Tackling Instance-Dependent Label Noise via a Universal Probabilistic Model

Qizhou Wang[1,2], **Bo Han**[2,*], **Tongliang Liu**[3], **Gang Niu**[4], **Jian Yang**[1], **Chen Gong**[1,5,*]

[1] Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of MoE, School of Computer Science and Engineering, Nanjing University of Science and Technology
[2] Department of Computer Science, Hong Kong Baptist University
[3] Trustworthy Machine Learning Lab, School of Computer Science, Faculty of Engineering, The University of Sydney
[4] RIKEN Center for Advanced Intelligence Project (AIP)
[5] Department of Computing, Hong Kong Polytechnic University

Abstract

The drastic increase of data quantity often brings the severe decrease of data quality, such as incorrect label annotations, which poses a great challenge for robustly training Deep Neural Networks (DNNs). Existing learning methods with label noise either employ ad-hoc heuristics or restrict to specific noise assumptions. However, more general situations, such as instance-dependent label noise, have not been fully explored, as scarce studies focus on their label corruption process. By categorizing instances into confusing and unconfusing instances, this paper proposes a simple yet universal probabilistic model, which explicitly relates noisy labels to their instances. The resultant model can be realized by DNNs, where the training procedure is accomplished by employing an alternating optimization algorithm. Experiments on datasets with both synthetic and real-world label noise verify that the proposed method yields significant improvements on robustness over state-of-the-art counterparts.

Introduction

DNNs have gained much popularity in the literature of machine learning (He et al. 2017; Jaderberg et al. 2015). However, one of the prominent factors for their success is the availability of large-scale training sample with clean annotations (Deng et al. 2009), where the acquisition process might be prohibitively expensive in practice. Hence, various low-cost surrogate strategies are provided to automatically collect labels (Li et al. 2017a; Xiao et al. 2015). These approaches make the acquisition of large-scale annotated data possible, but will also inevitably lead to noisy labels due to the imperfect results of search engines and crawling algorithms. Previous studies have suggested that noisy labels will hurt the test accuracy of DNNs (Arpit et al. 2017; Zhang et al. 2017). Thus, it would be desirable to develop robust methods for training DNNs with label noise.

In learning with noisy labels, existing robust learning methods consist of two general categories, according to whether the generation process for noisy labels is specified. The first category usually employs heuristic and ad-hoc rules without explicitly modeling the process of label corruption. For example, data cleansing techniques remove potential mislabeled instances before training (Angelova, Abu-Mostafam, and Perona 2005; Sun et al. 2007); risk reweighting strategies demote the adverse impact of un-trustworthy labels (Guo et al. 2018; Han et al. 2018b; Jiang et al. 2017; Lee et al. 2018; Wang et al. 2018; Han et al. 2020); label correction methods revise noisy labels based on model prediction (Reed et al. 2015; Tanaka et al. 2018; Yi and Wu 2019) or prior information (Gao et al. 2017; Li et al. 2017b; Veit et al. 2017). These methods have gained extensive popularity in the literature. However, without specifying the generation of label noise, these methods may lead to inferior or biased results (Xia et al. 2019; Berthon et al. 2020).

The above issue motivates us to explore the second category, which makes various assumptions on the process of label-noise generation. For example, the *random classification noise* (RCN) model assumes that labels are randomly corrupted without any relationship with their real classes and instance features. To handle this simple situation, various noise-tolerant loss functions (Ghosh, Kumar, and Sastry 2017; Manwani and Sastry 2013; Van Rooyen, Menon, and Williamson 2015) have been explored in the literature. Subsequently, the *class-conditional noise* (CCN) model is proposed to relate noisy labels to true labels, *i.e.*, some pairs of classes are more prone to be mislabeled. To depict such phenomenon, the noise transition matrix is introduced, which represents the probability of true labels flipping into noisy ones (Patrini et al. 2017; Chen et al. 2020). This matrix plays a significant role in attacking CCN, and its elements can be either tuned by cross validation (Natarajan et al. 2013) or estimated by various algorithms (Liu and Tao 2016; Goldberger and Ben-Reuven 2017; Han et al. 2018a; Xia et al. 2019; Yao et al. 2020).

However, (Xiao et al. 2015) suggest that mis-annotation is highly related to instance features in reality. In other words, some instances may be confusing subjectively, and thus suffer from mislabeling. This brings us the *instance-dependent noise* (IDN) model recently. To tackle this challenging problem, (Du and Cai 2015) consider the predetermined noise function that assumes mislabeled instances are close to the decision boundary; (Menon, Van Rooyen, and

*Corresponding authors. Emails: bhanml@comp.hkbu.edu.hk, chen.gong@njust.edu.cn.

(c) WOLF

Figure 1: (a) and (b) should be both labeled as DOG. However, one may confuse about the label of (b), whose pattern resembles WOLF in (c).

Natarajan 2016) propose an Isotron algorithm to decrease the difference between the clean and the corrupted distributions; (Cheng et al. 2020) cleanse data based on classification results on contaminated labels; (Berthon et al. 2020) assume the probability that the assigned label is correct has been given; and (Xia et al. 2020) explore the instance-dependent transition matrix that is weighted combination of parts-dependent matrices. Although these studies push the research field forward, their results are restricted to binary classification problems under various strong assumptions or rely on side information that may be unrealistic in practice, critically limiting their applications in real-world situations. Researchers also explore the underlying probabilistic process of label-noise generation, while the resultant probabilistic models rely on prior information (Xiao et al. 2015), or count on specific model architecture (Goldberger and Ben-Reuven 2017). Therefore, how to model the generation of IDN in a universal manner remains an important problem.

In this work, we focus on explicitly modeling the noisy label generation under the IDN assumption. Inspired by the fact that mislabeled instances often have particular patterns (Xiao et al. 2015), we assume that only those *confusing* instances are suffering from mislabeling. For example, in Fig. 1, (a) and (b) should be both labeled as DOG. However, one may confuse about (b), whose pattern resembles WOLF in (c). Thus, (b) might be mislabeled, while (a) is easy to be correctly labeled. Accordingly, in distinguishing confusing and unconfusing instances, we propose a probabilistic IDN model to depict the label noise behavior (*cf.*, Fig. 2). The model is realized by DNNs, and a novel alternating optimization algorithm is adopted to iteratively estimate true labels and update learnable parameters. Meanwhile, we investigate the learning behavior of our method that can find potentially confusing instances and correct their labels based on model prediction. We conduct experiments on *CIFAR-100* and *CIFAR-10* with synthetic label noise, and the empirical results demonstrate the state-of-the-art performance of our method. More importantly, we conduct real-world experiments on *Clothing1M* (Xiao et al. 2015), where the noisy labels are critically instance-dependent. The empirical results indicate that our method also achieves the superior test accuracy over baselines in the real-world setting.

