Learning Noise Transition Matrix from Only Noisy Labels via Total Variation Regularization

Yivan Zhang[1,2] Gang Niu[2] Masashi Sugiyama[2,1]

[1]The University of Tokyo [2]RIKEN AIP

Introduction

Problem

- **Noise transition matrix** is important in **learning from noisy labels**.
- However, it is usually unavailable or hard to obtain.
- Existing methods often depend on unreliable noisy class-posterior estimation.

Contribution

- We characterized the class-conditional label corruption process.
- We proposed a conceptually novel method for transition matrix estimation.

Methodology

- **Make probabilities more distinguishable**: total variation regularization
- **Capture uncertainties during training**: Dirichlet posterior update

Learning from Noisy Labels

Notation

- X: input features
- Y: true labels
- $\tilde{Y}$: noisy labels

Assumption

$$X \to Y \to \tilde{Y}$$

Class-Conditional Noise (CCN) assumes that the noisy label $\tilde{Y}$ is independent of the input feature X given the true label Y: $p(\tilde{Y}|Y,X) = p(\tilde{Y}|Y)$.

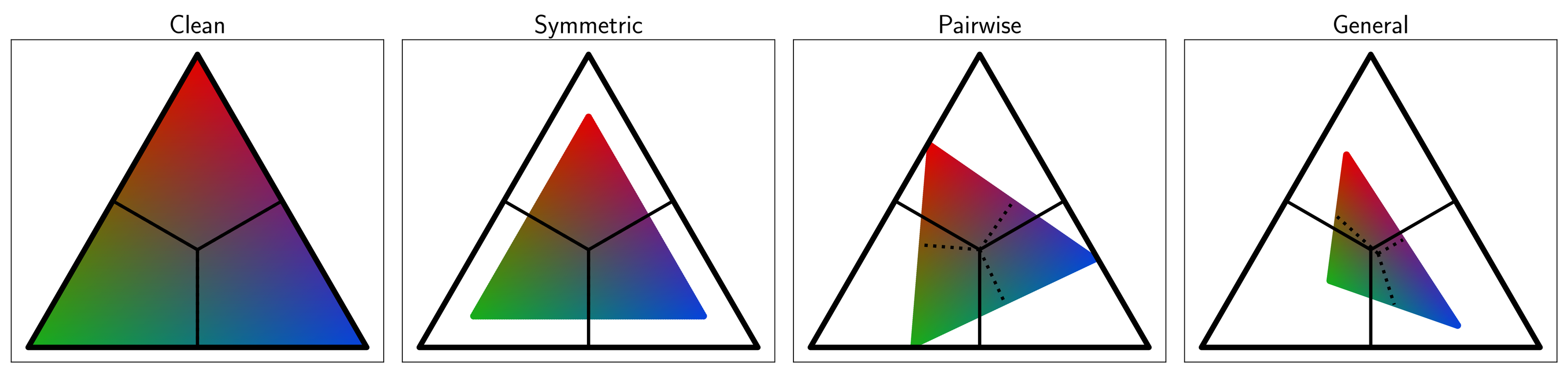
Noise Transition Matrix $T_{ij} = p(\tilde{Y}=j|Y=i)$

$$\begin{bmatrix} p(\tilde{Y}=1|X) \\ \vdots \\ p(\tilde{Y}=K|X) \end{bmatrix} = \begin{bmatrix} p(\tilde{Y}=1|Y=1) & \dots & p(\tilde{Y}=1|Y=K) \\ \vdots & \ddots & \vdots \\ p(\tilde{Y}=K|Y=1) & \dots & p(\tilde{Y}=K|Y=K) \end{bmatrix} \begin{bmatrix} p(Y=1|X) \\ \vdots \\ p(Y=K|X) \end{bmatrix}$$

$$\Downarrow$$

$$\boldsymbol{p}(\tilde{Y}|X) = \boldsymbol{T}^\top \boldsymbol{p}(Y|X)$$

Noise Transition Matrix



$$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \quad \begin{bmatrix} 0.7 & 0.3 & 0.0 \\ 0.0 & 0.7 & 0.3 \\ 0.3 & 0.0 & 0.7 \end{bmatrix} \quad \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$$

Class-conditional label corruption maps the probability simplex Δ^{K-1} to a convex hull $\mathrm{Conv}(\boldsymbol{T})$ of the rows of the noise transition matrix $\boldsymbol{T}$.

