Small-Vote Sample Selection for Label-Noise Learning

Youze Xu[1], Yan Yan (✉)[1][0000−0002−3674−7160], Jing-Hao Xue[2][0000−0003−1174−610X], Yang Lu[1][0000−0002−3497−9611], and Hanzi Wang[1][0000−0002−6913−9786]

[1] Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen, China
xuyouze@stu.xmu.edu.cn, {yanyan, luyang, hanzi.wang}@xmu.edu.cn
[2] Department of Statistical Science, University College London, London, UK
jinghao.xue@ucl.ac.uk

Abstract. The small-loss criterion is widely used in recent label-noise learning methods. However, such a criterion only considers the loss of each training sample in a mini-batch but *ignores* the loss distribution in the whole training set. Moreover, the selection of clean samples depends on a *heuristic* clean data rate. As a result, some noisy-labeled samples are easily identified as clean ones, and vice versa. In this paper, we propose a novel yet simple sample selection method, which mainly consists of a Hierarchical Voting Scheme (HVS) and an Adaptive Clean data rate Estimation Strategy (ACES), to accurately identify clean samples and noisy-labeled samples for robust learning. Specifically, we propose HVS to effectively combine the global vote and the local vote, so that both *epoch-level* and *batch-level* information is exploited to assign a hierarchical vote for each mini-batch sample. Based on HVS, we further develop ACES to *adaptively* estimate the clean data rate by leveraging a 1D Gaussian Mixture Model (GMM). Experimental results show that our proposed method consistently outperforms several state-of-the-art label-noise learning methods on both synthetic and real-world noisy benchmark datasets.

Keywords: Noisy labels · Label-noise learning · Sample selection

1 Introduction

In recent years, Deep Neural Networks (DNNs) based methods have achieved remarkable success in a variety of artificial intelligence-related tasks. Generally, these methods rely heavily on a large number of high-quality annotated training samples. Unfortunately, collecting large-scale samples with fully accurate annotations is labor-intensive and time-consuming, which unavoidably yields noisy labels [2, 11, 27]. Many DNNs-based methods easily overfit noisy-labeled samples, mainly due to the high learning capability of DNNs involving millions of parameters. A recent study [26] has shown that DNNs severely suffer from poor

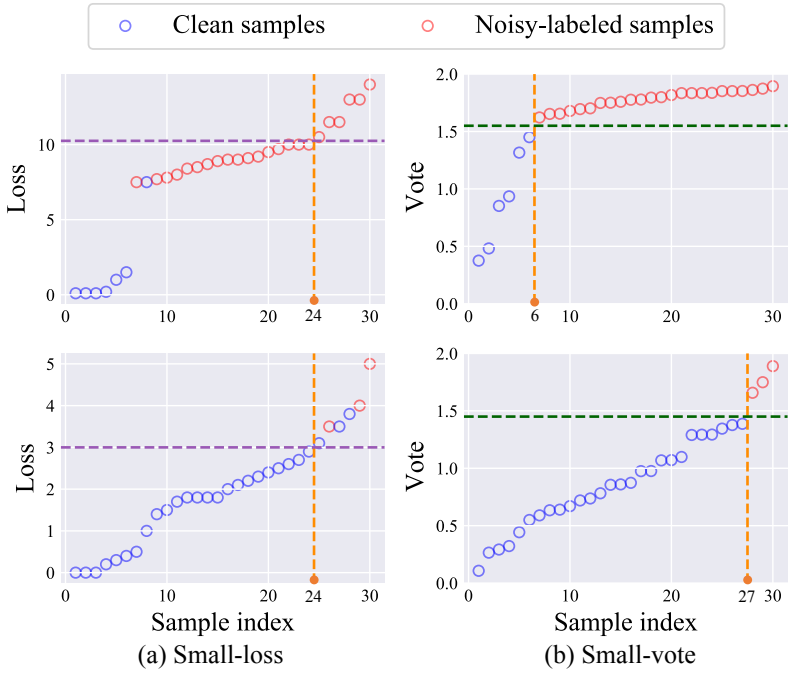


Fig. 1. Comparison between (a) the small-loss criterion and (b) our proposed small-vote sample selection method in the two randomly chosen mini-batches (the batch size is 30) during training. The purple dotted lines represent the thresholds obtained by the heuristic clean data rate (80% in two cases) and the green dotted ones represent the adaptive thresholds obtained by our method. The samples whose indices are smaller than or equal to the orange dotted lines are selected as clean samples.

generalization capability when they are trained on the samples containing noisy labels.

To alleviate the adverse effects of noisy labels, label-noise learning methods have been proposed to learn robust representations. Some methods [19, 14, 10, 24, 4] address the problem of noisy labels by selecting or weighting clean samples for each mini-batch during training. These methods usually take advantage of the small-loss criterion, which first identifies small-loss training samples as clean ones and then uses them for updating the network parameters [7, 20]. Such a criterion is well justified by the memorization effect that DNNs are able to learn simple and general patterns from clean samples before fitting noisy-labeled samples [1]. However, the small-loss criterion only considers the loss of training samples in each single mini-batch but ignores the loss distribution in the whole training set. Moreover, a heuristic clean data rate is used to select clean samples, whereas the noisy label distribution varies in randomly chosen mini-batches. Therefore, the small-loss criterion may not accurately identify clean samples and noisy-labeled samples, as illustrated in Figure 1(a). This raises the difficulty of learning robust

models. Hence, how to accurately distinguish clean samples from noisy-labeled samples remains a great challenge.

