## Learning with Noisy Supervision

Part II: Statistical Learning with Noisy Supervision

Tongliang Liu

Trustworthy Machine Learning Lab School of Computer Science University of Sydney

## Structure



## Structure



## Learning without noisy labels

**Problem setup:** 

Data: 
$$S = \{(x_1, y_1), ..., (x_n, y_n)\} \sim D^n$$
.

Aim: Learn a classifier  $f \in F$ , such that  $\forall (x, y) \sim D$ , f(x) is a good prediction for y.

#### What is the best classifier we can obtain?

w.r.t. accuracy

To measure the accuracy, we define loss function  $\ell(x, y), f \mapsto \ell(y, f(x)) \in \mathbb{R}$ . For example, 0-1 loss:  $\mathbf{1}(y \neq \operatorname{sign}(f(x)))$ .

The best classifier should be the one that has the smallest loss on all the possible data from the domain.

Theoretically,

$$R_{D,0-1}(f) = \mathbb{E}_{(X,Y)\sim D}[\mathbf{1}(Y \neq \operatorname{sign}(f(X)))]$$
$$= \iint P(X = \mathbf{x}, Y = y) \mathbf{1}(y \neq \operatorname{sign}(f(\mathbf{x})))d\mathbf{x}dy$$
$$= 1 - \iint P(X = \mathbf{x}, Y = y) \mathbf{1}(y = \operatorname{sign}(f(\mathbf{x})))d\mathbf{x}dy.$$

$$f_{\rho}(\boldsymbol{x}) = \arg \max_{\boldsymbol{y}} P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}).$$

## Expected risk, Bayes classifier

The expected risk:  $R_{D,0-1}(f) = \mathbb{E}_{(X,Y)\sim D}[\mathbf{1}(Y \neq \operatorname{sign}(f(X)))].$ 

Bayes risk: 
$$R_{D,0-1}^* = \inf_f R_{D,0-1}(f)$$
.

The Bayes decision rule (Bayes classifier):  $f_{\rho} = \arg \inf_{f} R_{D,0-1}(f)$ .

Restricted Bayes risk: 
$$f^* = \inf_{f \in \mathcal{F}} R_{D,0-1}(f)$$
.

**Empirically**,

In reality, we can only observe a sample of data  $S = \{(x_1, y_1), \dots, (x_n, y_n)\} \sim D^n$ .

We approximate the expected risk R(f) via the empirical risk:  $\widehat{R}_{D,\ell}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$ .

We minimize the empirical risk to find a predictor:  $f_n = \arg\min_{f \in \mathcal{F}} \widehat{R}_{D,\ell}(f).$ 

#### Statistically consistent classifier [1,2]:

With high probability, as  $n \to \infty$ , we have: $R_{D,\ell}(f_n) \to R_{D,\ell}(f^*)$ .

[1] Mohri et al. *Foundations of machine learning*. MIT press, 2018. [2] Devroye, et al. *A probabilistic theory of pattern recognition*. Vol. 31. Springer Science & Business Media, 2013.

#### Aim:

Designing algorithms whose outputs will approach  $f_{\rho}(\mathbf{x}) = \arg \max_{y} P(Y = y | X = \mathbf{x}).$ 

## Structure



## Learning with noisy labels

Noisy sample:  $\tilde{S} = \{(x_1, \tilde{y}_1), ..., (x_n, \tilde{y}_n)\} \sim \tilde{D}^n$ , where  $\tilde{y}$  stands for noisy labels and  $\tilde{D}$  the noisy distribution.

What is the best classifier we can learn?

Can we approach  $f_{\rho}(\mathbf{x}) = \arg \max_{y} P(Y = y | X = \mathbf{x})$ ?

## Learning with noisy labels

One category: extracting confident examples or correct labels. SOTA, e.g., Co-teaching [3]; Joint Optim [4].

Another category: label-noise learning [5]. Methodology, i.e., statistically consistent algorithms.

[3] Han, Bo, et al. "Co-teaching: Robust training of deep neural networks with extremely noisy labels." *NeurIPS* 2018.
 [4] Tanaka, Daiki, et al. "Joint optimization framework for learning with noisy labels." *CVPR* 2018.
 [5] Xia, Xiaobo, et al. "Are anchor points really indispensable in label-noise learning?." *NeurIPS* 2019.

## Learning with noisy labels

One category: extracting confident examples or correct labels. SOTA, e.g., Co-teaching [3]; Joint Optim [4].

Another category: label-noise learning [5]. Methodology, i.e., statistically consistent algorithms.

