Label Distribution for Learning with Noisy Labels

Yun-Peng Liu, Ning Xu, Yu Zhang and **Xin Geng***

MOE Key Laboratory of Computer Network and Information Integration, China
School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
{yunpengliu, xning, zhang_yu, xgeng}@seu.edu.cn

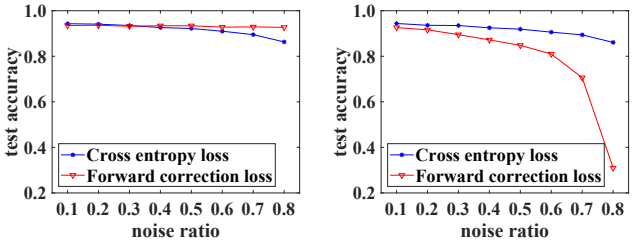
Abstract

The performances of deep neural networks (DNNs) crucially rely on the quality of labeling. In some situations, labels are easily corrupted, and therefore become noisy labels. Thus, designing algorithms that deal with noisy labels is of great importance for learning robust DNNs. However, it is difficult to distinguish between noisy labels and clean labels, which becomes the bottleneck of many methods. To address the problem, this paper proposes a novel method named *Label Distribution based Confidence Estimation* (LDCE). LDCE estimates the confidence of the observed labels based on *label distribution*. Then, the boundary between clean labels and noisy labels becomes clear according to confidence scores. To verify the effectiveness of the method, LDCE is combined with the existing learning algorithm to train robust DNNs. Experiments on both synthetic and real-world datasets substantiate the superiority of the proposed algorithm against state-of-the-art methods.

1 Introduction

Deep neural networks (DNNs) are the preferred choices for many classification tasks. A large number of labeled training instances are essential to training DNNs with high performance. It is convenient to obtain enough instances as well as labels with the assistance of the Internet and crawler [Divvala *et al.*, 2014], but noisy labels are inevitable. Training DNNs with noisy labels is challenging since the networks can easily overfit to the corrupted labels [Nettleton *et al.*, 2010].

Many prior works avoid overfitting to the corrupted data by correcting the noisy labels [Yi and Wu, 2019; Hendrycks *et al.*, 2018]. Note that the corrupted dataset inherently contains a large number of samples with clean labels. It is inevitable to make wrong corrections to clean labels due to the uncertainty of the boundary between clean labels and noisy labels. Such wrong operations will result in the decline of the performance. Moreover, current methods are weak at handling various noise patterns. When the noise pattern is changed, some methods will make unstable corrections. As shown in Fig. 1, we make a comparison between the model trained with *forward correction loss* [Patrini *et al.*, 2017], a classical label correction method, on samples containing noisy labels and the model trained with *cross entropy loss* on filtered samples with clean labels. Forward correction loss has good performances in asymmetric noise pattern but performs poorly under symmetric noise cases. In contrast, the model learned with only clean labels have stable performances on both noise patterns. It shows that the samples with clean labels are more important than the correcting operations under specific noise patterns. However, the uncertainty of labels makes it difficult to identify the samples with clean labels.



(a) Asymmetric noise (b) Symmetric noise

Figure 1: Performance comparison for model learned with *cross entropy loss* on samples with clean labels (i.e., filter out wrong-labeled samples) and model learned with *forward correction loss* on samples containing noisy labels under different noise patterns on CIFAR10.

To reduce the uncertainty of labels, a metric named *label confidence* is proposed in this paper for measuring the reliability of each label, in which clean labels get high confidence scores while noisy labels achieve low confidence scores. Note that *Label Distribution* (LD) naturally provides such a metric [Geng, 2016]. As shown in Fig. 2, LD assigns the description degree $d_{\boldsymbol{x}}^{y}$ to all classes in a distribution format, i.e., $d_{\boldsymbol{x}}^{y} \in [0,1]$ and $\sum_{y} d_{\boldsymbol{x}}^{y} = 1$. The description degree represents the degree to which y describes $\boldsymbol{x}$, which is naturally suitable to measure the label confidence. Moreover, compared with directly estimating the confidence score from the feature space, e.g., CleanNet [Lee *et al.*, 2018], the description degree is more reliable because the degree value is restricted by other classes in the distribution format. Motivated by this, this paper proposes a novel algorithm named

*Corresponding author

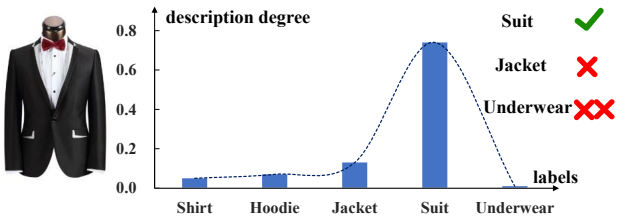


Figure 2: An example of Label Distribution.