Tackling Instance-Dependent Label Noise

In this paper, the scalar is in lowercase letter (*e.g.*, a), and the vector is in lowercase letter with boldface (*e.g.*, $\mathbf{a}$) whose element can be accessed by superscript (*e.g.*, $\mathbf{a}^j$).

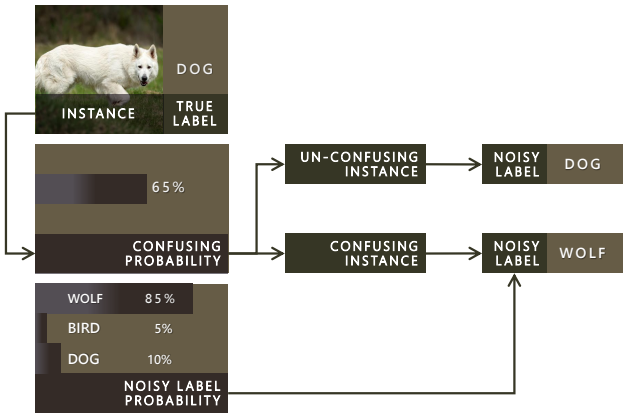


Figure 2: An example of the generation of the noisy label. Due to the label confusing probability, the instance has 65% to be a confusing instance. If it is confusing, the noisy label is generated by the noisy label probability itself, and is "WOLF" in this example. Conversely, if the instance is unconfusing, the noisy label will absolutely equal to the true label, *i.e.*, "DOG".

Under the c-class problem setting, we define the label space $\mathcal{Y} = \left\{\mathbf{y} : \mathbf{y} \in \{0,1\}^c, \mathbf{y}^\top \mathbf{y} = 1\right\}$ and the label distribution space $\mathcal{Q} = \left\{\mathbf{q} : \mathbf{q} \in [0,1]^c, \mathbf{1}^\top \mathbf{q} = 1\right\}$. Specifically, for a label $\mathbf{y} \in \mathcal{Y}$ and the label distribution $\mathbf{q} \in \mathcal{Q}$, $\mathbf{y}^j = 1$ indicates the instance is labeled to the j-th class, and $\mathbf{q}^j$ denotes the probability of $\mathbf{y}^j = 1$. Note that each label $\mathbf{y}$ has only one non-zero value at the coordinate of the underlying class $j \in [c]$, where $[c] = \{1, \ldots, c\}$ is the set that contains all the categories.

Label Noise Setting

In supervised classification, we consider the training sample $S = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ drawn i.i.d. from some unknown distribution with the associated labels $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$, where $\mathbf{y}_i \in \mathcal{Y}$ is the one-hot label for the instance $\mathbf{x}_i$. However, the true labels Y are inaccessible in the setting of label noise. Instead, we observe noisy labels $\widetilde{Y} = (\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_N)$, where $\widetilde{\mathbf{y}}_i \in \mathcal{Y}$ might be different from the corresponding true label $\mathbf{y}_i$. The task is to use the noisy-labeled sample $(S, \widetilde{Y})$ to select a softmax DNN classifier $\mathbf{h}_\mathbf{w}$ parameterized by $\mathbf{w}$, so that $\mathbf{h}_\mathbf{w}^j(\mathbf{x})$ properly estimates the true label probability $P(\mathbf{y}^j = 1|\mathbf{x})$ for each new instance $\mathbf{x}$.

In order to tackle instance-dependent label noise, we begin by specifying our label noise setting. Concretely, instances are categorized into *confusing* and *unconfusing* instances, and we assume that only confusing instances suffer from mislabeling. Mathematically, an instance is named as a confusing instance if $P(\widetilde{\mathbf{y}}_i = \mathbf{y}_i|\mathbf{y}_i, \mathbf{x}) < 1$, and as an unconfusing instance if $P(\widetilde{\mathbf{y}}_i = \mathbf{y}_i|\mathbf{y}_i, \mathbf{x}) = 1$. To facilitate the derivation of our probabilistic model, we introduce a binary variable $s_i \in \{0,1\}$ for each instance $\mathbf{x}_i$, such that $s_i = 1$ indicates the instance $\mathbf{x}_i$ is confusing, and $s_i = 0$ otherwise.

As aforementioned, given an unconfusing instance $\mathbf{x}_i$, the

noisy label $\widetilde{\mathbf{y}}_i$ equals to the true label $\mathbf{y}_i$ definitely, *i.e.*,

$$P(\widetilde{\mathbf{y}}_i|\mathbf{y}_i, \mathbf{x}_i, s_i = 0) = \mathbb{I}\{\mathbf{y}_i = \widetilde{\mathbf{y}}_i\}, \tag{1}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function that equals to 1 if the condition is true, and 0 otherwise. On the contrary, the noisy label $\widetilde{\mathbf{y}}_i$ might be incorrect if the instance is confusing. In this case, we assume that the noisy label is independent on the true label, namely,

$$P(\widetilde{\mathbf{y}}_i|\mathbf{y}_i, \mathbf{x}_i, s_i = 1) = P(\widetilde{\mathbf{y}}_i|\mathbf{x}_i). \tag{2}$$

