

# Multi-Class Classification from Noisy-Similarity-Labeled Data

Songhua Wu<sup>1\*</sup>, Xiaobo Xia<sup>1,2\*</sup>, Tongliang Liu<sup>1</sup>, Bo Han<sup>4,5</sup>,  
Mingming Gong<sup>3</sup>, Nannan Wang<sup>2</sup>, Haifeng Liu<sup>6</sup>, Gang Niu<sup>4</sup>

<sup>1</sup>University of Sydney; <sup>2</sup>Xidian University; <sup>3</sup>The University of Melbourne;  
<sup>4</sup>RIKEN; <sup>5</sup>Hong Kong Baptist University; <sup>6</sup>Brain-Inspired Technology Co., Ltd.

## Abstract

A similarity label indicates whether two instances belong to the same class while a class label shows the class of the instance. Without class labels, a multi-class classifier could be learned from similarity-labeled pairwise data by meta classification learning [Hsu et al., 2019]. However, since the similarity label is less informative than the class label, it is more likely to be noisy. Deep neural networks can easily remember noisy data, leading to overfitting in classification. In this paper, we propose a method for learning from only noisy-similarity-labeled data. Specifically, to model the noise, we employ a noise transition matrix to bridge the class-posterior probability between clean and noisy data. We further estimate the transition matrix from only noisy data and build a novel learning system to learn a classifier which can assign noise-free class labels for instances. Moreover, we theoretically justify how our proposed method generalizes for learning classifiers. Experimental results demonstrate the superiority of the proposed method over the state-of-the-art method on benchmark-simulated and real-world noisy-label datasets.

---

\*Equal contributions.

# 1 Introduction

Supervised classification crucially relies on the amount of data and the accuracy of corresponding labels. Since the data volume grows very quickly while supervision information cannot catch up with its growth, weakly supervised learning (WSL) is becoming more and more prominent [Zhou, 2017, Han et al., 2019, Wang et al., 2019, Li et al., 2017, 2018, Krause et al., 2016, Khetan et al., 2017, Hu et al., 2019a]. Among WSL, similarity-based learning is one of the hottest emerging problems [Bao et al., 2018, Hsu et al., 2019]. Compared with class labels, similarity labels are usually easier to obtain [Bao et al., 2018], especially when we encounter some sensitive issues, e.g., religion and politics. Take an illustrative example from Bao *et al.* [Bao et al., 2018]: for sensitive matters, people often hesitate to directly answer “What is your opinion on issue A?”, while they are more likely to answer “With whom do you share the same opinion on issue A?”. Intuitively, similarity information can not only alleviate embarrassment but also protect personal privacy to some degree.

Existing methods for similarity-based learning can be divided into two categories generally: semi-supervised clustering [Wagstaff et al., 2001, Xing et al., 2003] and weakly-supervised classification [Bao et al., 2018, Shimada et al., 2019]. The first category utilizes pairwise similarity and dissimilarity data for clustering. For example, pairwise links were used as constraints on clustering [Li and Liu, 2009]; Similar and dissimilar data pairs were used for metric learning, which learns a distance function over instances and can easily convert to clustering tasks [Niu et al., 2014]. The second category aims at classification, which not only separates different clusters but also identifies which class each cluster belongs to. For example, similarity and unlabeled (SU) learning proposed an unbiased estimator for binary classification [Bao et al., 2018]; Meta classification learning (MCL) showed a method to learn a multi-class classifier from only similarity data [Hsu et al., 2019].

All existing methods are based on the strong assumption that similarity labels are entirely accurate. However, similarity labels are hard to be fully accurate for many applications. For example, for some sensitive matters, people may not be willing to provide their real thoughts even when facing easy questions. It is commonly known that deep networks can memorize all the training data even there is noisy supervision, which tends to lead to the overfitting problem [Zhang et al., 2016, Zhong et al., 2019a, Li et al., 2019, Yi and Wu, 2019, Zhang et al., 2019, Tanno et al., 2019, Zhang et al., 2018]. Thus, if we directly employ the existing deep learning algorithms to deal with noisy similarity-based supervision, the test performance will inevitably degenerate because of overfitting. To the best of our knowledge, no pioneer work has been done to tackle the problem of binary classification with noisy similarity information, not to mention how to learn multi-class classifiers with theoretical guarantees.

In this paper, we study the problem of how to learn a Multi-class classifier from Noisy-Similarity-labeled data, which is called MNS classification. Specifically, we assume that latent clean class labels  $Y$  flip into latent noisy labels  $\bar{Y}$ , leading to noisy similarity labels  $\bar{S}$ . The corresponding graphical model, representing the interactions among variables, is

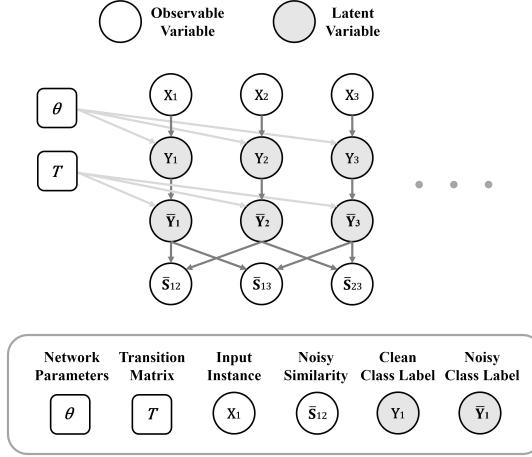


Figure 1: The assumed graphical representation for the proposed Multi-class classification with Noisy-Similarity-labeled data (called MNS classification), where  $X_i$  denotes the input instance;  $Y_i$  denotes the clean class label;  $\bar{Y}_i$  denotes the noisy class label;  $\bar{S}_{ii'}$  is noisy pairwise similarity supervision between  $(X_i, \bar{Y}_i)$  and  $(X_{i'}, \bar{Y}_{i'})$ ;  $\theta$  denotes the neural network parameters;  $T$  denotes the noise transition matrix. The latent variables are denoted by white circles and the observable variables are presented by grey circles.

shown in Figure 1. Based on this, we could further model the noise in the problem by using a transition matrix, i.e.,  $T_{ij}$  represents the probabilities that the clean class label  $i$  flips into the noisy class label  $j$  and  $T_{ij}(X) = P(\bar{Y} = j | Y = i, X)$ . We will show that under a mild assumption that *anchor points* (defined in 3.3) exist in the training data, we can estimate the transition matrix by only employing noisy-similarity-labeled data. Then, we build a deep learning system for multi-class classification from only noisy-similarity-labeled data. Note that if a good classifier can be learned, the corresponding method can be easily extended to learn metrics or clusters, because accurate labels and similar and dissimilar pairs can be assigned by the good classifier. In other words, the proposed method can not only learn a classifier from noisy-similarity-labeled data but metrics and clusters. The contributions of this paper are summarized as follows:

- We propose a deep learning system for multi-class classification to address the problem of how to learn from noisy-similarity-labeled data.
- We propose to model the noise by using the transition matrix based on a graphical model. We show that the transition matrix can be estimated from only noisy-similarity-labeled data. The effectiveness will be verified on both synthetic and real data.
- We theoretically establish a generalization error bound for the proposed MNS classification method, showing that the learned classifier will generalize well on unseen data.

- We empirically demonstrate that the proposed method can effectively reduce the side effect of noisy-similarity-labeled data. It significantly surpasses the baselines on many datasets with both synthetic noise and real-world noise <sup>1</sup>.