- Outer black triangle: probability simplex Δ^2
- Inner colored triangle: convex hull $\mathrm{Conv}(\boldsymbol{T})$

Good news: if the ground-truth noise transition matrix $\boldsymbol{T}$ is known, $\boldsymbol{p}(Y|X)$ is **identifiable** based on observations of $\boldsymbol{p}(\tilde{Y}|X)$ [Patrini et al., 2017].

Problem

Noise transition matrix is usually not available [Patrini et al., 2017].

Solution

Learn the noise transition matrix **from only noisy labels**.

Anchor Points

- An instance x is called an anchor point for class i if $p(Y=i|X=x)=1$.
- Based on anchor points, we can estimate $\boldsymbol{p}(\tilde{Y}|X)$ to obtain an estimate of $\boldsymbol{T}$.

$$\boldsymbol{p}(\tilde{Y}|X=x) = \boldsymbol{T}^\top \boldsymbol{p}(Y|X=x) = \boldsymbol{T}_i$$

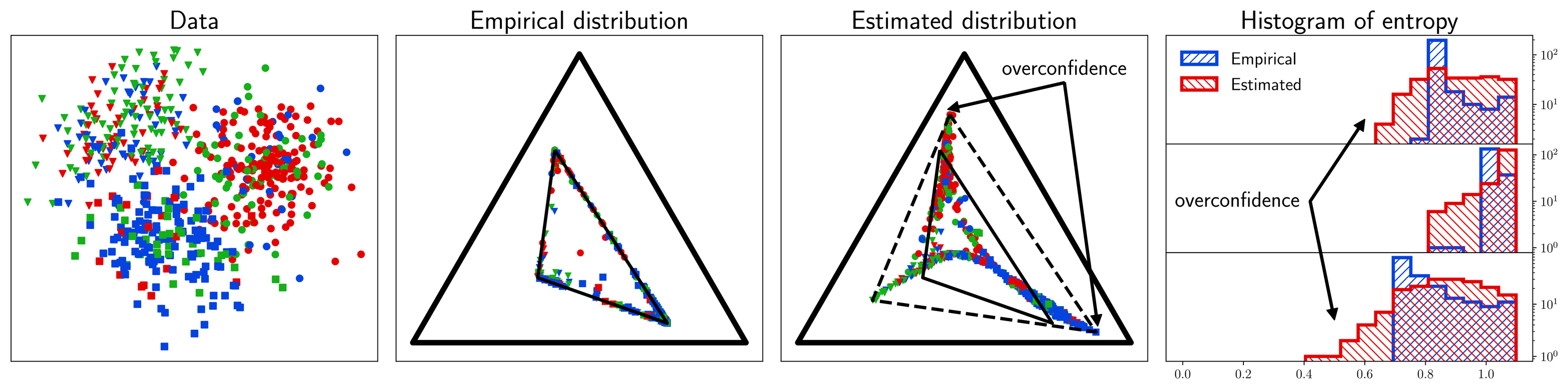
Problem

Anchor points are hard to obtain [Xia et al., 2019, Yao et al., 2020].

Solution

Do not rely on a separate set of anchor points.

Overconfidence



Problem

The estimation of the noisy class-posterior could be unreliable due to the **overconfidence** of deep neural networks [Guo et al., 2017, Hein et al., 2019].

Solution

Do not estimate the noisy class-posterior directly using neural networks.

