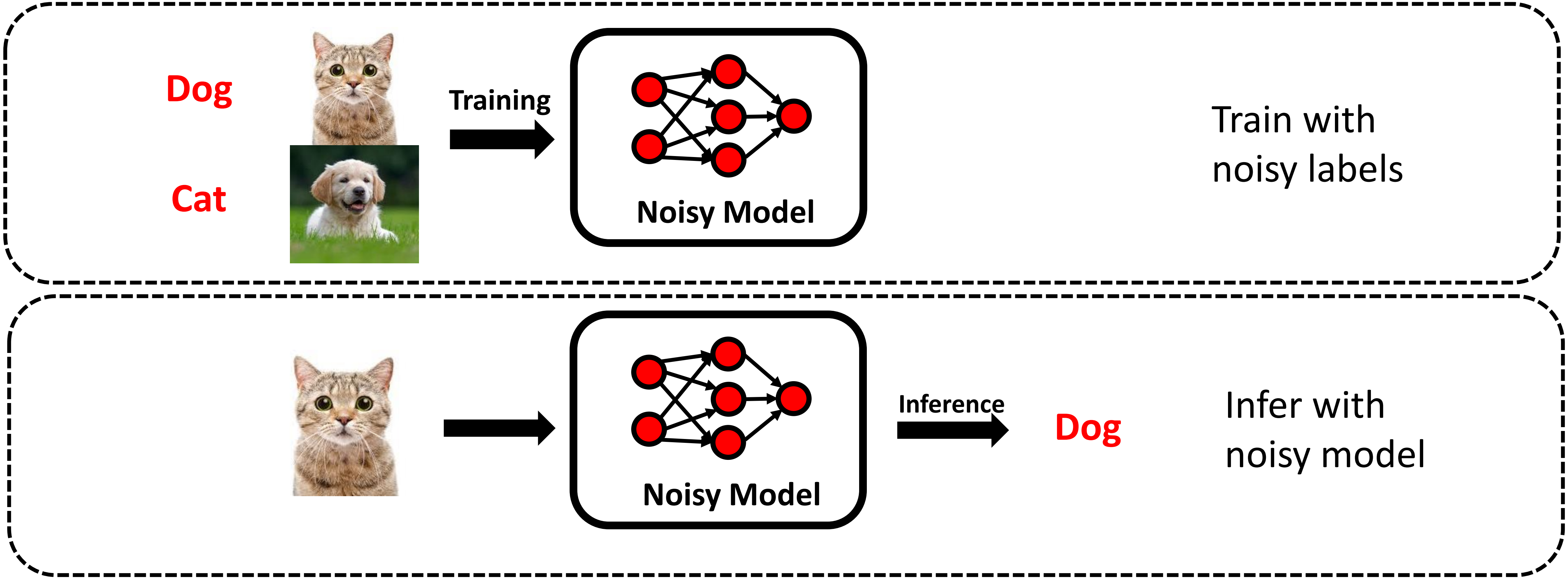
Error-Bounded Correction of Noisy Labels

Songzhu Zheng, Pengxiang Wu, Aman Goswami,
Mayank Goswami, Dimitris Metaxas, Chao Chen

The State University of New York at Stony Brook
Rutgers University
The City University of New York, Queen’s College

Label Noise is Ubiquitous and Troublesome



Label Noise can be Introduced by:

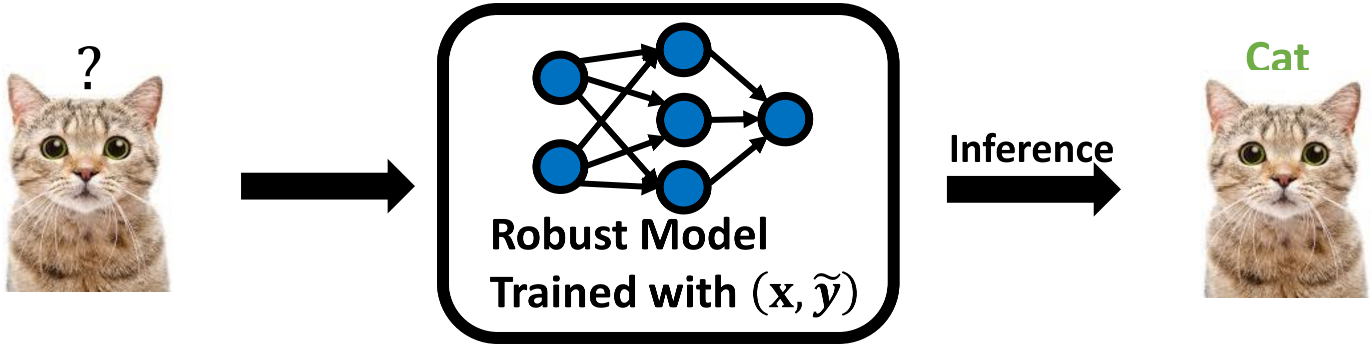
- Human or automatic annotators mistakenly (Yan et al. 2014; Veit et al. 2017)

Settings

- $\tilde{y}$ is noisy label (observed), y is clean label (unknown)
- Chanllenge:

Train with **noisy data** $(\mathbf{x}, \widetilde{\boldsymbol{y}})$.

But require to give **correct prediction** $\boldsymbol{y}$**.**

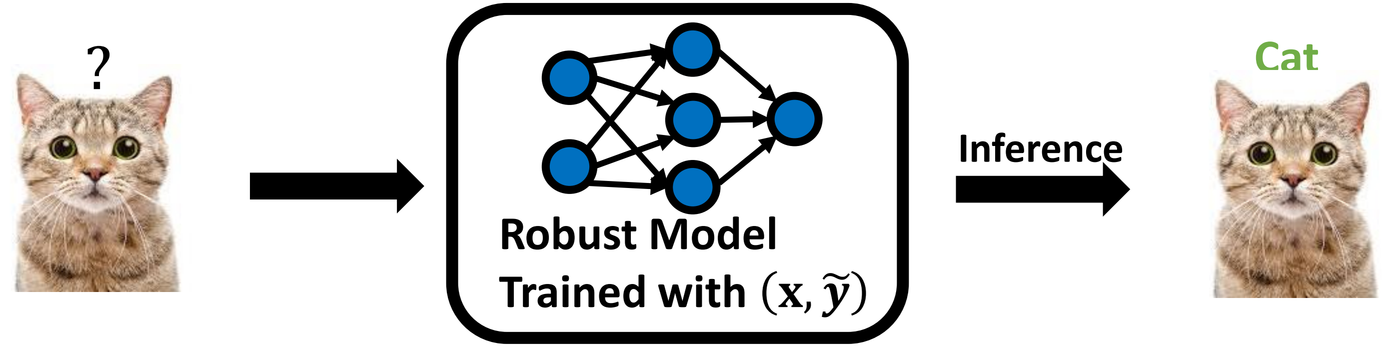


Settings

- $\tilde{y}$ is noisy label (observed), y is clean label (unknown)
- Chanllenge:

Train with **noisy data** $(\mathbf{x}, \tilde{\boldsymbol{y}})$.