Label Dsitribution based Confidence Estimation (LDCE) to estimate label confidence by generating label distribution.

Note that using some trusted samples is an effective approach to improving robustness in noisy label problems, and such small set can be fetched easily in many real-world applications [Hendrycks *et al.*, 2018]. LDCE estimates the label confidence via a small number of trusted samples, i.e., samples with clean labels. In this case, the training data is divided into two sets, i.e., a set with a few trusted samples and the other set with a large number of untrusted samples. Guided by the trusted samples, LDCE generates LD for each untrusted sample by measuring the similarity in feature space, i.e., the embedding space obtained from a feature encoder. Then, the confidence score of the observed label can be obtained from LD. After obtaining the label confidence, the samples with high confidence scores can be selected from the untrusted set, which is termed as *purified data* in this paper. Experiments show that the purified data mainly consists of the samples with clean labels.

Since the purified data is selected from the untrusted set, the risk of wrong operations to clean labels is mitigated. In this case, we combine the purified data with the existing correction method to train robust DNNs. The empirical results substantiate that the proposed method achieves favorable performances in both synthetic noise cases and real-world noise cases.

The contributions of this paper are as follows:

- A reliable metric *label confidence* is designed for measuring the reliability of labels based on *label distribution*.
- A practical algorithm for estimating the label confidence is proposed. Experimental results verify the efficiency of the estimation algorithm.
- A novel learning method using label confidence is designed for training robust DNNs. Experiments on three datasets show the superiority of the proposed method against state-of-the-art methods.

2 Related Work

Due to the presence of noisy labels, most of the learning algorithms based on the supervised learning framework can not accurately capture the mappings between instances and ground-truth labels. To deal with this problem, existing methods focus on mitigating the adverse effect of noisy labels.

One intuitive and easy approach is to remove the samples which are considered as wrong-labeled. For instance, [Han *et al.*, 2018; Chen *et al.*, 2019] attempt to filter out the unreliable samples during the training phase via a co-teaching framework. However, such methods do not explicitly deal with noisy labels. When the noise is severe, the performances of these methods are usually vulnerable.

An alternative approach is to correct noisy labels. [Tanaka *et al.*, 2018; Chen *et al.*, 2019] propose to replace the corrupted label with a more robust soft label which is in a distribution format. By converting a categorical label into a label distribution, the noisy label can be probabilistically corrected. Aside from correcting the labels directly, a loss correction strategy is proposed to revise the effects of noisy labels with the correction loss function [Patrini *et al.*, 2017]. However, wrong corrections to the clean labels will introduce extra noisy information during the learning process.

Other than the works mentioned above, some works notice the value of clean labels and turn to focus on strengthening the importance of the samples with clean labels. In [Guo *et al.*, 2018], CurriculumNet designs a learning schedule which starts from learning 'easy' subset to gradually adding 'complex' subset. In [Ren *et al.*, 2018; Shu *et al.*, 2019], sample reweighting strategy is used to improve the attention of clean labels. Our method belongs to this categroy. However, different from previous methods, we focus on combining the reweighting idea with correction methods.

3 The Proposed Methods

3.1 Notations Definition

First of all, some notations used in this paper are clarified as follows. The i-th instance is denoted by $\boldsymbol{x}_i$. The ground-truth label of the i-th instance is denoted by $\boldsymbol{y}_i \in \{0,1\}^c$ and $\mathbf{1}^{\mathrm{T}}\boldsymbol{y}_i = 1$, where c is the number of possible label values and $\mathbf{1}$ is a vector of all-ones. As the training set is corrupted, the observed label of the i-th instance is denoted by $\tilde{\boldsymbol{y}}_i \in \{0,1\}^c$ and $\mathbf{1}^{\mathrm{T}}\tilde{\boldsymbol{y}}_i = 1$.

In a c-class classification problem, $\mathcal{D}_u = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)|1 \leq i \leq N\}$ is a corrupted dataset, where the observed label $\tilde{\boldsymbol{y}}_i$ is considered as unreliable. Moreover, a trusted dataset is prepared as $\mathcal{D}_t = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)|1 \leq i \leq t\}$, where $t/(t+N) \ll 1$ is defined as the *trusted fraction*. To generate the *label confidence*, we introduce *label distribution* $\boldsymbol{d}_i$. The description degree of class j to instance $\boldsymbol{x}_i$ is denoted as $d_i^j \in [0,1]$, and $\sum_{j=1}^{c} d_i^j = 1$. The label confidence of the i-th sample is defined as c_i.

3.2 Confidence Estimation

As refered above, the bottleneck of current methods is the uncertainty on the untrusted set. Measuring the reliability of each label is a practical approach to reduce the uncertainty. Guided by this motivation, this paper designs a metric named *label confidence* based on *label distribution* [Geng, 2016] and proposes a practical method LDCE for estimating this metric.