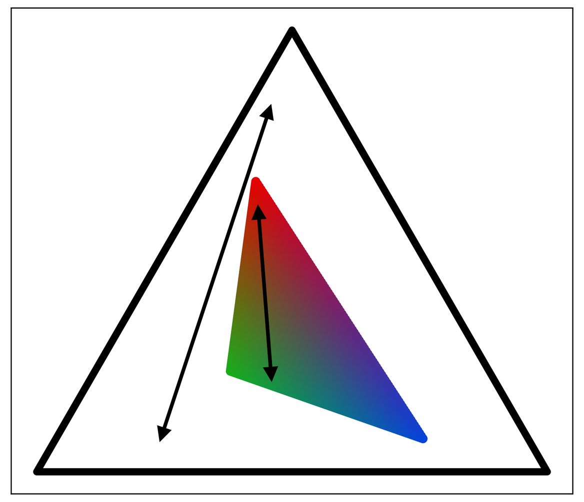
Motivation

Transition Matrix as a Contraction Mapping

The mapping $\Delta \to \mathrm{Conv}(\boldsymbol{U})$ defined by $\boldsymbol{p} \mapsto \boldsymbol{U}^\top \boldsymbol{p}$ is a **contraction mapping** over the simplex Δ relative to the total variation distance [Del Moral et al., 2003]:

$$\forall \boldsymbol{U} \in \mathcal{T}, \forall \boldsymbol{p}, \boldsymbol{q} \in \Delta,$$
$$d_{\mathrm{TV}}(\boldsymbol{U}^\top \boldsymbol{p}, \boldsymbol{U}^\top \boldsymbol{q}) \le d_{\mathrm{TV}}(\boldsymbol{p}, \boldsymbol{q})$$

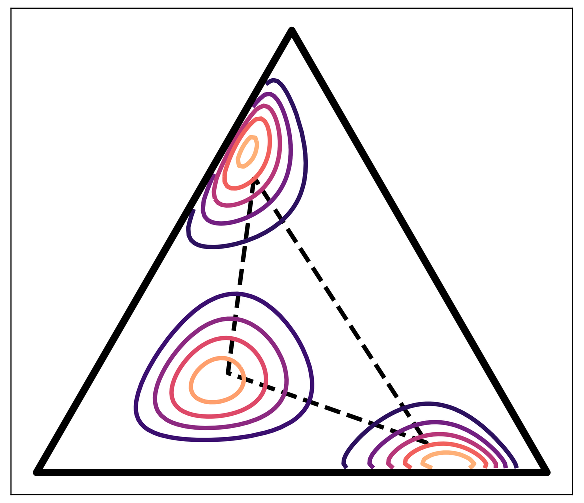
Clean class-posteriors are always more **distinguishable** from each other than noisy class-posteriors.