But require to give **correct prediction** $\boldsymbol{y}$**.**



- Noise Transition Matrix T. Each entry $\tau_{ij} = P(\tilde{y} = j \mid y = i)$:

$$T = \begin{array}{c} \text{True} \backslash \text{Noisy} \\ cat \\ dog \\ human \end{array} \begin{array}{c} \begin{array}{ccc} cat & dog & human \end{array} \\ \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \\ 0.3 & 0.3 & 0.4 \end{pmatrix} \end{array}$$

Uniform Noise

$$T = \begin{array}{c} \text{True} \backslash \text{Noisy} \\ cat \\ dog \\ human \end{array} \begin{array}{c} \begin{array}{ccc} cat & dog & human \end{array} \\ \begin{pmatrix} 0.6 & 0.4 & 0 \\ 0.4 & 0.6 & 0 \\ 0 & 0.4 & 0.6 \end{pmatrix} \end{array}$$

Pairwise Noise

Existing Solutions – Model Re-calibration

- Introduce new loss term to get robust model:
  1) Estimation of matrix T to correct the loss term (Goldberger & Ben-Reuven, 2017; Patrini et al., 2017)
  2) Robust deep learning layer (Van Rooyen et al., 2015)
  3) Reconstruction loss term (Reed et al., 2014)

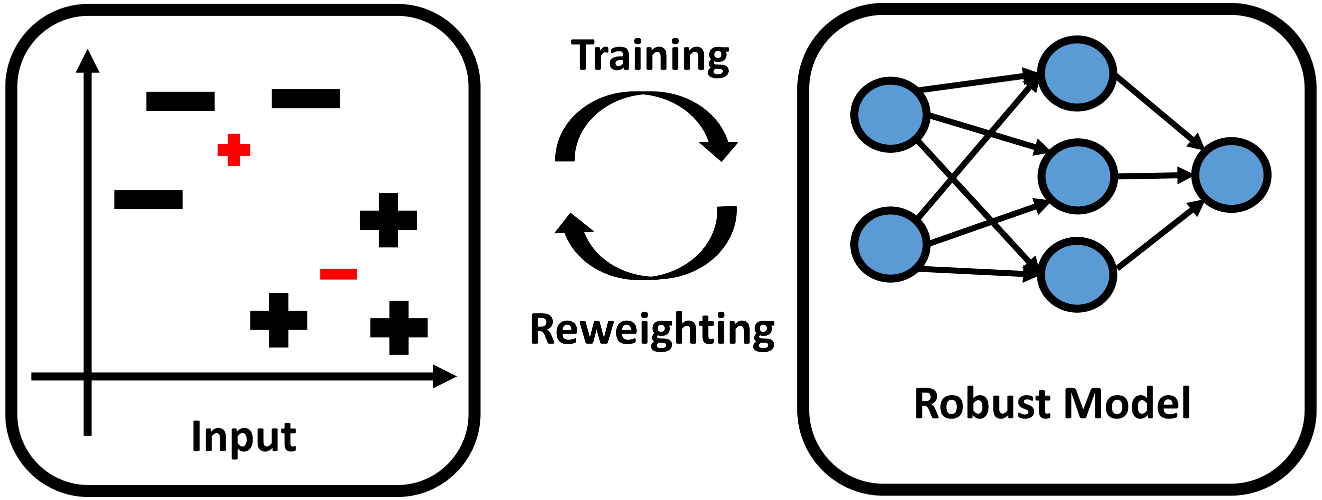
- Pros:

  Globally regularization; theoretical guarantee

- Cons:

  Not flexible enough; omit local information

Existing Solutions – Data Re-calibration



- Re-weighting or pick data point using noisy classifier
  - Noisy classifier's confidence determines the weight
  - Clean labels gain higher weight
  - Re-weighting and training happens jointly

- Pros:

  Better performance than model re-calibration model. Flexible enough to fully use point-wise information

- Cons:

  No theoretical support

Contribution

- The first theoretic explanation for data re-calibration method
  - Explained why noisy classifier to be used to decide whether a label is trustable or not.

- A theory inspired data re-calibrating algorithm
  - Easy to tune
  - Scalable
  - Label Correction

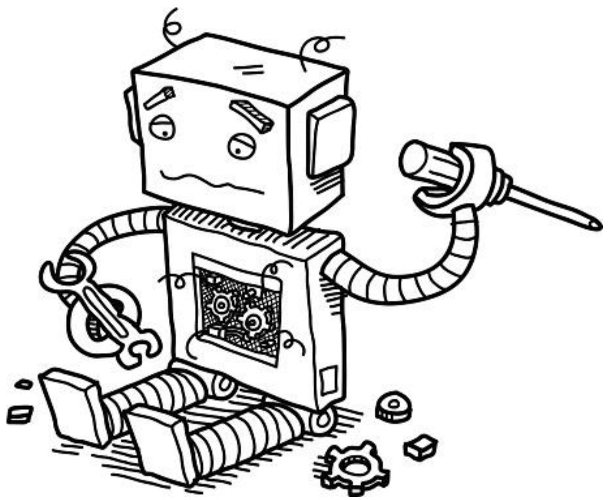


Image Source: https://media.istockphoto.com/vectors/hand-drawn-vector-cartoon-illustration-of-a-broken-robot-trying-to-vector-id1131797122?k=6&m=1131797122&s=612x612&w=0&h=H2fviprWr24dxlO2QPae1R8X3nrHB-J40NCunv2aE84=

(Noisy) Classifier and (Noisy) Posterior

Classification scoring function $\mathrm{f}(x)$ approximates posterior probability of labels:

- Clean (x, y) : $f(x)$ approximates **clean posterior** $\eta(x) = P(y = 1 \mid x)$
- Noisy $(x, \tilde{y})$: $f(x)$ approximates **noisy posterior** $\tilde{\eta}(x) = P(\tilde{y} = 1 \mid x)$

(Noisy) Classifier and (Noisy) Posterior

Classification scoring function $\mathrm{f}(x)$ approximates posterior probability of labels:

- Clean (x, y) : $f(x)$ approximates **clean posterior** $\eta(x) = P(y = 1 \mid x)$
- Noisy $(x, \tilde{y})$: $f(x)$ approximates **noisy posterior** $\tilde{\eta}(x) = P(y = 1 \mid x)$
- There is a linear relationship $\tilde{\eta}(x)$= $(1 - \tau_{10} - \tau_{01})\eta(x) + \tau_{01}$

Remember $\tau_{10} = P(\tilde{y} = 0 \mid y = 1)$ and $\tau_{01} = P(\tilde{y} = 1 \mid y = 0)$

Low Confidence of $\tilde{\eta}(x)$ Implies Noise

Theorem 1. Let $\epsilon := \|f - \tilde{\eta}\|_\infty$ and for $\Delta = \frac{1-|\tau_{10}-\tau_{01}|}{2}$, there exists constant $C, \lambda > 0$