To address the above challenge, in this paper, we propose a novel small-vote sample selection method to accurately select clean samples and noisy-labeled samples, and robustly train DNN simultaneously. Specifically, our proposed method mainly consists of a Hierarchical Voting Scheme (HVS) and an Adaptive Clean data rate Estimation Strategy (ACES). First, we develop HVS to assign a hierarchical vote for each mini-batch sample. Then, based on HVS, we introduce ACES to estimate the clean data rate by leveraging a 1D Gaussian Mixture Model (GMM). Some intermediate training results are given in Figure 1(b). Obviously, our proposed method adaptively estimates clean data rates reflecting well the different proportions of clean samples in the two randomly chosen mini-batches.

The contributions of this paper are summarized as follows:

- We propose a novel yet simple small-vote sample selection method, which performs noisy label detection and learns from noisy data in an end-to-end manner. In particular, we develop HVS to effectively combine the global vote from previous epochs and the local vote from the current mini-batch. In this way, both epoch-level and batch-level information can be fully exploited for voting. Based on HVS, we design ACES to adaptively and accurately identify clean samples, guiding the learning of a robust model.
- We conduct extensive experiments on four benchmark datasets (including MNIST, CIFAR-10, CIFAR-100, and Clothing 1M) with synthetic and real-world noisy labels. Without bells and whistles, our proposed method achieves excellent performance in terms of both test accuracy and label F1-score in comparison with state-of-the-art label-noise learning methods. Moreover, we show the good generalization capability of our method by introducing the proposed sample selection method into several representative label-noise learning methods.

2 Related Work

Roughly speaking, existing label-noise learning methods can be classified into three categories, including label transition matrix estimation, robust regularization, and sample selection.

Label transition matrix estimation. This category of methods is based on the estimation of the label transition matrix, which characterizes the label transition probabilities from a true class to an assigned one [22]. For example, Goldberger *et al.* [5] add an additional softmax layer in the neural network to model the label transition matrix. Patrini *et al.* [15] develop a two-step solution to heuristically estimate the label transition matrix. Yao *et al.* [23] introduce an intermediate class to decompose the original label transition matrix into the product of two easy-to-estimate transition matrices. Note that this category of methods cannot deal with a large number of labels and is fragile to a large ratio of noisy-labeled samples.

Robust regularization. This type of methods leverages robust regularization techniques to avoid overfitting on noisy labels and thus improve the generalization ability of DNNs. Pereyra *et al.* [16] estimate the marginalized effect of noisy labels during training and prevent DNN from assigning a full probability to the noisy-labeled sample, thereby reducing overfitting. Zhang *et al.* [28] regularize the DNN to favor simple linear behaviors in-between training samples to address the overfitting problem. Although these methods have achieved promising performance, they usually depend on additional hyperparameters that are sensitive to the data type [17, 13]. Moreover, some methods may completely memorize noisy-labeled samples with high capacity networks, since DNNs are often over-parameterized [6].

Sample selection. Recently, sample selection methods, which aim to select clean samples from noisy training data, have attracted considerable attention in label-noise learning. The small-loss criterion that identifies samples with small training losses as clean samples has been widely used in recent methods. For example, MentorNet [10] develops a collaborative learning paradigm, where a mentor network is first pre-trained and then used to select clean samples based on the small-loss criterion for guiding the training of a student network. Co-teaching [7] also involves two networks, where small-loss samples are selected by each network and fed into the peer network to update the network parameters in each mini-batch. Different from Co-teaching, Co-teaching+ [24] only selects small-loss samples with different prediction results from two networks. JoCoR [20] calculates a joint loss with co-regularization for each sample based on two networks, and then chooses the small-loss samples to simultaneously update the parameters of two networks. Yao *et al.* [22] use an AutoML method to dynamically determine the noise rate during the training process.

Different from conventional small-loss criterion based sample selection methods that only take into account the loss of each training sample in the current mini-batch, we also exploit the loss distribution in the whole training set at previous training epochs. This enables our method to more accurately identify clean samples or noisy-labeled samples in each mini-batch during training.

3 Proposed Method

In this section, we develop a novel and effective sample selection method for label-noise learning. After introducing preliminaries, we present the key components of our proposed method in detail.

3.1 Preliminaries

Considering a K-class classification problem with the noisy training data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $\mathbf{x}_i$ denotes the i-th sample (e.g., an image) in the training set and $y_i \in \{1, 2, \ldots, K\}$ represents the label corresponding to the sample $\mathbf{x}_i$. A sample is noisy-labeled when the corresponding label mismatches its ground-truth label. A large number of methods [10, 7, 24, 20] identify the samples that

are likely to be clean ones by using the small-loss criterion, and thus a robust model can be trained with small-loss samples in each mini-batch. The details of the small-loss criterion are described as follows.

A mini-batch data $\mathcal{D}_b^t = \{\mathbf{x}_j^b, y_j^b\}_{j=1}^J$ at epoch t is randomly drawn from the noisy training data $\mathcal{D}$, where $\mathbf{x}_j^b$ and y_j^b represent the j-th training sample in mini-batch b and the corresponding label, respectively. J denotes the batch size. The loss of each sample can be obtained by feeding $\mathcal{D}_b^t$ into the model and then used to identify clean samples. The selected clean data $\tilde{\mathcal{D}}_b^t$ can be formulated as

$$\tilde{\mathcal{D}}_b^t = \underset{\mathcal{D}':|\mathcal{D}'|\geq\lambda^t\cdot|\mathcal{D}_b^t|}{\arg\min} L(f_{\mathbf{\Theta}}, \mathcal{D}'), \tag{1}$$

where $f_{\mathbf{\Theta}}$ denotes the network with the parameters $\mathbf{\Theta}$, L represents the cross-entropy loss, and λ^t denotes the clean data rate which controls how many small-loss samples should be selected into $\tilde{\mathcal{D}}_b^t$. λ^t is heuristically defined as

$$\lambda^t = 1 - \tau \cdot \min(\frac{t}{T}, 1), \tag{2}$$

where τ is the estimated noise rate, which can be inferred using validation sets [12, 25]. The value of λ^t decreases quickly at the first T epochs until reaching $1-\tau$. Then, the selected clean data $\tilde{D}_b^t$ are used to calculate the average loss for updating the network parameters $\mathbf{\Theta}$.