The rest of this paper is organized as follows. In Section 2, we formalize the MNS classification problem, and in Section 3, we propose the MNS learning and practical implementation. Generalization error bound is analysed in Section 4. Experimental results are discussed in Section 5. We conclude our paper in Section 6.

## 2 Framing the MNS classification Problem

**Problem setup.** Let  $\mathcal{D}$  be the distribution of a pair of random variables  $(X, Y) \in \mathcal{X} \times [C]$ , where  $\mathcal{X} \subset \mathbb{R}^d$  and  $d$  represents the dimension;  $\mathcal{Y} = [C]$  is the label space and  $[C] = \{1, \dots, C\}$  is the number of classes. Our goal is to predict a label for any given instance  $X \in \mathcal{X}$ . Different from the traditional multi-class classification, in our setting, the class labels are not observable. Instead, we have noisy similarity labels  $\bar{S} \in \{0, 1\}$ . The clean similarity labels  $S$  indicate the similarities between examples, i.e.,  $S_{ii'} = \mathbb{1}[Y_i = Y_{i'}]$  where  $Y_i$  and  $Y_{i'}$  denote the class labels for instances  $X_i$  and  $X_{i'}$ . For noisy similarity labels, some of them are identical to the clean similarity labels, but some are different and we do not know which of them are clean. To the best of our knowledge, no existing work has discussed how to learn with the noisy similarity labels. We would like to review how the state-of-the-art work learns a classifier from the clean similarity labels.

**MCL classification** [Hsu et al., 2019]. Meta classification learning (MCL) utilizes the following likelihood to explain the similarity-based data

$$\mathcal{L}(\theta; X, Y, S) = P(X, Y, S; \theta) = P(S|Y)P(Y|X; \theta)P(X). \quad (1)$$

By introducing an independence assumption:  $S_{ii'} \perp S \setminus \{S_{ii'}\} | X_i, X_{i'}$  [Hsu et al., 2019, Appendix D], in other words,  $S_{ii'}$  and  $S \setminus \{S_{ii'}\}$  are independent to each other given  $X_i$  and  $X_{i'}$ ; they can simplify the likelihood expression as

$$\begin{aligned} \mathcal{L}(\theta; X, S) &\approx \prod_{i, i'} \left( \sum_{Y_i=Y_{i'}} \mathbb{1}[S_{ii'} = 1] P(Y_i|X_i; \theta) P(Y_{i'}|X_{i'}; \theta) \right. \\ &\quad \left. + \sum_{Y_i \neq Y_{i'}} \mathbb{1}[S_{ii'} = 0] P(Y_i|X_i; \theta) P(Y_{i'}|X_{i'}; \theta) \right). \end{aligned} \quad (2)$$

Then taking a negative logarithm on Equation 2, the final loss function can be derived as

$$\begin{aligned} L_{meta}(\theta) &= - \sum_{i, i'} S_{ii'} \log(g(X_i; \theta)^T g(X_{i'}; \theta)) \\ &\quad + (1 - S_{ii'}) \log(1 - g(X_i; \theta)^T g(X_{i'}; \theta)), \end{aligned} \quad (3)$$

---

<sup>1</sup>Datasets with real-world noise refer the noisy-similarity-labeled data where noisy similarity labels are generated using real-world data with label noise.

where  $g(X_i; \theta) = P(Y_i|X_i; \theta)$ , which can be learned from a neural network.

However, class label noise is ubiquitous in our daily life [Kaneko et al., 2019, Hu et al., 2019b, Zhong et al., 2019b, Acuna et al., 2019, Lee et al., 2018, Tanaka et al., 2018, Wang et al., 2018], not to mention the weaker supervision: similarity labels. The performance of classifiers will get worse if we still use the state-of-the-art methods designed for clean similarity labels. This motivates us to find a novel algorithm for learning from noisy-similarity-labeled data.

### 3 MNS Learning

In this section, we propose a method for multi-class classification from noisy-similarity-labeled data.

#### 3.1 Modeling noise in the supervision

To learn from the noisy-similarity-labeled data, we should model the noise. To model the noise, we introduce a graphic model in Figure 1 to describe the interactions among variables, where only input instances  $X$  and noisy similarity labels  $\bar{S}$  are observed while both clean class labels  $Y$  and noisy class labels  $\bar{Y}$  are latent. Rather than modeling the similarity-label noise directly, we assume that noise first occurs on latent class labels and as a consequence, similarity labels turn to noisy ones, i.e., noisy similarity labels  $\bar{S}_{ii'} \in \{0, 1\}$  indicate the similarities between noisy examples, and  $\bar{S}_{ii'} = \mathbb{1}[\bar{Y}_i = \bar{Y}_{i'}]$ . The assumption is reasonable. For example, in the sensitive matters, to hide one's thought on the question “With whom do you share the same opinion on issue A?”, people would like to randomly choose a fake opinion about the issue and answer the question conditioned on the fake opinion.

Specifically, to precisely describe label noise, we utilize a *noise transition matrix*  $T \in [0, 1]^{C \times C}$  [Cheng et al., 2017]. The transition matrix is generally dependent on instances, i.e.,  $T_{ij}(X) = P(\bar{Y} = j|Y = i, X)$ . Given only noisy examples, the instance-dependent transition matrix is non-identifiable without any additional assumption [Xia et al., 2019]. In this paper, we assume that given  $Y$ ,  $\bar{Y}$  is independent on instance  $X$  and  $P(\bar{Y} = j|Y = i, X) = P(\bar{Y} = j|Y = i)$ . This assumption considers the situations where noise relies only on the classes, which has been widely adopted in the class-label-noise learning community [Han et al., 2018, Xia et al., 2019]. Empirical results on real-datasets verify the efficiency of the assumptions.

We denote by  $\mathcal{D}_\rho$  the distribution of the noisy-similarity-labeled data  $(X_i, X_{i'}, \bar{S}_{ii'})$ , and the classifier is supposed to be learned from a training sample drawn from  $\mathcal{D}_\rho$ .

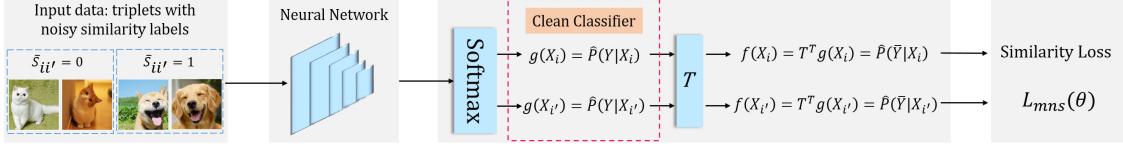


Figure 2: An overview of the proposed method. We add a noise transition matrix layer to model the noise. By minimizing the proposed loss  $L_{mns}(\theta)$ , a classifier  $g$  can be learned for assigning clean labels. The detailed structures of the Neural Network are provided in Section 5. Note that for the noisy similarity labels, some of them are correct and some are not. The similarity label for dogs is correct and the similarity label for cats is incorrect.