It means that annotators fail to determine the true label, and thus a confusing instance cannot be properly classified according to annotators' knowledge. Fig. 2 summarizes the key concepts of our IDN setting. Accordingly, the posterior of a noisy label $\widetilde{\mathbf{y}}_i$ can be summarized by:

$$\begin{aligned} P(\widetilde{\mathbf{y}}_i|\mathbf{y}_i, \mathbf{x}_i) &\overset{1}{=} \sum_{s\in\{0,1\}} P(\widetilde{\mathbf{y}}_i, s_i = s|\mathbf{y}_i, \mathbf{x}_i) \\ &\overset{2}{=} \sum_{s\in\{0,1\}} P(\widetilde{\mathbf{y}}_i|\mathbf{y}_i, \mathbf{x}_i, s_i = s)P(s_i = s|\mathbf{x}_i) \\ &\overset{3}{=} (1-\eta_i)\mathbb{I}\{\mathbf{y}_i = \widetilde{\mathbf{y}}_i\} + \eta_i\psi_i, \end{aligned} \tag{3}$$

where we define $\psi_i = P(\widetilde{\mathbf{y}}_i|\mathbf{x}_i)$ and term $\eta_i = P(s = 1|\mathbf{x}_i)$ as the *confusing probability*. Note that, the 2nd equation is based on the assumption $P(s_i = s|\mathbf{y}_i, \mathbf{x}_i) = P(s_i = s|\mathbf{x}_i)$, and the 3rd equation is derived by Eq. (1) and Eq. (2).

Remarks. Different from previous works (Yan et al. 2014) that find incorrectly labeled instances, we categorize instances into confusing and unconfusing instances. Without proper information about true labels, finding confusing instances is much easier than finding mislabeled ones (*cf.*, Fig. 1). This merit can also simplify the resulting probabilistic model, since we do not require to take different true-label cases into account. Moreover, the independency assumption for confusing instances in Eq. (2) can also be extended to true label dependent cases (Xiao et al. 2015). However, our simple assumption leads to lucid and explicable learning method in Algorithm 1.

Objective Function

In addition to parameters $\mathbf{w}$ of the classifier $\mathbf{h}_\mathbf{w}$, we need to estimate ψ_i and η_i for each instance $\mathbf{x}_i$. The probability of the noisy label ψ_i can easily be estimated by training a naive classifier on the original dataset with noisy labels (Berthon et al. 2020). By contrast, each confusing probability η_i has to be taken as a learnable parameter, and thus we aim to fit all trainable parameters $\Theta = \{\mathbf{w}, \eta_1, \ldots, \eta_N\}$ given only the noisy-labeled training sample $(S, \widetilde{Y})$. One common approach adopted for selecting parameters Θ is based on the maximum likelihood principle, which aims to find the optimal parameters by maximizing the log-likelihood:

$$\begin{aligned} \ell(\Theta) &= \sum_{i\in[N]} \log \hat{P}(\widetilde{\mathbf{y}}_i|\mathbf{x}_i; \Theta) \\ &= \sum_{i\in[N]} \log \sum_{j\in[c]} \hat{P}(\widetilde{\mathbf{y}}_i, \mathbf{y}_i^j = 1|\mathbf{x}_i; \Theta), \end{aligned} \tag{4}$$

where $\hat{P}(\cdot)$ denotes the estimated probability. In the next section, we describe the optimization algorithm for Eq. (4).

Before moving to the next section, we provide a simple derivation that is useful in below:

$$\begin{aligned} \hat{P}(\widetilde{\mathbf{y}}_i, \mathbf{y}_i^j = 1|\mathbf{x}_i; \Theta) &= \hat{P}(\widetilde{\mathbf{y}}_i|\mathbf{y}_i^j = 1, \mathbf{x}_i; \eta_i)\mathbf{h}_\mathbf{w}^j(\mathbf{x}_i) \\ &= \left[(1-\eta_i)\widetilde{\mathbf{y}}_i^j + \eta_i\psi_i\right]\mathbf{h}_\mathbf{w}^j(\mathbf{x}_i). \end{aligned} \tag{5}$$

The first equation is given by the definition of our target classifier, *i.e.,* $\mathbf{h}_\mathbf{w}^j(\mathbf{x}_i) = \hat{P}(\mathbf{y}_i^j = 1|\mathbf{x}_i; \mathbf{w})$; and the second equation is based on Eq. (3), where $\widetilde{\mathbf{y}}_i^j$ is in place of $\mathbb{I}\{\mathbf{y}_i = \widetilde{\mathbf{y}}_i\}$ since the vectors $\mathbf{y}_i, \widetilde{\mathbf{y}}_i$ each have only one non-zero element and we already have $\mathbf{y}_i^j = 1$.

Alternating Optimization

Due to the hidden variables (*i.e.,* unknown true labels), Eq. (4) is difficult to be optimized directly (Bishop 2006). Fortunately, based on Jensen's inequality, we can maximize its lower-bound to approach the optimal solution, namely,

$$\operatorname*{argmax}_\Theta \sum_{i\in[N]} \sum_{j\in[c]} \mathbf{q}_i^j \log \hat{P}(\widetilde{\mathbf{y}}_i, \mathbf{y}_i^j = 1|\mathbf{x}_i; \Theta), \tag{6}$$

where $\mathbf{q}_i^j$ is the estimated posterior of the true label defined by $\hat{P}(\mathbf{y}_i^j = 1|\widetilde{\mathbf{y}}_i, \mathbf{x}_i; \Theta')$, and Θ' is the current parameters.

As shown in the following, the optimization target $\mathbf{q}_i$ is related to the learnable parameters in Θ. Therefore, an alternating optimization framework is adopted to solve Eq. (6): It iteratively estimates the true label posteriors $\mathbf{q}_i$ and adaptively updates the learnable parameters in Θ. The resultant method proceeds to the optimal solution by alternating between two steps, namely, *predicting step* and *updating step*.

Predicting Step re-estimates the true label posterior $\mathbf{q}_i^j$ by the current parameters. Based on Bayes formula and Eq. (5), the estimated posterior of true label is:

$$\mathbf{q}_i^j = \frac{1}{K_i}\left[(1-\eta_i)\widetilde{\mathbf{y}}_i^j + \eta_i\psi_i\right]\mathbf{h}_\mathbf{w}^j(\mathbf{x}_i), \tag{7}$$

where $K_i = \sum_{j\in[c]} P(\widetilde{\mathbf{y}}_i, \mathbf{y}_i^j = 1|\mathbf{x}_i; \Theta)$ makes the elements of $\mathbf{q}_i$ sum to 1. Eq. (7) can be rewritten in vector as:

$$\mathbf{q}_i = \frac{1}{K_i}\mathbf{h}_\mathbf{w}(\mathbf{x}_i) * \left[(1-\eta_i)\widetilde{\mathbf{y}}_i + \eta_i\psi_i\mathbf{1}\right], \tag{8}$$

where $*$ denotes the element-wise product and $\mathbf{1}$ is a vector of all 1. Therein, to estimate the true label posterior $\mathbf{q}_i$, the noisy label $\widetilde{\mathbf{y}}_i$ is first linearly interpolated with $\mathbf{1}$ according to η_i and ψ_i, which integrates label ambiguity as in (Gao et al. 2017). Then, the smoothed label $(1-\eta_i)\widetilde{\mathbf{y}}_i + \eta_i\psi_i\mathbf{1}$ is weighted by $\mathbf{h}_\mathbf{w}(\mathbf{x}_i)$ to further combine the classifier prediction, and the normalizer K_i insures that $\mathbf{q}_i \in \mathcal{Q}$ is a valid label distribution.