Despite its popularity, the small-loss criterion suffers from the following two limitations. First, this criterion only considers the losses of mini-batch samples but ignores the loss distribution of all the samples in the whole training set. Such a way is not globally optimal. For example, as shown in the top row of Figure 1(a), when a mini-batch mainly contains noisy-labeled samples, the losses of this mini-batch may not be effectively used to indicate whether the label is noisy or clean. Second, the clean data rate λ^t is critical to exploit the memorization effect [1]. But according to Eq. (2), the value of λ^t is usually set without fully exploiting the knowledge on the data. In other words, λ^t is heuristic and it is often difficult to manually determine λ^t for each dataset. Therefore, some noisy-labeled samples may be improperly selected as clean ones, and vice versa (note that the mini-batch samples are randomly chosen from the whole training set). This clearly leads to a performance decrease.

The above limitations motivate us to formulate and design an effective small-vote sample selection method, which not only takes both the whole training set and the current mini-batch into account, but also adaptively selects clean samples. The proposed method mainly consists of a novel Hierarchical Voting Scheme (HVS) and an Adaptive Clean data rate Estimation Strategy (ACES).

3.2 Hierarchical Voting Scheme (HVS)

HVS combines the global vote (based on the loss distributions of all the samples at previous epochs) and the local vote (based on the losses of current mini-batch samples) to assign a hierarchical vote for each mini-batch sample. In general, we

compute and sort the loss of each sample in the whole training set after each epoch, and combine the normalized rank indices of each mini-batch sample at previous epochs as the global vote. Similarly, we view the normalized rank index of each sample in the current mini-batch as the local vote. Then, a hierarchical vote is performed by combining the global vote and the local vote from the epoch-level and batch-level, respectively.

To be specific, at epoch t, we compute the losses of all the samples in the whole training data. Suppose that the cross-entropy losses for the noisy training data are denoted as $\mathcal{L}^t = \{l_1^t, \cdots, l_N^t\}$, where l_i^t represents the loss of the i-th sample at epoch t. Then, we sort all the elements in $\mathcal{L}^t$ in the ascending order to obtain the sorted set $\mathcal{L}^t = \{l_{\mu_1}^t, \cdots, l_{\mu_N}^t\}$, where the permutation $\{\mu_1, \cdots, \mu_N\}$ is obtained such that $l_{\mu_1}^t \leq \cdots, \leq l_{\mu_N}^t$. Hence, the normalized rank index set $\mathcal{P}^t$ at epoch t can be formulated as

$$\mathcal{P}^t = \{p_{\mathbf{X}_{\mu_j}}^t | p_{\mathbf{X}_{\mu_j}}^t = j/N, \forall \ \ \mathbf{x}_{\mu_j} \in \mathcal{D}\}, \tag{3}$$

where $p_{\mathbf{X}_{\mu_j}}^t \in [0, 1]$ represents the normalized rank index of the μ_j-th sample.

For the global vote, we vote each mini-batch sample based on the normalized rank index set obtained at previous C training epochs, which can be formulated as

$$\mathcal{G}_b^t = \{g_{\mathbf{x}_j^b}^t | g_{\mathbf{x}_j^b}^t = \frac{1}{C} \sum_{c=1}^{C} p_{\mathbf{x}_j^b}^{(t-c)}, \forall \ \ \mathbf{x}_j^b \in \mathcal{D}_b^t\}, \tag{4}$$

where $g_{\mathbf{x}_j^b}^t \in [0, 1]$ indicates the global vote of $\mathbf{x}_j^b$ for mini-batch b at epoch t. C is a hyper-parameter used to control how many previous epochs are used to perform the global vote, and $p_{\mathbf{x}_j^b}^{(t-c)}$ represents the normalized rank index of $\mathbf{x}_j^b$ at epoch $(t - c)$.

Similarly, the losses for mini-batch b at epoch t are denoted as $\mathcal{L}_b^t = \{l_{1b}^t, \cdots, l_{Jb}^t\}$, where l_{jb}^t represents the loss of the j-th sample in the mini-batch. All the elements in $\mathcal{L}_b^t$ are sorted in the ascending order to obtain the sorted set $\mathcal{L}_b^t = \{l_{\nu_1 b}^t, \cdots, l_{\nu_J b}^t\}$, where the permutation $\{\nu_1, \cdots, \nu_J\}$ is obtained such that $l_{\nu_1 b}^t \leq \cdots, \leq l_{\nu_J b}^t$. The normalized rank index set $\hat{\mathcal{P}}_b^t$ in the mini-batch is