[3] Han, Bo, et al. "Co-teaching: Robust training of deep neural networks with extremely noisy labels." *NeurIPS* 2018.
 [4] Tanaka, Daiki, et al. "Joint optimization framework for learning with noisy labels." *CVPR* 2018.
 [5] Xia, Xiaobo, et al. "Are anchor points really indispensable in label-noise learning?." *NeurIPS* 2019.

## Why called "label-noise learning"?

## Model label noise

#### Transition matrix:

$$\begin{bmatrix} P(\tilde{Y} = 1 | Y = 1, \mathbf{x}) & \cdots & P(\tilde{Y} = 1 | Y = C, \mathbf{x}) \\ \vdots & \ddots & \vdots \\ P(\tilde{Y} = C | Y = 1, \mathbf{x}) & \cdots & P(\tilde{Y} = C | Y = C, \mathbf{x}) \end{bmatrix}.$$

## Transition matrix

$$\begin{bmatrix} P(\tilde{Y} = 1 | \boldsymbol{x}) \\ \vdots \\ P(\tilde{Y} = C | \boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y} = 1 | Y = 1, \boldsymbol{x}) & \cdots & P(\tilde{Y} = 1 | Y = C, \boldsymbol{x}) \\ \vdots \\ P(\tilde{Y} = C | \boldsymbol{x}) \end{bmatrix} \begin{bmatrix} P(Y = 1 | \boldsymbol{x}) \\ \vdots \\ P(Y = C | Y = 1, \boldsymbol{x}) & \cdots & P(\tilde{Y} = C | Y = C, \boldsymbol{x}) \end{bmatrix} \begin{bmatrix} P(Y = 1 | \boldsymbol{x}) \\ \vdots \\ P(Y = C | \boldsymbol{x}) \end{bmatrix}$$

## Why called "label-noise learning"?

- Label-noise learning [5]
- Noisy-label learning
- Learning with noisy labels [6]

[5] Xia, Xiaobo, et al. "Are anchor points really indispensable in label-noise learning?." *NeurIPS* 2019.[6] Natarajan, Nagarajan, et al. "Learning with noisy labels." *NeurIPS* 2013.

## Model Label Noise

(1)Random Classification Noise (RCN) [7]:  $\rho_{\tilde{Y},Y}(X) = P(\tilde{Y}|Y,X) = P(\tilde{Y}|Y) = \rho, \forall Y \neq \tilde{Y}.$ 

(2)Class-conditional Noise (CCN) [6]:  $\rho_{\tilde{Y},Y}(X) = P(\tilde{Y}|Y,X) = P(\tilde{Y}|Y).$ 

(3) Instance-dependent Noise (IDN) [8,9]:  $\rho_{\tilde{Y},Y}(X) = P(\tilde{Y}|Y,X).$ 

[7] Angluin, Dana, and Philip Laird. "Learning from noisy examples." *Machine Learning* 2.4: 343-370, 1988.
 [8] Cheng, Jiacheng, et al. "Learning with bounded instance and label-dependent label noise." *ICML* 2020.
 [9] Berthon, Antonin, et al. "Confidence scores make instance-dependent label-noise learning possible." *ICML*, 2021 20

## Model Label Noise

(1)Random Classification Noise (RCN) [7]:  $\rho_{\tilde{Y},Y}(X) = P(\tilde{Y}|Y,X) = P(\tilde{Y}|Y) = \rho, \forall Y \neq \tilde{Y}.$ 

(2)Class-conditional Noise (CCN) [6]:  $\rho_{\tilde{Y},Y}(X) = P(\tilde{Y}|Y,X) = P(\tilde{Y}|Y).$ 

(3) Instance-dependent Noise (IDN) [8,9]:  $\rho_{\tilde{Y},Y}(X) = P(\tilde{Y}|Y,X).$ 

[7] Angluin, Dana, and Philip Laird. "Learning from noisy examples." *Machine Learning* 2.4: 343-370, 1988.
 [8] Cheng, Jiacheng, et al. "Learning with bounded instance and label-dependent label noise." *ICML* 2020.
 [9] Berthon, Antonin, et al. "Confidence scores make instance-dependent label-noise learning possible." *ICML*, 2021 21

## Random Classification Noise (RCN)

**Theorem 1.** The losses satisfying the following symmetric criterion is robust to RCN:

$$L(f(X), +1) + L(f(X), -1) = C,$$

where *C* is a constant. That is

$$\arg\min_{f} R_{D,L}(f) = \arg\min_{f} R_{\widetilde{D},L}(f).$$

Because:  $R_{\widetilde{D},L}(f) = \mathbb{E}_{(X,\widetilde{Y})\sim\widetilde{D}}[L(f(X),\widetilde{Y})] = (1-2\rho)R_{D,L}(f) + \rho C.$ 