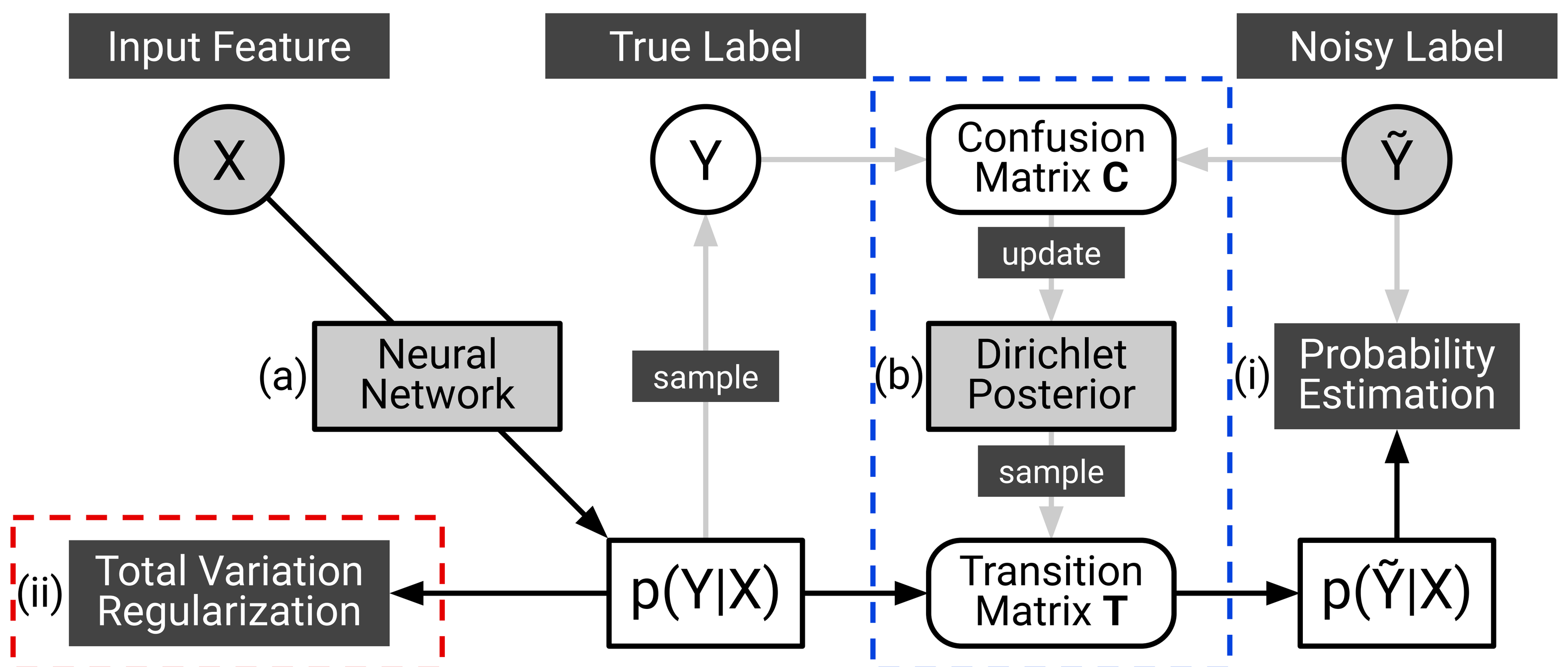
Transition Matrix Estimation

In addition to the gradient information, the **confusion matrix** can be used to estimate the transition matrix.

To capture uncertainties during training, we could use **Dirichlet distributions** to accumulate information of confusion matrices, which leads us to a **derivative-free** approach for transition matrix estimation.



Proposed Method



Our model has two modules:

- (a) a **neural network** for predicting $\boldsymbol{p}(Y|X)$
- (b) a **Dirichlet posterior** for the noise transition matrix $\boldsymbol{T}$

The learning objective also contains two parts:

- (i) the usual **cross-entropy loss** for classification from noisy labels
- (ii) a **total variation regularization** term for the predicted probability

Implementation

Total Variation Regularization

We **sample a fixed number of pairs** to reduce the additional computational cost.

$$d_{\mathrm{TV}}(\boldsymbol{p}, \boldsymbol{q}) := \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{q}\|_1$$
$$R(W) := \mathop{\mathbb{E}}_{X_1 \sim p(X)} \mathop{\mathbb{E}}_{X_2 \sim p(X)} [d_{\mathrm{TV}}(\boldsymbol{p}_1, \boldsymbol{p}_2)]$$
$$\text{where } \boldsymbol{p}_i := \boldsymbol{p}(Y|X_i; W) \quad i = 1, 2$$

```
p = model(x)  # probability [batch_size, num_classes]
idx_1, idx_2 = randint(0, batch_size, (2, num_pairs))
tv = 0.5 * l1_norm(p[idx_1] - p[idx_2], dim=1).mean()
```

Dirichlet Posterior Update

Inspired by the closed-form posterior update rule for the Dirichlet-multinomial conjugate, we **update the concentration parameters** $\boldsymbol{A}$ during training using the confusion matrix $\boldsymbol{C}$, where (β_1, β_2) are fixed hyperparameters.