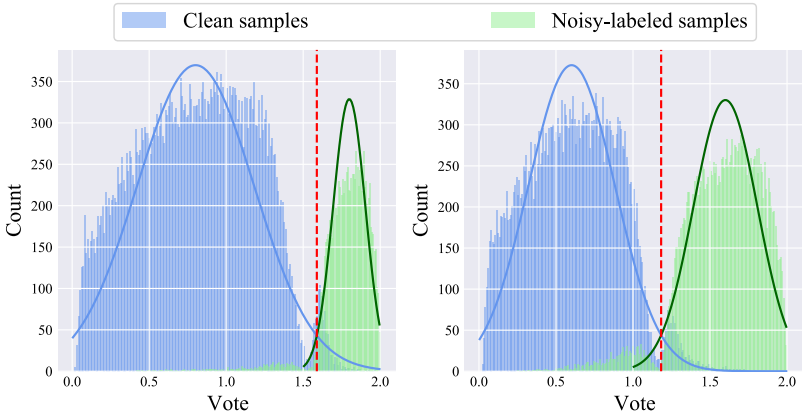
$$\hat{\mathcal{P}}_b^t = \{\hat{p}_{\mathbf{X}_{\nu_j}^b}^t | \hat{p}_{\mathbf{X}_{\nu_j}^b}^t = j/J, \forall \ \ \mathbf{x}_{\nu_j}^b \in \mathcal{D}_b^t\}, \tag{5}$$

where $\hat{p}_{\mathbf{X}_{\nu_j}^b}^t \in [0, 1]$ indicates the local vote of $\mathbf{x}_{\nu_j}^b$.

Finally, since both the global vote and the local vote have the same value range, we define the hierarchical votes of mini-batch samples at epoch t by simply combining them, i.e.,

$$\mathcal{V}_b^t = \{v_{\mathbf{x}_j^b}^t | v_{\mathbf{x}_j^b}^t = g_{\mathbf{x}_j^b}^t + \hat{p}_{\mathbf{x}_j^b}^t, \forall \ \ \mathbf{x}_j^b \in \mathcal{D}_b^t\}, \tag{6}$$

where $v_{\mathbf{x}_j^b}^t \in [0, 2]$ represents the hierarchical vote of $\mathbf{x}_j^b$ for mini-batch b at epoch t. It is worth noting that instead of directly relying on the loss of the sample, we



(a) Symmetry-20%. (b) Symmetry-40%.

Fig. 2. Histograms of hierarchical votes at an epoch on the training sets involving (a) symmetry-20% and (b) symmetry-40% noisy labels from the CIFAR-10 dataset. The ground-truth noisy-labeled samples and clean samples are marked with different colors. The red dotted lines denote the thresholds estimated by ACES.

take advantage of the normalized rank index for both the global vote and the local vote. Such a manner is able to address the problem of different scales of loss distributions at epochs and mini-batches.

3.3 Adaptive Clean Data Rate Estimation Strategy (ACES)

Intuitively, a noisy-labeled sample tends to have a higher hierarchical vote than a clean sample (note that a sample will be assigned with a high hierarchical vote if the corresponding local and global votes show large normalized rank indices). In Figure 2, we visualize the histograms of hierarchical votes at an epoch on the training sets involving two different levels of noisy labels from the CIFAR-10 dataset. We can find that the histogram of samples at an epoch shows two distinct modes which correspond to noisy-labeled samples and clean samples, respectively.

The λ^t defined in Eq. (2) depends on a fixed noise rate τ, which is heuristic. As a result, some noisy-labeled samples are incorrectly identified as clean ones, and vice versa. Such a manner leads to a performance decrease. Therefore, we develop ACES to adaptively estimate the clean data rate by taking advantage of a 1D Gaussian Mixture Model (GMM) to model the histogram with two modes.

More specifically, given the noisy training data $\mathcal{D}$ and the corresponding hierarchical votes $\mathcal{V}^{(t-1)}$ at epoch $(t-1)$, we fit these votes using a 1D GMM with two components:

$$F(\mathcal{V}^{(t-1)}) = \sum_{i=1,2} \pi_i^t \mathcal{N}\{\mathcal{V}^{(t-1)} | m_i^t, \gamma_i^t\}, \tag{7}$$

Algorithm 1: The small-vote sample selection method

Input: Network $f_{\mathbf{\Theta}}$ with the parameters $\mathbf{\Theta}$, the number of epochs C for the global vote, the maximum number of epochs T_{max}, the maximum number of iterations I_{max}, noisy training data $\mathcal{D}$, the learning rate η;
Output: training parameters $\mathbf{\Theta}$;

```
for t = 1, 2, ..., T_max do
    Shuffle the noisy training data D;
    if t ≤ C then
        Update the parameters Θ according to the small-loss criterion;
    else
        for b = 1, 2, ..., I_max do
            Fetch a mini-batch D_b^t from D;
            Compute G_b^t via Eq. (4);
            Compute P̂_b^t via Eq. (5);
            Update V_b^t based on the global vote and the local vote via Eq. (6);
            Select the training data D̂_b^t by using α^(t−1) via Eq. (8) from D_b^t;
            Update the parameters Θ via gradient descent based on the selected
            training data;
        end
    end
    Compute and store P^t via Eq. (3);
    Compute and store the threshold α^t;
    Update the learning rate η;
end
return Θ
```

where $\mathcal{N}$ denotes a Gaussian distribution, m_i^t and γ_i^t are the mean and standard deviation of the i-th component, respectively. π_i^t represents the weight of the i-th component. The parameters of 1D GMM can be estimated by using the EM algorithm [3]. Then, the threshold $\alpha^{(t-1)}$ is determined by finding the intersection point of two Gaussians.