### 3.2 Likelihood-based estimator

Intuitively, according to figure 1, we can explain the noisy-similarity-based data by using the following likelihood model

$$\begin{aligned}\mathcal{L}(\theta; X, Y, \bar{Y}, \bar{S}) &= P(X, Y, \bar{Y}, \bar{S}; \theta) \\ &= P(\bar{S}|\bar{Y})P(\bar{Y}|Y)P(Y|X; \theta)P(X).\end{aligned}\quad (4)$$

In order to calculate the above likelihood, we have to marginalize the clean class label  $Y$  and noisy class label  $\bar{Y}$ . Thanks to our proposed deep learning system (summarized in Figure 2),  $P(\bar{Y}|Y)$ , modeled by a noise transition matrix  $T$ , could be learned only from noisy data (shown in Section 3.3). Therefore, we only need to marginalize noisy class label  $\bar{Y}$ . With the independence assumption  $S_{ii'} \perp S \setminus \{S_{ii'}\} | X_i, X_{i'}$ , we can calculate the likelihood with the following expression

$$\begin{aligned}\mathcal{L}(\theta; X, Y, \bar{Y}, \bar{S}) &\propto \sum_{\bar{Y}} \sum_Y P(\bar{S}|\bar{Y})P(\bar{Y}|Y)P(Y|X; \theta) \\ &= \prod_{i,i'} \left( \sum_{\bar{Y}_i=\bar{Y}_{i'}} \mathbb{1}[\bar{S}_{ii'} = 1] \sum_Y P(\bar{Y}_i|Y)P(Y|X_i; \theta) \right. \\ &\quad \sum_Y P(\bar{Y}_{i'}|Y)P(Y|X_{i'}; \theta) \\ &\quad \left. + \sum_{\bar{Y}_i \neq \bar{Y}_{i'}} \mathbb{1}[\bar{S}_{ii'} = 0] \sum_Y P(\bar{Y}_i|Y)P(Y|X_i; \theta) \right. \\ &\quad \left. \sum_Y P(\bar{Y}_{i'}|Y)P(Y|X_{i'}; \theta) \right) \\ &= \prod_{i,i'} \left( \sum_{\bar{Y}_i=\bar{Y}_{i'}} \mathbb{1}[\bar{S}_{ii'} = 1] P(\bar{Y}_i|X_i; \theta)P(\bar{Y}_{i'}|X_{i'}; \theta) \right. \\ &\quad \left. + \sum_{\bar{Y}_i \neq \bar{Y}_{i'}} \mathbb{1}[\bar{S}_{ii'} = 0] P(\bar{Y}_i|X_i; \theta)P(\bar{Y}_{i'}|X_{i'}; \theta) \right)\end{aligned}\quad (5)$$

where the proportion relationship holds because  $P(X)$  is constant for given  $X$  such that can be omitted. Note that

$$\begin{aligned} P(\bar{Y}_i|X_i; \theta) &= \sum_Y P(\bar{Y}_i|Y)P(Y|X_i; \theta) \\ &= \sum_{k=1}^C T_{ki}P(Y = k|X_i; \theta). \end{aligned} \quad (6)$$

Let  $g(X) = P(Y|X; \theta)$  and  $f(X) = P(\bar{Y}|X; \theta)$ , we have

$$f(X) = P(\bar{Y}|X; \theta) = T^\top P(Y|X; \theta) = T^\top g(X). \quad (7)$$

Then by taking a negative logarithm on Equation 5 and substituting  $P(\bar{Y}|X; \theta)$  with  $f(X)$ , we obtain the objective function of the proposed method, i.e.,

$$\begin{aligned} L_{mns}(\theta; X_i, X_{i'}, \bar{S}_{ii'}) &= - \sum_{i,i'} \log \left( \sum_{\bar{Y}_i=\bar{Y}_{i'}} \mathbb{1}[\bar{S}_{ii'} = 1] P(\bar{Y}_i|X_i; \theta) P(\bar{Y}_{i'}|X_{i'}; \theta) \right. \\ &\quad \left. + \sum_{\bar{Y}_i \neq \bar{Y}_{i'}} \mathbb{1}[\bar{S}_{ii'} = 0] P(\bar{Y}_i|X_i; \theta) P(\bar{Y}_{i'}|X_{i'}; \theta) \right) \\ &= - \sum_{i,i'} \bar{S}_{ii'} \log(f(X_i; \theta)^T f(X_{i'}; \theta)) + \\ &\quad (1 - \bar{S}_{ii'}) \log(1 - f(X_i; \theta)^T f(X_{i'}; \theta)). \end{aligned} \quad (8)$$

Let us look inside Equation 8. Intuitively,  $f(X; \theta)$  outputs the predicted noisy categorical distribution of instance  $X$  and  $f(X_i; \theta)^T f(X_{i'}; \theta)$  is exactly the predicted noisy similarity, indicating the probability of data pairs belonging to the same noisy class. For clarity, we visualize the predicted noisy similarity in Figure 3. If  $X_i$  and  $X_{i'}$  are predicted belonging to the same class, i.e.,  $\text{argmax}_{m \in C} f_m(X_i; \theta) = \text{argmax}_{n \in C} f_n(X_{i'}; \theta)$ , the predicted noisy similarity should be relatively high ( $\hat{\bar{S}}_{ii'} = 0.30$  in Figure 3(a)). By contrast, if  $X_i$  and  $X_{i'}$  are predicted belonging to different classes, the predicted noisy similarity should be relatively low ( $\hat{\bar{S}}_{ii'} = 0.0654$  in Figure 3(b)).

Further, let  $\hat{\bar{S}}_{ii'} = f(X_i; \theta)^T f(X_{i'}; \theta)$ , denoting the predicted noisy similarity. Substituting  $\hat{\bar{S}}_{ii'}$  into Equation 8,  $L_{mns}$  can convert into a binary cross-entropy loss version, i.e.,

$$L_{mns}(\theta) = - \sum_{i,i'} \bar{S}_{ii'} \log \hat{\bar{S}}_{ii'} + (1 - \bar{S}_{ii'}) \log(1 - \hat{\bar{S}}_{ii'}). \quad (9)$$

Let us look inside Equation 9. We could treat  $\ell(\hat{\bar{S}}_{ii'}, \bar{S}_{ii'}) = -\bar{S}_{ii'} \log \hat{\bar{S}}_{ii'} + (1 - \bar{S}_{ii'}) \log(1 - \hat{\bar{S}}_{ii'})$  as the loss function denoting the loss of using  $\hat{\bar{S}}_{ii'}$  to predict  $\bar{S}_{ii'}$ . Then,

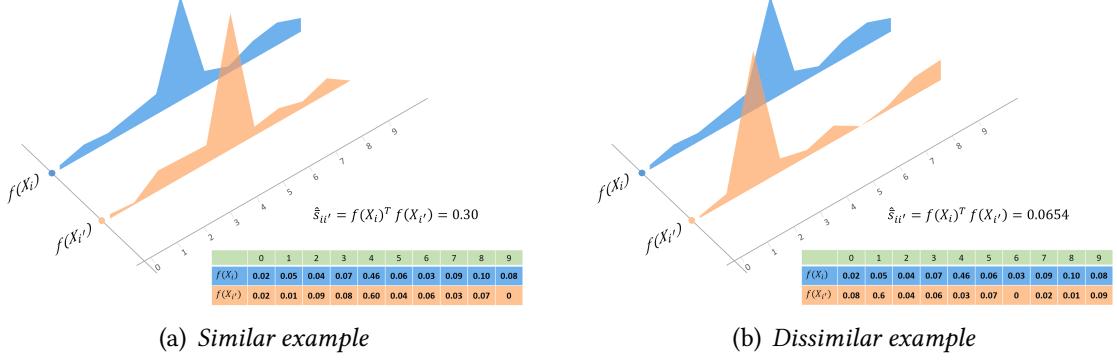


Figure 3: Examples of predicted noisy similarity. Assume class number is 10;  $f(X_i)$  and  $f(X_{i'})$  are categorical distribution of instances  $X_i$  and  $X_{i'}$  respectively, which are shown above in the form of area charts.  $\hat{S}_{ii'} = f(X_i)^T f(X_{i'}) = 0.30$  for the similar example and  $\hat{S}_{ii'} = f(X_i)^T f(X_{i'}) = 0.0654$  for the dissimilar example.

our problem can be formulated in the traditional risk minimization framework [Mohri et al., 2018]. The expected and empirical risks of employing estimator  $f$  can be defined as

$$R(f) = \mathbb{E}_{(X_i, X_{i'}, \bar{S}_{ii'}) \sim \mathcal{D}_\rho} [\ell(f(X_i), f(X_{i'}), \bar{S}_{ii'})], \quad (10)$$

and

$$R_n(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}), \quad (11)$$

where  $n$  is training sample size of the noisy-similarity-labeled data.