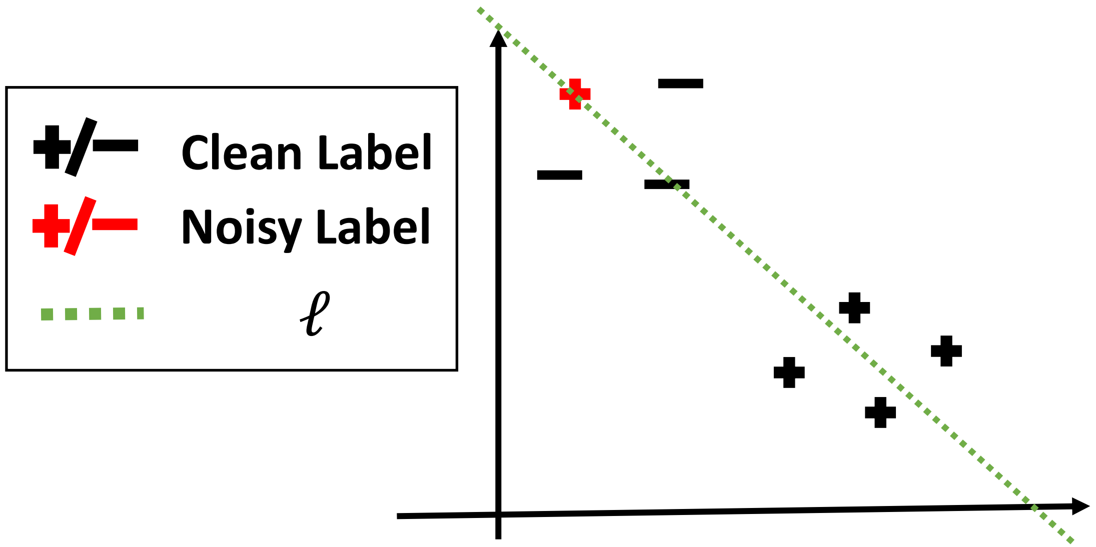
such that:

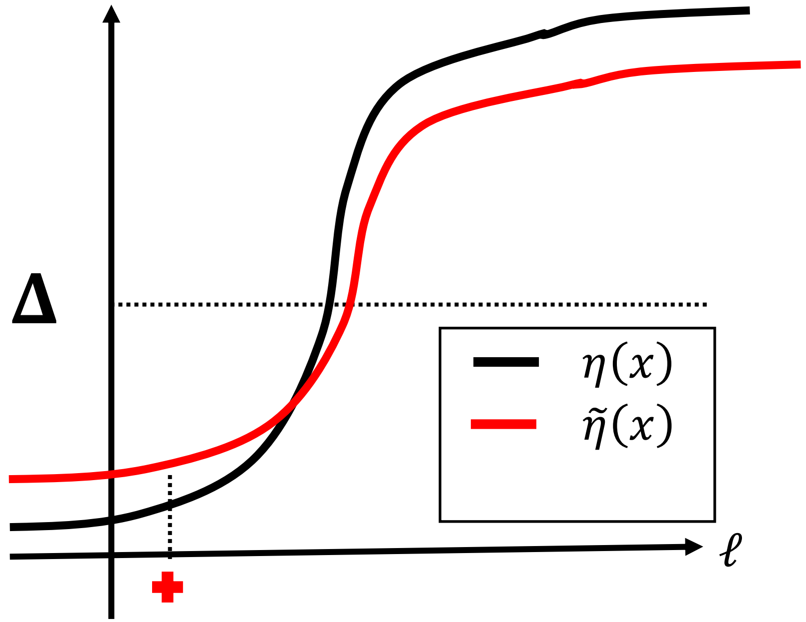
- $\tilde{y} = 1$: $Prob[f(x) \leq \Delta, \tilde{y}\ is\ clean] \leq C[O(\epsilon)]^\lambda$
- $\tilde{y} = 0$: $Prob[1 - f(x) \leq \Delta, \tilde{y}\ is\ clean\ ] \leq C[O(\epsilon)]^\lambda$

Low Confidence of $\tilde{\eta}(x)$ Implies Noise

Theorem 1. Let $\epsilon := \|f - \tilde{\eta}\|_\infty$, there exists constant $C, \lambda > 0$ and $\Delta \in (0,1)$, such that:

- $\boxed{\tilde{y} = 1: \ Prob[f(x) \le \Delta, \tilde{y} \text{ is clean}] \ \le C[O(\epsilon)]^\lambda}$
- $\tilde{y} = 0: \ Prob[1 - f(x) \le \Delta, \tilde{y} \text{ is clean}] \ \le C[O(\epsilon)]^\lambda$

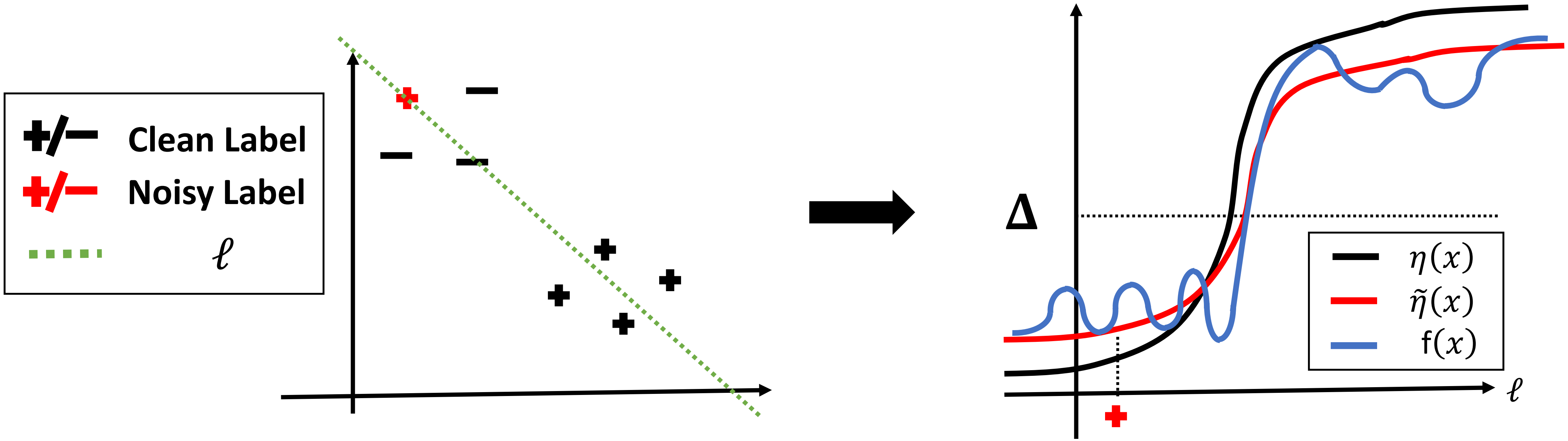




Low Confidence of $\tilde{\eta}(x)$ Implies Noise

Theorem 1. Let $\epsilon := \|f - \tilde{\eta}\|_\infty$, there exists constant $C, \lambda > 0$ and $\Delta \in (0,1)$, such that:

- $\boxed{\tilde{y} = 1 : \; Prob[f(x) \le \Delta, \tilde{y} \text{ is clean}] \; \le C[O(\epsilon)]^\lambda}$
- $\tilde{y} = 0 : \; Prob[1 - f(x) \le \Delta, \tilde{y} \text{ is clean}] \; \le C[O(\epsilon)]^\lambda$