After obtaining the threshold $\alpha^{(t-1)}$, we can select the mini-batch samples whose votes are lower than the threshold as clean samples at epoch t. Mathematically, the selected training data $\hat{D}_b^t$ can be obtained as

$$\hat{D}_b^t = \{\mathbf{x}_j^b | v_{\mathbf{x}_j^b}^t \leq \alpha^{(t-1)}, \ \ \forall \ \ \mathbf{x}_j^b \in D_b^t\}, \tag{8}$$

where $v_{\mathbf{x}_j^b}^t$ is the hierarchical vote of $\mathbf{x}_j^b$ for mini-batch b at epoch t. Hence, the clean data rate is estimated as $|\hat{D}_b^t|/J$.

The overall procedure of our proposed small-vote sample selection method is shown in Algorithm 1. It is worth noting that our proposed method can be viewed as an extension of the small-loss criterion. When only the local vote (defined in Eq. (5)) and the heuristic clean data rate (defined in Eq. (2)) are adopted, our proposed method degenerates to the small-loss criterion.

Table 1. The details of four benchmark datasets.

Datasets	# of train	# of test	# of class	size
MNIST	$60K$	$10K$	10	28×28
CIFAR-10	$50K$	$10K$	10	32×32
CIFAR-100	$50K$	$10K$	100	32×32
Clothing1M	$1M$	$10K$	14	256×256

4 Experiments

We conduct experiments on several commonly used benchmark datasets and compare the proposed method with several state-of-the-art label-noise learning methods.

Datasets. We show the effectiveness of our proposed method on four benchmark datasets, MNIST, CIFAR-10, CIFAR-100, and Clothing 1M [21]. These datasets are widely used for evaluating the performance of label-noise learning methods [7, 20]. The details of the four benchmark datasets are shown in Table 1.

For MNIST, CIFAR-10, and CIFAR-100, we follow the common settings to add synthetic noise into the training sets, as done in the literature [7, 9]. Specifically, the noisy labels are modified in the following two ways: (1) Symmetry flipping: a sample is assigned to a uniform random label rather than its true label with the probability p_s, where $p_s = 20\%$ or 40% in our experiments, as done in [18]; (2) Pair flipping: a sample in one class is assigned to have the same label of another class [7]. The probability p_k of sample mislabelling in a class is simply set to 40% in our experiments due to space limits. Similar results can be observed for other values of p_s or p_k.

For Clothing 1M, we follow the same settings as [20]. We use $1M$ images with noisy labels as the training set and $10K$ clean images as the test set. For each image in the Clothing 1M dataset, we resize it to 256×256 and crop the middle 224×224 as the input of the model.

Competing Methods. We compare our proposed small-vote sample selection method (called as small-vote) with the following state-of-the-art label-noise learning methods, including Co-teaching [7], Co-teaching+ [24], O2U-Net [9], and JoCoR [20]. The baseline method that trains the standard Convolutional Neural Network (CNN) with the small-loss criterion is also used.

Network Structure and Optimizer. For a fair comparison, we re-implement all the methods based on the open-source codes by PyTorch and conduct all the experiments on a NIVIDIA 2080Ti GPU. For MNIST, CIFAR-10, and CIFAR-100, we adopt a 9-layer CNN [7]. For Clothing 1M, we use ResNet-18 [8]. All experiments are trained for 200 epochs with the Adam optimizer ($momentum = 0.9$). The batch size is set to 128 for each dataset. The number of epochs C for the global vote is set to 10. Similarly to O2U-Net [9], we use a linear decrease function to cyclically adjust the learning rate, which linearly decreases from 0.1 to 0.001 in a cycle round. All our code will be released soon.

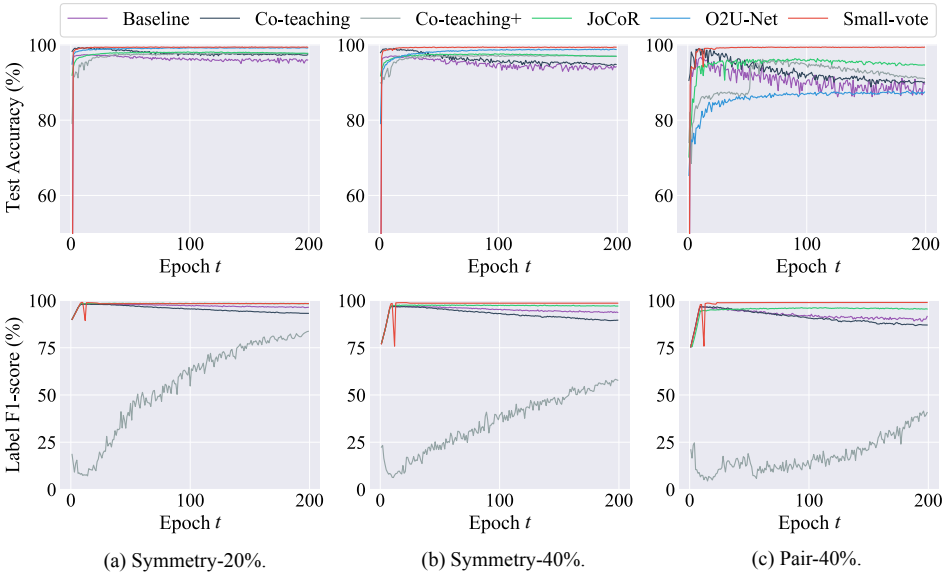


Fig. 3. Results on the MNIST dataset under different settings of noisy labels. Top: test accuracy (%), vs. epochs; bottom: label F1-score (%) vs. epochs.