$$\boldsymbol{A}^{(\text{posterior})} = \boldsymbol{A}^{(\text{prior})} + \boldsymbol{C}^{(\text{observation})}$$
$$\boldsymbol{A} \leftarrow \beta_1 \boldsymbol{A} + \beta_2 \boldsymbol{C}$$

```
y = Categorical(p).sample()  # predicted labels
C = confusion_matrix(y, y_)  # confusion matrix
A = beta_1 * A + beta_2 * C  # update
```

Optimization

For each batch of data, we **sample a transition matrix** from the Dirichlet posterior.

$$\boldsymbol{T}_i \sim \mathrm{Dirichlet}(\boldsymbol{A}_i) \quad (i = 1, \dots, K)$$
$$L_0(W, \boldsymbol{T}) := \mathop{\mathbb{E}}_{X \sim p(X)} \left[D_{\mathrm{KL}}\left(\boldsymbol{p}(\tilde{Y}|X) \,\middle\|\, \boldsymbol{T}^\top \boldsymbol{p}(Y|X; W) \right) \right]$$
$$\mathcal{L}(W, \boldsymbol{T}) := L_0(W, \boldsymbol{T}) - \gamma R(W)$$

```
T = Dirichlet(A).sample()  # transition matrix
loss = cross_entropy(p @ T, y_) - gamma * tv
```

Experiments

Improved classification performance, measured by **accuracy**.

		(a) Clean	(b) Symm.	(c) Pair	(d) Pair^2	(e) Trid.	(f) Rand.
CIFAR100	MAE	11.23(1.02)	7.89(0.67)	6.94(1.11)	6.60(0.74)	7.45(0.55)	7.15(0.98)
	CCE	**70.58(0.29)**	42.94(0.47)	44.00(0.71)	41.37(0.27)	46.55(0.54)	42.41(0.48)
	GCE	57.10(0.85)	48.66(0.58)	45.27(0.85)	43.67(0.94)	50.98(0.33)	48.66(0.63)
	Forward	**70.58(0.28)**	44.32(0.64)	44.17(0.57)	42.07(0.55)	47.48(0.40)	43.15(0.53)
	T-Revision	**70.47(0.26)**	46.52(0.57)	44.08(0.42)	42.01(0.52)	47.59(0.60)	45.33(0.40)
	Dual-T	**70.56(0.28)**	55.92(0.60)	46.22(0.72)	44.74(0.65)	61.68(0.51)	57.92(0.50)
	TVG	70.02(0.30)	**57.33(0.42)**	45.68(0.85)	44.38(0.72)	54.23(0.53)	**59.85(0.61)**
	TVD	69.93(0.21)	52.54(0.45)	**56.02(0.82)**	**49.18(0.53)**	**62.45(0.44)**	53.95(0.47)

Improved transition matrix estimation, measured by **average total variation**.

		(a) Clean	(b) Symm.	(c) Pair	(d) Pair^2	(e) Trid.	(f) Rand.
CIFAR-100	Forward	**0.00(0.00)**	48.62(0.11)	39.81(0.03)	43.57(0.04)	40.92(0.07)	49.06(0.10)
	T-Revision	0.46(0.05)	31.58(0.46)	39.45(0.03)	42.77(0.06)	40.01(0.09)	39.49(0.26)
	Dual-T	3.10(0.08)	17.10(0.18)	33.26(0.20)	33.79(0.26)	**23.56(0.43)**	22.59(0.23)
	TVG	1.59(0.02)	**13.11(0.10)**	37.79(0.30)	38.83(0.34)	30.80(0.51)	**16.47(0.18)**
	TVD	21.98(0.11)	26.46(0.15)	**29.47(0.26)**	**31.34(0.30)**	23.86(0.22)	35.37(0.30)

References

Pierre Del Moral, Michel Ledoux, and Laurent Miclo. On contraction properties of markov kernels. *Probability theory and related fields*, 126(3):395–420, 2003.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330, 2017.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems*, pages 6838–6849, 2019.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual T: Reducing estimation error for transition matrix in label-noise learning. In *Advances in Neural Information Processing Systems*, pages 7260–7271, 2020.