Updating Step updates trainable parameters Θ via Eq. (6), where $\mathbf{q}_i$ is the re-estimated posterior given by Eq. (7). Ignoring the constant parts, maximizing Eq. (6) w.r.t. the confusing probability η_i is equivalent to:

$$\operatorname*{argmax}_{0\le\eta_i\le 1,\ \forall i} \sum_{i\in[N]} \sum_{j\in[c]} \mathbf{q}_i^j \log\left[(1-\eta_i)\,\widetilde{\mathbf{y}}_i^j + \eta_i\psi_i\right]. \tag{9}$$

Similarly, maximizing Eq. (6) w.r.t. the DNN parameters $\mathbf{w}$ can be written as:

$$\underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i\in[N]} \sum_{j\in[c]} \mathbf{q}_i^j \log \mathbf{h}_{\mathbf{w}}^j(\mathbf{x}_i), \tag{10}$$

which is the canonical maximum likelihood estimation that directly uses true label distributions as optimization targets.

Remarks. As we can see from Eq. (8), the noisy label $\widetilde{\mathbf{y}}_i$ is directly taken as the correct one if $\eta_i = 0$. As the value of η_i increases, the model prediction $\mathbf{h}_{\mathbf{w}}(\mathbf{x}_i)$ gradually dominates the true label prediction. Moreover, the model prediction is directly used for estimating the probability of the true label if η_i can equal to 1. Generally, η_i can be viewed as a learnable trade-off parameter, automatically determining the degree of learning from the noisy label and the model prediction.

Updating Rules

The constrained optimization Eq. (9) can be solved by Projected Gradient Ascent (PGA) (Nocedal and Wright 2006), as the constraint set $\{\eta_i : 0 \leq \eta_i \leq 1\}$ is convex in nature. In each iteration, each η_i is first updated by gradient ascent:

$$\eta_i \leftarrow \eta_i + \alpha_1 \frac{[\mathbf{1} + (\psi_i\eta_i - \eta_i - 1)\widetilde{\mathbf{y}}_i]^\top \mathbf{q}_i}{\eta_i + \epsilon}, \tag{11}$$

where α_1 is the learning rate and ϵ is fixed to 0.0001 to avoid dividing by zero. Then, PGA chooses the value of η_i that is nearest to the constraint $\{\eta : 0 \leq \eta \leq 1\}$ via:

$$\eta_i \leftarrow \min(\max(\eta_i, 0), 1), \tag{12}$$

such that the updated value of η_i in Eq. (11) still meets the constraint in Eq. (9). For the DNN parameters $\mathbf{w}$, if gradient ascent is adopted, the updating rule in solving Eq. (10) is:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_2 \sum_{i\in[N]} \sum_{j\in[c]} \frac{\mathbf{q}_i^j}{\mathbf{h}_{\mathbf{w}}^j(\mathbf{x}_i)} \nabla_{\mathbf{w}} \mathbf{h}_{\mathbf{w}}^j(\mathbf{x}_i), \tag{13}$$

where α_2 is the learning rate.

Overall Algorithm

In alternating optimization, to prevent inaccurate true label estimation at beginning, each confusing probability η_i is initialized by a small positive value (*e.g.*, $\eta^{\text{INIT}} = 0.01$). Accordingly, the estimated label distribution $\mathbf{q}_i$ is dominantly determined by the noisy label to avoid unstable model prediction. Besides, it is well-known that DNNs avoid learning label noise in the early training phase (Arpit et al. 2017; Tanaka et al. 2018), so this initialization strategy can effectively prevent our classifier from misleading initial training.

For large-scale learning tasks, such as image classification, the aforementioned alternating optimization is adopted in a mini-batch manner. The resultant learning method is summarized in Algorithm 1. In each iteration, a mini-batch Ξ is fetched uniformly at random from the training sample. In step 6, we re-estimate true label posteriors by applying Eq. (8) for instances in Ξ. In step 8-11, PGA executes (if necessary) one iteration to the subset thereof confusing probabilities only to the mini-batch Ξ. In step 12, the DNN parameters $\mathbf{w}$ are updated by mini-batch gradient ascent.

Algorithm 1 The Overall Algorithm.

Input: the noisy-labeled training sample $(S, \widetilde{Y})$; the estimated probabilities $\{\psi_1, \ldots, \psi_N\}$; number of training steps *num_steps*; and hyperparameters η^{INIT}, α_1, α_2.
Output: the optimal DNN parameters $\mathbf{w}$.

1: Initialize $\eta_i = \eta^{\text{INIT}}, \forall i$;
2: Initialize parameters $\mathbf{w}$;
3: **for** $t \leftarrow 1$ to *num_steps* **do**
4: Fetch a mini-batch Ξ uniformly at random;
5: *# predicting step*
6: $\mathbf{q}_i = \frac{1}{K_i}\mathbf{h}_{\mathbf{w}}(\mathbf{x}_i) * [(1-\eta_i)\widetilde{\mathbf{y}}_i + \eta_i\psi_i\mathbf{1}], i \in \Xi$;
7: *# updating step*
8: **if** update posterior **then**
9: $\eta_i \leftarrow \eta_i + \alpha_1 \frac{[\mathbf{1} + (\psi_i\eta_i - \eta_i - 1)\widetilde{\mathbf{y}}_i]^\top \mathbf{q}_i}{\eta_i + \epsilon}$;
10: $\eta_i \leftarrow \min(\max(\eta_i, 0), 1), i \in \Xi$;
11: **end if**
12: $\mathbf{w} \leftarrow \mathbf{w} + \alpha_2 \sum_{i\in\Xi} \sum_{j\in[c]} \frac{q_i^j}{h_{\mathbf{w}}^j(\mathbf{x}_i)} \nabla_{\mathbf{w}} h_{\mathbf{w}}^j(\mathbf{x}_i)$;
13: **end for**
14: **return** $\mathbf{w}$

Experiments

We first test the proposed method on *CIFAR-100* and *CIFAR-10* (Krizhevsky, Hinton et al. 2009) with synthetic label noise, and then conduct real-world label noise experiments on *Clothing1M* (Xiao et al. 2015).