Evaluation Metrics. We use two commonly used evaluation metrics: test accuracy (i.e., *test accuracy = (# of correct predictions) / (# of test data)*) and label F1-score (i.e., *Label F1-score* $= \frac{2\times P\times R}{P+R}$, where P = *(# of clean labels) / (# of all selected labels)* is the label precision and R = *(# of clean labels) / (# of all clean samples)* is the label recall).

4.1 Comparisons with State-of-the-Arts

We evaluate the performance obtained by all the competing methods on the synthetic noisy labels by using MNIST, CIFAR-10, and CIFAR-100. Figures 3-5 show the results on the three datasets, respectively. We report the test accuracy vs. the number of epochs and the label F1-score vs. the number of epochs on the three datasets under different settings of noisy labels (including Symmetry-20%, Symmetry-40%, and Pair-40%).

Moreover, we also show the superiority of our proposed method on the real-world noisy labels by using the Clothing1M dataset. The comparison results are given in Table 2, where "best" and "last" respectively denote the trained models at the epoch (when the validation accuracy is optimal) and at the end of training epochs.

Results on MNIST. As shown in Figure 3, our proposed small-vote method outperforms the other competing methods in terms of both test accuracy and label F1-score for different settings of noisy labels. This is because HVS effectively assigns a hierarchical vote for each mini-batch sample and ACES adaptively

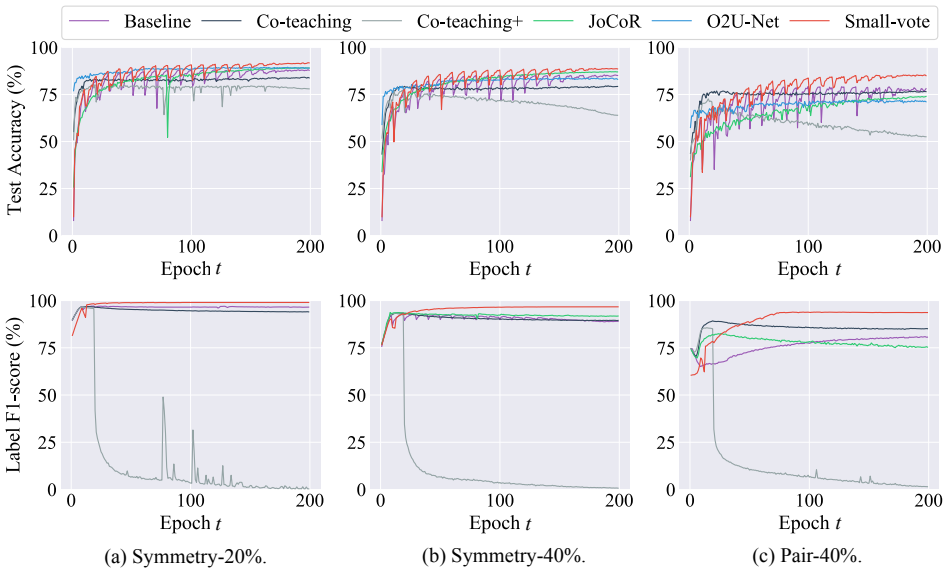


Fig. 4. Results on the CIFAR-10 dataset under different settings of noisy labels. Top: test accuracy (%), vs. epochs; bottom: label F1-score (%) vs. epochs.

estimates the clean data rate, leading to accurate identification of both noisy-labeled samples and clean samples. O2U-Net involves two stages, where noisy labels are detected in the first stage and then a model is trained on clean data in the second stage. However, O2U-Net does not fully use the memorization effect to alleviate the negative impact of noisy-labeled samples for training. Hence, the test accuracy obtained by O2U-Net is inferior to our method. Note that O2U-Net is not shown in the second row of Figure 3 since no sample selection is used in O2U-Net.

Results on CIFAR-10. As shown in the first row of Figure 4, our proposed small-vote method outperforms the other competing methods with a large margin. The recent state-of-the-art JoCoR obtains much worse performance than our method in terms of average test accuracy over the last ten epochs (about 11.96%, 2.56%, and 2.49% decrease under three settings of noisy labels). From the second row of Figure 4, the label F1-scores obtained by Co-teaching, Co-teaching+, and JoCoR gradually decline after several epochs. In contrast, our small-vote method not only achieves high label F1-scores under all the settings, but also shows better performance at larger epochs. This can be ascribed to the effectiveness of our method for discriminating noisy-labeled samples from clean samples, enabling the capability of learning a more robust model.

Results on CIFAR-100. CIFAR-100 is more challenging than MNIST and CIFAR-10. The overall test accuracy obtained by all the methods on CIFAR-100 is much lower than that on CIFAR-10, since there are more classes in CIFAR-100. As shown in Figure 5, the label F1-score obtained by other competing method-