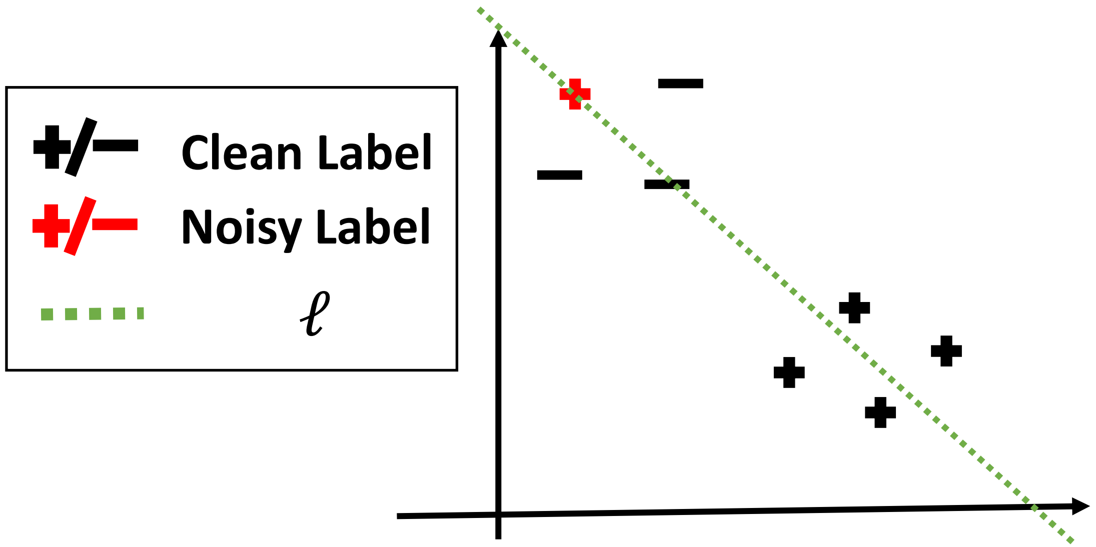


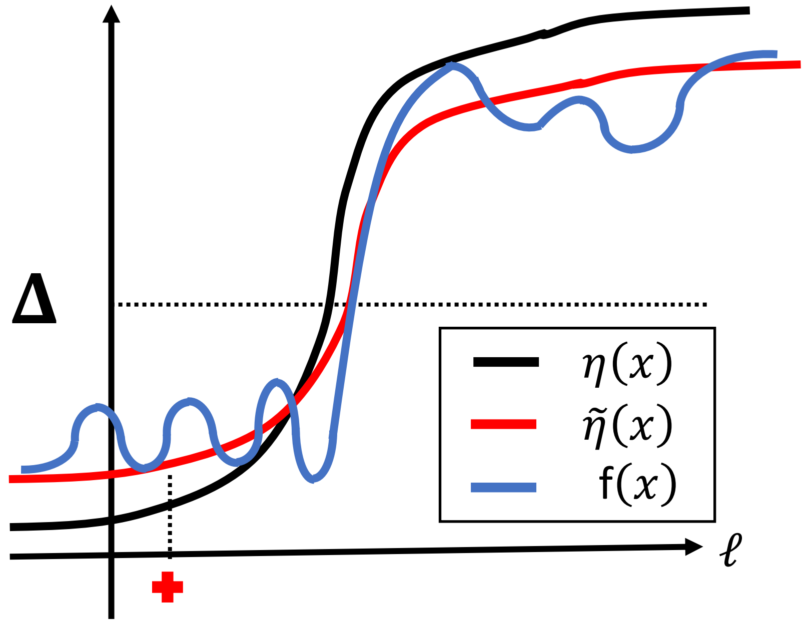
Low Confidence of $\tilde{\eta}(x)$ Implies Noise

Theorem 1. Let $\epsilon := \|f - \tilde{\eta}\|_\infty$, there exists constant $C, \lambda > 0$ and $\Delta \in (0,1)$, such that:

- $\tilde{y} = 1:\ Prob[f(x) \le \Delta, \tilde{y}\ is\ clean]\ \le C[O(\epsilon)]^\lambda$
- $\tilde{y} = 0:\ Prob[1 - f(x) \le \Delta, \tilde{y}\ is\ clean\ ]\ \le C[O(\epsilon)]^\lambda$

$\tilde{\eta}(x) = (1 - \tau_{10} - \tau_{01})\eta(x) + \tau_{01}$

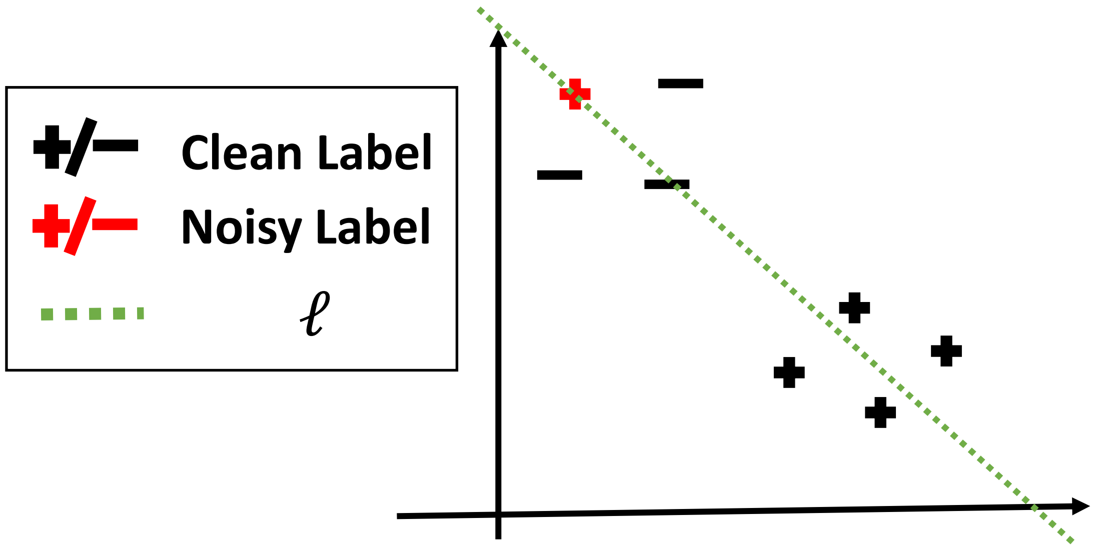


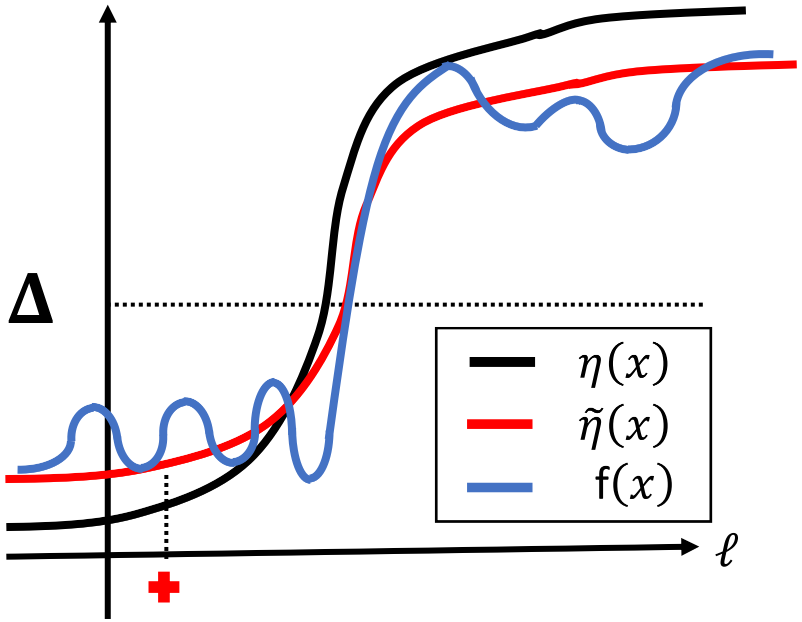


Low Confidence of $\tilde{\eta}(x)$ Implies Noise

Theorem 1. Let $\epsilon := \|f - \tilde{\eta}\|_\infty$, there exists constant $C, \lambda > 0$ and $\Delta \in (0,1)$, such that:

- $\boxed{\tilde{y} = 1: \; Prob[f(x) \leq \Delta, \tilde{y} \text{ is clean}] \; \leq C[O(\epsilon)]^\lambda}$
- $\tilde{y} = 0: \; Prob[1 - f(x) \leq \Delta, \tilde{y} \text{ is clean}] \; \leq C[O(\epsilon)]^\lambda$

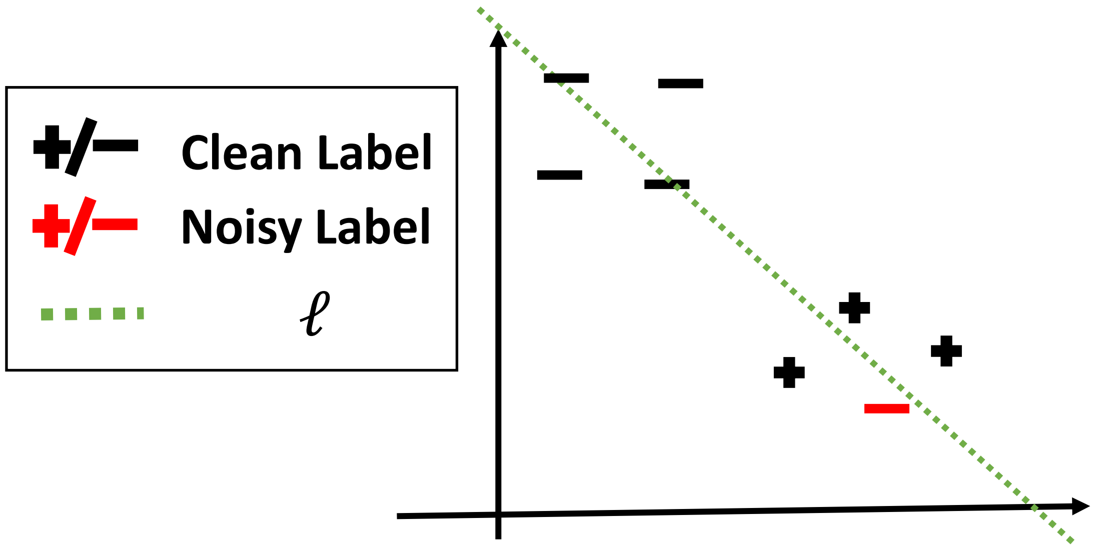


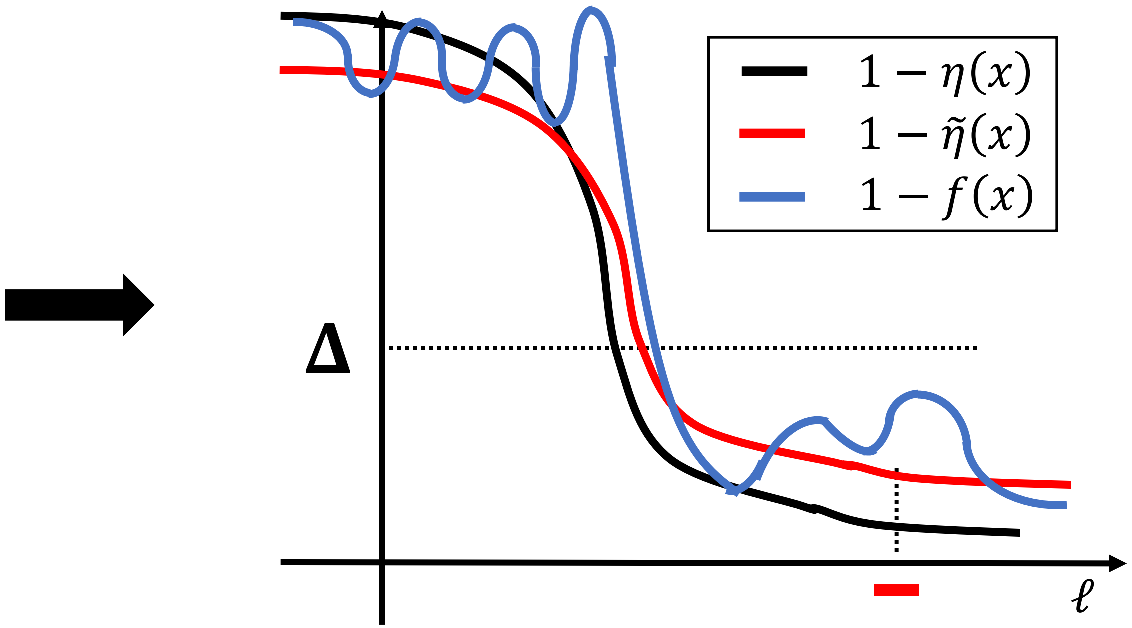


Inconfidence of $\tilde{\eta}(x)$ Implies Noise

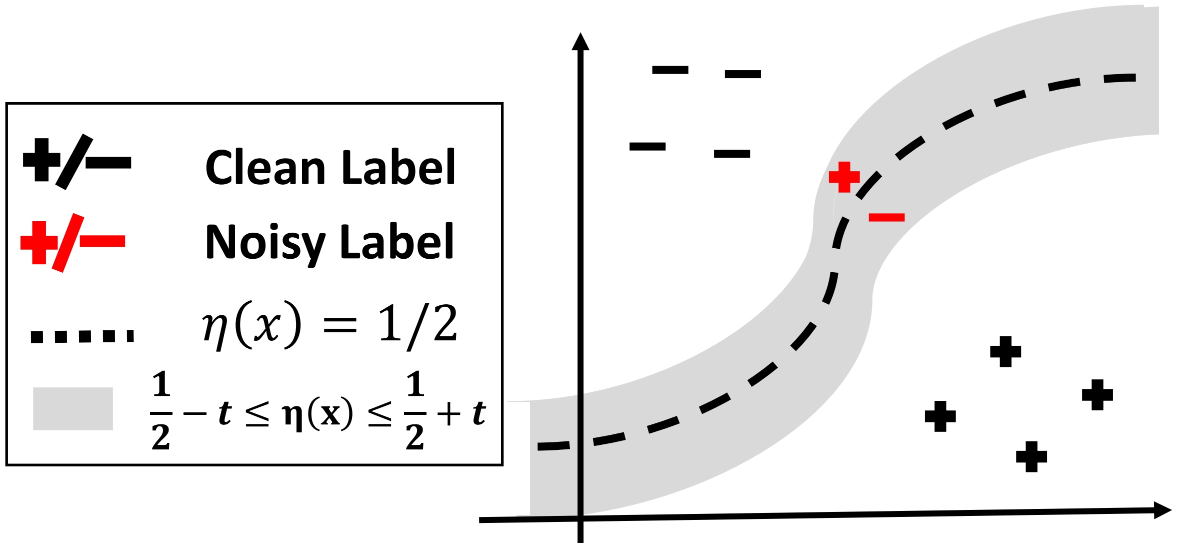
Theorem 1. Let $\epsilon := \|f - \tilde{\eta}\|_\infty$, there exists constant $C, \lambda > 0$ and $\Delta \in (0,1)$, such that:

- $\tilde{y} = 1$: $Prob[f(x) \leq \Delta, \tilde{y}\ is\ clean] \leq C[O(\epsilon)]^\lambda$
- $\boxed{\tilde{y} = 0 : Prob[1 - f(x) \leq \Delta, \tilde{y}\ is\ clean] \leq C[O(\epsilon)]^\lambda}$



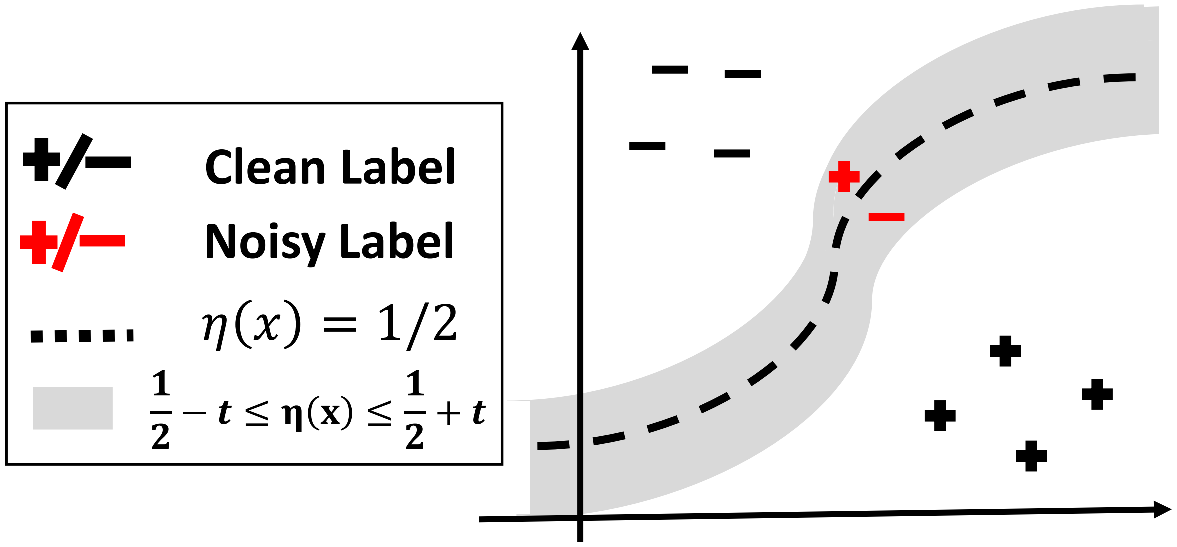


Tsybakov Condition



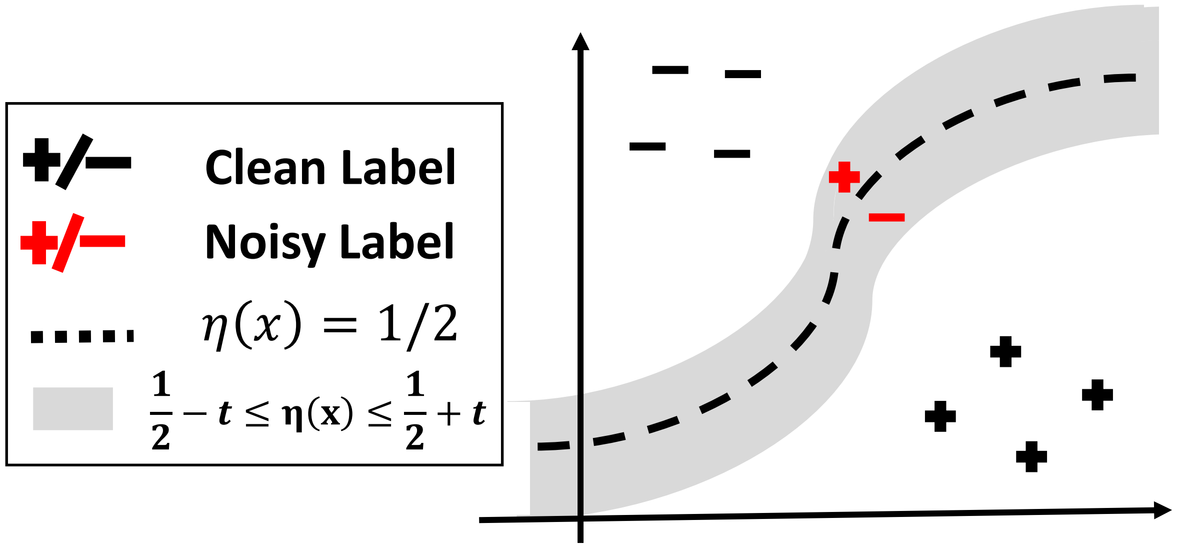
Tsybakov Condition

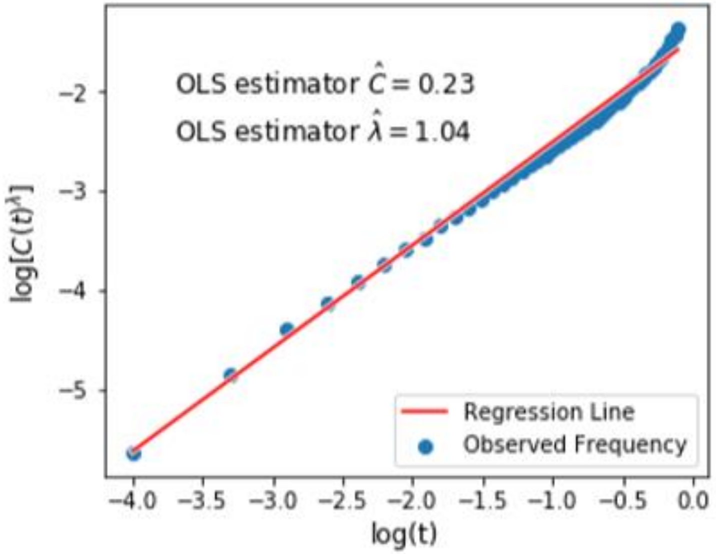
- **Tsybakov Condition** [2004]. There exists constants $C, \lambda > 0$ and $t_0 \in \left(0, \frac{1}{2}\right]$, such that for all $t \leq t_0$, $P\left[\left|\eta(x) - \frac{1}{2}\right| \leq t\right] \leq Ct^{\lambda}$



Tsybakov Condition

- **Tsybakov Condition** [2004]. There exists constants $C, \lambda > 0$ and $t_0 \in \left(0, \frac{1}{2}\right]$, such that for all $t \leq t_0$, $P\left[\left|\eta(x) - \frac{1}{2}\right| \leq t\right] \leq Ct^{\lambda}$
- Empirical Verification (CIFAR-10) : $\hat{C} = 0.32$ and $\hat{\lambda} = 1.04$. Statistically Significant





Inconfidence of $\tilde{\eta}(x)$ Implies Noise

Theorem 1. Let $\epsilon := \|f - \tilde{\eta}\|_\infty$, there exists constant $C, \lambda > 0$ and $\Delta \in (0, 1)$, such that:

- $\tilde{y} = 1: \ Prob[f(x) \le \Delta, \tilde{y}\ is\ clean] \ \le 0.23[O(\epsilon)]^{1.04}$
- $\tilde{y} = 0: \ Prob[1 - f(x) \le \Delta, \tilde{y}\ is\ clean\,] \ \le 0.23[O(\epsilon)]^{1.04}$

Theory-Inspired Algorithm

Procedure `LRT-Correction (Simplified)`

Input: $(\boldsymbol{x}, \widetilde{y}), f(\boldsymbol{x}), \delta = \frac{\Delta}{1-\Delta}$.