Label Distribution Generation

LD offers a numerical metric *description degree* for each class in label space. As shown in Fig. 2, a high degree usually denotes more reliable labeling. Thus, the description degree

on the observed label is naturally suitable to be the label confidence metric.

Note that samples sharing the similar features tend to have the same label, similarity in feature space has been successfully applied in recovering LD [Xu *et al.*, 2018]. In this paper, the feature similarity is calculated with a small batch of trusted samples. Then, LD is generated according to the similarity scores.

In detail, it is the first step to sample a support set and two query sets from the training data. Sampling subsets to construct meta-task is commonly used in few-shot learning algorithms [Wang and Yao, 2019], which is helpful to learn from limited data. Then, the *membership degree* [Xu *et al.*, 2018] to class j for instance $\boldsymbol{x}_i$ is calculated by

$$m_i^j = \frac{1}{|\boldsymbol{S}_j|} \sum_{j=1}^{|\boldsymbol{S}_j|} s_{ij}, \tag{1}$$

where $\boldsymbol{S}_j$ denotes the samples of class j in the support set, and s_{ij} denotes the similarity score between instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Finally, the membership degrees to different classes are normalized into a label distribution $\boldsymbol{d}_i = [d_i^1, d_i^2, \ldots, d_i^c]$ via a softmax layer

$$d_i^j = \frac{\exp(m_i^j)}{\sum_{k=1}^{c} \exp(m_i^k)}. \tag{2}$$

After obtaining the label distribution, the label confidence c_i is updated iteratively according to

$$c_i^{(t+1)} = \alpha c_i^{(t)} + (1-\alpha) \boldsymbol{d}_i^{\mathrm{T}} \tilde{\boldsymbol{y}}_i, \tag{3}$$

where α is the step size.

Framework of Estimation Model

In order to get accurate label confidence from the corrupted data, it is important to train a reliable feature encoder for similarity calculation. This paper designs a unified learning framework to estimate the label confidence as well as learn a reliable feature encoder. As shown in Fig. 3, the framework is composed of a feature encoder $f_{\boldsymbol{\theta}}$, a metric module g, and a fully connected (FC) layer $f_{\boldsymbol{\phi}}$.

The estimation model is learned with a multi-task strategy [Ruder, 2017], which consists of a metric learning task and a classification task. The output of the feature encoder is denoted as $\boldsymbol{z}_i = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$, which refers to the feature embedding of instance $\boldsymbol{x}_i$. Then, the similarity score between two instances is measured as $s_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|$.

After obtaining the label confidence according to Eq.1-2, the loss function of the metric learning task is calculated by the cross entropy loss on the trusted query set according to Eq. 4.

$$\mathcal{L}_{sim} = -\frac{1}{m_1} \sum_{i=1}^{m_1} \sum_{j=1}^{c} y_i^j \log d_i^j, \tag{4}$$

where m_1 denotes the number of the query data sampled from the trusted set.

For the classification task, the linear layer $f_{\boldsymbol{\phi}}$ is used to predict the label of instance $\boldsymbol{x}_i$, and the predicted result is denoted as $p(\boldsymbol{y}_i|\boldsymbol{x}_i) = f_{\boldsymbol{\phi}}(\boldsymbol{z}_i)$. For data sampled from the trusted set, the loss is calculated directly with the cross entropy loss according to Eq. 5.

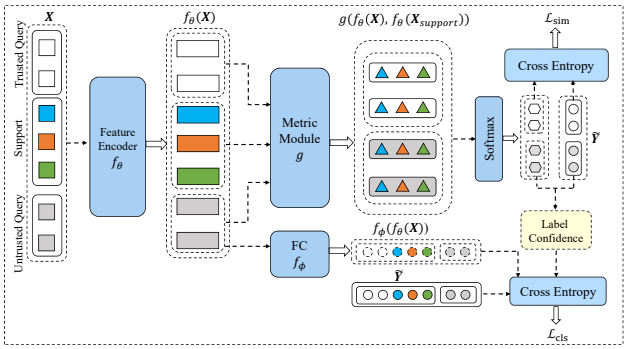


Figure 3: A conceptual illustration of the estimation model.

$$\mathcal{L}_t = -\frac{1}{n+m_1} \sum_{i=1}^{n+m_1} \sum_{j=1}^{c} y_i^j \log(p(y_i^j|\boldsymbol{x}_i)), \tag{5}$$

where n and m_1 denotes the number of support data and query data sampled from the trusted set. For data sampled from the untrusted set, the loss is calculated by Eq. 6 based on the attention mechanism [Vaswani *et al.*, 2017].