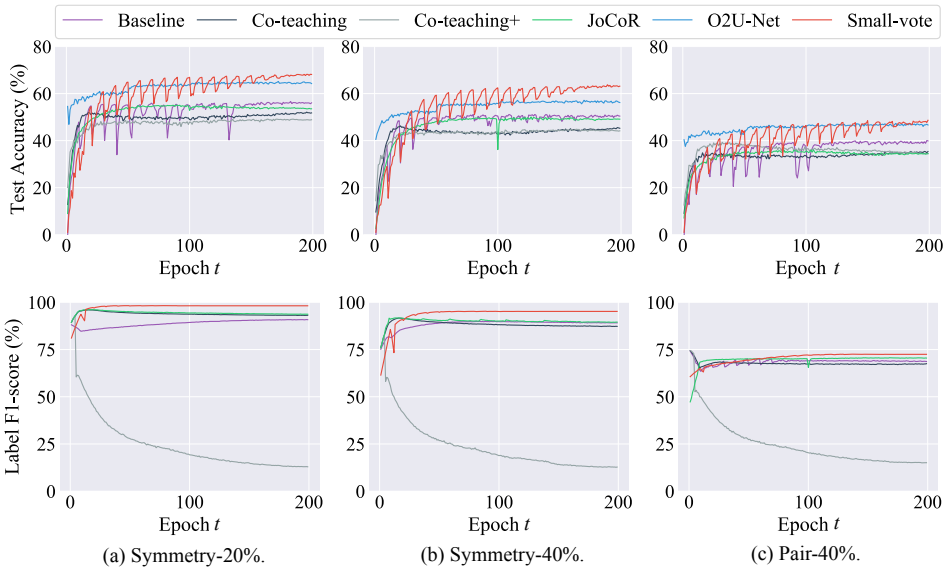


Fig. 5. Results on the CIFAR-100 dataset under different settings of noisy labels. Top: test accuracy (%), vs. epochs; bottom: label F1-score (%) vs. epochs.

s first increases and then gradually decreases during the training. This shows that the ability to identify noisy-labeled samples is limited for these methods. However, the label F1-score obtained by the proposed method keeps very stable after several epochs under different settings, leading to better test accuracy. Compared with the baseline method, our proposed method obtains much better performance, which shows the excellent classification ability of our method. In general, our proposed method performs favorably against the other competing methods.

Results on Clothing 1M. From Table 2, our proposed small-vote method obtains better results than the other competing methods on best. Moreover, small-vote achieves a significant improvement in accuracy of 11.53% over Co-teaching+, and an improvement of 0.53% over the JoCoR on last. Therefore, small-vote can effectively identify noisy-labeled samples and clean samples on the dataset containing real-world noisy labels. In summary, our proposed method achieves state-of-the-art results against several competing methods on four datasets with synthetic and real-world noisy labels.

4.2 Ablation studies

In this subsection, we perform ablation studies to analyze the effectiveness of key components of small-vote and the influence of the key parameter on CIFAR-10. Moreover, we show the generalization capability of our method by integrating small-vote with several state-of-the-art label-noise learning methods.

Table 2. Results on the Clothing 1M dataset. The test accuracy (%) is used for performance comparison. The best results are highlighted in bold.

Methods	best	last
Co-teaching	69.21	68.51
Co-teaching+	59.32	58.79
JoCoR	70.30	69.79
Baseline	68.72	68.21
Small-vote	**70.43**	**70.32**

Table 3. Ablation study of key components of our small-vote on CIFAR-10. The average test accuracy (%) over the last ten epochs is used for performance comparison. The best results are highlighted in bold.

Methods	Symmetry-20%	Symmetry-40%	Pair-40%
HVS-L	89.67	86.54	80.46
HVS-G	90.12	87.53	83.12
HVS	90.99	88.27	84.51
Small-vote	**91.56**	**89.59**	**85.70**

First, to evaluate the importance of two main components (i.e., HVS and ACES) in small-vote, we compare the following variants. The HVS-G and HVS-L methods denote our proposed method based on the global vote and the local vote, respectively, without using ACES (we use λ^t defined in Eq. (2) to select clean samples instead). The HVS method denotes our proposed method based on HVS without using ACES. The results are given in Table 3. Note that HVS-L is equivalent to the baseline method.

Compared with HVS-L, HVS achieves a performance boost. HVS-L only relies on the local information from the current mini-batch and uses a heuristic clean date rate. On the contrary, HVS effectively combines the global information from previous epochs and the local information from the current mini-batch. HVS also obtains higher test accuracy than HVS-G. This shows that both the global vote and the local vote play an important role in sample selection. Small-vote outperforms HVS under different settings of noisy labels. This can be ascribed to the adoption of ACES. ACES is a data-dependent clean data rate estimation strategy, which can effectively address the problem of unknown noisy label distribution in the mini-batch (note that the mini-batch is randomly chosen from the whole training data). Therefore, small-vote is able to identify clean and noisy-labeled samples more accurately, thereby leading to a more robust DNN model.

Second, we evaluate the influence of the key parameter C defined in Eq. (4) on the final performance. We set the values of C to 0, 5, 10, and 50. The results are given in Table 4. We can see that our small-vote achieves the best performance when the value of C is set to 10. When the value of C is set to

Table 4. Influence of different values of C on CIFAR-10. The average test accuracy (%) over the last ten epochs is used for performance comparison. The best results are highlighted in bold.

C	Symmetry-20%	Symmetry-40%	Pair-40%
0	89.67	86.54	80.46
5	91.05	88.10	84.95
10	**91.56**	**89.59**	**85.70**
50	90.87	88.50	82.05

Table 5. Performance comparison between small-loss and small-vote in the frameworks of Co-teaching, Co-teaching+, and JoCoR on the CIFAR-10 dataset. The average test accuracy (%) over the last ten epochs is used for performance comparison. The best results are highlighted in bold.

Methods	Symmetry-20%	Symmetry-40%	Pair-40%
Co-teaching (small-loss)	83.92	79.25	73.93
Co-teaching (small-vote)	**85.44**	**81.35**	**75.17**
Co-teaching+ (small-loss)	78.09	64.27	52.79
Co-teaching+ (small-vote)	**80.32**	**66.24**	**66.67**
JoCoR (small-loss)	89.07	86.75	73.73
JoCoR (small-vote)	**90.32**	**88.09**	**82.50**

0, only the local vote is used. When the value of C is large, too old historical information is exploited to perform the global vote. For these two extreme cases, the performance drops. Therefore, in all the experiments, we fix the value of C to 10.