The whole pipeline is summarized in Figure 2. The softmax function outputs an estimator for the clean class posterior, i.e.,  $g(X) = \hat{P}(Y|X)$ , where  $\hat{P}(Y|X)$  denotes the estimated posterior. After the softmax layer, a noise transition matrix layer is added. According to Equation 7, by pre-multiplying the transpose of the transition matrix, we can obtain a predictor  $f(X) = \hat{P}(\bar{Y}|X)$  for the noisy class posterior, which can be further used to compute the prediction of the noisy similarity label, i.e.,  $\hat{S}_{ii'}$ . Therefore, by minimizing  $L_{mns}$ , as the training data goes to infinity,  $\hat{S}$  will converge to noisy similarity  $\bar{S}$  and  $f(X; \theta)$  will converge to the optimal classifier for predicting noisy class labels. Meanwhile, given the true transition matrix,  $g(X)$  will converge to the optimal classifier for predicting clean class labels.

### 3.3 Estimate noise transition matrix $T$

However, the transition matrix is unknown. We will discuss how to estimate the transition matrix for the noisy-similarity-labeled data in this subsection.

---

**Algorithm 1** MNS Learning Algorithm.

---

**Input:** noisy-similarity-labeled training data; noisy-similarity-labeled validation data.

**Stage 1: Learn  $\hat{T}$**

1: Learn  $f(X) = \hat{P}(\bar{Y}|X)$  by training the network in Figure 2 without the noise transition matrix layer;

2: Estimate  $\hat{T}$  according to Equation (12) by using instances with the highest  $\hat{P}(\bar{Y}|X)$  as anchor points;

**Stage 2: Learn the classifier  $g(X) = \hat{P}(Y|X)$**

3: Fix the transition matrix layer in Figure 2 by using the estimated transition matrix;

4: Minimize  $L_{mns}$  to learn  $g$  and stop when  $\hat{P}(\bar{Y}|X)$  corresponds to the minimum classification error on the noisy validation set;

**Output:**  $g$ .

---

*Anchor points* [Liu and Tao, 2015, Patrini et al., 2017, Yu et al., 2018] have been widely used to estimate the transition matrix for noisy-class-labeled data [Niu et al., 2018]. We illustrate that they can also be used to estimate the transition matrix for the noisy-similarity-labeled data. Specifically, an anchor point  $x$  for class  $y$  is defined as  $P(Y = y|X = x) = 1$  and  $P(Y = y'|X = x) = 0, \forall y' \in \mathcal{Y} \setminus \{y\}$ . Let  $x$  be an anchor point for class  $i$  such that  $P(Y = i|X = x) = 1$  and for  $k \neq i$ ,  $P(Y = k|X = x) = 0$ . Then we have

$$\begin{aligned} P(\bar{Y} = j|X = x) &= \sum_{k=1}^C T_{kj} P(Y = k|X = x) \\ &= T_{ij} P(Y = i|X = x) = T_{ij}. \end{aligned} \tag{12}$$

Equation 12 shows that given anchor points for each class and the noisy class posterior distribution, the transition matrix can be estimated. Note that the noisy class posterior can be estimated by  $f(x) = \hat{P}(\bar{Y}|X)$  using the pipeline in Figure 2 without the transition matrix layer. However, it is a bit strong to have access to anchor points. Instead, we assume that anchor points exist in the training data but unknown to us. Empirically, we select examples with the highest  $\hat{P}(\bar{Y} = i|X = x)$  as anchor points for the  $i$ -th class.

### 3.4 Implementation

Given the true transition matrix, we can directly build a neural network as shown in Figure 2 to learn a multi-class classifier only from the noisy-similarity-labeled data. When the true transition matrix is unknown, we estimate it with the method proposed in Section 3.3 and then we can train the whole network as normal. The proposed algorithm is summarized in Algorithm 1.

## 4 Generalization error

In this section, we will theoretically analyze the generalization ability of the proposed method. Although it looks complex, we will show that it will generalize well.

Assume that the neural network has  $d$  layers with parameter matrices  $W_1, \dots, W_d$ , and the activation functions  $\sigma_1, \dots, \sigma_{d-1}$  are Lipschitz continuous, satisfying  $\sigma_j(0) = 0$ . We denote by  $h : X \mapsto W_d\sigma_{d-1}(W_{d-1}\sigma_{d-2}(\dots\sigma_1(W_1X))) \in \mathbb{R}^C$  the standard form of the neural network. Then the output of the softmax function is defined as  $g_i(X) = \exp(h_i(X)) / \sum_{j=1}^C \exp(h_j(X))$ ,  $i = 1, \dots, C$ , and  $f(X) = T^\top g(X)$  is the output of the noise transition matrix layer. Let  $\hat{f} = \text{argmax}_{i \in \{1, \dots, C\}} \hat{f}_i$  be the classifier learned from the hypothesis space  $\mathcal{F}$  determined by the neural network, i.e.,  $\hat{f} = \text{argmin}_{f \in \mathcal{F}} R_n(f)$ . Note that the risks are defined in Section 3.2.

**Theorem 1.** *Assume the parameter matrices  $W_1, \dots, W_d$  have Frobenius norm at most  $M_1, \dots, M_d$ , and the activation functions are 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Assume the transition matrix is given, and the instances are upper bounded by  $B$ , i.e.,  $\|X\| \leq B$  for all  $X$ , and the loss function  $\ell(\hat{S}_{ii'}, \bar{S}_{ii'})$  is upper bounded by  $M^2$ . Then, for any  $\delta > 0$ , with probability at least  $1 - \delta$ ,*

$$R(\hat{f}) - R_n(\hat{f}) \leq \frac{2BC(\sqrt{2d\log 2} + 1)\prod_{i=1}^d M_i}{\sqrt{n}} + M\sqrt{\frac{\log 1/\delta}{2n}}. \quad (13)$$

A detailed proof is provided in Appendix.

Theorem 1 implies that if the training error is small and the training sample size is large, the expected risk  $R(\hat{f})$  of the learned classifier for noisy classes will be small. If the transition matrix is well estimated, the learned classifier for the clean class will also have a small risk according to Equation 7. This theoretically justifies why the proposed method works well. In the experiment section, we will show that the transition matrices will be well estimated and that the proposed method will significantly outperform the baselines.

## 5 Experiments

In this section, we empirically investigate the performance of noise transition matrix estimation and the proposed method for MNS classification on three synthetic noisy datasets and two real-world noisy datasets.

### 5.1 Experiments on synthetic noisy datasets

**Datasets.** We synthesize noisy-similarity-labeled data by employing three widely used datasets, i.e., *MNIST* [LeCun, 1998], *CIFAR-10*, and *CIFAR-100* [Krizhevsky et al., 2009].

---

<sup>2</sup>The assumption holds because deep neural networks will always regulate the objective to be a finite value and thus the corresponding loss functions are of finite values.