Datasets

CIFAR-100: To generate the simulated dataset with instance-dependent label noise, we train a Multi-Layer Preceptron (MLP) on training sample with original labels. Then, we re-label the clean sample by the predicted labels given by MLP. The training accuracy of the MLP is 69.29%, resulting in a training dataset with label noise. These noisy labels are instance-dependent, as they are determined by the decision boundaries of the MLP in the instance space.

CIFAR-10: We also leverage the classification results with low capacity models to synthesize noisy labels. To exploit a different IDN setting, we apply an unsupervised clustering algorithm to generate noisy labels. First, we map training instances into embedding features that are outputs of pool-5 layer of ResNet-50 (He et al. 2016) pre-trained on *ImageNet*. Then, we apply k-means++ (Arthur and Vassilvitskii 2007) to these embedding features. Finally, instances in the same cluster are uniformly re-labeled by majority voting of their original labels, and the accuracy of the resulting pseudo labels is only 65.75%.

Additionally, we conduct synthetic CCN experiments to further demonstrate the universality of our method: noisy labels are generated by mapping TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, and CAT $\leftrightarrow$ DOG with different noise transition probabilities r. For example, an instance of BIRD has probability r to be mislabeled by

AIRPLANE. It mimics the confusion of similar classes, and the label noise is class-conditional here.

Clothing1M: *Clothing1M* is a large-scale dataset with noisy labels. It contains more than one million images of clothing crawled from several shopping websites, which are classified into 14 classes. Labels are automatically generated according to surrounding texts of these instances. It is reported that the labels suffer from instance-dependent noise (Xiao et al. 2015). Hence, we conduct experiments on *Clothing1M* to examine the performance of our method in a real-world IDN setting. The overall label precision for the training dataset is approximately 60%, and it also contains 50k, 14k, and 10k correctly labeled instances for auxiliary training, validation, and test. Besides, this dataset exhibits serious data imbalance phenomenon. For example, more than 88k instances are labeled to SHIRT, but, in contrast, only 19k instances have the label SWEATER.

Implementation Details [1]

CIFAR-100 & CIFAR-10: We employ ResNet-32 (He et al. 2016) for fair comparison with existing methods. Mean-subtraction, horizontal random flip, and 32×32 random crop are performed for data pre-processing. Then, we use mini-batch gradient ascent with a momentum of 0.9; a weight decay of 10^{-4}; and a batch size of 256. The alternating optimization algorithm is executed for 160 epochs. For the classifier parameters, we begin with a learning rate $\alpha_2 = 0.05$ and divide it by 10 per 40 epochs. For label confusing probabilities, they are updated every 5 epochs from the 35^{th} epoch. We test model performance with different learning rate α_1 on *CIFAR-100*, and fix $\alpha_1 = 0.7$ on *CIFAR-10*.

Clothing1M: As a common setting, ResNet-50 pre-trained on *ImageNet* is utilized as the backbone model. In addition to the common data pre-processing techniques (*i.e.*, mean subtraction, horizontal random flip, and 224×224 random crop), oversampling is applied by adding more copies for training instances that are labeled to each minority class. Oversampling is widely used for tackling data imbalance problem, while the oversampled dataset is not truly balanced, because the observed labels are not the true labels. We use mini-batch gradient ascent with a momentum of 0.9, a weight decay of 10^{-3}, and a batch size of 32. The alternating optimization algorithm is executed for 15 epochs. For the label confusing probabilities, their values are updated from the second epoch with a fixed learning rate $\alpha_1 = 0.05$. For the classifier parameters, the learning rate α_2 is set to be 5×10^{-3} and divided by 10 per 5 epochs.

The Adopted Baseline Methods

We compare with the following label noise learning algorithms in our experiments: "Co-teaching" (Han et al. 2018b) is the state-of-the-art sample selection algorithm which can avoid distribution bias; "Forward" (Patrini et al. 2017) and "$\mathcal{L}_{\text{DMI}}$" (Xu et al. 2019) are two noise-robust methods under CCN assumption; "Bootstrapping" (Reed et al. 2015) is the pioneer that handles the label noise by label correction; and "Tanaka" (Tanaka et al. 2018) as well as "PENCIL" (Yi and Wu 2019) are two advanced methods that also use model predictions in correcting noisy labels.

There have been recent advances that incorporate the methodology of meta-learning (Ren et al. 2018; Shu et al. 2019) and designing specific network architectures (Lee et al. 2018). However, for fair comparison, their results are not listed here as they typically rely on additional training instances with manually verified labels. Besides, for fair comparison, we re-implement "Bootstrapping" and "Co-teaching" by using ResNet-32 on *CIFAR-100* and *CIFAR-10*; and using ResNet-50 on *Clothing1M*.

[1]Note that, same backbone models and hyperparameters are used for directly training DNNs with noisy labels in estimating the probability of the observed noisy labels, *i.e.,* $\{\psi_1, \ldots, \psi_N\}$.

	Method	Test Accuracy (%)
#1	Co-teaching (Han et al. 2018b)	45.15±0.53
#2	Forward (Patrini et al. 2017)	44.97±0.77
#3	$\mathcal{L}_{\text{DMI}}$ (Xu et al. 2019)	45.07±0.42
#4	Bootstrapping (Reed et al. 2015)	44.52±0.35
#5	Tanaka (Tanaka et al. 2018)	46.02±0.42
#6	PENCIL (Yi and Wu 2019)	45.57±0.40
#7	Ours	**47.51±0.28**

Table 1: Average test accuracy and standard deviation (5 trials) on *CIFAR-100* with synthetic label noise.