[10] Du Plessis, Marthinus C. et al. "Analysis of learning from positive and unlabeled data." NeurIPS 2014

## Random Classification Noise (RCN)

The symmetric losses that are robust to RCN:

(1) 0-1 Loss:  $L(f(X), Y) = \mathbf{1}(sign(f(X)) \neq Y);$ 

(2) Unhinged Loss: L(f(X), Y) = 1 - Yf(X);

(3) Sigmoid Loss: 
$$L(f(X), Y) = \frac{1}{1 + e^{Yf(X)}}$$
;

(4) Ramp Loss:  $L(f(X), Y) = \frac{1}{2} \max(0, \min(2, 1 - Yf(X))) \dots$ 

## Class-conditional Noise (CCN)

The loss correction method: Modify  $\ell$  to be  $\tilde{\ell}$  such that

$$\mathbb{E}_{(X,\tilde{Y})\sim \widetilde{D}}\big[\widetilde{\ell}\big(f(X),\widetilde{Y}\big)\big] = \mathbb{E}_{(X,Y)\sim Y}[\ell(f(X),Y)]$$
  
By exploiting the model of label noise:

$$\begin{bmatrix} P(\tilde{Y}=1|\boldsymbol{x})\\ \vdots\\ P(\tilde{Y}=C|\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y}=1|Y=1) & \cdots & P(\tilde{Y}=1|Y=C)\\ \vdots & \ddots & \vdots\\ P(\tilde{Y}=C|Y=1) & \cdots & P(\tilde{Y}=C|Y=C) \end{bmatrix} \begin{bmatrix} P(Y=1|\boldsymbol{x})\\ \vdots\\ P(Y=C|\boldsymbol{x}) \end{bmatrix}$$

$$\begin{bmatrix} P(\tilde{Y}=1|\boldsymbol{x})\\ \vdots\\ P(\tilde{Y}=C|\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y}=1|Y=1) & \cdots & P(\tilde{Y}=1|Y=C)\\ \vdots & \ddots & \vdots\\ P(\tilde{Y}=C|Y=1) & \cdots & P(\tilde{Y}=C|Y=C) \end{bmatrix} \begin{bmatrix} P(Y=1|\boldsymbol{x})\\ \vdots\\ P(Y=C|\boldsymbol{x}) \end{bmatrix}$$

#### **Unbiased estimator (binary classification)** [6]:

$$\tilde{\ell}_{ue}(f(\mathbf{x}), y) = \frac{\left(1 - \rho_{y, -y}\right)\ell(f(\mathbf{x}), y) - \rho_{-y, y}\ell(f(\mathbf{x}), -y)}{1 - \rho_{-1, +1} - \rho_{+1, -1}}$$

The idea is that  $\mathbb{E}_{\tilde{y}|y}[\tilde{\ell}_{ue}(f(\boldsymbol{x}), \tilde{y})] = \ell(f(\boldsymbol{x}), y).$ 

Thus, 
$$\mathbb{E}_{(X,\tilde{Y})\sim\widetilde{D}}\left[\tilde{\ell}_{ue}(f(X),\tilde{Y})\right] = \mathbb{E}_{(X,Y)\sim D}\left[\ell(f(X),Y)\right]$$

[6] Natarajan, Nagarajan, et al. "Learning with noisy labels." NeurIPS 2013.



$$\begin{bmatrix} P(\tilde{Y}=1|\boldsymbol{x})\\ \vdots\\ P(\tilde{Y}=C|\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y}=1|Y=1) & \cdots & P(\tilde{Y}=1|Y=C)\\ \vdots & \ddots & \vdots\\ P(\tilde{Y}=C|Y=1) & \cdots & P(\tilde{Y}=C|Y=C) \end{bmatrix} \begin{bmatrix} P(Y=1|\boldsymbol{x})\\ \vdots\\ P(Y=C|\boldsymbol{x}) \end{bmatrix}$$

#### **Importance reweighting** [11]:

$$\tilde{\ell}_{ir}(f(\boldsymbol{x}), \boldsymbol{y}) = \frac{P(\boldsymbol{x}, \boldsymbol{y})}{\tilde{P}(\boldsymbol{x}, \boldsymbol{y})} \ell(f(\boldsymbol{x}), \boldsymbol{y}) = \frac{\boldsymbol{g}_{\boldsymbol{y}}(\boldsymbol{x})}{(T^{\top}\boldsymbol{g})_{\boldsymbol{y}}(\boldsymbol{x})} \ell(f(\boldsymbol{x}), \boldsymbol{y}),$$
  
where  $f(\boldsymbol{x}) = \arg \max_{j \in \{1, \dots, C\}} \boldsymbol{g}_{j}(\boldsymbol{x}).$ 