Output: $\widetilde{y}_{new}$

if $\widetilde{y} = 1$ **then**

  $\mathrm{LR}(f, \boldsymbol{x}, \widetilde{y}) := \frac{f(\boldsymbol{x})}{1-f(\boldsymbol{x})}$

else

  $\mathrm{LR}(f, \boldsymbol{x}, \widetilde{y}) := \frac{1-f(\boldsymbol{x})}{f(\boldsymbol{x})}$

end if

if $\mathrm{LR}(f, \boldsymbol{x}, \widetilde{y}) \leq \delta$ **then**

  $\widetilde{y}_{new} = 1 - \widetilde{y}$

else

  $\widetilde{y}_{new} = \widetilde{y}$

end if

Theory-Inspired Algorithm

Procedure `LRT-Correction (Simplified)`

Input: $(\boldsymbol{x}, \widetilde{y}), f(\boldsymbol{x}), \delta = \frac{\Delta}{1-\Delta}$.
Output: $\widetilde{y}_{new}$

if $\widetilde{y} = 1$ **then**
    $\mathrm{LR}(f, \boldsymbol{x}, \widetilde{y}) := \frac{f(\boldsymbol{x})}{1-f(\boldsymbol{x})}$
else
    $\mathrm{LR}(f, \boldsymbol{x}, \widetilde{y}) := \frac{1-f(\boldsymbol{x})}{f(\boldsymbol{x})}$
end if
if $\mathrm{LR}(f, \boldsymbol{x}, \widetilde{y}) \leq \delta$ **then**
    $\widetilde{y}_{new} = 1 - \widetilde{y}$
else
    $\widetilde{y}_{new} = \widetilde{y}$
end if

Corollary 1. Let $\epsilon := \max|f(x) - \tilde{\eta}(x)|$.
If $\tilde{y}_{new}$ denotes the output of the *LRT-Correction* with input $(\mathrm{x}, \tilde{\mathrm{y}})$, f and δ then $\exists \mathrm{C}, \lambda > 0$:

$$Prob[\tilde{y}_{new} \text{ is clean}] > 1 - C[O(\epsilon)]^{\lambda}$$

Remark:

The extension to multi-class would be natural

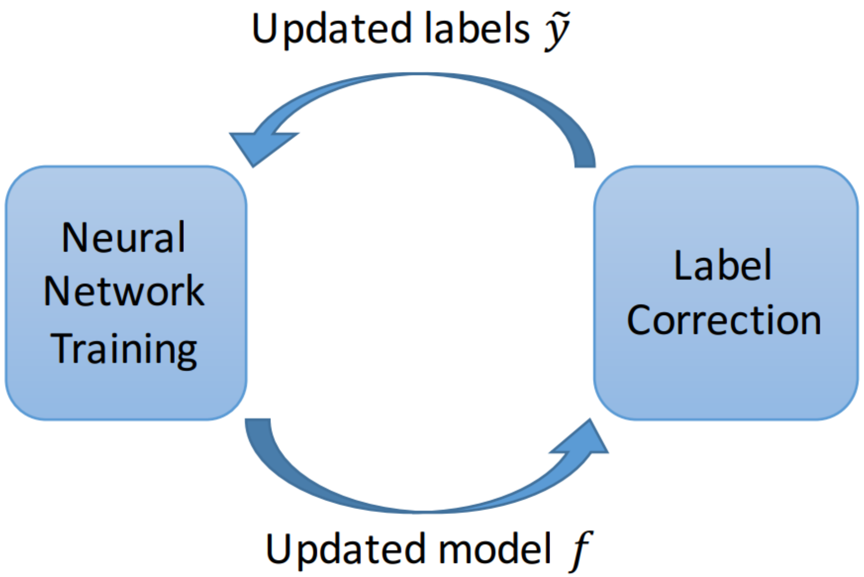
AdaCorr: Using LRT-Correction During Training

Step 1: Train $f(x)$ using $(x, \tilde{y})$

Step 2: Applying LRT-Correction using $(x, \tilde{y})$, $f(x)$ and δ

Step 3: Let $\tilde{y} = \tilde{y}_{new}$

Step 4: Repeat Step 1~3



Remark:
In step 1, to get a good approximation of $\tilde{\eta}(x)$, we train $f(x)$ with $(x, \tilde{y})$ for several warm-up epochs

Experiment - Setting

Data Sets:

- MNIST (LeCun & Cortes, 2010);
- CIFAR-10/CIFAR-100 (Krizhevsky et al., 2009);
- ModelNet40 (Z. Wu & Xiao, 2015)
- Clothing 1M (Xiao et al., 2015)

Base Lines:

- Forward Correction (Patrini et al., 2017)
- Decoupling (Malach & Shalev-Schwartz 2017)
- Forgetting (Arpit et al., 2017)
- Co-teaching (Han et al., 2018)
- MentorNet (Jiang et al., 2018)
- Abstention (Thulasidasan et al., 2019)

Backbone for every baseline:

- Preactive ResNet-34 (He et al., 2016) for MNIST; CIFAR10/100.
- ModelNet40 (Qi et al.) for Point Cloud.
- ResNet-50 for Cloth 1M

Epochs for every baseline: 180 epochs

Optimizer for every baseline: RAdam (Liu et al., 2019)

Learning Rate: 0.001 at beginning and decayed 0.5 for every 60 epochs

Hyper-parameter for AdaCorr:

- 30 epochs as Burning-in Period
- Initial $1/\delta$ is set to be 1.2 and decreased by 0.02 every epoch