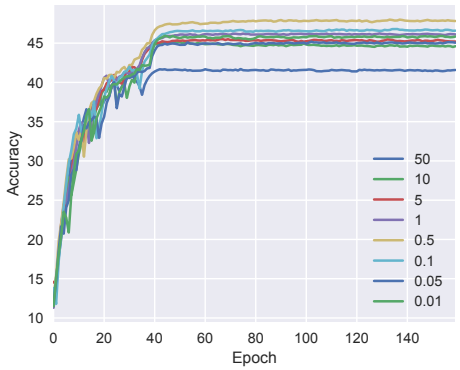


Figure 3: Average validation accuracy curves (5 trials) on *CIFAR-100* with different learning rate α_2 of the confusing probabilities. Note that, the standard deviations are not provided for clarity.

Experiments on *CIFAR-100*

To evaluate the performance of our method in IDN, we first test our proposed method on *CIFAR-100* with synthetic label noise. Our method and the baselines are implemented five times, and the average results are reported in Table **??**. As we can see, "$\mathcal{L}_{\text{DMI}}$" and "Forward" reveal relatively inferior per-

	Method	IDN(%)	CCN(%)				
			0.1	0.2	0.3	0.4	0.5
#1	Co-teaching	66.93±0.21	91.31±0.05	89.01±0.14	83.56±0.30	79.67±0.59	73.30±0.41
#2	Forward	64.29±1.07	91.55±0.08	90.64±0.12	87.46±0.17	85.65±0.23	**80.22±0.39**
#3	$\mathcal{L}_{\text{DMI}}$	66.15±0.43	91.73±0.05	90.07±0.11	91.25±0.35	88.80±0.27	76.15±0.37
#3	Bootstrapping	66.24±0.44	88.53±0.13	84.01±0.13	80.53±0.20	78.95±0.17	72.07±0.83
#4	Tanaka	67.77±0.57	92.19±0.02	91.97±0.13	91.64±0.11	90.39±0.16	68.39±1.25
#5	PENCIL	69.51±0.63	93.27±0.10	92.86±0.15	91.29±0.20	89.25±0.23	76.18±0.73
#6	Ours	**69.82±0.20**	**93.81±0.05**	**93.06±0.10**	**92.79±0.21**	**90.48±0.16**	78.03±0.55

Table 2: Average test accuracy and standard deviation (5 trials) on *CIFAR-10* with synthetic label noise.

formance due to their non-adaptiveness to IDN cases. The performance of "Bootstrapping" is also not effective, in part because of its inherent difficulty in hyperparameter tuning. By contrast, "Co-teaching" and two model prediction based methods "Tanaka" and "PENCIL" show much preferred results, while our method still outperform them by 1.49% to 1.94%. These results clearly demonstrate the merits of specifying the noisy label generation process in handling IDN.

In our method, the learning rate α_2 for confusing probabilities plays a key role in tackling the label noise. To provide some guidelines for hyperparameter tuning, we show the validation accuracy curves during training in Fig. 3, given the learning rates α_2 with different orders of magnitude. At the beginning, all the curves precariously vibrate due to the relatively large initial learning rate, while they are suddenly increase and become stable when the confusing probabilities start to update near 35^{th} epoch. As we can see, the performance reaches the highest result when the learning rate equals to 0.5. Moreover, with the rapidly increase of α_2, the model performance drops quickly (*e.g.*, $\alpha_2 = 50$). The reason is that, with a large learning rate, too many instances are taken as confusing at beginning, and the imperfect model predictions might be directly taken as the true label distributions. By contrast, though validation accuracy with extremely small learning rates (*e.g.*, $\alpha_2 = 0.01$) can also hurt the model performance, the impact is not as severe as the large cases. Here, most of the instances are taken as unconfusing ones, and it simply degenerates to a naive situation that learns directly from the noisy labels.

Experiments on *CIFAR-10*

We further conduct IDN experiments on *CIFAR-10*. The average results of five individual trials are summarized in the third column of Table **??**. According to the results, our method achieves overall higher test accuracy, again demonstrating the superiority of our learning method in handling IDN situations. The advanced algorithms implicitly handle instance-dependent noise based on heuristic rules, *e.g.*, self-learning and small-loss assumption, while our method still outperforms these baselines by 0.31% to 3.58%. By contrast, two CCN based methods, *i.e.*, "$\mathcal{L}_{\text{DMI}}$" and "Forward", perform relatively inferior to other baselines and our method, which reflects the significance in studying IDN cases.

Additionally, to show the proposed method can also properly handle class-conditional label noise, we conduct CCN experiments with noise transition probabilities r ranging from 0.1 to 0.5. Table **??** reports the average empirical results from column 3 to 7. Our setup is generally in line with (Yi and Wu 2019). However, to demonstrate the robustness of each method, we do not adaptively change their hyperparameters under various r cases. Compared with "Forward" and "$\mathcal{L}_{\text{DMI}}$", two robust learning algorithms in handling CCN, our method can outperform their results in most cases. The reason for the superior performance of "Forward" when $r = 0.5$ is that it uses the ground-truth noise transition matrix, which is unrealistic in real situations. Our method outperforms "Co-teaching" and "Bootstrapping", of which the results show extreme overfitting phenomenon. Furthermore, the average test accuracy of our method is also slightly better than "Tanaka" as well as "PENCIL", which are two state-of-the-art methods in these settings.

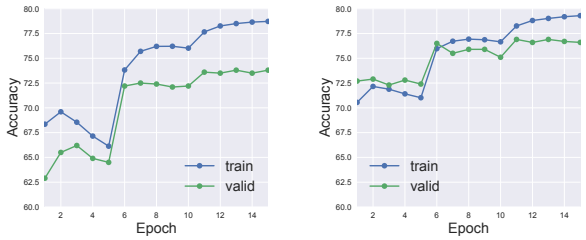
Experiments on *Clothing1M*

We further conduct experiments in a real-world setting, and the results on *Clothing1M* are summarized in Table **??**. Row #2 and #5 are quoted from (Patrini et al. 2017) and (Tanaka et al. 2018), and their results both rely on side information. In Row #2, Patrini et al. exploit 50k extra clean training instances to estimate the noise transition matrix. In Row #5, Tanaka et al. use the distribution prior of true labels to relieve the class imbalance problem. In contrast, we do not rely on any side information, while our method (Row #7) achieves 1.79% better test accuracy than that of "Tanaka" and 4.18% higher than "Forward". Our method also achieves substantial improvement over "Bootstrapping", "Co-teaching", and "$\mathcal{L}_{\text{DMI}}$", and pushes the best test accuracy reported by "PENCIL" from 73.49% to 74.02%.

