mathbf{p}}}{\partial \hat{\mathbf{y}}} \cdot \frac{\partial \ell}{\partial \hat{\mathbf{p}}}$ // initialize

for $k = l, \ldots, 1$ **do** // backward: accumulate derivatives

  $\frac{\partial \ell}{\partial \mathbf{z}_k} \leftarrow \frac{\partial \mathbf{h}_k}{\partial \mathbf{z}_k} \cdot \frac{\partial \ell}{\partial \mathbf{h}_k}$ // $\frac{\partial \mathbf{h}_k}{\partial \mathbf{z}_k} = \mathrm{diag}(\sigma'(\mathbf{z}_k))$

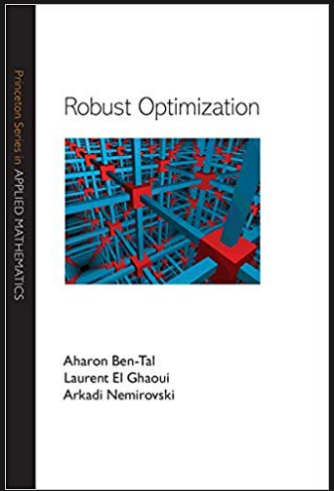
  $\frac{\partial \ell}{\partial W_k} \leftarrow \frac{\partial \mathbf{z}_k}{\partial W_k} \cdot \frac{\partial \ell}{\partial \mathbf{z}_k} + \frac{\partial r}{\partial W_k}$ // $\frac{\partial \mathbf{z}_k}{\partial W_k} = \sum_j [\mathbf{h}_{k-1}^\top \otimes \mathbf{e}_j] \otimes \mathbf{e}_j$

  $\frac{\partial \ell}{\partial \mathbf{h}_{k-1}} \leftarrow \frac{\partial \mathbf{z}_k}{\partial \mathbf{h}_{k-1}} \cdot \frac{\partial \ell}{\partial \mathbf{z}_k}$ // $\frac{\partial \mathbf{z}_k}{\partial \mathbf{h}_{k-1}} = W_k^\top$

Adversarial Training

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

- $\epsilon = 0$ reduces to the usual training
- $\epsilon \neq 0$ amounts to a change in the loss
- A mim-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



A. Madry et al. "Towards Deep Learning Model Resisstant to Adversarial Attacks". In: *International Conference on Learning Representations (ICLR)*. 2018, A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski. "Robust Optimization". Princeton University Press, 2009.

Lasso Revisited

- Consider the linear regression problem:

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

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

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

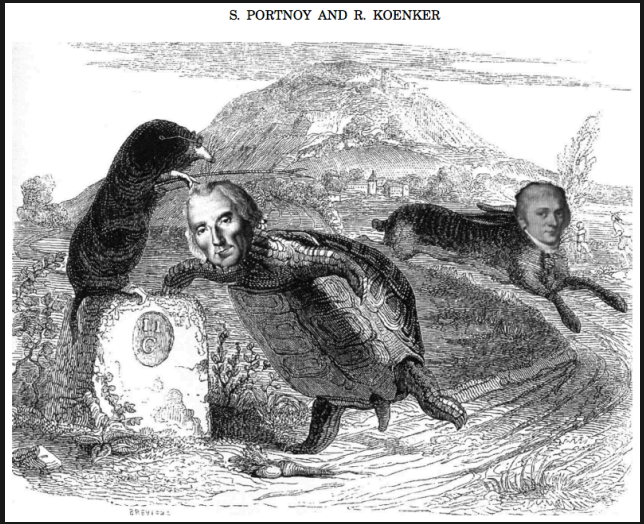
- Prove that robust linear regression is exactly equivalent to (square-root) Lasso:

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

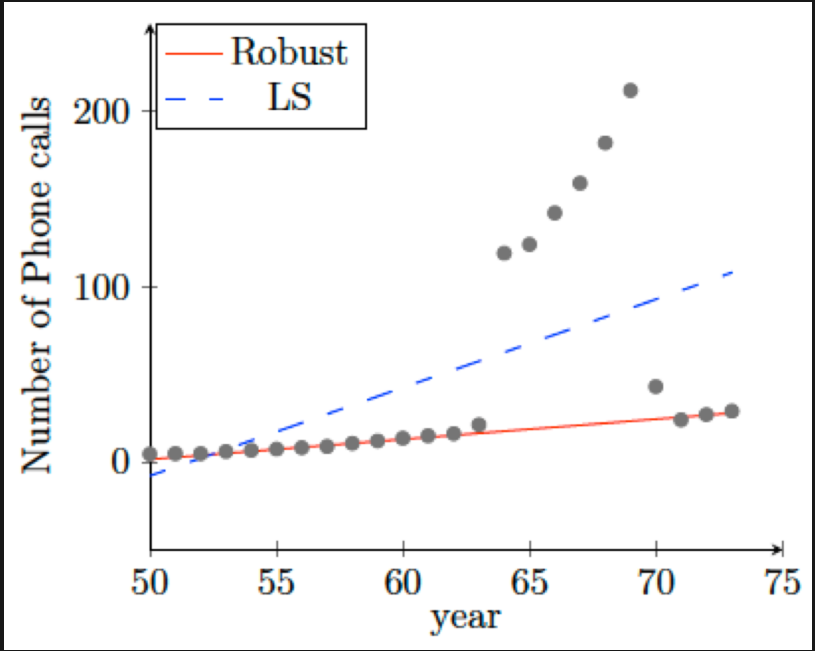
Robustness More Broadly

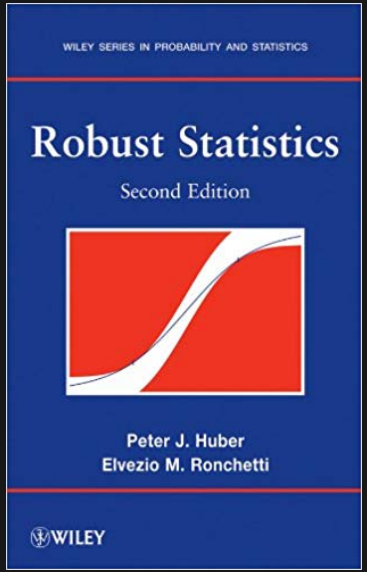
$$\|X\mathbf{w} - \mathbf{y}\|_2^2 \qquad vs. \qquad \|X\mathbf{w} - \mathbf{y}\|_1$$

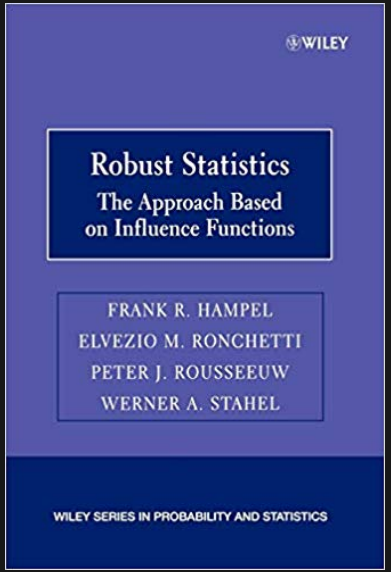
- Least-squares (Gauss) vs. least absolute deviation (Laplace)

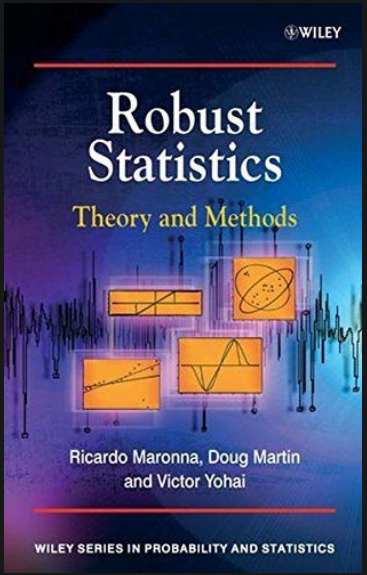


S. Portnoy and R. Koenker. "The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators". *Statistical Science*, vol. 12, no. 4 (1997), pp. 279–300.