Experiment - Performance

Data Set	Method	Noise Level of Uniform Flipping				Noise Level of Pair Flipping		
		0.2	0.4	0.6	0.8	0.2	0.3	0.4
MNIST	Standard	99.0 ± 0.2	98.7 ± 0.4	98.1 ± 0.3	91.3 ± 0.9	99.3 ± 0.1	99.2 ± 0.1	98.8 ± 0.1
	Forgetting	99.0 ± 0.1	98.8 ± 0.1	97.7 ± 0.2	62.6 ± 8.9	99.3 ± 0.1	96.5 ± 2.0	89.7 ± 1.9
	Forward	99.1 ± 0.1	98.7 ± 0.2	98.0 ± 0.4	89.6 ± 4.8	99.4 ± 0.0	99.2 ± 0.2	96.5 ± 4.4
	Decouple	99.3 ± 0.1	99.0 ± 0.1	98.5 ± 0.2	94.6 ± 0.2	99.4 ± 0.0	99.3 ± 0.1	99.1 ± 0.2
	MentorNet	99.2 ± 0.2	98.7 ± 0.1	98.1 ± 0.1	87.5 ± 5.2	98.6 ± 0.4	99.1 ± 0.1	98.9 ± 0.1
	Coteach	99.1 ± 0.2	98.7 ± 0.3	98.2 ± 0.3	95.7 ± 0.7	99.1 ± 0.1	99.0 ± 0.2	98.9 ± 0.2
	Abstention	94.0 ± 0.3	76.8 ± 0.3	49.6 ± 0.1	21.2 ± 0.5	94.3 ± 0.3	88.5 ± 0.3	81.4 ± 0.2
	AdaCorr	$\mathbf{99.5 \pm 0.0}$	$\mathbf{99.4 \pm 0.0}$	$\mathbf{99.1 \pm 0.0}$	$\mathbf{97.7 \pm 0.2}$	$\mathbf{99.5 \pm 0.0}$	$\mathbf{99.6 \pm 0.0}$	$\mathbf{99.4 \pm 0.0}$
CIFAR10	Standard	87.5 ± 0.2	83.1 ± 0.4	76.4 ± 0.4	47.6 ± 2.0	88.8 ± 0.2	88.4 ± 0.3	84.5 ± 0.3
	Forgetting	87.1 ± 0.2	83.4 ± 0.2	76.5 ± 0.7	33.0 ± 1.6	89.6 ± 0.1	83.7 ± 0.1	86.4 ± 0.5
	Forward	87.4 ± 0.8	83.1 ± 0.8	74.7 ± 1.7	38.3 ± 3.0	89.0 ± 0.5	87.4 ± 1.1	84.7 ± 0.5
	Decouple	87.6 ± 0.4	84.2 ± 0.5	77.6 ± 0.1	48.5 ± 0.9	90.6 ± 0.3	89.1 ± 0.3	86.3 ± 0.5
	MentorNet	90.3 ± 0.3	83.2 ± 0.5	75.5 ± 0.7	34.1 ± 2.5	90.4 ± 0.2	88.9 ± 0.1	83.3 ± 1.0
	Coteach	90.1 ± 0.4	87.3 ± 0.5	80.9 ± 0.5	25.0 ± 3.6	91.8 ± 0.1	89.9 ± 0.2	80.1 ± 0.7
	Abstention	85.3 ± 0.4	82.0 ± 0.7	68.8 ± 0.4	33.8 ± 7.7	88.5 ± 0.0	83.1 ± 0.5	77.4 ± 0.4
	AdaCorr	$\mathbf{91.0 \pm 0.3}$	$\mathbf{88.7 \pm 0.5}$	$\mathbf{81.2 \pm 0.4}$	$\mathbf{49.2 \pm 2.4}$	$\mathbf{92.2 \pm 0.1}$	$\mathbf{91.3 \pm 0.3}$	$\mathbf{89.2 \pm 0.4}$

Experiment - Performance

Data Set	Method	Noise Level of Uniform Flipping				Noise Level of Pair Flipping		
		0.2	0.4	0.6	0.8	0.2	0.3	0.4
CIFAR100	Standard	58.9 ± 0.8	52.1 ± 1.0	42.1 ± 0.7	20.8 ± 1.0	59.5 ± 0.4	52.9 ± 0.6	44.7 ± 1.3
	Forgetting	59.3 ± 0.8	53.0 ± 0.2	40.9 ± 0.5	7.7 ± 1.1	61.4 ± 0.9	54.6 ± 0.6	37.7 ± 4.6
	Forward	58.4 ± 0.5	52.2 ± 0.3	41.1 ± 0.5	20.6 ± 0.6	58.3 ± 0.7	53.2 ± 0.6	44.4 ± 2.8
	Decouple	59.0 ± 0.7	52.2 ± 0.7	40.2 ± 0.4	18.5 ± 0.8	60.8 ± 0.7	56.1 ± 0.7	48.4 ± 1.0
	MentorNet	63.6 ± 0.5	51.4 ± 1.4	38.7 ± 0.8	17.4 ± 0.9	64.7 ± 0.2	57.4 ± 0.8	47.4 ± 1.7
	Coteach	66.1 ± 0.5	60.0 ± 0.6	$\mathbf{48.3 \pm 0.1}$	16.1 ± 1.1	63.4 ± 0.9	57.6 ± 0.3	49.2 ± 0.3
	Abstention	$\mathbf{75.1 \pm 5.4}$	60.0 ± 0.8	51.1 ± 0.8	10.3 ± 0.5	65.4 ± 0.5	56.8 ± 0.5	47.3 ± 0.3
	AdaCorr	67.8 ± 0.1	$\mathbf{60.2 \pm 0.8}$	46.5 ± 1.2	$\mathbf{24.6 \pm 1.1}$	$\mathbf{68.3 \pm 0.2}$	$\mathbf{61.1 \pm 0.5}$	$\mathbf{49.8 \pm 0.7}$
ModelNet40	Standard	79.1 ± 2.6	75.3 ± 3.3	70.0 ± 3.0	57.9 ± 2.3	84.4 ± 1.2	82.3 ± 1.3	78.9 ± 0.7
	Forgetting	80.1 ± 1.8	73.9 ± 0.6	69.0 ± 0.7	26.2 ± 4.8	83.3 ± 1.1	62.0 ± 3.0	59.5 ± 2.9
	Forward	52.3 ± 5.1	49.4 ± 6.8	43.5 ± 5.2	28.2 ± 5.5	48.1 ± 6.8	48.0 ± 3.7	49.1 ± 4.4
	Decouple	82.5 ± 2.2	80.7 ± 0.7	72.9 ± 1.0	55.4 ± 2.7	85.7 ± 1.4	84.3 ± 1.0	80.5 ± 2.4
	MentorNet	86.5 ± 0.5	75.4 ± 1.8	70.9 ± 1.9	52.7 ± 3.1	83.7 ± 1.8	81.0 ± 1.5	79.3 ± 2.1
	Coteach	85.6 ± 0.9	84.2 ± 0.8	$\mathbf{81.8 \pm 1.1}$	68.9 ± 2.8	85.7 ± 0.8	79.1 ± 3.0	69.1 ± 2.4
	Abstention	78.1 ± 0.6	65.6 ± 0.5	45.6 ± 1.5	23.5 ± 0.5	82.3 ± 0.5	80.4 ± 0.6	65.6 ± 0.5
	AdaCorr	$\mathbf{86.9 \pm 0.3}$	$\mathbf{85.1 \pm 0.6}$	78.6 ± 1.4	$\mathbf{72.1 \pm 1.1}$	$\mathbf{87.6 \pm 0.4}$	$\mathbf{84.6 \pm 0.5}$	$\mathbf{83.7 \pm 0.5}$

Experiment - Performance

Table 1. Performance on Clothing 1M Dataset

Method	Accuracy(%)
Standard	68.94
Forward	69.84
Backward	69.13
AdaCorr	71.74 ± 0.12

Conclusion

- We addressed the training with label noise problem
- We provided the first theoretical justification for data re-calibration methods
  - We prove that noisy classifier can be used to decide the purity of the label
- We proposed a new theory inspired algorithm
  - scalable ; easy to tune; good performance.

Code will be available on GitHub: https://github.com/pingqingsheng/LRT

Thanks for watching