We further exploit 50k training instances with correct labels by mixing them with the noisy labelled training sample. To make fully use of these clean labels, each mini-batch consists of the same number of instances drawn from the noisy-labeled and clean training instances. The confusing probabilities for the clean instances are fixed to 0 throughout the training procedure, and the result is reported in Row #8. As we can see, there is a noticeable improvement compared with the result in Row #7 that only utilizes the noisy-labeled instances. Obviously, training instances with correct labels can provide much reliable guide information during the training procedure. Besides, the clean training instances can also be used for finetuning the resulting classifier in #7, and the new state-of-the-art #10 outperforms the similar finetuned result of "Forward" in #9 more than 0.3%.

	Method	Test Accuracy(%)	Side Information
#1	Co-teaching	66.35	-
#2	Forward	69.84	transition matrix
#3	$\mathcal{L}_{\text{DMI}}$	72.46	-
#4	Bootstrapping	67.55	-
#5	Tanaka	72.23	label distribution
#6	PENCIL	73.49	-
#7	Ours	**74.02**	-
#8	Ours	**77.55**	+50k (train)
#9	Forward	80.38	+50k (finetune)
#10	Ours	**80.68**	+50k (finetune)

Table 3: Similar to previous works, we report the best test accuracy on *Clothing1M*. #2, #9 are quoted from (Patrini et al. 2017), #5 is quoted from (Tanaka et al. 2018), and #6 is quoted from (Yi and Wu 2019).



(a) 1M (b) 1M + 50k

Figure 4: Training and validation accuracy curves on *Clothing1M* with/without clean data.

To provide some insights into the learning behavior of the classifier in our method, we plot validation and training accuracy curves in Fig. 4 with two different learning strategies. Note that the training accuracy is calculated w.r.t. the noisy labels, instead of the clean labels as in validation. Abnormally high training accuracy typically indicates that the classifier overfits the noisy labels, while we always wish the validation accuracy is as high as possible. Fig. 4(a) reports the result that only uses noisy-labeled instances. The training and validation accuracy both increase in the first two epochs, since the noisy labels are directly taken as the correct ones for true label posteriors. Then, model predictions gradually participate into the prediction of true label posteriors. Accordingly, the training accuracy curve declines sharply, while the validation accuracy curve keeps more steady. This indicates that our method can effectively avoid the classifier in overfitting the label noise. Subsequently, both curves increase with the decrease of the learning rate, and finally become stable. Similar shape of the accuracy curves can be observed in Fig. 4(b), of which the result also exploits 50k clean training instances as in #8. Although the training accuracy in both cases are stable at nearly 80%, there is an obvious improvement for the validation accuracy curve in Fig. 4(b). This demonstrates the benefit of utilizing clean labels in training classifiers.

In Fig. 5, we give examples of the training instances

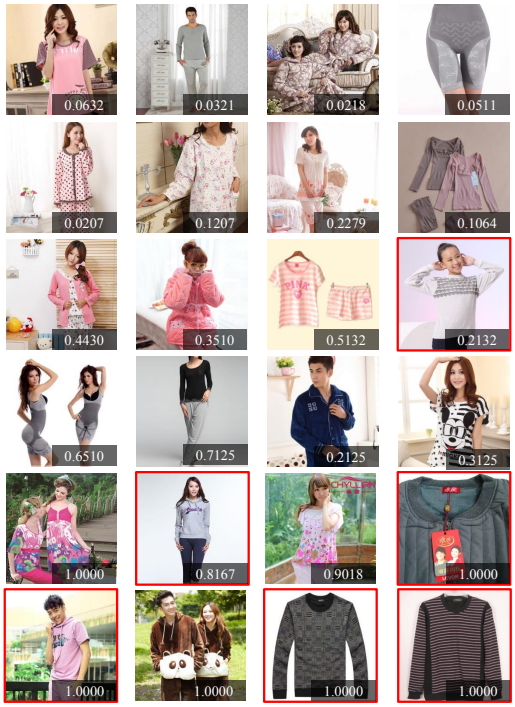


Figure 5: Examples of the training instances with label UNDERWEAR on *Clothing1M*. The estimated confusing probabilities are in the bottom right corner, and the mislabeled instances are in red box.

whose noisy labels are assigned to UNDERWEAR with the estimated confusing probabilities in their lower right corner. For instances with relatively small confusing probabilities in the first two rows, they are all correctly labeled, and can roughly be viewed as unconfusing instances. With larger confusing probabilities in the third and fourth rows, the corresponding instance pattern becomes diverse, which also contain an instance that is mislabeled. Finally, for instances with extremely large confusing probabilities in the last two rows, instances with rare patterns and incorrect labels become common.

Conclusion

By categorizing training instances into confusing and unconfusing instances, we propose a novel probabilistic model to describe the generation of instance-dependent label noise. Our model is realized by DNNs with additional trainable parameters in finding the potentially confusing instances, and we deploy an alternating optimization algorithm to iteratively correct noisy labels and update trainable parameters. The learning behavior is explainable, which can correct the noisy labels of confusing instances according to the model predictions. In the future, we will focus on the estimation algorithms for confusing probabilities, and extend our model to more weakly supervised learning scenarios (Kiryo et al. 2017; Yu et al. 2018).

Acknowledgments

BH was supported by the RGC Early Career Scheme No. 22200720, NSFC Young Scientists Fund No. 62006202, HKBU Tier-1 Start-up Grant, HKBU CSD Start-up Grant, and HKBU CSD Departmental Incentive Grant. TLL was supported by Australian Research Council Project DE-190101473. JY was supported by NSFC No. U1713208 and "111 Program"under Grant No. AH92005. CG was supported by NSFC No. 61973162, the Fundamental Research Funds for the Central Universities No. 30920032202, CCF-Tencent Open Fund No. RAGR20200101, the "Young Elite Scientists Sponsorship Program" by CAST No. 2018QNRC001, and Hong Kong Scholars Program No. XJ2019036.