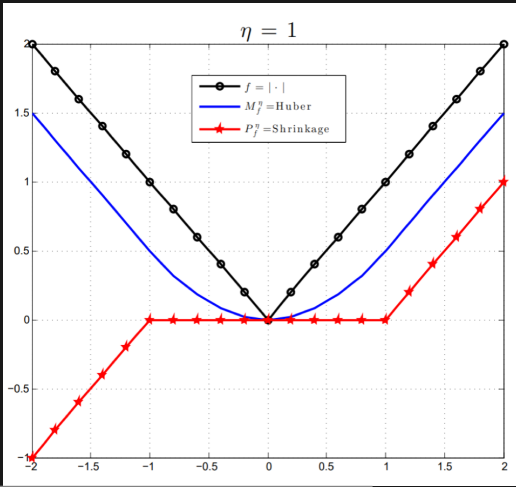

Robust
LS
Number of Phone calls
200
100
0
50
55
60
65
70
75
year

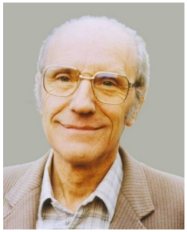

WILEY SERIES IN PROBABILITY AND STATISTICS
Robust Statistics
Second Edition
Peter J. Huber
Elvezio M. Ronchetti
WILEY


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



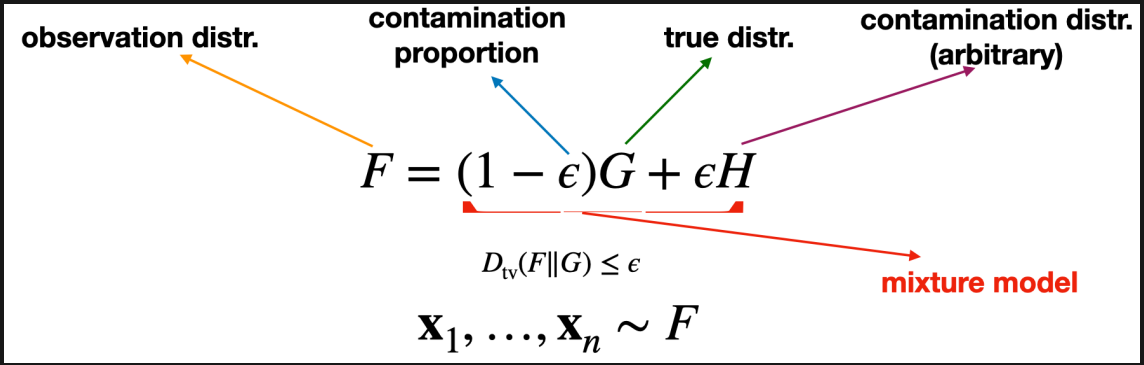


$$\mathsf{M}_{|\cdot|}^{\eta}(t) = \min_{s} \tfrac{1}{2}(s-t)^2 + \eta|s|$$

$$= \begin{cases} \frac{1}{2}t^2, & |t| \le \eta \\ \eta|t| - \frac{1}{2}\eta^2, & |t| \ge \eta \end{cases}$$

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.

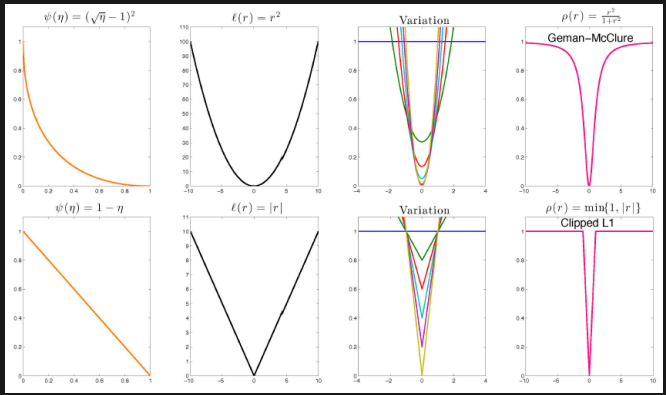
Huber's ϵ-contamination Model



- 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.

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, Ö. Aslan, and D. Schuurmans. "A Polynomial-time Form of Robust Regression". In: *Advances in Neural Information Processing Systems 26 (NIPS)*. 2012.