*MNIST* has  $28 \times 28$  grayscale images of 10 classes including 60,000 training images and 10,000 test images. *CIFAR-10* and *CIFAR-100* both have  $32 \times 32 \times 3$  color images including 50,000 training images and 10,000 test images. *CIFAR-10* has 10 classes while *CIFAR-100* has 100 classes. For all the three benchmark datasets, we leave out 10% of the training examples as a validation set, which is for model selection.

**Noisy similarity labels generation.** First, we artificially corrupt the class labels of training and validation sets according to noise transition matrices. Specifically, for each instance with clean label  $i$ , we replace its label by  $j$  with a probability of  $T_{ij}$ . After that, we assign data pairs  $((X_i, \bar{Y}_i), (X_{i'}, \bar{Y}_{i'}))$  noisy similarity labels  $\bar{S}_{ii'}$  and remove  $\bar{Y}_i$  and  $\bar{Y}_{i'}$ . In this paper, we consider the symmetric noisy setting defined in Appendix. Noise-0.5 generates severe noise which means almost half labels are corrupted while Noise-0.2 generates slight noise which means around 20% labels are corrupted.

**Baselines.** We compare our proposed method with state-of-the-art methods and conduct all the experiments with default parameters by PyTorch on NVIDIA Tesla V100. Specifically, we compare with the following two algorithms:

- Meta Classification Likelihood (MCL) [Hsu et al., 2019], which is the state-of-the-art method for multi-classification from clean-similarity-labeled data.
- KLD-based Contrastive Loss (KCL) [Hsu and Kira, 2016], which is a strong baseline. It uses Kullback–Leibler divergence to measure the distance between two distributions.

**Network structure.** For *MNIST*, we use LeNet. For *CIFAR-10*, we use pre-trained ResNet-32. For *CIFAR-100*, we use VGG8. For all networks, as shown in Figure 2, the output number of the last fully connected layer is set to be the number of classes. We add a noise transition matrix layer after the softmax. Since the loss functions of MNS, MCL and KCL are designed for instance pairs, a pairwise enumeration layer [Hsu et al., 2018] is adapted before calculating the loss.

**Optimizer.** We follow the optimization method in [Patrini et al., 2017] to learn the noise transition matrix  $\hat{T}$ . To learn  $g$ , we use the Adam optimizer with initial learning rate 0.001. On *MNIST*, the batch size is 128 and the learning rate decays every 10 epochs by a factor of 0.1 with 30 epochs in total. On *CIFAR-10*, the batch size is also 128 and the learning rate decays every 40 epochs by a factor of 0.1 with 120 epochs in total. On *CIFAR-100*, the batch size is 1000 and the learning rate drops at epoch 80 and 160 by a factor of 0.1 with 200 epochs in total.

**Results.** The results in Tables 1, 2, and 3 demonstrate the test accuracy and stability of four algorithms on three benchmark datasets. Overall, we can see that when similarity labels are corrupted,  $\text{MNS}(\hat{T})$  achieves the best performance among three similarity-based learning methods, approaching or even exceeding  $\text{MNS}(T)$  which is given the true noise transition matrix. Specifically, On *MNIST* and *CIFAR10*, when the noise rates are high,  $\text{MNS}(\hat{T})$  performs better than  $\text{MNS}(T)$ . This should be because that  $\hat{T}$  and the networks are learned jointly as shown in Algorithm 1.

Table 1: Average Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on *MNIST*. KCL, MCL and MNS only have access to noisy similarity labels. Specifically, MCL( $\hat{T}$ ) denotes the method in which we estimate noise transition matrix first and then use the estimated  $\hat{T}$  for training while MCL( $T$ ) skips the first step and directly use the true noise transition matrix.

Noise	0.2	0.3	0.4	0.5	0.6
KCL	<b>99.20±0.02</b>	<b>99.06±0.05</b>	95.97±3.65	90.61±0.78	85.20±4.69
MCL	98.51±0.10	98.28±0.06	97.92±0.24	97.54±0.09	96.94±0.20
MNS( $\hat{T}$ )	98.56±0.07	98.29±0.16	<b>98.01±0.15</b>	<b>97.61±0.41</b>	<b>97.26±0.23</b>
MNS( $T$ )	98.75±0.07	98.69±0.11	98.32±0.09	98.18±0.13	94.48±4.49

Table 2: Average Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on *CIFAR10*.

Noise	0.2	0.3	0.4	0.5	0.6
KCL	19.14±1.27	17.67±2.15	18.58±1.28	17.96±3.41	15.14±1.67
MCL	75.58±3.64	68.90±0.32	63.38±1.32	61.67±0.98	44.55±2.96
MNS( $\hat{T}$ )	<b>78.83±1.81</b>	<b>76.80±1.33</b>	<b>70.35±1.21</b>	<b>68.87±0.97</b>	<b>50.99±2.88</b>
MNS( $T$ )	82.42±0.37	77.42±0.46	70.71±0.33	69.28±0.41	40.24±0.61

Table 3: Average Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on *CIFAR100*.

Noise	0.2	0.3	0.4	0.5	0.6
KCL	13.32±1.57	7.982±0.57	5.406±0.15	3.738±0.45	3.208±0.55
MCL	48.38±0.38	40.48±0.79	32.75±0.77	26.48±0.36	21.94±0.19
MNS( $\hat{T}$ )	<b>48.78±0.74</b>	<b>43.90±0.39</b>	<b>40.26±0.93</b>	<b>35.14±0.69</b>	<b>31.40±0.26</b>
MNS( $T$ )	51.95±0.44	48.97±0.25	46.45±1.00	42.01±0.78	36.50±0.45

On *MNIST*, when the noise rate is relatively low (under 0.4), KCL has the highest accuracy; MCL and MNS also perform well. Intuitively, compared with inner product, Kullback-Leibler divergence measures the similarity between two distributions better, but it may introduce bad local minima or small gradients for learning [Hsu et al., 2019] such that it has poor performances on more complex datasets or higher noise rate. For exam-

ple, when the noise rate increases (beyond 0.3), the accuracy of KCL drops dramatically, falling from 99.06 at Noise-0.3 to 85.20 at Noise-0.6. By contrast, MNS and MCL are more robust to noise. Both methods decrease slightly as the noise rate rises while our method is always a little better than the state-of-the-art method MCL.

On *CIFAR-10* and *CIFAR-100*, there is a significant decrease in the accuracy of all methods and our method achieves the best results across all noise rate, i.e., at Noise-0.6, MNS gives an accuracy uplift of about 6.5% and 10% on *CIFAR-10* and *CIFAR-100* respectively compared with the state-of-the-art method MCL.

## 5.2 Experiments on real-world noisy datasets

**Datasets.** We verify the effectiveness of the proposed method on two real-word datasets with noisy supervision, i.e., *Clothing1M* [Xiao et al., 2015] and *Food-101* [Bossard et al., 2014]. Specifically, *Clothing1M* has 1M images with real-world noisy labels and additional 50k, 14k, 10k images with clean labels for training, validation and testing. We only use noisy training set in training phase and leave out 10% as validation set for model selection and test our model on 10k testing set. *Food-101* consists of 101 food categories, with 101,000 images. For each class, 250 manually reviewed clean test images are provided as well as 750 training images with real-world noise. For *Food-101*, we also leave out 10% for validation. In particular, we use *Random Crop* and *Random Horizontal Flip* for data augmentation. Since datasets contain some amount of class label noise already, we do not need to corrupt the labels artificially. We generate noisy-similarity-labeled data by using the noisy-class-labeled data directly.