References

Angelova, A.; Abu-Mostafam, Y.; and Perona, P. 2005. Pruning training sets for learning of object categories. In *CVPR*.

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. In *ICML*.

Arthur, D.; and Vassilvitskii, S. 2007. k-means++: The advantages of careful seeding. In *ACM-SIAM*.

Berthon, A.; Han, B.; Niu, G.; Liu, T.; and Sugiyama, M. 2020. Confidence scores make instance-dependent label-noise learning possible. *arXiv preprint arXiv:2001.03772* .

Bishop, C. M. 2006. Pattern recognition and machine learning. *Springer* .

Chen, P.; Ye, J.; Chen, G.; Zhao, J.; and Heng, P.-A. 2020. Robustness of accuracy metric and its inspirations in learning with noisy labels. *arXiv preprint arXiv:2012.04193* .

Cheng, J.; Liu, T.; Ramamohanarao, K.; and Tao, D. 2020. Learning with bounded instance-and label-dependent label noise. In *ICML*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Du, J.; and Cai, Z. 2015. Modelling class noise with symmetric and asymmetric distributions. In *AAAI*.

Gao, B.-B.; Xing, C.; Xie, C.-W.; Wu, J.; and Geng, X. 2017. Deep Label Distribution Learning With Label Ambiguity. *IEEE Trans. Image Process.* 26(6): 2825–2838. doi: 10.1109/TIP.2017.2689998.

Ghosh, A.; Kumar, H.; and Sastry, P. 2017. Robust loss functions under label noise for deep neural networks. In *AAAI*.

Goldberger, J.; and Ben-Reuven, E. 2017. Training deep neural-networks using a noise adaptation layer. In *ICLR*.

Guo, S.; Huang, W.; Zhang, H.; Zhuang, C.; Dong, D.; Scott, M. R.; and Huang, D. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *ECCV*.

Han, B.; Niu, G.; Yu, X.; Yao, Q.; Xu, M.; Tsang, I.; and Sugiyama, M. 2020. Sigua: Forgetting may make learning with noisy labels more robust. In *ICML*.

Han, B.; Yao, J.; Niu, G.; Zhou, M.; Tsang, I.; Zhang, Y.; and Sugiyama, M. 2018a. Masking: A new perspective of noisy supervision. In *NeurIPS*.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; and Sugiyama, M. 2018b. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. In *NeurIPS*.

Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; and Fei-Fei, L. 2017. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*.

Kiryo, R.; Niu, G.; du Plessis, M. C.; and Sugiyama, M. 2017. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Lee, K.-H.; He, X.; Zhang, L.; and Yang, L. 2018. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*.

Li, W.; Wang, L.; Li, W.; Agustsson, E.; and Van Gool, L. 2017a. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862* .

Li, Y.; Yang, J.; Song, Y.; Cao, L.; Luo, J.; and Li, L.-J. 2017b. Learning from noisy labels with distillation. In *ICCV*.

Liu, T.; and Tao, D. 2016. Classification with noisy labels by importance reweighting. *IEEE Trans. Pattern Anal. Mach. Intell.* 38(3): 447–461. doi:10.1109/TPAMI.2015.2456899.

Manwani, N.; and Sastry, P. S. 2013. Noise Tolerance Under Risk Minimization. *IEEE Trans. Cybern.* 43(3): 1146–1151. doi:10.1109/TSMCB.2012.2223460.

Menon, A. K.; Van Rooyen, B.; and Natarajan, N. 2016. Learning from binary labels with instance-dependent corruption. *Mach. Learn* .

Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; and Tewari, A. 2013. Learning with noisy labels. In *NeurIPS*.

Nocedal, J.; and Wright, S. 2006. Numerical optimization. *Springer* .

Patrini, G.; Rozza, A.; Krishna Menon, A.; Nock, R.; and Qu, L. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*.

Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2015. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to reweight examples for robust deep learning. In *ICML*.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*.

Sun, J.-w.; Zhao, F.-y.; Wang, C.-j.; and Chen, S.-f. 2007. Identifying and correcting mislabeled training instances. In *FGCN*.

Tanaka, D.; Ikami, D.; Yamasaki, T.; and Aizawa, K. 2018. Joint optimization framework for learning with noisy labels. In *CVPR*.

Van Rooyen, B.; Menon, A.; and Williamson, R. C. 2015. Learning with symmetric label noise: The importance of being unhinged. In *NeurIPS*.

Veit, A.; Alldrin, N.; Chechik, G.; Krasin, I.; Gupta, A.; and Belongie, S. 2017. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*.

Wang, Y.; Liu, W.; Ma, X.; Bailey, J.; Zha, H.; Song, L.; and Xia, S.-T. 2018. Iterative learning with open-set noisy labels. In *CVPR*.

Xia, X.; Liu, T.; Han, B.; Wang, N.; Gong, M.; Liu, H.; Niu, G.; Tao, D.; and Sugiyama, M. 2020. Parts-dependent label noise: Towards instance-dependent label noise. In *NeurIPS*.

Xia, X.; Liu, T.; Wang, N.; Han, B.; Gong, C.; Niu, G.; and Sugiyama, M. 2019. Are anchor points really indispensable in label-noise learning? In *NeurIPS*.

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X. 2015. Learning from massive noisy labeled data for image classification. In *CVPR*.

Xu, Y.; Cao, P.; Kong, Y.; and Wang, Y. 2019. L_DMI: A novel information-theoretic loss function for training deep nets robust to label noise. In *NeurIPS*.

Yan, Y.; Rosales, R.; Fung, G.; Ramanathan, S.; and Dy, J. G. 2014. Learning from multiple annotators with varying expertise. *Mach. Learn* .

Yao, Y.; Liu, T.; Han, B.; Gong, M.; Deng, J.; Niu, G.; and Sugiyama, M. 2020. Dual T: Reducing estimation error for transition matrix in label-noise learning. In *NeurIPS*.

Yi, K.; and Wu, J. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*.

Yu, X.; Liu, T.; Gong, M.; and Tao, D. 2018. Learning with biased complementary labels. In *ECCV*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.