Thus, 
$$\mathbb{E}_{(X,\tilde{Y})\sim\widetilde{D}}\left[\tilde{\ell}_{ir}(f(X),\tilde{Y})\right] = \mathbb{E}_{(X,Y)\sim D}\left[\ell(f(X),Y)\right]$$

$$\begin{bmatrix} P(\tilde{Y}=1|\boldsymbol{x})\\ \vdots\\ P(\tilde{Y}=C|\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y}=1|Y=1) & \cdots & P(\tilde{Y}=1|Y=C)\\ \vdots & \ddots & \vdots\\ P(\tilde{Y}=C|Y=1) & \cdots & P(\tilde{Y}=C|Y=C) \end{bmatrix} \begin{bmatrix} P(Y=1|\boldsymbol{x})\\ \vdots\\ P(Y=C|\boldsymbol{x}) \end{bmatrix}$$

#### **Forward correction** [12]:



[12] Patrini, Giorgio, et al. "Making deep neural networks robust to label noise: A loss correction approach." CVPR 2017.27

## A summary of consistent algorithms

- Many methods for dealing with noisy labels Loss correction, Sample selection, label correction, ...
- Model label noise
  Random Classification Noise (RCN)
  Class-conditional Noise (CCN)
  Instance-dependent Noise (IDN)
- Symmetric loss functions are robust to RCN A loss function is symmetric if  $\sum_{y} \ell(f(x), y) = c$
- Three loss correction methods Unbiased estimator, importance reweighting, forward correction

## Structure



# How to estimate the transition matrix

Given the noisy data  $\tilde{S} = \{(x_1, \tilde{y}_1), \dots, (x_n, \tilde{y}_n)\} \sim \tilde{D}.$ 

How to estimate the transition matrix *T*?

### Anchor point assumption [11]

Rearrange the relationship among the noisy class posterior, the clean class posterior, and the transition matrix, we have

$$P(\tilde{Y} = 1 | \mathbf{x}) = (1 - \beta_{+1,-1} - \beta_{-1,+1})P(Y = 1 | \mathbf{x}) + \beta_{-1,+1}$$
$$P(\tilde{Y} = -1 | \mathbf{x}) = (1 - \beta_{+1,-1} - \beta_{-1,+1})P(Y = -1 | \mathbf{x}) + \beta_{+1,-1}$$

We designed the following estimator:  $\beta_{-y,+y} = \min_{x \in X} P(\tilde{Y} = y | x).$ 

#### Definition

## If $P(Y = i | \mathbf{x}^i) = 1$ , then $\mathbf{x}^i$ is called the anchor point for the *i*-th class.

## Anchor point assumption

$$\begin{bmatrix} P(\tilde{Y}=1|X)\\ \vdots\\ P(\tilde{Y}=C|X) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y}=1|Y=1) & \cdots & P(\tilde{Y}=1|Y=C)\\ \vdots\\ P(\tilde{Y}=C|Y=1) & \cdots & P(\tilde{Y}=C|Y=C) \end{bmatrix} \begin{bmatrix} P(Y=1|X)\\ \vdots\\ P(Y=C|X) \end{bmatrix}$$

$$\begin{bmatrix} P(\tilde{Y} = 1 | X = \mathbf{x}^{1}) \\ \vdots \\ P(\tilde{Y} = C | X = \mathbf{x}^{1}) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y} = 1 | Y = 1) \\ \vdots \\ P(\tilde{Y} = C | Y = 1) \end{bmatrix} \cdots \begin{bmatrix} P(\tilde{Y} = 1 | Y = C) \\ \vdots \\ P(\tilde{Y} = C | Y = 1) \end{bmatrix} \cdots \begin{bmatrix} P(\tilde{Y} = C | Y = C) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} P(\tilde{Y} = 1 | X = \mathbf{x}^{i}) \\ \vdots \\ P(\tilde{Y} = C | X = \mathbf{x}^{i}) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y} = 1 | Y = i) \\ \vdots \\ P(\tilde{Y} = C | Y = i) \end{bmatrix}$$

#### How to find anchor points

Binary classification, find the anchor points:  $x^{y} = \underset{x \in X}{\operatorname{argm} ax} P(\tilde{Y} = y | x).$ 