$$\mathcal{L}_u = -\sum_{i=1}^{m_2} \sum_{j=1}^{c} a_i \tilde{y}_i^j \log(p(y_i^j|\boldsymbol{x}_i)), \tag{6}$$

where a_i denotes the attention value obtained from the label confidence, and m_2 is the number of the query data sampled from the untrusted set. Since the encoder is trained from scratch, the estimation result is unstable in the former iterations. In this case, we initialize the label confidence to 0 and calculate the attention value with a threshold δ.

$$a_i = \begin{cases} c_i & if \quad c_i \geq \delta, \\ 0 & otherwise. \end{cases} \tag{7}$$

Then, the loss function for the classification task is designed as follows:

$$\mathcal{L}_{cls} = \frac{\mathcal{L}_t + \mathcal{L}_u}{n + m_1 + \sum_{i=1}^{m_2} \mathbb{I}(a_i)}, \tag{8}$$

where $\mathbb{I}(.)$ is the indicator function.

Based on the above analysis, Eq. 4 and Eq. 8 are combined to form the loss function for training the estimation model.

$$\mathcal{J}_{LDCE} = \mathcal{L}_{sim} + \mathcal{L}_{cls}. \tag{9}$$

Algorithmic detailes are shown in Algorithm 1.

3.3 Learning with Purified Data

After obtaining the label confidence, the boundary between clean labels and noisy labels becomes clear by assuming that clean labels get higher label confidence. In this case, the samples with high confidence scores are selected with the same threshold in the estimation model, and the selected samples are named as *purified data*. Then, we combine the purified data with the classical correction method GLC [Hendrycks *et al.*, 2018] by proposing a revised correction loss, and the learning method is named as *Purified Data based Loss Correction* (PDLC).

Algorithm 1 Label Distribution based Confidence Estimation

Input: Trusted data $\mathcal{D}_t$, Untrusted data $\mathcal{D}_u$, $f_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\phi}}$.
Parameter: batch size n, m_1, m_2, max iterations T, step size α, threshold δ.

Initialize model parameter $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and label confidence.
for $i = 1$ **to** T **do**
  $\{\boldsymbol{x}^{(s)}, \boldsymbol{y}^{(s)}\} \leftarrow$ SampleMiniBatch($\mathcal{D}_t$, n)
  $\{\boldsymbol{x}^{(q)}, \boldsymbol{y}^{(q)}\} \leftarrow$ SampleMiniBatch($\mathcal{D}_t$, m_1)
  $\{\boldsymbol{x}^{(\tilde{q})}, \tilde{\boldsymbol{y}}^{(\tilde{q})}\} \leftarrow$ SampleMiniBatch($\mathcal{D}_u$, m_2)
  $\boldsymbol{z}_i^{(s)} \leftarrow f_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(s)})$, $\boldsymbol{z}_i^{(q)} \leftarrow f_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(q)})$, $\boldsymbol{z}_i^{(\tilde{q})} \leftarrow f_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(\tilde{q})})$.
  Calculate the label distribution $\boldsymbol{d}_i$ by Eq. 1-2.
  Formulate the learning function by Eq. 4-9.
  Update model parameter $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ in backward process.
  Update label confidence by Eq.3.
end for

Revised Forward Correction Loss

In [Patrini *et al.*, 2017], *forward correction loss* is proposed to tackle the noisy label problem. The loss function is shown as

$$\ell_{corr}(p(\boldsymbol{y}_i|\boldsymbol{x}_i), \tilde{\boldsymbol{y}}_i = \boldsymbol{e}^k) = -\log \sum_{j=1}^{c} \boldsymbol{C}_{jk} p(y_i^j|\boldsymbol{x}_i), \quad (10)$$

where $\boldsymbol{e}^k$ denotes the k-th standard canonical vector, i.e., $\boldsymbol{e}^k \in \{0,1\}^c$ and $\mathbf{1}^{\mathrm{T}}\boldsymbol{e}^k = 1$. $\boldsymbol{C} \in \mathbb{R}^{c\times c}$ is the noise transition matrix and $\boldsymbol{C}_{jk} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^k|\boldsymbol{y} = \boldsymbol{e}^j)$. How to obtain $\boldsymbol{C}$ is the same as [Hendrycks *et al.*, 2018].

Since the purified data contains mainly samples with clean labels, forward correction loss does not perform well on purified data when compared with cross entropy loss. However, simply replacing the correction loss with the cross entropy loss will mitigate the correction effects on the remaining untrusted samples. In this case, we design a revised forward correction loss by combining forward correction loss with cross entropy loss. The combination loss function is shown as

$$\ell_{purified} = \lambda \ell_{ce} + (1-\lambda)\ell_{corr}, \quad (11)$$

Specifically, $\lambda \in (0,1)$ is the hyperparameter to balance the cross entropy loss and the initial correction loss, which is selected according to the model performance on each noise pattern. We observe that the case with a high noise ratio favors a small λ value, while the case with a small noise ratio prefers a high λ value.