**Baselines.** The same as the synthetic experiment part.

**Network structure and optimizer.** For all experiments, we use pre-trained ResNet-50. On *Clothing1M*, the batch size is 256 and the learning rate drops every 5 epochs by a factor of 0.1 with 10 epochs in total. On *Food-101*, the batch size is 1000 and the learning rate drops at epoch 80 and 160 by a factor of 0.1 with 200 epochs in total. Other settings are the same as the synthetic experiment part.

Table 4: Classification Accuracy on real-world noisy dataset *Clothing1M*.

KCL	MCL	MNS( $\hat{T}$ )
9.49	66.20	<b>67.50</b>

Table 5: Classification Accuracy on real-world noisy dataset *Food-101*.

KCL	MCL	MNS( $\hat{T}$ )
30.17	48.08	<b>71.18</b>

**Results.** From Table 4 and 5, We can see that on *Clothing1M*, MNS( $\hat{T}$ ) achieves the best accuracy. On *Food-101*, MNS( $\hat{T}$ ) also performs distinguishedly, uplifting about 23% in accuracy compared with MCL. Specifically, the gap between MCL and MNS( $\hat{T}$ ) is huge in Table 5 while is not in Table 4. Let us review the definition of similarity-labeled data: if two instances belong to the same class, they will have similarity label  $S = 1$ , otherwise  $S = 0$ . That is to say, for a  $k$ -class dataset, only around  $\frac{1}{k}$  of similarity-labeled data has similarity labels  $S = 1$ , and the rest  $1 - \frac{1}{k}$  has similarity labels  $S = 0$ . For *Clothing1M* (Table 4), the  $k = 14$ . For *Food-101* (Table 5), the  $k = 101$ . Therefore, the generated similarity-labeled data from *Food-101* is much more unbalanced than that from *Clothing1M*. As a result, the baseline performed badly on *Food-101*, making the gap huge in Table 5.

### 5.3 Noise transition matrix estimation

To estimate  $T$ , we first learn the noisy predictor  $f(X) = \hat{P}(\bar{Y}|X)$ . For each dataset, the network and optimizer remain the same as above but the noise transition matrix layer is exclude.  $T$  is then estimated using the method proposed in Section 3.3.

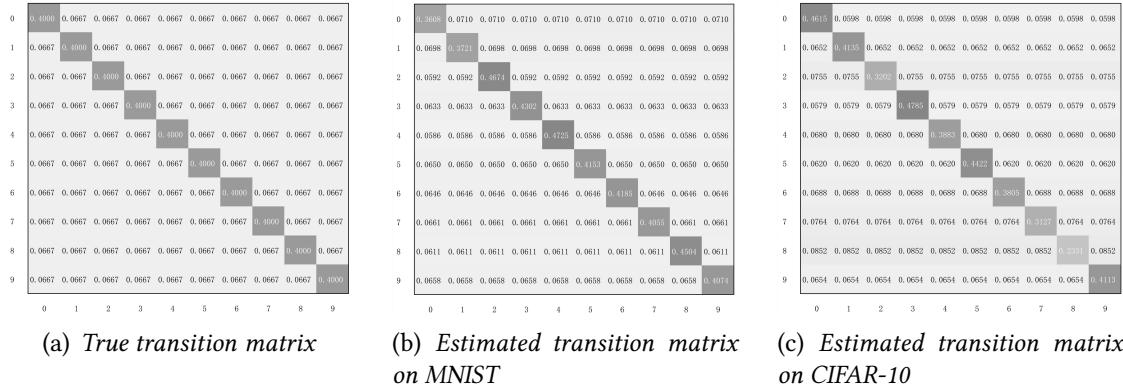


Figure 4: True transition matrix  $T$  at Noise-0.6 and corresponding  $\hat{T}$  of two datasets with 10 classes: *MNIST* and *CIFAR-10*.

Here we only show the estimated transition matrices of three synthetic noisy datasets because we have the exact values of the true transition matrices such that we could assess the estimation accuracies. Estimated transition matrices of real-world noisy datasets are provided in Appendix. From Figure 4 and 5, we can see that transition matrices estimated with the proposed method are very close to the true one. By employing the calculation method of estimation error as  $\epsilon = ||T - \hat{T}||_1 / ||T||_1$ , *MNIST*, *CIFAR-10* and *CIFAR-100* achieve 0.0668, 0.1144 and 0.1055 in error respectively.

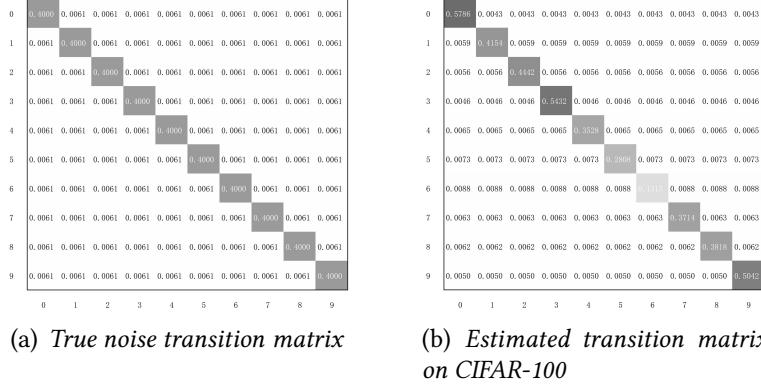


Figure 5: True transition matrix  $T$  at Noise-0.6 and corresponding  $\hat{T}$  of *CIFAR-100*. Note that we only show the first 10 rows and columns of the matrix.

## 6 Conclusion

This paper proposes a noisy-similarity-based multi-class classification algorithm (called MNS classification) by designing a novel deep learning system exploiting only noisy-similarity-labeled data. MNS classification provides an effective way for making predictions on sensitive matters where it is difficult to collect high-quality data such that similarities with noise could be all the information available. The core idea is to model the noise in the latent noisy class labels by using a noise transition matrix while only noisy similarity labels are observed. By adding a noise transition matrix layer in the deep neural network, it turns to robust to similarity label noise. We also present that noise transition matrix can be estimated in this setting. Experiments are conducted on benchmark-simulated and real-world label-noise datasets, demonstrating our method can excellently solve the above weakly supervised problem. In future work, investigating different types of noise for diverse real-life scenarios might prove important.

## References

- David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11075–11083, 2019.
- Han Bao, Gang Niu, and Masashi Sugiyama. Classification from pairwise similarity and unlabeled data. In *International Conference on Machine Learning*, pages 461–470, 2018.
- Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.
- Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.
- Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. Learning with bounded instance-and label-dependent label noise. *arXiv preprint arXiv:1709.03768*, 2017.
- Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.
- Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.
- Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5138–5147, 2019.
- Yen-Chang Hsu and Zsolt Kira. Neural network-based clustering using pairwise constraints. In *ICLR workshop*, 2016. URL <https://arxiv.org/abs/1511.06321>.
- Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *International Conference on Learning Representations (ICLR)*, 2018. URL <https://openreview.net/forum?id=ByRWCqvT->.
- Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. *arXiv preprint arXiv:1901.00544*, 2019.
- Mengying Hu, Hu Han, Shiguang Shan, and Xilin Chen. Weakly supervised image classification through noise regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11517–11525, 2019a.

- Wei Hu, Yangyu Huang, Fan Zhang, and Ruirui Li. Noise-tolerant paradigm for training face recognition cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11887–11896, 2019b.
- Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. Label-noise robust generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2476, 2019.
- Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*, 2017.
- Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *European Conference on Computer Vision*, pages 301–320. Springer, 2016.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.
- Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.
- Chenglong Li, Chengli Zhu, Yan Huang, Jin Tang, and Liang Wang. Cross-modal ranking with soft consistency and noisy labels for robust rgb-t tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 808–823, 2018.
- Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019.
- Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1910–1918, 2017.
- Zhenguo Li and Jianzhuang Liu. Constrained clustering by spectral kernel learning. In *2009 IEEE 12th International Conference on Computer Vision*, pages 421–427. IEEE, 2009.
- Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

Gang Niu, Bo Dai, Makoto Yamada, and Masashi Sugiyama. Information-theoretic semi-supervised metric learning via entropy regularization. *Neural computation*, 26(8):1717–1762, 2014.