Finally, to demonstrate the generalization ability of our small-vote sample selection method, we replace the small-loss criterion used in several state-of-the-art label-noise learning methods (including Co-teaching, Co-teaching+, and JoCoR) with our proposed small-vote. Table 5 shows the test accuracy obtained by different methods on CIFAR-10. As we can see, our small-vote sample selection outperforms the small-loss criterion with a moderate margin for different types and levels of noisy labels in the frameworks of Co-teaching, Co-teaching+, and JoCoR. Therefore, our small-vote sample selection is able to distinguish clean samples from noisy-labeled samples more effectively than the small-loss criterion, leading to performance improvements. The above results further show the great generalization of small-vote for label-noise learning.

5 Conclusion

In this paper, we have proposed a simple yet effective small-vote sample selection method for label-noise learning. The proposed method is comprised of two main components, including a Hierarchical Voting Scheme (HVS) and an

Adaptive Clean data rate Estimation Strategy (ACES). HVS effectively combines the global vote from previous epochs and the local vote from the current mini-batch to assign hierarchal votes for each mini-batch. Based on HVS, ACES adaptively estimates the clean data rate, so that clean samples and noisy-labeled samples can be accurately identified in the mini-batch. Experimental results on noisy-labeled data from four benchmark datasets including MNIST, CIFAR-10, CIFAR-100, and Clothing1M have shown the superiority of our proposed method over several state-of-the-art methods. Moreover, the good generalization capability of our method has been verified by incorporating our small-vote method into representative label-noise learning methods.

Acknowledgments

This work was supported by the Open Research Projects of Zhejiang Lab under Grant 2021KB0AB03, by the National Natural Science Foundation of China under Grants 62071404 and 61872307, by the Natural Science Foundation of Fujian Province under Grant 2020J01001, and by the Youth Innovation Foundation of Xiamen City under Grant 3502Z20206046.

References

1. Arpit, D., Jastrzķebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: ICML. pp. 233–242 (2017)
2. Bootkrajang, J., Kabán, A.: Label-noise robust logistic regression and its applications. In: ECML-PKDD. pp. 143–158 (2012)
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)
4. Feng, L., Shu, S., Lin, Z., Lv, F., Li, L., An, B.: Can cross entropy loss be robust to label noise. In: IJCAI. pp. 2206–2212 (2020)
5. Goldberger, J., Ben-Reuven, E.: Training deep neural-networks using a noise adaptation layer. In: ICLR (2017)
6. Han, B., Niu, G., Yu, X., Yao, Q., Xu, M., Tsang, I., Sugiyama, M.: Sigua: Forgetting may make learning with noisy labels more robust. In: ICML. pp. 4006–4016 (2020)
7. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: NeurIPS. pp. 8527–8537 (2018)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
9. Huang, J., Qu, L., Jia, R., Zhao, B.: O2u-net: A simple noisy label detection approach for deep neural networks. In: ICCV. pp. 3326–3334 (2019)
10. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: ICML. pp. 2304–2313 (2018)

11. Kumar, A., Shah, A., Raj, B., Hauptmann, A.: Learning sound events from webly labeled data. In: IJCAI. pp. 2772–2778 (2019)
12. Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(3), 447–461 (2015)
13. Luo, Y., Han, B., Gong, C.: A bi-level formulation for label noise learning with spectral cluster discovery. In: IJCAI. pp. 2605–2611 (2020)
14. Malach, E., Shalev-Shwartz, S.: Decoupling "when to update" from "how to update". In: NeurIPS. pp. 960–970 (2017)
15. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: CVPR. pp. 1944–1952 (2017)
16. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. In: ICLRW (2017)
17. Song, H., Kim, M., Park, D., Lee, J.G.: Learning from noisy labels with deep neural networks: A survey. arXiv preprint arXiv:2007.08199 (2020)
18. Van Rooyen, B., Menon, A., Williamson, R.C.: Learning with symmetric label noise: The importance of being unhinged. In: NeurIPS. pp. 10–18 (2015)
19. Vembu, S., Zilles, S.: Interactive learning from multiple noisy labels. In: ECML-PKDD. pp. 493–508 (2016)
20. Wei, H., Feng, L., Chen, X., An, B.: Combating noisy labels by agreement: A joint training method with co-regularization. In: CVPR. pp. 13726–13735 (2020)
21. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: CVPR. pp. 2691–2699 (2015)
22. Yao, Q., Yang, H., Han, B., Niu, G., Kwok, J.T.Y.: Searching to exploit memorization effect in learning with noisy labels. In: ICML. pp. 10789–10798 (2020)
23. Yao, Y., Liu, T., Han, B., Gong, M., Deng, J., Niu, G., Sugiyama, M.: Dual t: Reducing estimation error for transition matrix in label-noise learning. In: NeurIPS (2020)
24. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I.W., Sugiyama, M.: How does disagreement help generalization against label corruption? In: ICML. pp. 7164–7173 (2019)
25. Yu, X., Liu, T., Gong, M., Batmanghelich, K., Tao, D.: An efficient and provable approach for mixture proportion estimation using linear independence assumption. In: CVPR. pp. 4480–4489 (2018)
26. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: ICLR (2017)
27. Zhang, H., Long, D., Xu, G., Zhu, M., Xie, P., Huang, F., Wang, J.: Learning with noise: Improving distantly-supervised fine-grained entity typing via automatic relabeling. In: IJCAI. pp. 3808–3815 (2020)
28. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: Beyond empirical risk minimization. In: ICLR (2018)