Final Objective

After obtaining the purified data, the whole training set is divided into three parts. The first part is the pre-acquired trusted samples $\mathcal{D}_t$ with ground-truth labels, and the loss function is the cross entropy loss ℓ_{ce}. The second part is the purified data $\mathcal{D}_p$ with high label confidence, and the loss on $\mathcal{D}_p$ is the revised forward correction loss $\ell_{purified}$. The last part is the remaining samples with relatively low label confidence that recorded as $\tilde{\mathcal{D}}_u$, and the loss on $\tilde{\mathcal{D}}_u$ is calculated by the forward correction loss ℓ_{corr}.

$$\mathcal{L}_{trusted} = \sum_{i=1}^{|\mathcal{D}_t|} \ell_{ce}(f_{\boldsymbol{\varphi}}(\boldsymbol{x}_i), \boldsymbol{y}_i), \quad (12)$$

$$\mathcal{L}_{purified} = \sum_{i=1}^{|\mathcal{D}_p|} \ell_{purified}(f_{\boldsymbol{\varphi}}(\boldsymbol{x}_i), \tilde{\boldsymbol{y}}_i)), \quad (13)$$

$$\mathcal{L}_{untrusted} = \sum_{i=1}^{|\tilde{\mathcal{D}}_u|} \ell_{corr}(f_{\boldsymbol{\varphi}}(\boldsymbol{x}_i), \tilde{\boldsymbol{y}}_i), \quad (14)$$

$$\mathcal{J} = \frac{\mathcal{L}_{trusted} + \mathcal{L}_{purified} + \mathcal{L}_{untrusted}}{|\mathcal{D}_t| + |\mathcal{D}_p| + |\tilde{\mathcal{D}}_u|}. \quad (15)$$

The DNN model is denoted as $f_{\boldsymbol{\varphi}}$. By minimizing Eq. 15, the optimal parameter $\boldsymbol{\varphi}$ of the DNN model can be obtained.

4 Experiments

4.1 Experimental Setup

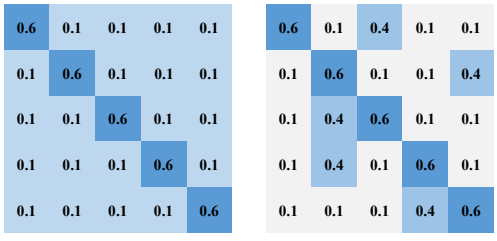
Datasets

The experiments are conducted on CIFAR10 and CIFAR100 [Krizhevsky *et al.*, 2009] with synthetic label noise and Clothing1M [Xiao *et al.*, 2015] with real-world label noise.

CIFAR10 & CIFAR100 are two datasets consists of 32 × 32 color images. The two datasets both contain 50,000 training samples and 10,000 test samples. CIFAR10 assigns the samples with 10 classes, while CIFAR100 assigns the samples with 100 classes.

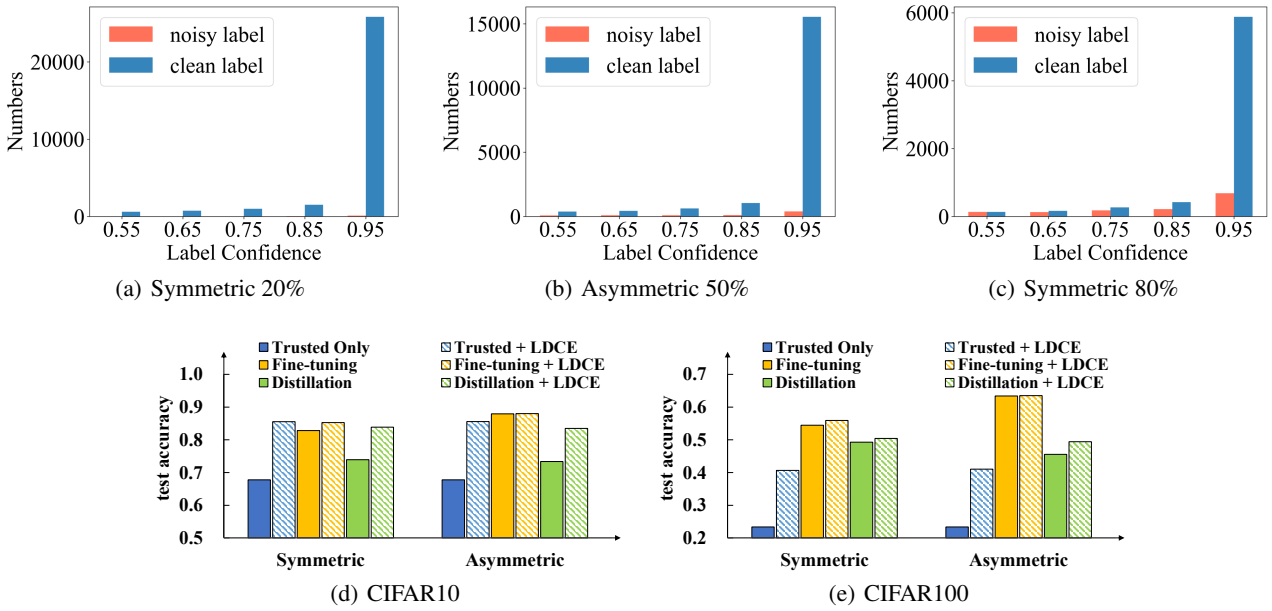
Since a trusted set is essential in the learning settings, the training set is split into two parts with the trusted fraction of 5% and 10%. Then, the synthetic label noise is added into the untrusted set. Following the previous literature [Chen *et al.*, 2019], experiments are conducted on two representative types of label noise: symmetric noise and asymmetric noise. As illustrated in Fig. 4, the label noise is uniformly distributed among all other classes in the symmetric case. In the asymmetric case, label noise is generated by flipping a label to a different class. The noise ratio ε denotes the proportion of wrong labels. In this paper, we test noise ratio 20%, 50% and 80% for both symmetric and asymmetric noise.