Li Niu, Qingtao Tang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Learning from noisy web data with category-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7689–7698, 2018.

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.

Takuya Shimada, Han Bao, Issei Sato, and Masashi Sugiyama. Classification from pairwise similarities/dissimilarities and unlabeled data via empirical risk minimization. *arXiv preprint arXiv:1904.11717*, 2019.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. *arXiv preprint arXiv:1902.03680*, 2019.

Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8688–8696, 2018.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 322–330, 2019.

Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *arXiv preprint arXiv:1906.00189*, 2019.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003.

Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. *arXiv preprint arXiv:1903.07788*, 2019.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–83, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9029–9038, 2018.

Weihe Zhang, Yali Wang, and Yu Qiao. Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2019.

Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1237–1246, 2019a.

Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7812–7821, 2019b.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.

# Appendices

## A Proof of Theorem 1

We have defined

$$R(f) = \mathbb{E}_{(X_i, X_{i'}, \bar{S}_{ii'}) \sim \mathcal{D}_\rho} [\ell(f(X_i), f(X_{i'}), \bar{S}_{ii'})], \quad (14)$$

and

$$R_n(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}), \quad (15)$$

where  $n$  is training sample size of the noisy-similarity-labeled data.

First we bound the generalization error with Rademacher complexity [Bartlett and Mendelson, 2002].

**Theorem 2** ([Bartlett and Mendelson, 2002]). *Let the loss function be upper bounded by  $M$ . Then, for any  $\delta > 0$ , with the probability  $1 - \delta$ , we have*

$$\sup_{f \in \mathcal{F}} |R(f) - R_n(f)| \leq 2\mathfrak{R}_n(\ell \circ \mathcal{F}) + M \sqrt{\frac{\log 1/\delta}{2n}}, \quad (16)$$

where  $\mathfrak{R}_n(\ell \circ \mathcal{F})$  is the Rademacher complexity defined by

$$\mathfrak{R}_n(\ell \circ \mathcal{F}) = \mathbb{E} \left[ \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}) \right], \quad (17)$$

and  $\{\sigma_1, \dots, \sigma_n\}$  are Rademacher variables uniformly distributed from  $\{-1, 1\}$ .

Before further upper bound the Rademacher complexity  $\mathfrak{R}_n(\ell \circ \mathcal{F})$ , we discuss the special loss function and its *Lipschitz continuity* w.r.t  $h_j(X_i)$ ,  $j = \{1, \dots, C\}$ .

**Lemma 1.** *Given transition matrix  $T$ , loss function  $\ell(f(X_i), f(X_{i'}), \bar{S}_{ii'})$  is 1-Lipschitz with respect to  $h_j(X_i)$ ,  $j = \{1, \dots, C\}$ ,*

$$\left| \frac{\partial \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'})}{\partial h_j(X_i)} \right| < 1. \quad (18)$$

Detailed proof of Lemma 1 can be found in Section A.1.

Based on Lemma 1, we can further upper bound the Rademacher complexity  $\mathfrak{R}_n(\ell \circ \mathcal{F})$  by the following lemma.

**Lemma 2.** Given transition matrix  $T$  and assume loss function  $\ell(f(X_i), f(X_{i'}), \bar{S}_{ii'})$  is 1-Lipschitz with respect to  $h_j(X_i)$ ,  $j = \{1, \dots, C\}$ , we have

$$\begin{aligned}\mathfrak{R}_n(\ell \circ \mathcal{F}) &= \mathbb{E} \left[ \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}) \right] \\ &\leq C \mathbb{E} \left[ \sup_{h \in H} \frac{1}{n} \sum_{i=1}^n \sigma_i h(X_i) \right],\end{aligned}\tag{19}$$

where  $H$  is the function class induced by the deep neural network.

Detailed proof of Lemma 2 can be found in Section A.2.

The right hand part of the above inequality, indicating the hypothesis complexity of deep neural networks, can be bound by the following theorem.

**Theorem 3** ([Golowich et al., 2017]). Assume the Frobenius norm of the weight matrices  $W_1, \dots, W_d$  are at most  $M_1, \dots, M_d$ . Let the activation functions be 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Let  $X$  is upper bounded by  $B$ , i.e., for any  $X$ ,  $\|X\| \leq B$ . Then,

$$\mathbb{E} \left[ \sup_{h \in H} \frac{1}{n} \sum_{i=1}^n \sigma_i h(X_i) \right] \leq \frac{B(\sqrt{2d \log 2} + 1) \prod_{i=1}^d M_i}{\sqrt{n}}.\tag{20}$$

Combining Lemma 1,2, and Theorem 2, 3, Theorem 1 is proven.

## A.1 Proof of Lemma 1

Recall that

$$\begin{aligned}\ell(f(X_i), f(X_{i'}), \bar{S}_{ii'} = 1) &= -\log(f(X_i)^T f(X_{i'})). \\ &= -\log((T^T g(X_i))^T (T^T g(X_{i'}))),\end{aligned}\tag{21}$$

where

$$\begin{aligned}g(X_i) &= [g_1(X_i), \dots, g_c(X_i)] \\ &= \left[ \left( \frac{\exp(h_1(X))}{\sum_{j=1}^c \exp(h_j(X))} \right), \dots, \left( \frac{\exp(h_c(X))}{\sum_{i=j}^c \exp(h_j(X))} \right) \right]^T.\end{aligned}\tag{22}$$

Take the derivative of  $\ell(f(X_i), f(X_{i'}), \bar{S}_{ii'} = 1)$  w.r.t.  $h_j(X_i)$ , we have

$$\frac{\partial \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'} = 1)}{\partial h_j(X_i)} = \frac{\partial \ell(f(X_i), f(X_{i'}), 1)}{\partial f(X_{i'})^T f(X_i)} \frac{\partial f(X_{i'})^T f(X_i)}{\partial f(X_i)} \frac{\partial f(X_i)}{\partial g(X_i)} \frac{\partial g(X_i)}{\partial h_j(X_i)},\tag{23}$$

where

$$\begin{aligned}\frac{\partial \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'} = 1)}{\partial f(X_{i'})^T f(X_i)} &= -\frac{1}{f(X_{i'})^T f(X_i)} \\ \frac{\partial f(X_{i'})^T f(X_i)}{\partial f(X_i)} &= f(X_{i'})^T \\ \frac{\partial f(X_i)}{\partial g(X_i)} &= T^T \\ \frac{\partial g(X_i)}{\partial h_j(X_i)} &= g'(X_i) = [g'_1(X_i), \dots, g'_c(X_i)].\end{aligned}$$

Note that the derivative of the softmax function has some properties, i.e., if  $m \neq j$ ,  $g'_m(X_i) = -g_m(X_i)g_j(X_i)$  and if  $m = j$ ,  $g'_j(X_i) = (1 - g_j(X_i))g_j(X_i)$ .