Multi-classification, approximate the anchor points for multi-class learning:  $x^y \approx \underset{x \in X}{\operatorname{argmax}} P(\tilde{Y} = y | x).$ 

#### *T* estimator vs Dual-*T* estimator [13]

$$T \text{ estimator:} \begin{bmatrix} P(\tilde{Y} = 1 | X = \mathbf{x}^i) \\ \vdots \\ P(\tilde{Y} = C | X = \mathbf{x}^i) \end{bmatrix} = \begin{bmatrix} P(\tilde{Y} = 1 | Y = i) \\ \vdots \\ P(\tilde{Y} = C | Y = i) \end{bmatrix}$$

Estimation error:  $|P(\tilde{Y} = c | \mathbf{x}) - \hat{P}(\tilde{Y} = c | \mathbf{x})| = \Delta_1$ .

[13] Yao Y, et al. Dual T: Reducing estimation error for transition matrix in label-noise learning. NeurIPS 2020.



[13] Yao Y, et al. Dual T: Reducing estimation error for transition matrix in label-noise learning. NeurIPS 2020.

#### T estimator vs Dual-T estimator

We let  $P(Y' = y | \mathbf{x}) = \hat{P}(\tilde{Y} = y | \mathbf{x})$ , where Y' is a variable for intermediate class.

Dual-*T* estimator:

$$T_{ij} = P(\tilde{Y} = j | Y = i) = \sum_{l=1}^{C} P(\tilde{Y} = j | Y' = l, Y = i) P(Y' = l | Y = i)$$
$$= \sum_{l=1}^{C} T_{lj}^{\clubsuit} (Y = i) T_{il}^{\clubsuit}.$$

[13] Yao Y, et al. Dual T: Reducing estimation error for transition matrix in label-noise learning. NeurIPS 2020.

## Estimation error of transition matrix



[13] Yao Y, et al. Dual T: Reducing estimation error for transition matrix in label-noise learning. NeurIPS 2020.

# Sufficiently scattered assumption vs anchor point assumption



[14] Li, Xuefeng, et al. "Provably end-to-end label-noise learning without anchor points." ICML 2021.

## VolMinNet [14]



$$\min_{\widehat{T} \in \mathbb{T}} \operatorname{vol} (\widehat{T})$$
  
s. t.  $\widehat{T}h_{\theta} = P(\widetilde{Y}|X)$ 

[14] Li, Xuefeng, et al. "Provably end-to-end label-noise learning without anchor points." ICML 2021.



[14] Li, Xuefeng, et al. "Provably end-to-end label-noise learning without anchor points." ICML 2021.

## T revision [15]



If 
$$P(\tilde{Y}|X = x) = [0.141; 0.189; 0.239; 0.281; 0.15],$$
  
then,  $P(Y|X = x) = (T^{\top})^{-1}P(\tilde{Y}|X = x) = [0.15; 0.28; 0.25; 0.3; 0.02].$   
 $P(Y|X = x) = (\tilde{T}^{\top})^{-1}P(\tilde{Y}|X = x) = [0.1587; 0.2697; 0.2796; 0.2593; 0.0325].$ 

## *T* revision [15]



Weighted loss 
$$= \frac{1}{n} \sum_{i=1}^{n} \frac{g_{\tilde{y}_i}(x_i)}{(T^{\mathsf{T}}g)_{\tilde{y}_i}(x_i)} L(f(x_i), \tilde{y}_i)$$
,  
where  $f(x) = \operatorname{argmax}_{i \in \{1, \dots, C\}} g_i(x)$ .



[15] Xia X, et al. Are Anchor Points Really Indispensable in Label-Noise Learning? NeurIPS. 2019

## A summary of estimating transition matrix

- How to estimate the transition matrix given only noisy data? Method: T estimator (by exploiting anchor points)
- Large estimation error of the noisy class posterior Method: Dual-T estimator (by decomposing the matrix)
- How about if there is no anchor points? Method: VolMinNet (using the sufficiently scattered assumption)
- How to deal with poorly estimated transition matrix Method: T revision (revising the matrix by using a slack variable)

## Conclusion and future directions

- Conclusion
  - Statistically consistent algorithms: the classifier learned by using noisy data will converge to the optimal one defined by using clean data
  - Statistically consistent algorithms are robust to the data distribution and label noise type
  - Modelling the label noise and estimating the transition matrix are cores in label-noise learning
- Future directions
  - Design effectively loss correction methods for deep learning
  - How to address the finite/small sample problem
  - How to use a small set of clean data to better estimate the transition matrix
  - How to model and estimate the instance-dependent label noise (IDN)