Clothing1M is a dataset collected with real-world label noise. The training set consists of 1M images with noisy labels from 14 fashion classes and 47,570 images with manually refined labels. The validation set and test set have 14,313 and 10,526 images respectively. The images with manually refined labels in training set are used as trusted samples.



(a) Symmetric Noise (b) Asymmetric Noise

Figure 4: Examples of noise transition matrix $\boldsymbol{C}$ (taking 5 classes and noise ratio $\varepsilon = 40\%$ as an example).



(a) Symmetric 20% (b) Asymmetric 50% (c) Symmetric 80%

(d) CIFAR10 (e) CIFAR100

Figure 5: (a)-(c) Distribution of label confidence in the interval $[\delta, 1]$ on CIFAR10 with 5% trusted fraction. (d)-(e) Performance comparison for methods learned without purified data and with purified data on CIFAR10 and CIFAR100 with 5% trusted fraction and 50% noise ratio. The striped bars denote the methods learned with purified data.

Implementation Details

The experiments are implemented with PyTorch framework. Detailed implementations for each dataset are as follows.

CIFAR10 & CIFAR100. For estimation model, we use a ResNet-32 [He *et al.*, 2016] as the feature encoder. The learning rate is 0.1 with a deacy step 60 and a decay rate 0.1. The hyper-parameters is α=0.6, δ=0.5. We observe that the hyper-parameters, which are selected by experience, are not very sensitative to different noise patterns. Thus, the hyper-parameters are fixed for all noise patterns. For classifier model, we adopt a Wide Residual Network [Zagoruyko and Komodakis, 2016] of depth 40 and a widening factor of 2. The learning rate is 0.1 with a multi-step deacy [60, 80, 90] and a deacy rate 0.2. For both estimation model and classifier model, we use SGD optimizer with 0.9 momentum, a ℓ_2 weight decay 1×10^{-4} and train the models for 100 epochs.

Clothing1M. Following the previous works [Tanaka *et al.*, 2018; Shu *et al.*, 2019], we use ResNet-50 [He *et al.*, 2016] pre-trained on ImageNet for both the feature encoder and classifier model. The hyper-parameters is the same with CIFAR10 and CIFAR100. The learning rate is 0.01 with a decay step 5 and a deacy rate 0.1. We use SGD optimizer with a momentum 0.9, a ℓ_2 weight decay 1×10^{-3} and train the models for 10 epochs. For preprocessing, we resize each image to 256×256, crop the middle 224×224 as input, and perform normalization.

Baselines

We compare our algorithm with **Trusted Only**, referring to learning DNNs with only trusted samples, and **Fine-tuning**, referring to fine-tuning DNNs trained on corrupted data with trusted samples [Shu *et al.*, 2019]. Other than the two methods above, classical comparison methods with the same learning settings include: **Distillation** [Li *et al.*, 2017], **MentorNet** [Jiang *et al.*, 2018], **L2RW** [Ren *et al.*, 2018], **MW-Net** [Shu *et al.*, 2019], **GLC** [Hendrycks *et al.*, 2018].

For fair comparisons, all the contrast methods are evaluated with the same setup. To ensure that the empirical results are reliable, we repeat each experiment on synthetic noise cases 5 times with different random seeds.

4.2 Experimental Results

Results on CIFAR10 & CIFAR100

The label confidence obtained from the estimation model is critical for the learning method PDLC. To investigate the performance of the estimation model, we firstly illustrate the distribution of label confidence in the interval $[\delta, 1]$ (δ = 0.5) on CIFAR10 with 5% trusted fraction. As can be seen from the Fig. 5(a)-5(c), for both symmetric and asymmetric noise types, the purified data, i.e., samples with label confidence in the interval $[\delta, 1]$, mainly consists of the samples with clean labels. In other words, only limited wrong-labeled samples exist in the purified data, which verifies the effectiveness of the estimation model and the capability of label confidence in identifying the clean labels.