We denote by  $\text{Vector}[m]$  the  $m - \text{th}$  element in  $\text{Vector}$  for those complex vectors. Because  $0 < g_m(X_i) < 1, \forall m \in \{1, \dots, c\}$  and  $T_{ij} > 0, \forall i, j \in \{1, \dots, c\}$ , we have

$$g'_m(X_i) \leq |g'_m(X_i)| < g_m(X_i), \quad \forall m \in \{1, \dots, c\}; \quad (24)$$

$$T^T g'(X_i)[m] < T^T |g'(X_i)|[m] < T^T g(X_i)[m], \quad \forall m \in \{1, \dots, c\}. \quad (25)$$

Since  $0 < f_m(X_{i'})^T < 1, \forall m \in \{1, \dots, c\}$ , similarly we have

$$f(X_{i'})^T T^T |g'(X_i)| < f(X_{i'})^T T^T g(X_i) = f(X_{i'})^T f(X_i). \quad (26)$$

Therefore,

$$\begin{aligned}& \left| \frac{\partial \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'} = 1)}{\partial h_j(X_i)} \right| \\ &= \left| \frac{\partial \ell(f(X_i), f(X_{i'}), 1)}{\partial f(X_{i'})^T f(X_i)} \frac{\partial f(X_{i'})^T f(X_i)}{\partial f(X_i)} \frac{\partial f(X_i)}{\partial g(X_i)} \frac{\partial g(X_i)}{\partial h_j(X_i)} \right| \\ &= \left| -\frac{f(X_{i'})^T T^T g'(X_i)}{f(X_{i'})^T f(X_i)} \right| \\ &\leq \left| \frac{f(X_{i'})^T T^T |g'(X_i)|}{f(X_{i'})^T f(X_i)} \right| \\ &< \left| \frac{f(X_{i'})^T f(X_i)}{f(X_{i'})^T f(X_i)} \right| = 1.\end{aligned} \quad (27)$$

Similarly, we can proof

$$\left| \frac{\partial \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'} = 0)}{\partial h_j(X_i)} \right| < 1. \quad (28)$$

Combining Eq.27 and Eq.28, we obtain

$$\left| \frac{\partial \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'})}{\partial h_j(X_i)} \right| < 1. \quad (29)$$

## A.2 Proof of Lemma 2

$$\begin{aligned}
& \mathbb{E} \left[ \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}) \right] \\
&= \mathbb{E} \left[ \sup_g \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}) \right] \\
&= \mathbb{E} \left[ \sup_{\arg\max\{h_1, \dots, h_C\}} \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}) \right] \\
&= \mathbb{E} \left[ \sup_{\max\{h_1, \dots, h_C\}} \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}) \right] \\
&\leq \mathbb{E} \left[ \sum_{k=1}^C \sup_{h_k \in H} \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}) \right] \\
&= \sum_{k=1}^C \mathbb{E} \left[ \sup_{h_k \in H} \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(X_i), f(X_{i'}), \bar{S}_{ii'}) \right] \\
&\leq C \mathbb{E} \left[ \sup_{h_k \in H} \frac{1}{n} \sum_{i=1}^n \sigma_i h_k(X_i) \right] \\
&= C \mathbb{E} \left[ \sup_{h \in H} \frac{1}{n} \sum_{i=1}^n \sigma_i h(X_i) \right],
\end{aligned}$$

where the first three equations hold because given  $T, f, g$  and  $\max\{h_1, \dots, h_C\}$  give the same constraint on  $h_j(X_i), j = \{1, \dots, C\}$ ; the sixth inequality holds because of the Lemma [Ledoux and Talagrand, 2013].

## B Definition of transition matrix

Symmetric noisy setting is defined as follows, where  $C$  is the number of classes.

$$\text{Noise-}\rho: \quad T = \begin{bmatrix} 1 - \rho & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & \frac{\rho}{C-1} \\ \frac{\rho}{C-1} & 1 - \rho & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} \\ \vdots & & \ddots & & \vdots \\ \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & 1 - \rho & \frac{\rho}{C-1} \\ \frac{\rho}{C-1} & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & 1 - \rho \end{bmatrix}. \quad (30)$$

## C Estimation of transition matrix on real-world noisy datasets

Here we show the estimated transition matrices of *Clothing1M* and the first ten classes of *Food-101*. For *Clothing1M*, we use additional 50k images with clean labels to learn the transition matrix such that the left  $\hat{T}$  in Figure 1 is very close to the true one. The right  $\hat{T}$  in Figure 1 was estimated only from noisy-similarity-labeled data, which learned most of the features of true transition matrix. For *Food-101*, both  $\hat{T}$  were estimated from noisy-labeled data. From Figure 2 we can see that the result close to the result which verifies the effectiveness of our method.

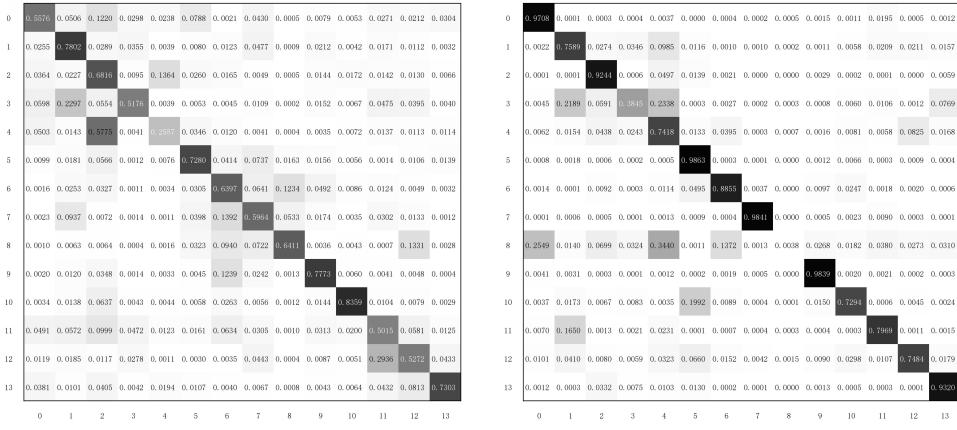


Figure 6:  $\hat{T}$  of *Clothing1M*; the one in the left hand is  $\hat{T}$  estimated from class labels, the one in the right hand is  $\hat{T}$  estimated from noisy similarity labels.

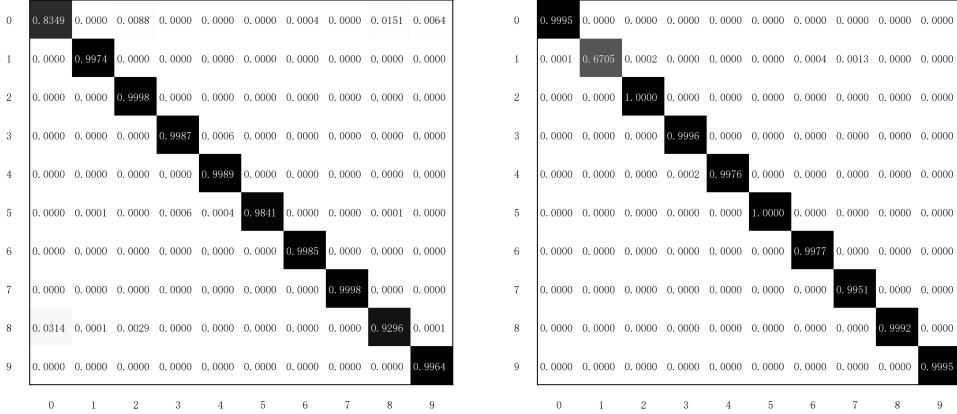


Figure 7:  $\hat{T}$  of the first ten classes of *Food-101*; the one in the left hand is  $\hat{T}$  estimated from class labels, the one in the right hand is  $\hat{T}$  estimated from noisy similarity labels