To further investigate the effectiveness of the purified data, we combine the purified data with methods **Trusted Only**, **Fine-tuning** and **Distillation**. Since the three methods are easy to implement, we only need to enrich the trusted samples with purified data. Fig. 5(d)-5(e) summarize the results. It can be observed that the three methods combined with the purified data all achieved performance gain, which verifies the effectiveness of the purified data.

dataset (trusted fraction)	noise type	noise ratio	method							
			Trusted Only	Fine-tuning	Distillation	MentorNet	L2RW	MW-Net	GLC	PDLC
CIFAR10 (5%)	symmetric	20%	67.78±0.58	87.52±0.20	84.04±0.40	91.26±0.17	88.49±0.29	91.54±0.40	92.06±0.11	**92.14±0.11**
		50%		82.82±0.36	73.92±2.36	85.82±0.27	83.39±0.71	86.37±0.38	87.10±0.31	**87.36±0.43**
		80%		65.90±1.78	66.52±3.33	44.48±6.62	56.61±1.75	64.06±0.65	70.42±1.43	**77.80±2.23**
	asymmetric	20%	67.78±0.58	88.88±0.16	84.56±0.62	92.52±0.27	89.64±0.12	92.73±0.28	93.38±0.22	**93.42±0.18**
		50%		87.94±0.44	73.36±2.51	72.90±3.06	87.47±0.45	69.30±3.10	**93.00±0.43**	92.96±0.35
		80%		87.82±0.40	66.64±3.47	–	82.42±0.80	–	**92.70±0.37**	92.40±0.33
CIFAR10 (10%)	symmetric	20%	79.38±0.67	88.80±0.25	87.20±0.32	91.42±0.16	87.88±0.13	91.22±0.26	92.16±0.15	**92.28±0.15**
		50%		85.12±0.46	79.18±1.84	86.34±0.23	83.75±0.44	86.28±0.33	87.66±0.19	**88.56±0.22**
		80%		76.48±1.54	73.84±2.81	63.90±1.81	64.59±0.88	72.89±0.84	80.72±0.49	**84.18±0.82**
	asymmetric	20%	79.38±0.67	89.86±0.09	86.44±0.64	92.24±0.17	88.97±0.23	92.25±0.41	93.52±0.16	**93.62±0.22**
		50%		89.04±0.17	79.06±2.43	82.40±1.21	87.05±0.32	83.87±1.77	93.24±0.22	**93.32±0.23**
		80%		88.88±0.40	73.50±3.41	–	83.26±0.40	–	92.78±0.41	**92.86±0.19**
CIFAR100 (5%)	symmetric	20%	23.40±0.51	62.38±0.24	66.52±0.33	69.52±0.26	60.98±1.32	69.28±0.21	71.24±0.17	**71.48±0.41**
		50%		54.48±0.19	49.28±0.50	59.44±0.57	51.29±1.87	61.44±2.03	62.76±0.42	**63.48±0.63**
		80%		30.18±1.76	26.90±1.26	15.50±2.48	23.51±3.03	37.40±4.80	34.26±1.08	**40.46±1.21**
	asymmetric	20%	23.40±0.51	64.66±0.74	68.18±0.63	71.78±0.24	61.12±1.80	68.11±0.33	74.86±0.19	**74.90±0.20**
		50%		63.40±0.87	45.58±1.06	43.34±1.37	54.54±1.75	41.05±0.76	74.28±0.36	**74.44±0.43**
		80%		62.72±0.55	22.00±1.08	–	33.08±3.01	–	**74.12±0.31**	73.72±0.42
CIFAR100 (10%)	symmetric	20%	38.70±0.29	64.28±0.39	64.66±0.28	69.58±0.67	59.60±0.79	68.26±0.70	71.76±0.22	**72.06±0.42**
		50%		58.04±0.64	52.36±0.66	60.70±0.72	50.83±2.28	60.33±3.26	64.82±0.39	**65.30±0.31**
		80%		41.32±1.49	39.14±1.64	23.70±4.40	28.64±2.56	47.98±0.67	47.78±0.93	**52.42±0.46**
	asymmetric	20%	38.70±0.29	66.28±0.43	66.04±0.67	71.92±0.13	60.71±0.89	66.82±0.29	74.88±0.22	**74.98±0.20**
		50%		65.54±0.23	51.16±0.77	49.78±1.69	55.23±1.35	45.17±1.00	74.24±0.18	**74.54±0.27**
		80%		64.48±0.31	36.20±1.37	–	37.73±0.78	–	74.14±0.15	**74.24±0.30**

Table 1: Average test accuracy (%, 5 runs) with standard deviation on CIFAR10 and CIFAR100 under symmetric noise with ratio 20%, 50%, 80%, and asymmetric noise with ratio 20%, 50%, 80%. The best test accuracy is bolded.

Next, we evaluate the performance of PDLC by comparing the method with seven contrast methods on different noise patterns and trusted fractions. Different from the above experiments that simply enrich the trusted samples with purified data, PDLC leverages the purified data with a revised correction loss function. Table 1 summarizes the experimental results. It can be observed that PDLC achieves favorable performances among different noise patterns.

For symmetric noise cases, PDLC outperforms all the comparison methods in all noise ratios and trusted fractions. When the noise ratio is small (e.g., 20%), most of the comparison methods can achieve high test accuracies. Even in such cases, PDLC still achieves higher test accuracies. When the noise ratio is high (e.g., 80%), the performances of the classifier models drop significantly. In this case, PDLC shows strong superiority over the contrast methods.

For asymmetric noise cases, PDLC also achieves better test accuracies in most cases. Since the forward correction loss used in both GLC and PDLC is well-designed for asymmetric noise, both the two methods can achieve high test accuracies. In this case, the purified data used in PDLC can play a limited role in performance improvements, which explains why PDLC does not rank first in some cases.

Results on Clothing1M

To verify the effectiveness of the proposed method on real-world data, experiments are conducted on Clothing1M, which is a dataset with real-world label noise. We compare PDLC with several methods, including Cross Entropy, Forward [Patrini *et al.*, 2017], LCCN [Yao *et al.*, 2019], PENCIL [Yi and Wu, 2019], MW-Net [Shu *et al.*, 2019] and GLC [Hendrycks *et al.*, 2018]. The results are summarized in Tabel 2. Row 1 to 4 and row 6 are quoted from [Shu *et al.*, 2019], and row 5 is quoted from [Yi and Wu, 2019]. Row 7 to 8 are obtained by our own implementations. It can be observed that PDLC achieves the best performance against the other methods.

#	method	accuracy	#	method	accuracy
1	Cross Entropy	68.96	5	PENCIL	73.49
2	Forward	69.84	6	MW-Net	73.72
3	LCCN	73.03	7	GLC	73.53
4	MLNT	73.47	8	PDLC	74.15

Table 2: Test accuracy (%) on Clothing1M.

5 Conclusion

In this paper, a novel method LDCE is proposed to estimate label confidence. The label confidence is a metric designed for measuring the reliability of labels. LDCE estimates the label confidence by generating label distribution. Then, the samples with high confidence scores are selected as purified data. To verify the effectiveness of LDCE, we design a learning method PDLC by leveraging the purified data. The experiments conducted on both synthetic and real-world datasets substantiate the superiority of the learning method.

This paper has shown that estimating the label confidence from the corrupted data is a feasible strategy in the noisy label problem. In the future, we will explore more effective approaches for estimating and utilizing the label confidence.

Acknowledgments

This research was supported by the National Key Research & Development Plan of China (No. 2018AAA0100104), the National Science Foundation of China (61622203), the Collaborative Innovation Center of Novel Software Technology and Industrialization, the Collaborative Innovation Center of Wireless Communications Technology, and the National Natural Science Foundation of China (61702095).

References

[Chen *et al.*, 2019] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070, 2019.

[Divvala *et al.*, 2014] Santosh K Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3270–3277, 2014.

[Geng, 2016] Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1734–1748, 2016.

[Guo *et al.*, 2018] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision*, pages 135–150, 2018.

[Han *et al.*, 2018] 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.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hendrycks *et al.*, 2018] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in neural information processing systems*, pages 10456–10465, 2018.

[Jiang *et al.*, 2018] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2309–2318, 2018.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[Lee *et al.*, 2018] 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.

[Li *et al.*, 2017] 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.

[Nettleton *et al.*, 2010] David F Nettleton, Albert Orriols-Puig, and Albert Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial intelligence review*, 33(4):275–306, 2010.

[Patrini *et al.*, 2017] 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.

[Ren *et al.*, 2018] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4331–4340, 2018.

[Ruder, 2017] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[Shu *et al.*, 2019] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pages 1917–1928, 2019.

[Tanaka *et al.*, 2018] 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.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Wang and Yao, 2019] Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *arXiv preprint arXiv:1904.05046*, 2019.

[Xiao *et al.*, 2015] 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.

[Xu *et al.*, 2018] Ning Xu, An Tao, and Xin Geng. Label enhancement for label distribution learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2926–2932, Stockholm, Sweden, 2018.

[Yao *et al.*, 2019] Jiangchao Yao, Hao Wu, Ya Zhang, Ivor W Tsang, and Jun Sun. Safeguarded dynamic label regression for noisy supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

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

[Zagoruyko and Komodakis, 2016] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.